

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações

Este exemplar corresponde a redação final da tese
defendida por Gustavo Abraham
Hirchoren e aprovada pela Comissão
Julgada em 03/09/1999
Dalton Soares Arantes
Orientador

Predição e Estimação de Parâmetros de Processos
Auto-Similares para Redes de Faixa Larga

Gustavo Abraham Hirchoren

Orientador:

Prof. Dr. Dalton Soares Arantes

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do Título de Doutor em Engenharia Elétrica.

Campinas, SP - Brasil
1999

20003564

UNIDADE	Be
N.º CHAMADA:	H614p
V.	Ex.
TOMBO BC/	40476
PROC.	278/00
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	\$11,00
DATA	19/03/00
N.º CPD	

CM-00135071-2

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

H614p Hirchoren, Gustavo Abraham
Predição e estimação de parâmetros de processos auto-
similares para redes de faixa larga / Gustavo Abraham
Hirchoren.--Campinas, SP: [s.n.], 1999.

Orientador: Dalton Soares Arantes.

Tese (doutorado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Redes de computação. 2. Teoria da previsão. 3.
Estimativa de parâmetro. 4. Processos gaussianos. I.
Arantes, Dalton Soares. II. Universidade Estadual de
Campinas. Faculdade de Engenharia Elétrica e de
Computação. III. Título.

RESUMO

O objetivo deste trabalho é o estudo de técnicas ótimas e sub-ótimas, no sentido quadrático médio, para a caracterização estatística de processos auto-similares, visando aplicações em redes de faixa larga. Especificamente, o trabalho aborda métodos ótimos e sub-ótimos de predição e estimação de parâmetros para processos Brownianos Fracionários (fBm) não estacionários, que têm sido considerados os mais representativos na modelagem de processos com dependência de longo prazo. Preditores ótimos variantes no tempo para processos fBm discretos são obtidos, e sua aplicação no controle, gerenciamento e policiamento de tráfego é discutida. Demonstra-se que uma técnica de baixa complexidade pode ser obtida para o cômputo da transformada wavelet sequencial, que pode ser utilizada com vantagem na estimação dos parâmetros do fBm. O problema do policiamento de tráfego em redes ATM é estudado à luz das características de dependência de longo prazo dos processos auto-similares. Para isso, estuda-se o comportamento do algoritmo "leaky bucket" para o policiamento de tráfego e as possíveis vantagens do uso de algoritmos ótimos para processos auto-similares. Neste caso, desenvolve-se uma técnica baseada em filtros de Kalman para a estimação dos parâmetros mais relevantes para o policiamento de tráfego.

ABSTRACT

The objective of this work is the study of optimal and suboptimal techniques, in the mean-square sense, for the statistical characterization of self-similar processes, with applications to broadband networks. Specifically, the work is about optimal and suboptimal methods of prediction and estimation of parameters for fractional Brownian motions (fBm), a class of nonstationary processes that have been considered for long-range dependent processes modeling. Time-variant optimal predictors for discrete fBm processes are obtained and their applications in traffic control, management and policy are discussed. It is demonstrated that a technique of low complexity can be obtained for the computation of sequential wavelet transform, that can be used with advantages in estimation of fBm parameters. The problem of traffic policing in ATM networks is studied on the basis of the characteristics of the long-range dependence of self-similar processes. The leaky bucket algorithm for traffic policing is discussed and the possible advantages of using an optimal algorithm for self-similar processes are studied. In this case, a technique based on Kalman filters is presented for the estimation of the relevant parameters for traffic policing.

Agradecimentos

Desejo expressar a minha eterna gratidão ao **Prof. Dr. Dalton Soares Arantes**, cujo apoio permanente e colaboração foram essenciais para a execução deste trabalho. Em sua pessoa encontrei não somente um grande pesquisador e professor mas uma grande amizade.

Gostaria de agradecer ao meu amigo Ernesto Barrientos pelo grande apoio com as séries auto-similares geradas com o método de Paxson.

Gostaria de agradecer também aos meus pais **Benjamin e Rosa** e ao meu irmão **Hernán** pela compreensão, apoio e incentivo.

Gustavo Abraham Hirchoren

Campinas, Julho de 1999.

Conteúdo

1	Introdução	1
2	Controle de Tráfego em Redes ATM	5
2.1.	Introdução	5
2.2.	Técnicas de controle de tráfego propostas para redes ATM	6
2.3.	Algoritmos de controle de admissão	8
2.3.1.	Aproximação Gaussiana	8
2.3.2.	Reserva de fila rápida.	9
2.3.3.	Capacidade equivalente	11
2.3.4.	Aproximação de fluxo para a probabilidade de perda de células.	12
2.3.5.	Método não paramétrico.	13
2.3.6.	Aproximação de tráfego pesado.	13
2.3.7.	Comparação das técnicas de admissão de conexão	14
2.3.8.	Conformação do tráfego	14
2.4.	Policimento de tráfego	14
2.4.1.	“Leaky Bucket”	15
2.4.2.	Técnicas baseadas em janelas.	16
2.4.3.	Eficiência dos métodos baseados em janela	17
2.5.	Descarte seletivo	18
2.5.1.	“Push-out”	18
2.5.2.	“Threshold”	18
2.6.	Mecanismos de controle reativo de congestionamento.	19
2.6.1.	Técnicas de notificação ao nó final.	19
2.6.2.	Controle adaptativo de taxa.	20
2.6.3.	Negociação de parâmetros na chamada.	20
2.6.4.	Codificação dinâmica de fonte.	20
2.7.	Resumo	21
3	Análise do Leaky Bucket	22
3.1.	O método de fluxo	23
3.2.	Análise do desempenho do Leaky Bucket com o método do fluxo.	24
3.3.	Seletividade do Leaky Bucket.	25

3.4.	Controle da taxa de bit de pico (situação I)	28
3.5.	Controle da taxa de bit média (situação II).	29
3.6.	Controle da duração do surto (situação III).	31
3.7.	Conclusões.	31
4	Predição Ótima para o Movimento Browniano Fracionário	32
4.1.	Introdução	32
4.2.	Predição ótima do fBm em tempo discreto	34
4.3.	Projeto do preditor ótimo a N passos e de ordem p do fBm e análise do desempenho	44
4.4.	Conclusões	47
5	Estimação dos Parâmetros do fBm Usando Transformada de Wavelet	48
5.1.	Introdução	48
5.2.	Propriedades da correlação dos coeficientes wavelet do movimento Browniano fracionário	49
5.3.	Estimação do parâmetro H e da variância σ^2 de uma série auto-similar	51
5.3.1.	Resultados das simulações	53
5.4.	Banco de preditores de Wiener multi-escala usando wavelets	55
6	Policiamento de Tráfego com Filtro de Kalman	57
6.1.	Introdução	57
6.2.	Estrutura da correlação do processo $i(n)$	59
6.3.	Caso I: estimação de K_1 , K_2 e $i(n)$ com um preditor de Kalman. O parâmetro H é suposto já estimado com a técnica de wavelets	60
6.4.	Caso II: estimação de K_1 e $i(n)$ com um preditor de Kalman. Os parâmetros H e K_2 são supostos já estimados com a técnica de wavelets	63
6.5.	Caso III: estimação de $Y(n)$ com um preditor de Kalman. Todos os parâmetros H , K_1 e K_2 são supostos desconhecidos.	65
6.6.	Resultados das simulações	67
7	Estimação Sequencial com Transformada de Wavelet de Baixa Complexidade	74
7.1.	Introdução	74
7.2.	Complexidade da decomposição em wavelets sequencial usando wavelets de Haar	75
7.3.	Complexidade da decomposição em wavelets sequencial usando wavelets com qualquer L	77
7.4.	Conclusões	82
8	Conclusões	83

Lista de Figuras

2.1	Multiplexagem determinística versus multiplexagem estatística.	7
2.2	Quatro versões do leaky bucket.	16
3.1	O leaky bucket como um dispositivo de policiamento de tráfego.	22
3.2	O processo chegada de células para uma fonte “on-off”.	23
3.3	Probabilidade de perda B em função da taxa de bit de pico b/b_0 para o serviço “Packet Voice” ($a/b_0 = 0.8$, $N = 135$).	25
3.4	Probabilidade de perda B em função da duração média do surto h/h_0 para o serviço “Still Picture” (Caso B).	27
3.5	Probabilidade de perda B , representando o desempenho do Leaky Bucket na situação I para o serviço “Packet Voice” ($a/b_0 = 0.99$, $M = 1696$ bits).	29
3.6	Probabilidade de perda B , representando o desempenho do Leaky Bucket na situação II para o serviço “Packet Voice” ($a/b_0 = 0.38$, $M = 1107064$ bits).	30
4.1	Coefficiente ótimo $c_n^2(0)$, variante no tempo, em função de n , para $H = 0, 8$	35
4.2	Coefficiente ótimo $c_n^2(0)$, variante no tempo, em função de n , para $H = 0, 8$	37
4.3	Coefficiente ótimo $c_n^2(1)$, variante no tempo, em função de n , para $H = 0, 8$	37
6.1	Estimação da taxa média para a série 1 com o filtro de Kalman (—) e com o leaky bucket (...).	68
6.2	Processo $Y(n)$ (—) e sua predição a um passo (...), para a série 1.	69
6.3	Estimação da taxa média para a série 2 com o filtro de Kalman (—) e com o leaky bucket (...).	69
6.4	Processo $Y(n)$ (—) e sua predição a um passo (...), para a série 2.	70
6.5	Processo $Y(n)$ (—) e sua predição a um passo (...), para as primeiras 350 amostras da série BC-Oct89Ext agregada em intervalos de 1 seg.	73
7.1	$Complexidade_{seq}$ e seu limitante superior para $N = 2048$ e $L = 10$, em função de K	81

Lista de Tabelas

2.1	Parâmetros das conexões 1 e 2.	8
3.1	Características dos serviços.	24
3.2	Limiar N do contador Leaky Bucket para $B = 10^{-9}$	24
3.3	Comportamentos de usuário.	26
3.4	Valores de N usados nas curvas da Figura 3.4 para cada valor de a/b_0	27
4.1	$NMSE$ para a predição ótima a 5 passos do fBm com $H = 0.9$ e diferentes ordens p nos preditores.	45
4.2	Coefficientes ótimos e o correspondente $NMSE$ teórico do preditor ótimo a N passos para $H = 0.6$ e diferentes valores de N	46
4.3	Coefficientes ótimos e o correspondente $NMSE$ teórico do preditor ótimo a N passos para $H = 0.7$ e diferentes valores de N	46
4.4	Coefficientes ótimos e o correspondente $NMSE$ teórico do preditor ótimo a N passos para $H = 0.8$ e diferentes valores de N	46
4.5	Coefficientes ótimos e o correspondente $NMSE$ teórico do preditor ótimo a N passos para $H = 0.9$ e diferentes valores de N	46
5.1	Valores obtidos de \hat{H} , ordenada à origem (\hat{b}) e $\hat{\sigma}^2$ para distintas simulações.	53
5.2	Médias.	53
5.3	Desvios padrões.	53
5.4	Desvio padrão/Média.	53
5.5	Valores obtidos de \hat{H} , ordenada à origem (\hat{b}) e $\hat{\sigma}^2$, para distintas simulações. Foi utilizada uma regressão linear ponderada.	54
5.6	Médias.	55
5.7	Desvios padrões.	55
5.8	Desvio padrão/Média.	55
5.9	Valores teóricos do $NMSE$ e os obtidos nas simulações para predição a 5 passos de $d_j[n]$ para diferentes valores do parâmetro H . Foi usado $p = 6$	56
6.1	Comparação dos valores obtidos nas simulações para as séries 1 e 2 no caso II.	70
6.2	Comparação dos valores obtidos nas simulações para as séries 1 e 2 no caso III.	71

6.3	Comparação dos valores obtidos nas simulações para as séries $h05$, $h06$, $h07$, $h08$ e $h09$ no caso III.	72
6.4	Valores obtidos nas simulações para as séries BC-pOct89 e BC-Oct89Ext no caso III.	72
7.1	Quantidade de coeficientes de detalhamento (igual à quantidade de coeficientes de aproximação) que não deve ser calculada novamente N_e	75
7.2	Valores do K que resultam em $Complexidade_{seq} < 10\sqrt{N}$ para diferentes valores de N e L	80

Capítulo 1

Introdução

A evolução das atuais redes de comunicação em direção à chamada Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL) trouxe consigo a necessidade de se projetar redes capazes de suportar uma grande variedade de serviços com diferentes taxas de bit e características. Em particular, o modo de transferência conhecido como ATM (Asynchronous Transfer Mode) tem sido adotado como alternativa direta para a implementação da RDSI-FL.

Basicamente, a rede ATM utiliza a comutação de pacotes de comprimento fixo (chamados de células) e otimiza os recursos da rede através da multiplexagem estatística. Isto requer mecanismos de controle de congestionamento a fim de se evitar excessiva perda de informação para poder cumprir com as exigências de QoS (qualidade de serviço) das diferentes conexões.

O problema se agrava se considerarmos que os grandes tributários da rede ATM, serão, seguramente, as aplicações de vídeo a taxa de bit variável (VBR) e de redes locais, que geram tráfego auto-similar [12]. Já foi demonstrado [12] que, infelizmente, nenhum dos modelos comumente utilizados para modelagem de tráfego é capaz de captar este comportamento auto-similar ou fractal.

A grande dificuldade hoje consiste em que os parâmetros de tráfego necessários para caracterizar uma fonte é, em grande parte, ainda um problema em aberto. Em consequência, a determinação dos parâmetros a serem controlados também é um problema em aberto. Mesmo assim, várias técnicas de controle de tráfego já foram propostas para redes ATM [16].

No Capítulo 2 deste trabalho são discutidos os problemas do controle de tráfego em redes ATM e analisadas distintas técnicas de controle de admissão, de policiamento do tráfego e de controle reativo de congestionamento.

No Capítulo 3 é analisada em profundidade a técnica recomendada nos padrões internacionais para policiamento de tráfego: o “leaky bucket”.

O “leaky bucket” pode ser considerado como um dispositivo que faz uma prova estatística sobre as células que são oferecidas à rede. A hipótese de prova é que os valores dos parâmetros do tráfego estejam dentro dos limites contratados no “call setup”. Uma curva ideal de probabilidade de perda para a função de policiamento deve levar em consideração dois requisitos:

1. Quando a hipótese é verdadeira (a fonte é bem comportada) o dispositivo de policiamento não deve introduzir perda de células (transparência).

2. Quando a hipótese de prova é falsa (uma fonte supera os parâmetros contratados) todas suas células deveriam ser descartadas ou marcadas.

O comportamento ideal acima não pode ser obtido com dispositivos práticos de policiamento. Não obstante, o desempenho de uma técnica de policiamento pode ser avaliado através da curva da probabilidade de perda B como uma função dos parâmetros da fonte. O dispositivo é bom se a probabilidade de perda é baixa quando a hipótese de prova é verdadeira, e a probabilidade de perda é alta quando a hipótese de prova é falsa.

No caso do “leaky bucket”, podemos variar as características da prova ajustando os dois parâmetros de controle: a capacidade do “buffer” M e a taxa de drenagem a .

No Capítulo 3 são consideradas duas fontes “bursty” do tipo “on/off” com durações dos períodos de atividade e de silêncio com distribuições exponenciais. Para a segunda fonte o grau de surto é muito maior do que para a primeira.

Utilizando o método do fluxo, é estudado o desempenho do leaky bucket em três situações:

1. Controle da taxa de bit de pico.
2. Controle da taxa de bit média.
3. Controle da duração do surto.

As conclusões das simulações são as seguintes:

- Foi observado que os parâmetros do leaky bucket dependem fortemente das características da fonte a ser controlada.
- A taxa de bit de pico pode ser controlada facilmente adotando a taxa de drenagem perto da taxa de pico da fonte.
- O controle da taxa de bit média requer uma taxa de drenagem perto da taxa média da fonte.
- O comprimento do surto não pode ser controlado pelo leaky bucket quando a fonte incrementa na mesma proporção as durações do surto e de silêncio.
- No caso do controle da taxa de bit média é requerido um limiar muito alto no contador do leaky bucket para se obter uma probabilidade de perda aceitável em condições nominais. Isto implica que são necessários tempos muito grandes para se detetar uma violação da taxa de bit média. Este problema é tanto mais grave quanto maior for a duração dos surtos (ou seja, com tráfego muito “bursty”).

Os problemas da técnica leaky bucket aumentam notavelmente com tráfego auto-similar devido ao alto grau de surto que está presente em todas as escalas de tempo. No Capítulo 4 utiliza-se o processo movimento Browniano fracionário (fBm) para modelar processos auto-similares. Norros [1] propôs o chamado tráfego Browniano fracionário para modelar o tráfego em redes locais. Utilizando o modelo do tráfego Browniano fracionário, o grau de auto-similaridade (ou “burstiness”) do tráfego pode ser medido somente com um parâmetro: o parâmetro de Hurst H do fBm. Em [2] apresenta-se o preditor ótimo do fBm no sentido quadrático médio em tempo contínuo, bem como o correspondente erro quadrático médio. Os resultados são interessantes e enfatizam algumas propriedades peculiares dos preditores

para processos auto-similares. Entretanto, dado que na prática só as amostras do processo são conhecidas, neste trabalho desenvolvemos o preditor ótimo do fBm em tempo discreto a N passos e de ordem p , no sentido quadrático médio. Nossos resultados também verificam as características distintivas dos preditores mencionadas em [1] [2], mas para sinais em tempo discreto, de maior utilidade prática.

Além disso, no Capítulo 4 são obtidos os valores numéricos dos coeficientes e do correspondente erro quadrático médio do preditor ótimo, para diferentes valores dos parâmetros H , N e p .

Foram desenvolvidos na literatura diferentes estimadores de H no domínio do tempo e no domínio da frequência. No Capítulo 5 destacamos as vantagens da técnica de estimação de H baseada em wavelets [11], [19]. Analisamos também a estimação usando wavelets da constante de variância do fBm.

Demonstramos que fazendo um desenvolvimento wavelets do fBm é possível:

1. Estimar a constante de variância do fBm.
2. Obter uma completa predição dos incrementos do fBm em diferentes instantes e escalas de tempo usando um banco de preditores de Wiener multi-escala.

A análise com wavelets é, portanto, uma ferramenta bastante apropriada para processos fBm.. Ela fornece uma precisa estimação do parâmetro H do fBm, que é muito útil para caracterizar o grau de auto-similaridade do tráfego. Usando-se o valor de H estimado, pode-se obter as predições do tráfego no espaço tempo-escala para aplicação em funções de gerenciamento de tráfego em tempo real como mecanismos de controle de congestionamento dinâmicos. Por exemplo, um esquema de controle lógico pode escolher a predição na escala adequada de acordo com o nível da fila num certo instante de tempo.

O processo tráfego Browniano fracionário é completamente caracterizado pelos três parâmetros básicos H , m e a , com as seguintes interpretações: $m > 0$ é a taxa média do tráfego, $a > 0$ é um coeficiente de variância e $0.5 \leq H < 1$ é o parâmetro de Hurst. O uso de modelos parcimoniosos como este, é de grande interesse na modelagem de tráfego.

Deve-se enfatizar que o estudo de Norros [1], de predição de tráfego, considera que os parâmetros a , m e H são conhecidos, o que reduz o problema da predição do tráfego ao problema da predição do fBm.

Na verdade, estes parâmetros também devem ser estimados e comparados com os valores declarados pelo usuário no contrato de tráfego. O objetivo do Capítulo 6 é estudar o uso de filtros de Kalman para realizar a função de controle de tráfego da rede, não só estimando o tráfego a ser recebido como também os seus principais parâmetros.

No Capítulo 6 são analisados diversos casos. Demonstra-se que, infelizmente, não é possível usar um filtro de Kalman estendido (EKF) porque o sistema não é totalmente observável. A conclusão final é usar a técnica baseada em wavelets para estimar os parâmetros H e a , e usar um preditor de Kalman linear para estimar conjuntamente tanto o processo incremento de tráfego $i(n)$ como o parâmetro de taxa média do tráfego. Para isso, são apresentadas várias simulações. Demonstra-se que a estimação da taxa média do tráfego obtida com esta técnica proposta é superior à dada pelo leaky bucket.

Finalmente, no Capítulo 7 desenvolve-se uma técnica que permite fazer a decomposição em wavelets na forma sequencial, para blocos de N amostras, com uma complexidade de

apenas $O(\sqrt{N})$ ao invés de $O(N)$, se um número adequado K de amostras no bloco anterior é substituído por outro número de K amostras no bloco seguinte.

Demonstra-se que para os valores de N e L (comprimento dos filtros wavelet usados) de interesse prático, o intervalo possível de valores de K é bastante amplo, o que oferece flexibilidade para escolher o valor de K de acordo com os requisitos de uma certa aplicação e ao mesmo tempo trabalhar com uma baixa complexidade $O(\sqrt{N})$.

Esta técnica, combinada com a estimação baseada em wavelets do parâmetro de Hurst e da variância do fBm mencionada acima, constitui-se numa poderosa ferramenta para, em tempo real, acompanhar as alterações na dependência de longo prazo do sinal de tráfego.

A principal contribuição deste trabalho consiste na originalidade das técnicas desenvolvidas nos Capítulos 4, 5, 6 e 7 e na importância dos resultados obtidos, que são de utilidade não somente para análise e predição de tráfego auto-similar mas também para outras muitas aplicações com processos auto-similares.

Capítulo 2

Controle de Tráfego em Redes ATM

2.1. Introdução

O objetivo de um sistema de controle inteligente para Redes ATM, é controlar o tráfego da rede para evitar congestionamento que produz grandes atrasos e transbordamento de células nos buffers.

Em redes de comutação de circuitos é suficiente uma técnica de controle de admissão muito simples porque os recursos são reservados. Por exemplo, se um canal de 64kbits está disponível entre as partes, a conexão é estabelecida, caso contrário é rechaçada.

Em redes de comutação de pacotes, o controle do tráfego é mais complicado devido a que os recursos são compartilhados. A técnica usada é o controle de fluxo. Cada nó da rede pode regular o fluxo de tráfego em seus enlaces de entrada para reduzir ou interromper o tráfego de acordo com o nível do buffer.

O controle do tráfego em redes ATM apresenta os seguintes problemas [16]:

1. As fontes B-ISDN (“broadband integrated services digital network”) geram tráfego a taxas significativamente diferentes (Kbps a Mbps).
2. Uma única fonte pode gerar muitos tipos de tráfego (voz, dados, imagens) com diferentes características.
3. Além da perda de pacotes e bloqueio de chamada, as redes ATM devem lidar com a variação de atraso, com o atraso máximo e com o atraso diferencial (“skewness”).
4. Os diferentes serviços têm diferentes exigências de qualidade de serviço (QoS).
5. Para vários tipos de serviços, as características do tráfego ainda não podem ser modeladas ou entendidas completamente.
6. O grande valor do produto (faixa)x(atraso de propagação) diminui a eficiência das técnicas de controle reativo.
7. A grande velocidade de transmissão limita o tempo disponível nos nós para processamento.

O objetivo então é procurar o equilíbrio entre a eficiente utilização dos recursos e o cumprimento das exigências de QoS para os diversos serviços.

2.2. Técnicas de controle de tráfego propostas para redes ATM

O objetivo desta seção é descrever sucintamente as principais técnicas de controle de tráfego propostas para redes ATM. Os conceitos aqui apresentados, bem como os das próximas seções, serão necessários para uma melhor compreensão dos temas da tese abordados nos Capítulos 4, 5, 6 e 7.

– Gerenciamento de recursos.

Quando as características do tráfego na rede variam com o tempo, devem-se redefinir os VP (virtual path)'s e realocar suas faixas. O problema de se definir um conjunto de VP's e seus parâmetros para uma dada topologia de rede, ainda é um problema em aberto na literatura.

– Controle de admissão de chamada.

Quando uma solicitação de nova chamada é recebida na rede, deve-se decidir pela aceitação ou rejeição da chamada. Uma chamada é aceita se e somente se a rede tem suficientes recursos para prover a requerida QoS da nova chamada, sem alterar as QoS's das chamadas existentes. Para isso, duas perguntas devem ser respondidas:

- a) Como pode ser determinada a largura de faixa requerida pela nova conexão?
- b) Como é possível ter certeza que os QoS's das chamadas existentes não serão afetadas pela nova chamada?

A técnica utilizada deve:

- 1) Responder a essas perguntas.
- 2) Operar em tempo real.
- 3) Maximizar a utilização dos recursos da rede.

A pergunta a) ainda é, em grande parte, uma questão em aberto, mas vamos inicialmente supor que existem modelos tratáveis (do tipo Markoviano) para caracterizar fontes VBR.

– Superposição dos modelos das fontes.

Teoricamente, é possível desenvolver um modelo Markoviano da rede de comunicações e resolvê-lo numericamente com a nova conexão para determinar se a nova conexão deve ser aceita ou rejeitada. Na prática, é impossível resolver isto em tempo real, mesmo para redes pequenas (devido às atuais restrições de capacidade e processamento).

– Alocação de faixa.

Existem duas técnicas básicas para alocação de faixa:

- Multiplexagem determinística: é alocada a faixa de pico produzindo grande desperdício para tráfego muito esporádico ("bursty"), mas eliminando quase totalmente o congestionamento.
- Multiplexagem estatística: é alocada a faixa estatística da conexão (maior do que a taxa média e menor do que a taxa de pico). A dificuldade é o cálculo da faixa estatística para garantir as QoS's das conexões, porque isto depende das características próprias da nova conexão e das características das conexões já existentes na rede.

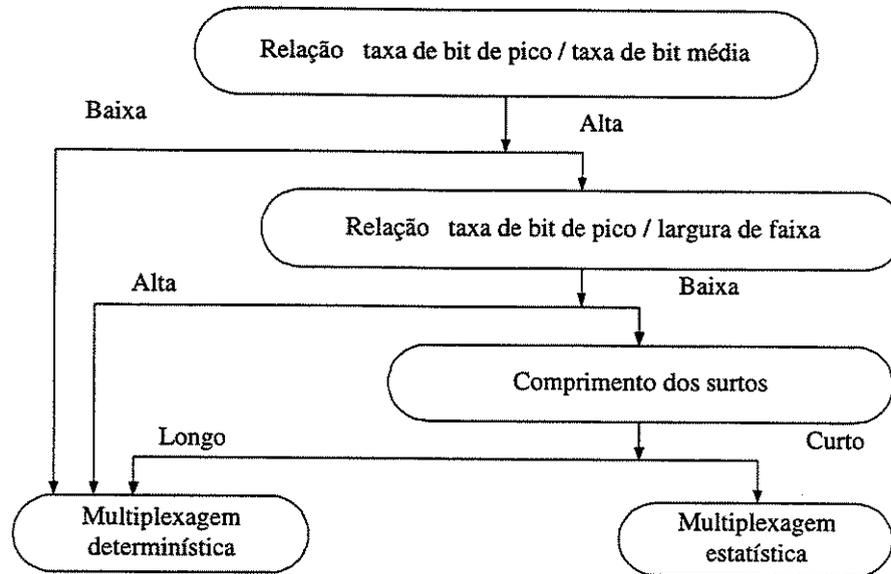


Figura 2.1: Multiplexagem determinística versus multiplexagem estatística.

A decisão entre multiplexagem determinística e estatística depende dos seguintes fatores (Figura 2.1):

1. Relação entre taxa de bit de pico e taxa de bit média: se esta relação é alta, é conveniente usar multiplexagem estatística.
2. Comprimentos de surtos: sejam duas conexões “bursty”, mostradas na Tabela 2.1 [16], com a mesma taxa de bit média e de pico e, portanto, iguais utilizações de fonte de 0.5 e que podem ser modeladas por IBP’s (Interrupted Bernoulli Process), sendo:

α : probabilidade de que um “slot” tenha uma célula válida, no período de atividade.

p : probabilidade de permanência no estado “on”.

q : probabilidade de permanência no estado “off”.

n_a : duração do período de atividade.

n_s : duração do período de silêncio.

A duração média dos períodos de atividade e silêncio é maior para a fonte 1 do que para a fonte 2. Portanto, usando “burstiness” $\equiv c^2$ (o coeficiente quadrático de variação dos tempos entre chegadas de células) temos que a fonte 1 é mais “bursty” do que a fonte 2 e, portanto, requer mais recursos da rede para igual grau de serviço (Cell Loss Priority).

3. Relação entre taxa de bit de pico da conexão e largura de faixa da linha: o tempo total que um pacote consome no nó é menor com ATM se um número de células do pacote já foi transmitido antes da chegada da última célula. É por isto que o ATM é muito eficiente: permite multiplexar muitas conexões com pequenas restrições

Parâmetro	Conexão 1	Conexão 2
α	1	1
p	0.75	0.5
q	0.75	0.5
$E(n_a) = E(n_s) = 1/(1-p)$	4 "slots"	2 "slots"
c^2	1.5	0.5

Tabela 2.1: Parâmetros das conexões 1 e 2.

nas filas dos nós intermediários. Isso é o esperado em B-ISDN onde a faixa da rede será muito maior do que a velocidade na interface de usuário.

2.3. Algoritmos de controle de admissão

O grande problema dos algoritmos de controle de admissão, é que eles alocam mais faixa do que a requerida (perda de recursos) ou menos faixa do que a requerida (perda de QoS). A seguir são apresentados e analisados alguns algoritmos de controle de admissão [16].

2.3.1. Aproximação Gaussiana

Uma aproximação Gaussiana é frequentemente utilizada para um cálculo simplificado nos algoritmos de controle de admissão.

– Sejam:

- (m_i, σ_i) : taxa de bit média e desvio padrão para a conexão i .
- A : variável aleatória GAUSSIANA que representa a taxa de bit agregada de n conexões.
- O problema é determinar a faixa c_o necessária para n conexões para que $P[A > c_o] \leq \epsilon$.
Pode-se demonstrar que

$$c_o = m + \alpha\sigma;$$

$$\alpha = \sqrt{2\ln(1/\epsilon) - \ln 2\pi}.$$

Procedimento de admissão de chamada:

Depois de receber um pedido de conexão com parâmetros (m_{n+1}, σ_{n+1}) sobre um enlace com parâmetros de tráfego agregado m e σ :

- 1) Calcular os novos parâmetros do tráfego agregado: $(m', \sigma') = (m + m_{n+1}, \sqrt{\sigma^2 + \sigma_{n+1}^2})$.
- 2) Calcular $c_o = m' + \alpha\sigma'$.
- 3) No caso $c_o \leq C$ (a faixa do enlace) aceitar a conexão, caso contrário rejeitar a conexão.

Dificuldades do método:

- 1) A suposição Gaussiana é aceitável somente se um grande número de conexões com parâmetros similares (m_i, σ_i) são multiplexadas. Entretanto, nas redes ATM os σ_i podem ser muito diferentes.
- 2) Todas as conexões são tratadas para a mesma exigência de perda de células, quando em verdade podem diferir significativamente.
- 3) O método não explora totalmente o ganho de multiplexagem estatística que pode ser obtido, devido a que não considera o tamanho do buffer.

2.3.2. Reserva de fila rápida.

- Caracterização da fonte “bursty” por modelo Markoviano de 2 estados.
- Quando uma fonte é ativa, um certo número de “slots” são reservados na fila do enlace. Ao terminar o período de atividade, os “slots” reservados são liberados. Podem usar-se células marcadas para indicar a transição entre períodos de atividade e silêncio.

- Sejam:

B : número de “slots” disponíveis na fila do enlace.

B_i : número de “slots” a serem reservados para conexão i .

b_i : número de “slots” em uso pela conexão i .

s_i : estado da conexão i (atividade ou silêncio).

Procedimento do método:

- 1) Diminuir b_i com cada transmissão de uma célula de conexão i da fila do enlace.
- 2) Se a conexão está no estado de silêncio e:
 - a) Se a célula recebida indica o começo do período de atividade:
 - (1) Se $B_i > B$ então descartar a célula;
 - (2) Se $B_i \leq B$ então:
 - (a) Ressetar o relógio;
 - (b) Chavear s_i de silêncio a atividade;
 - (c) $B = B - B_i$;
 - (d) Se $b_i < B_i$, então:
 - (i) $b_i = b_i + 1$;
 - (ii) Colocar a célula na fila como uma célula não marcada.
 - (e) Se $b_i = B_i$, então colocar a célula na fila como uma célula marcada.
 - b) Se a célula recebida não indica o começo de um período de atividade, então descartar a célula.
- 3) Se a conexão está no estado de atividade, então:
 - a) Se a célula recebida não indica o fim do período de atividade, então:
 - (1) Ressetar o relógio;
 - (2) Se $b_i < B_i$, então:
 - (a) $b_i = b_i + 1$;

(b) Colocar a célula na fila como uma célula não marcada.

(3) Se $b_i = B_i$, então colocar a célula na fila como uma célula marcada.

b) Se a célula recebida indica o fim do período de atividade ou o relógio expira, então:

(1) Chavear s_i de atividade a silêncio;

(2) $B = B + B_i$.

Observações:

- Ao receber a célula de começo de surto e se o espaço requerido de fila não está disponível, então todas as células do surto são perdidas. Isto é adequado só para tráfego sensível à perda.

- Para tráfego tolerante à perda, cada célula do surto menos a última é transmitida como uma célula de começo de surto.

- Uma técnica para introduzir o tipo de célula no cabeçalho é usar o bit CLP (Cell Loss Priority) da seguinte maneira:

i) No estado de silêncio:

CLP=1: surto de uma célula ou célula de baixa prioridade.

CLP=0: célula de começo de surto

ii) No estado de atividade:

CLP=1: célula de fim de surto.

CLP=0: célula intermediária de surto.

Notar que com este esquema as células de baixa prioridade não podem ser enviadas dentro do surto.

Sejam:

λ_i : taxa de bit de pico da conexão i ;

μ_i : taxa de bit média da conexão i ;

x_i : variável aleatória que representa o número de “slots” necessários para conexão i na fila. No estado de silêncio $x_i = 0$. No estado de atividade $x_i = B_i$.

Assim,

$$p_i = P[x_i = B_i] = \frac{\mu_i}{\lambda_i};$$
$$P[x_i = 0] = 1 - \frac{\mu_i}{\lambda_i}.$$

- Considerar um enlace com n conexões e com demandas de fila x_1, x_2, \dots, x_n . A demanda de fila total é dada por

$$X = \sum_{i=1}^n x_i.$$

- Para decidir se uma nova conexão pode ser multiplexada com as existentes, é calculada a probabilidade de requerer-se mais “slots” do que os disponíveis na fila (L), dada por $p = \text{Probabilidade de demanda excessiva} = P[X > L]$.

- No caso dessa probabilidade ser menor do que um valor predefinido, a nova conexão é aceita, caso contrário é rejeitada.

Dificuldade deste método:

Sejam duas conexões que requerem todos os “slots” da fila quando estão ativas e que $p_1 = 0.9$ e $p_2 = 0.01$. Com as duas conexões, resulta $p = 0.009$. Portanto, a conexão 2 será aceita, mas somente em 10% de seus surtos a conexão 2 poderá transmitir.

Isto pode ser evitado trabalhando-se com a probabilidade:

$p =$ Probabilidade de que o número de “slots” requeridos não esteja disponível quando a fonte i transmite seu surto,

em lugar da probabilidade de demanda excessiva.

2.3.3. Capacidade equivalente

- Usa um modelo de conexão para cada fonte de dois estados “on/off”, com durações de cada estado independentes e com distribuições exponenciais. No estado ativo a fonte transmite à sua taxa de pico e no período de silêncio a taxa é zero. É baseado em um modelo de fluxo.

Sejam:

Z : variável aleatória que representa a taxa de bit de entrada no buffer do enlace.

A : velocidade de transmissão do enlace.

R : taxa de bit de pico da conexão.

b : duração média do período ativo.

ρ : utilização da fonte (probabilidade de que a fonte esteja ativa).

μ : taxa de transição de saída do estado ativo ($\mu = 1/b$).

λ : taxa de transição de saída do estado de silêncio ($\lambda = \rho / [b(1 - \rho)]$).

c : velocidade do enlace requerida por uma conexão (capacidade equivalente).

X : tamanho do buffer.

A dinâmica da fila do enlace é a seguinte:

1. Se $z < A$ e:

a. A fila está vazia, então permanece vazia;

b. A fila não está vazia, então seu conteúdo diminui a uma taxa constante de $A - z$.

2. Se $z = A$, então o conteúdo da fila não se altera.

3. Se $z > A$ e:

a. A fila não está cheia, então seu conteúdo incrementa a uma taxa constante de $z - A$.

b. A fila está cheia, então existe perda de células a uma taxa constante de $z - A$.

Pode-se demonstrar que para uma conexão com parâmetros (R, ρ, b) como entrada em uma fila de tamanho X , a velocidade do enlace deve ser:

$$c = R \frac{y - X + \sqrt{(y - X)^2 + 4X\rho y}}{2y},$$

$$y = \ln(1/\epsilon)b(1 - \rho)R.$$

Neste modelo, a capacidade total para n conexões multiplexadas é

$$C = \sum_{i=1}^n c_i.$$

Observação:

- C superestima a requerida capacidade para o tráfego agregado porque não é levada em consideração a interação entre conexões individuais. É possível melhorar o modelo fazendo a combinação com a aproximação Gaussiana:

$$C = \min[m + \alpha\sigma, \sum_{i=1}^n c_i].$$

Procedimento do método:

1. Dados os parâmetros de uma nova conexão $(R_{n+1}, \rho_{n+1}, b_{n+1})$ e os valores atuais do tráfego agregado m, σ e C :
 - a. Calcular c_{n+1} .
 - b. Calcular os novos valores de m' e σ' .
2. Calcular o novo valor de C' ; se é menor do que a taxa do enlace, aceitar a nova conexão, caso contrário rejeitar a conexão.

2.3.4. Aproximação de fluxo para a probabilidade de perda de células.

- Considera uma fonte VBR modelada como “on-off”. Não é considerada nenhuma suposição sobre as distribuições dos períodos de atividade e silêncio.

Sejam:

- p_i : taxa de pico de células.
- m_i : taxa média de células.
- N fontes independentes com taxa r_i .
- C células/seg: taxa de transmissão do enlace.
- M : tamanho do buffer.
- $R = \sum_{i=1}^N r_i$: taxa de células do tráfego agregado.
- q : número de slots disponíveis na fila.
- L : taxa de perda de células.

$$L = \sum_{x>C} (x - C) Pr(R = x) [1 - Pr(q > 0 / R = x)].$$

Um supremo de L é dado por

$$OF = \sum_{x>C} (x - C)Pr(R = x).$$

Portanto, um limitante superior da probabilidade de perda de células é dado por

$$PV = \frac{OF}{\sum_{i=1}^N m_i}.$$

PV pode ser usado para fazer admissão de conexão.

Observações:

- Não usa o tamanho do buffer (pois trabalha com o supremo de L) e, portanto, isto limita o ganho de multiplexagem estatística que pode ser obtido.
- Não leva em consideração as possíveis diferenças entre as exigências de perda de células das diferentes conexões. Isto pode ser feito utilizando

$$OF_n = \sum_{x>C} (x - C)Pr(R = x) \frac{p_n}{x}, n = 1, \dots, N.$$

$$PV_n = \frac{OF_n}{m_n}.$$

2.3.5. Método não paramétrico.

- Não precisa conhecer as distribuições do processo de chegada.

Sejam:

- C células/seg: velocidade do enlace.
- $1/C$ seg: tempo do slot de célula.
- R'_i, a'_i : taxas de pico e média de células da conexão i .
- r slots: período de observação da fila do enlace.
- $R = rR'_i/C, ra'_i/C$: números de células máximo e médio que chegam durante o período de observação r .

Procedimento de admissão de conexão: calcular a probabilidade de perda de células com a conexão $n + 1$: $U(n + 1, r)$. Se é menor que a perda desejada aceitar; caso contrário, rejeitar a nova conexão.

2.3.6. Aproximação de tráfego pesado.

- É baseada no comportamento assintótico da cauda (“tail”) da distribuição do comprimento de fila. Considera tempo de serviço constante e chegadas Markovianas. Então, para i suficientemente grande tem-se

$$P(\text{comprimento da fila} > i) = \alpha(1/z^*)^i.$$

A probabilidade $P(\text{comprimento da fila} > i) = p$ pode ser usada para fazer admissão de conexão.

O problema é determinar as constantes α e z^* . Aproximações dessas podem ser obtidas.

2.3.7. Comparação das técnicas de admissão de conexão

Foram comparadas na literatura 3 técnicas: capacidade equivalente, método não paramétrico, e aproximação de tráfego pesado. As conclusões são as seguintes [16]:

- O método da capacidade equivalente fornece o maior ganho de multiplexagem estatística para uma dada CLR.
- A aproximação de tráfego pesado é eficaz quando a relação (tamanho de buffer)/(comprimento de surto) aumenta; se o tamanho do buffer é pequeno, então coincide com a alocação de pico (alocação determinística).
- Com o método não paramétrico o ganho de multiplexagem estatística é menos sensível ao tamanho do buffer.
- Quando o tamanho do buffer tende ao infinito, tanto o método da capacidade equivalente como o método da aproximação de tráfego pesado tendem ao mesmo ganho estatístico.

2.3.8. Conformação do tráfego

A quantidade de recursos necessários para uma conexão é maior do que a taxa média e menor do que a taxa de pico.

É possível reduzir a taxa de pico de uma conexão colocando-se numa fila as células antes de ingressar na rede. O problema é que esta técnica não pode ser usada para tráfego sensível ao atraso e que em geral a redução possível de taxa de pico é limitada pela quantidade de atraso tolerado.

2.4. Policiamento de tráfego

A admissão de conexão não é suficiente para evitar congestionamento na rede devido a que os usuários podem não cumprir com os parâmetros negociados, pois

1. Os usuários não conhecem ou subestimam os requisitos da conexão.
2. O equipamento do usuário não funciona bem.
3. Os usuários podem intencionalmente subestimar suas exigências para pagar menos.
4. Os usuários podem intencionalmente tentar “romper” a rede.

Dessa forma, a rede deve fazer a monitoração das fontes a fim de garantir o cumprimento dos parâmetros negociados. Esta função é chamada de *policimento de tráfego ou controle de parâmetros de usuário (UPC)*. O UPC é realizado na UNI (“user-network interface”) e deve detectar o mais rápido possível as fontes que não cumprem o contrato, mas por outro lado deve ser transparente para as fontes que cumprem.

As possíveis ações da função de policimento para as fontes que não cumprem são:

1. Descartar células que não cumprem.
2. Atrasar células que não cumprem, de tal forma a forçarem que cumpram o contrato.
3. Marcar células que não cumprem para serem tratadas de forma diferente das demais quando ocorrer congestionamento.
4. Adaptivamente controlar o tráfego informando à fonte quando ela começa a violar o contrato de tráfego.

A dificuldade maior hoje consiste em que os parâmetros de tráfego necessários para caracterizar uma fonte é ainda, em grande parte, um problema em aberto. Isto é devido ao fato, recentemente demonstrado em [12], de que as aplicações que serão os grandes tributários da rede ATM geram tráfego auto-similar. Portanto, a especificação de quais parâmetros devem ser controlados também é um problema em aberto.

Assim mesmo, várias técnicas de policimento já foram propostas para o controle da taxa de pico, da taxa média e do comprimento do período ativo. Entre elas se destacam o “Leaky Bucket” e as técnicas baseadas em janelas, que serão discutidas a seguir.

2.4.1. “Leaky Bucket”

- O princípio de funcionamento dessa técnica é simples. Uma célula, antes de entrar na rede, deve obter um “token”. Os “tokens” são gerados a uma taxa constante e são colocados em um “token pool”. O tamanho do “token pool” é limitado e “tokens” gerados quando o “token pool” está cheio são descartados. O tamanho do “token pool” impõe um limitante superior ao comprimento do surto. Por exemplo, para controlar um VP (Virtual Path) de taxa de pico de 10 Mbps, é requerida uma taxa de geração de “tokens” de um em cada $42.4 \mu \text{ seg.}$, e o tamanho do “pool” pode compensar variações de taxa.

Existem basicamente quatro versões do “leaky bucket”, conforme mostradas na Figura 2.2 abaixo.

- No caso da Figura 2.2 (a), as células que chegam quando não existem “tokens” são descartadas.
- No caso da Figura 2.2 (b), é introduzida uma fila para que as células aguardem quando não existem “tokens”.
- No caso das Figuras 2.2 (c), (d), como alternativa ao descarte de células, são marcadas com o bit CLP em 1 e ingressadas na rede para incrementar a utilização dos recursos da rede.

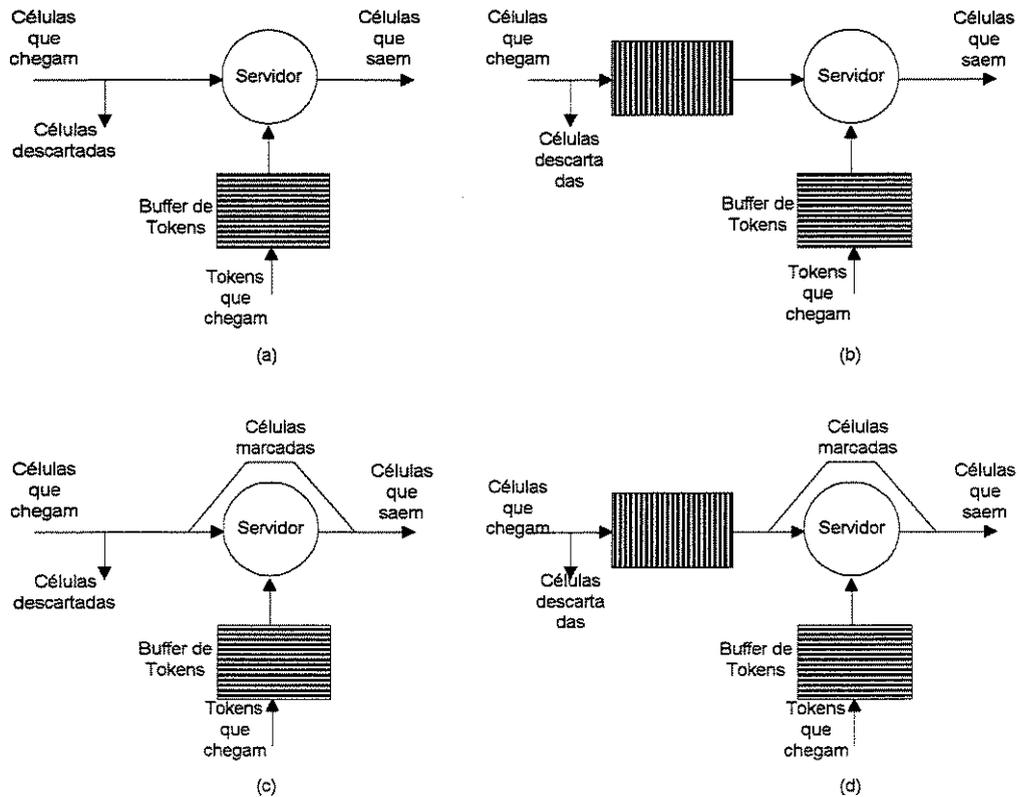


Figura 2.2: Quatro versões do leaky bucket.

O controle da taxa média é mais complicado. Fazer a taxa de geração de “tokens” igual à taxa média de células não é suficiente; um surto de células à taxa de pico pode ser descartado ou marcado, embora cumpra o contrato.

A dificuldade é definir o tempo T sobre o qual a taxa média estatística é conhecida. Para que a operação do “leaky bucket” seja transparente às fontes que cumprem o contrato, a probabilidade de perda de células no “leaky bucket” deve ser menor que 10^{-10} .

No caso das figuras b, d , o atraso que produz-se deve ser limitado. Este pode ser reduzido aumentando a taxa de geração de “tokens” e o tamanho do “token pool”, mas a eficiência do controle de tráfego é reduzida.

2.4.2. Técnicas baseadas em janelas.

– “Jumping window” (JW).

Esta técnica impõe um limitante superior m ao número de células aceitas de uma fonte durante um intervalo de tempo fixo T , chamado janela. As células em excesso podem ser descartadas ou marcadas com o bit CLP. A taxa de células oferecida à rede é $\lambda_p = m/T$. Portanto, temos duas variáveis independentes mas só uma equação. Com $m = 1$ e $T = N$ (o tempo de geração de “token”) temos a técnica de “leaky bucket”.

Em geral, a taxa de pico ou a taxa média pode ser controlada (mas não as duas) escolhendo m , e portanto $T = m/\lambda_p$, onde λ_p é a taxa a ser controlada. A escolha de m não pode ser totalmente arbitrária. Por exemplo, $m = 5, T = 10$ produzem a mesma taxa que $m = 10, T = 20$, mas quando T aumenta, gasta-se mais tempo para detectar que a taxa controlada é excedida e quando T diminui o esquema não é transparente aos usuários que cumprem os parâmetros. Considere uma fonte que gera de forma determinística 10 células durante seu período ativo e permanece inativa durante um tempo equivalente à transmissão de 30 células. Então, um sistema com $m = 1$ e $T = 4$ leva a um descarte de 3 em cada 4 células.

A probabilidade de que o sistema JW perca células de uma fonte que cumpre os parâmetros, é dada por

$$pv = \frac{\sum_{i=1}^{\infty} ix_{m+i}(T)}{\sum_{i=1}^{\infty} ix_i(T)},$$

onde $x_i(T)$ é a probabilidade de que sejam geradas i células durante o intervalo T .

A equação acima pode ser usada para encontrar o parâmetro T a ser usado, tal que pv seja suficientemente baixa (sistema transparente).

Existem algumas variantes da técnica JW que são apresentadas a seguir.

- “Triggered jumping window mechanism” (TJW). Neste caso a janela está sincronizada com a atividade da fonte: a janela começa com a primeira célula do período de atividade, portanto as janelas não necessariamente são consecutivas no tempo.
- “Moving window” (MW). É similar a JW mas pode ser interpretada como uma janela que é móvel ao longo do tempo.
- “Exponentially weighted moving average” (EWMA). É similar a JW. A única diferença é que o número de células aceitas durante uma janela i varia com o tempo e é uma soma exponencialmente ponderada do número de células aceitas na janela precedente e o número médio de células. Esta variação dá mais flexibilidade à técnica quanto à variabilidade (“burstness”) do tráfego.

2.4.3. Eficiência dos métodos baseados em janela

Em geral, o “leaky bucket” é superior aos métodos baseados em janela e é recomendado pela ITU-T e ATM Forum como técnica de policiamento de tráfego.

O “leaky bucket” e “EWMA” parecem ser as técnicas mais efetivas que podem lidar com flutuações de curto prazo no tráfego [16].

Na realidade, nenhuma técnica resolve o problema de controle de todos os parâmetros que caracterizam uma fonte: taxa de pico e taxa média, duração de surto, distribuições dos períodos de atividade e silêncio.

Alguns problemas são:

- A taxa média de células deve ser estimada a partir de um número pequeno de amostras, o que produz erros grandes e portanto decisões de policiamento incorretas.

- O tempo para reagir aumenta com o tamanho da janela.
- Devido à incerteza na caracterização dos parâmetros da fonte, os parâmetros das técnicas de policiamento não são adequadamente escolhidos, o que limita a eficiência do método.

2.5. Descarte seletivo

O bit CLP pode ser usado pelo usuário para estabelecer prioridades. Por exemplo, células que têm sincronização de vídeo são transmitidas como de alta prioridade. O bit CLP pode ainda ser usado para marcar células recebidas pela rede para compensar incertezas devido à natureza estatística do tráfego e permitir maior utilização dos recursos. Existem duas técnicas de descarte seletivo:

2.5.1. “Push-out”

- Células de alta e baixa prioridades são aceitas na rede sempre que existir espaço disponível na fila.
- Uma célula de baixa prioridade que chega quando a fila está cheia é descartada.
- Uma célula de alta prioridade que chega quando a fila está cheia, será descartada se não há células de baixa prioridade na fila. No caso de haver célula de baixa prioridade, esta é substituída pela célula de alta prioridade (a menos que a célula esteja sendo transmitida).
- A principal desvantagem desta técnica é sua complexidade de implementação e a dificuldade de assegurar o sequenciamento das células; a fila deve ser gerenciada não como FIFO.

2.5.2. “Threshold”

- Um valor de limiar menor do que a capacidade da fila é usado para regular a ocupação da fila entre células de alta e baixa prioridade.
 - Os dois tipos de células são admitidos na fila; no entanto, o número total de células é menor do que o limiar.
 - Quando o número de células supera o limiar, todas as células de baixa prioridade que chegam são descartadas até que a capacidade seja menor do que o limiar.
 - Células de alta prioridade continuam ingressando na fila sempre que existir espaço disponível.
 - O principal problema é a determinação do valor do limiar; o qual é um compromisso entre descarte desnecessário de células marcadas e degradação da performance de células de alta prioridade. O limiar pode ser adaptativo.
- A vantagem em relação ao “push-out” é que a técnica do limiar pode ser implementada como FIFO e portanto com menor complexidade.

2.6. Mecanismos de controle reativo de congestionamento.

Embora as técnicas preventivas reduzam a probabilidade de transbordamento de filas, não é possível eliminar a perda de células totalmente devido à natureza estatística do tráfego.

O principal objetivo das técnicas reativas é evitar que períodos momentâneos de sobrecarga convertam-se em períodos sustentados, evitando-se que a vazão da rede tenda a zero.

O problema das técnicas reativas em redes ATM é o grande valor do produto (atraso de propagação) \times (faixa), devido à alta velocidade dos enlaces. Por exemplo, o número de células em trânsito sobre um enlace de 5000 km OC-3 é maior do que 9000 !!, supondo um atraso de propagação de 5μ seg/km.

O tamanho requerido das filas para guardar um número grande de células pode ser excessivamente caro, ou o atraso que elas produzem pode não ser tolerado. O instante em que a fonte é informada do congestionamento pode ser tarde demais para uma reação efetiva. Portanto, os mecanismos reativos não são tão eficazes em redes ATM como em redes de pacotes de baixa velocidade. Todavia, mecanismos reativos são necessários como segurança e podem potencialmente ser usados para incrementar a utilização dos recursos em relação ao uso de técnicas preventivas que geralmente alocam mais recursos do que o realmente requerido.

2.6.1. Técnicas de notificação ao nó final.

- Estimação pelos nós finais.
 - O nó fonte transmite periodicamente células de prova com a indicação do instante de tempo de transmissão.
 - Estas células são usadas pelo nó destino para medir o atraso das células.
 - Quando o nó destino detecta congestionamento, ele notifica ao nó fonte para ajustar sua taxa.

A principal desvantagem é o tráfego extra gerado na rede.

- Notificação de congestionamento explícita “backward” (EBCN).
 - Cada nó monitora a ocupação das filas de seus troncos.
 - Quando é alcançado um dado valor de limiar, envia-se uma célula especial aos nós fonte de todas as conexões que passam pelo nó congestionado.
 - Este processo continua até que pode chegar eventualmente ao nó do usuário.

Não foi adotada nos padrões devido à complexidade de processamento nos nós intermediários de células especiais.

- Notificação de congestionamento explícita “forward” (EFCN).
 - Cada nó monitora a ocupação das filas de seus troncos.
 - Quando é alcançado um dado valor de limiar, todas as células que passam através desse tronco são marcadas até que o congestionamento termina (o tamanho do buffer é menor do que outro limiar).

- Os nós receptores não reagem imediatamente à indicação, mas são calculadas estatísticas e no caso de determinar-se que o congestionamento é sustentado, é enviada uma notificação à fonte.

Existe um bit no cabeçalho da célula para esta função.

2.6.2. Controle adaptativo de taxa.

- A taxa é variada pela fonte de acordo com a informação de congestionamento.

- Essa informação pode ser obtida com EFCN, onde o nó destino é que efetivamente controla a taxa da fonte.

2.6.3. Negociação de parâmetros na chamada.

Para alguns tipos de aplicações, é requerida somente a transmissão de poucos surtos, como por exemplo, interconexão de LAN's, e-mail, transferência de pequenos arquivos, etc. O objetivo desta técnica é minimizar o "overhead" do estabelecimento da conexão. O ATM é orientado para conexão, mas uma alternativa é estabelecer VP's entre os correspondentes nós finais e, portanto, definir uma rede virtual entre os usuários desses serviços. A principal desvantagem é o uso ineficiente dos recursos da rede. Uma alternativa é a reserva rápida de faixa. Durante o estabelecimento da chamada, é determinado o caminho da fonte ao destino e depois uma célula especial é enviada pelo caminho para reservar a faixa de pico da conexão. Cada nó pode aceitar, negar ou reduzir a faixa requerida. No caso de aceitar-se o pedido de conexão, o surto é transmitido. A faixa reservada é liberada em cada nó imediatamente depois da última célula. Com este método, a fase de terminação da conexão é eliminada e o "overhead" durante o estabelecimento é reduzido.

2.6.4. Codificação dinâmica de fonte.

Quando é recebida uma indicação de congestionamento pela fonte, a taxa pode ser reduzida ou interrompida temporariamente. Para tráfego não sensível ao atraso, as células podem ser atrasadas na fila da fonte. Para tráfego em tempo real sensível ao atraso, isto não é aceito. Uma opção é usar o bit CLP; por exemplo, em tráfego de vídeo os quadros com informação de sincronização são transmitidos com mais alta prioridade do que os que contêm a diferença em relação ao quadro de referência. Outra opção é incrementar o passo de quantização, de forma a reduzir a taxa da fonte.

A técnica EFCN pode ser usada para informar às fontes sobre o congestionamento da rede.

A eficiência desta técnica e, portanto, a de codificação dinâmica de fonte, é limitada pelo grande produto (atraso de propagação)x(faixa).

2.7. Resumo

- O objetivo dos algoritmos de controle é atingir uma alta utilização dos recursos da rede e cumprir com os parâmetros de QoS.
- As conhecidas técnicas de controle de congestionamento usadas nas redes de pacotes não são adequadas em redes ATM, devido ao grande produto (atraso de propagação) \times (faixa).
- Os protocolos usados devem ser muito simples para permitir sua implementação em alta velocidade.
- Como consequência do grande número de células em trânsito, são necessárias filas muito grandes. Isto pode ser usado para tráfego não sensível ao atraso, mas não é aceito para tráfego em tempo real.
- Técnicas preventivas têm obtido limitado êxito, devido ao fato de serem muito sensíveis aos parâmetros de caracterização da fonte, o que é ainda um problema em aberto.
- O problema é maior devido à existência de diferentes aplicações e variadas exigências de QoS.

Capítulo 3

Análise do Leaky Bucket

O leaky bucket é um contador que é incrementado cada vez que uma célula é gerada pela fonte e é decrementado periodicamente com uma adequada taxa de drenagem. Quando uma célula chega e o contador está no seu limiar N , a célula é descartada ou marcada.

O Leaky Bucket pode ser modelado como uma fila $G/D/1/N$ com espaço finito N . Cada conexão tem seu próprio contador, que é inicializado no “call setup”. No diagrama mostrado na Figura 3.1, nenhuma célula é colocada na fila e o fluxo de células aceitas não é alterado pelo Leaky Bucket.

Os dois parâmetros de controle do Leaky Bucket são:

- A taxa de drenagem a .
- A capacidade do buffer N em células (ou M em bits).

Vamos supor para o processo chegada de células o padrão da Figura 3.2, correspondente a uma fonte “bursty” do tipo “on/off”. Cada surto tem uma duração Z (variável aleatória) e uma taxa de bit de b bits/seg (taxa de bit de pico). A duração do período de silêncio é a variável aleatória L .

A fonte pode ser caracterizada pelo seguinte conjunto de parâmetros:

- A taxa de bit de pico (do surto): b
- A duração média do surto: h
- A duração média do silêncio: k

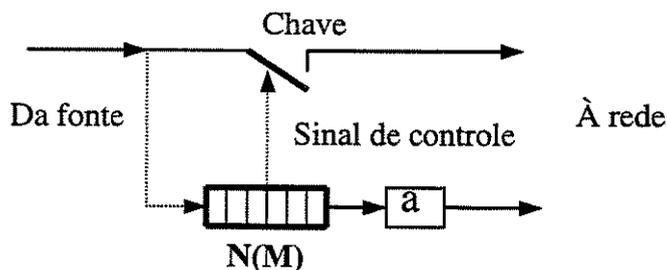


Figura 3.1: O leaky bucket como um dispositivo de policiamento de tráfego.

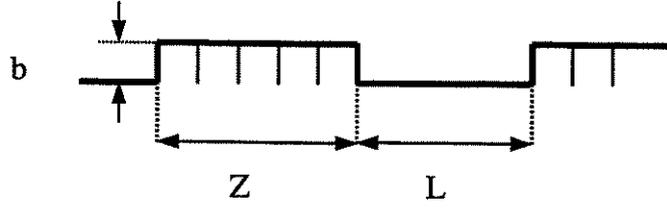


Figura 3.2: O processo chegada de células para uma fonte “on-off”.

- O grau de surto da fonte: $\beta = (h + k)/h$
- A taxa de bit média: $m = bh/(h + k)$
- A duração média do ciclo: $c = h + k$
- O comprimento médio do surto em bits: $t = bh$

3.1. O método de fluxo

Durante um surto de duração Z o contador do Leaky Bucket é incrementado pelo número (Zb/l) de células que chegaram e é decrementado pelo número (Za/l) de células atendidas, onde l é o comprimento da célula em bits. Portanto, ao final do surto o contador é incrementado por

$$Q = \frac{Z(b - a)}{l}.$$

Analogamente, durante o período de silêncio de duração L o contador, se não for nulo, é decrementado por

$$R = \frac{La}{l}.$$

Deve-se observar que Q e R deveriam ser variáveis aleatórias discretas.

A aproximação do método de fluxo considera o fluxo de bit como uma variável aleatória contínua e permite obter uma expressão explícita para a probabilidade de perda de células B no Leaky Bucket. Considerando distribuições exponenciais para as variáveis aleatórias Z e L , pode-se demonstrar que B é dada por [9]

$$B = \frac{b - a}{b} \frac{\lambda_1 - \lambda_2}{\lambda_1 e^{(\lambda_1 - \lambda_2)M} - \lambda_2}, \quad (3.1)$$

onde

$$\lambda_1 = \frac{1}{h(b - a)},$$

$$\lambda_2 = \frac{1}{ka}.$$

Parâmetro	“Packet voice”	“Still picture”
b	32 Kb/s	2 Mb/s
h	352 ms	500 ms
k	650 ms	11000 ms
m	11.2 kb/s	87 kb/s
β	2.85	23

Tabela 3.1: Características dos serviços.

a/b	“Packet voice”		“Still picture”	
	a/m	N	a/m	N
0.990	2.83	17	22.76	388
0.900	2.57	60	20.69	4369
0.800	2.29	135	18.39	9120
0.700	2.00	202	16.09	14080
0.600	1.71	342	13.79	19253
0.500	1.43	644	11.49	24704
0.400	1.14	1862	9.19	30609
0.360	1.03	—	8.28	33173
0.200	—	—	4.60	46819
0.100	—	—	2.30	72199
0.045	—	—	1.03	1124200

Tabela 3.2: Limiar N do contador Leaky Bucket para $B = 10^{-9}$.

3.2. Análise do desempenho do Leaky Bucket com o método do fluxo.

No método exato Q e R são variáveis aleatórias discretas, ao passo que no método de fluxo esses estados são considerados contínuos. Nas discussões que se seguem, h e k foram truncadas de maneira que Q e R tomam valores discretos.

Vamos considerar dois padrões “bursty” de chegadas de células, correspondentes aos serviços “Packet voice” e “Still picture”. Os serviços são caracterizados pelos parâmetros mostrados na Tabela 3.1 [9].

Na Tabela 3.2 [9] é dado o limiar N (em células) do contador Leaky Bucket correspondente a uma probabilidade de perda de células de 10^{-9} , para alguns valores da taxa de drenagem a .

Observa-se na Tabela 3.2 que o número de estados aumenta fortemente quando o grau de surto do serviço é grande (tráfego muito “bursty”, como é o caso do serviço “Still Picture”) ou quando a taxa de drenagem aproxima-se da taxa de bit média m da fonte. Portanto, o modelo exato não pode ser usado nestes casos para analisar o desempenho do Leaky Bucket.

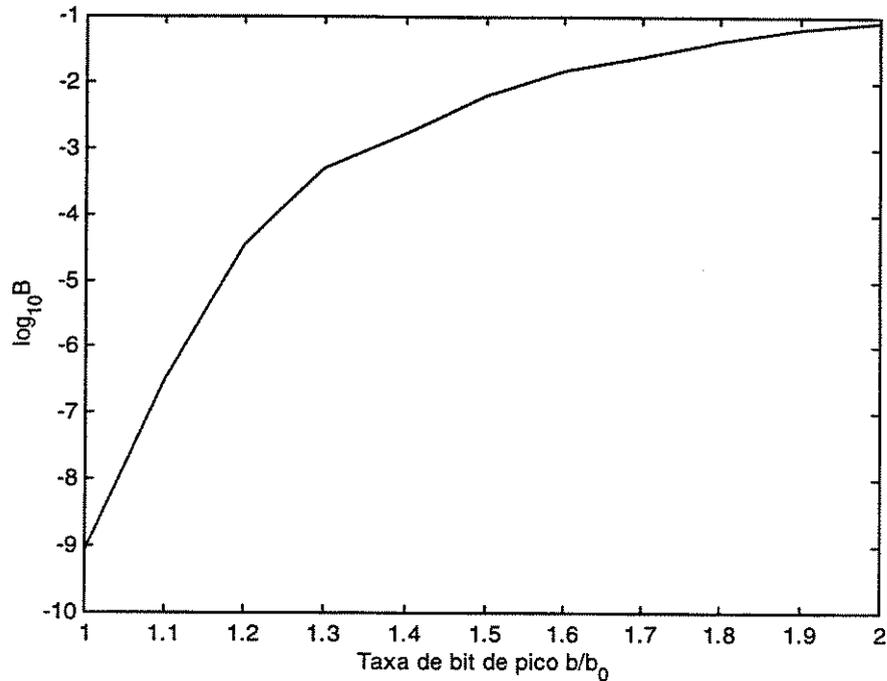


Figura 3.3: Probabilidade de perda B em função da taxa de bit de pico b/b_0 para o serviço “Packet Voice” ($a/b_0 = 0.8$, $N = 135$).

Na Figura 3.3 a probabilidade de perda de células B , dada pelo método de fluxo, é mostrada em função da taxa de bit de pico da fonte b , com M , h e k constantes.

Na próxima seção vamos utilizar a expressão de B para analisar o desempenho do Leaky Bucket como método de policiamento.

3.3. Seletividade do Leaky Bucket.

O Leaky Bucket pode ser considerado como um dispositivo que faz uma prova estatística sobre as células que são oferecidas à rede. A hipótese de prova é que os parâmetros b (taxa de bit de pico), h (duração média do surto) e k (duração média do silêncio) estão dentro dos limites contratados no “call setup” ($b \leq b_0$, $h \leq h_0$, $k \geq k_0$).

Uma curva ideal de probabilidade de perda para a função de policiamento deve levar em consideração dois requisitos:

1. Quando a hipótese é verdadeira (a fonte é bem comportada), o dispositivo de policiamento não deve introduzir perda de células (transparência).
2. Quando a hipótese de prova é falsa (uma fonte supera os parâmetros contratados), todas suas células deveriam ser descartadas ou marcadas.

Caso	Variável principal	Constantes	Região de controle
A	b	h, k, c, β	$a \approx m_0, b_0$
B	h	k, b	$a \approx m_0$
C	k	h, b, t	$a \approx m_0$
D	β	m, c, t	$a \approx b_0$
E	c	m, b, β	?

Tabela 3.3: Comportamentos de usuário.

O comportamento ideal acima não pode ser obtido com dispositivos práticos de policiamento. Contudo, o desempenho de uma técnica de policiamento pode ser avaliada através da curva da probabilidade de perda B como uma função dos parâmetros da fonte. O dispositivo é bom se a probabilidade de perda é baixa quando a hipótese de prova é verdadeira, e a probabilidade de perda é alta quando a hipótese de prova é falsa.

No caso do leaky bucket, podemos variar as características da prova ajustando os dois parâmetros de controle: a capacidade do buffer M e a taxa de drenagem a .

A capacidade do buffer M pode ser determinada se a é fixada para uma certa probabilidade de perda B_0 desejada, quando os parâmetros são os nominais, isto é, $b = b_0$, $h = h_0$, $k = k_0$. Uma expressão explícita para M pode ser obtida de (3.1)

$$M = \frac{1}{(\lambda_1 - \lambda_2)} \ln \left[\frac{(1 - \frac{a}{b})(\lambda_1 - \lambda_2) + B\lambda_2}{B\lambda_1} \right].$$

Vamos analisar 5 comportamentos de usuário, conforme Tabela 3.3. Nos casos A, B e C variam-se, respectivamente, a taxa de bit de pico, a duração média do surto e o período médio de silêncio. No caso D é variada a taxa de bit de pico, juntamente com a duração do surto, de tal maneira que a taxa de bit média é constante. No caso E as durações do surto e de silêncio são variadas na mesma proporção e de forma a manter a taxa de bit média constante.

O conteúdo principal da Tabela 3.3 é a região de controle da taxa de drenagem a , onde a seletividade do Leaky Bucket é boa para cada comportamento de usuário. A região é determinada observando-se a forma da curva de B como uma função dos parâmetros da fonte para cada valor da taxa de drenagem a .

Um exemplo é mostrado na Figura 3.4, onde a região efetiva de controle é dada por $a/b_0 \approx 0.045$, ou seja, $a \approx m_0$ (pois a curva de seletividade é mais pronunciada).

Na Tabela 3.4 são mostrados os valores de N , usados nas curvas da Figura 3.4, para cada correspondente valor de a/b_0 para se obter uma probabilidade de perda de células de 10^{-9} quando $h = h_0$.

A seguir vamos considerar três situações correspondentes ao controle dos parâmetros mais importantes, que são a taxa de bit de pico, a taxa de bit média e a duração do surto.

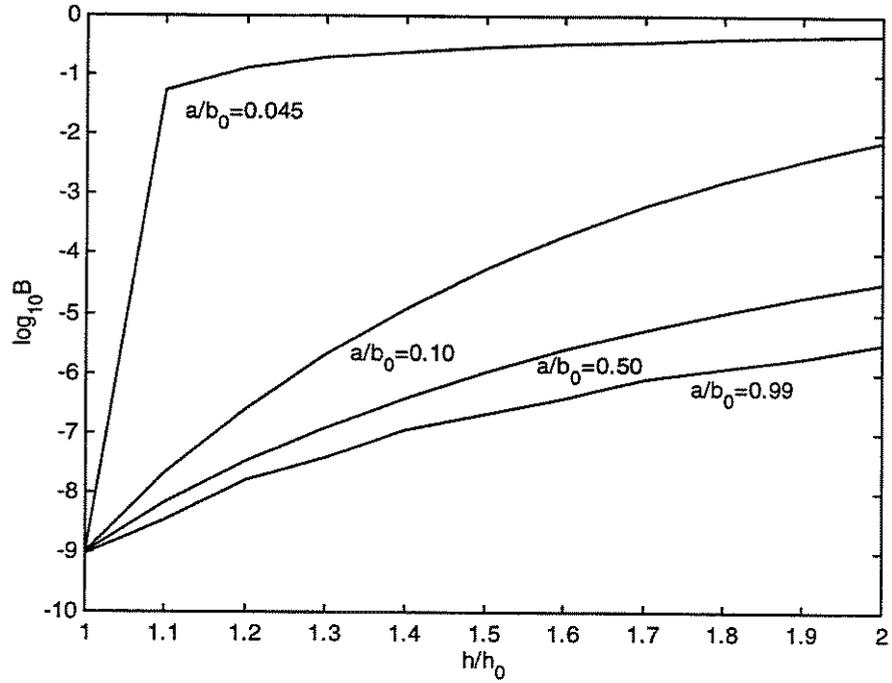


Figura 3.4: Probabilidade de perda B em função da duração média do surto h/h_0 para o serviço "Still Picture" (Caso B).

a/b_0	N
0.990	388
0.500	24704
0.100	72199
0.045	1124200

Tabela 3.4: Valores de N usados nas curvas da Figura 3.4 para cada valor de a/b_0 .

3.4. Controle da taxa de bit de pico (situação I)

A taxa de bit de pico é variada nos casos A e D da Tabela 3.3 .

Para o projeto do Leaky Bucket devemos fixar as seguintes condições:

1. Para os parâmetros nominais da fonte b_0 , h_0 e k_0 , a probabilidade de perda B deve ser B_0 .
2. Para um dado valor da taxa de bit de pico $b_1 > b_0$ (caso A), a probabilidade de perda B deve exceder um limiar B_1 .
3. Para um dado valor de grau do surto da fonte $\beta_1 > \beta_0$ (caso D), a probabilidade de perda B deve exceder um limiar B_2 .

Além disso, para se obter um melhor comportamento dinâmico a capacidade do buffer M deve ser a menor possível.

A solução está na região $a \approx b_0$. No caso do serviço “Packet Voice”, impondo-se

$$\begin{aligned} B_0 &\approx 10^{-9} \quad \text{para } b = b_0, h = h_0, k = k_0, \\ B_1 &\approx 0.1 \quad \text{para } b_1 = 1.2b_0 \text{ (caso A),} \\ B_2 &\approx 0.1 \quad \text{para } \beta_1 = 1.2\beta_0, \text{ (caso D),} \end{aligned}$$

obtem-se $a = 0.99b_0$, $M = 0.12t_0 = 4538 = 1696$ bits, pois

$$\begin{aligned} t_0 &= b_0 h_0 = 32 \text{ kbps } 352 \text{ msec} = 11264 \text{ bits} \\ M &= 0.12t_0 = 1351.68 \Rightarrow N = \frac{M}{538} = 3.19 \\ &\Rightarrow M = 4538 = 1696 \text{ bits.} \end{aligned}$$

Para considerar a “seletividade” no caso de variação de β (caso D), deve-se variar b , h e k , pois

$$\begin{aligned} \beta &= \frac{h+k}{h} = \frac{\text{constante}}{h} = \frac{b}{m}, \\ \beta \uparrow &\Rightarrow b \uparrow, h \downarrow, k \uparrow = (h_0 + k_0) - (h \downarrow). \end{aligned}$$

Para considerar a “seletividade” no caso de variação de c (caso E), deve-se variar h e k , pois

$$c \uparrow \Rightarrow h \uparrow, k \uparrow.$$

Na Figura 3.5 é mostrado o desempenho obtido. A seletividade é boa para os parâmetros b (caso A) e β (caso D), mas é ruim para h (caso B), k (caso C) e c (caso E).

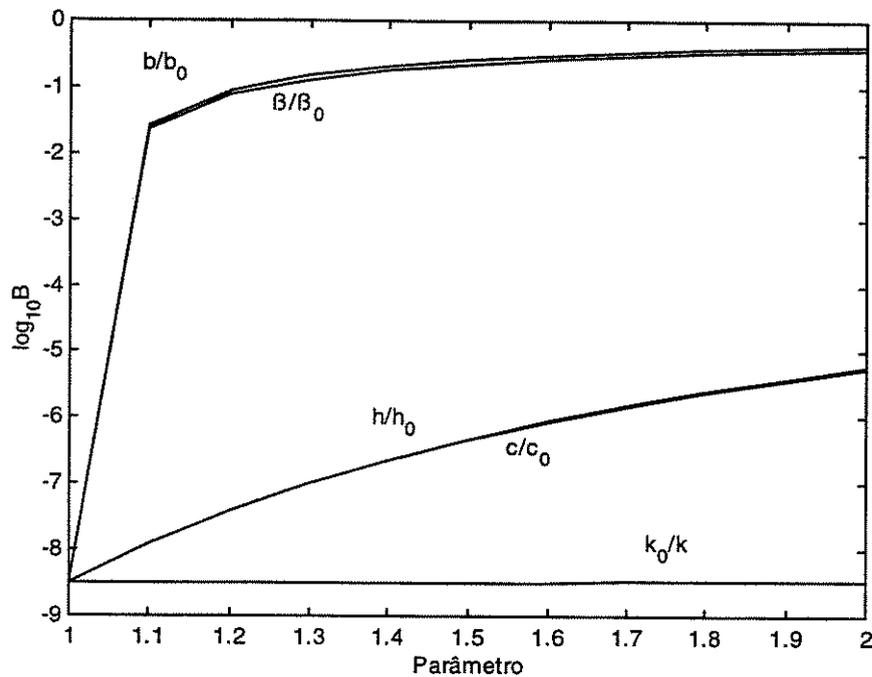


Figura 3.5: Probabilidade de perda B , representando o desempenho do Leaky Bucket na situação I para o serviço “Packet Voice” ($a/b_0 = 0.99$, $M = 1696$ bits).

3.5. Controle da taxa de bit média (situação II).

A taxa de bit média é variada nos casos A, B e C da Tabela 3.3. Portanto, uma prova de boa seletividade pode ser obtida impondo-se:

1. Para os parâmetros nominais da fonte b_0 , h_0 e k_0 , a probabilidade de perda B deve ser B_0 .
2. Para um dado valor da taxa de bit de pico $b_1 > b_0$ (caso A), a probabilidade de perda B deve exceder um limiar B_1 .
3. Para um dado valor de duração de surto $h_1 > h_0$ (caso B), a probabilidade de perda B deve exceder um limiar B_2 .
4. Para um dado valor de duração de silêncio $k_1 < k_0$ (caso C), a probabilidade de perda B deve exceder um limiar B_3 .

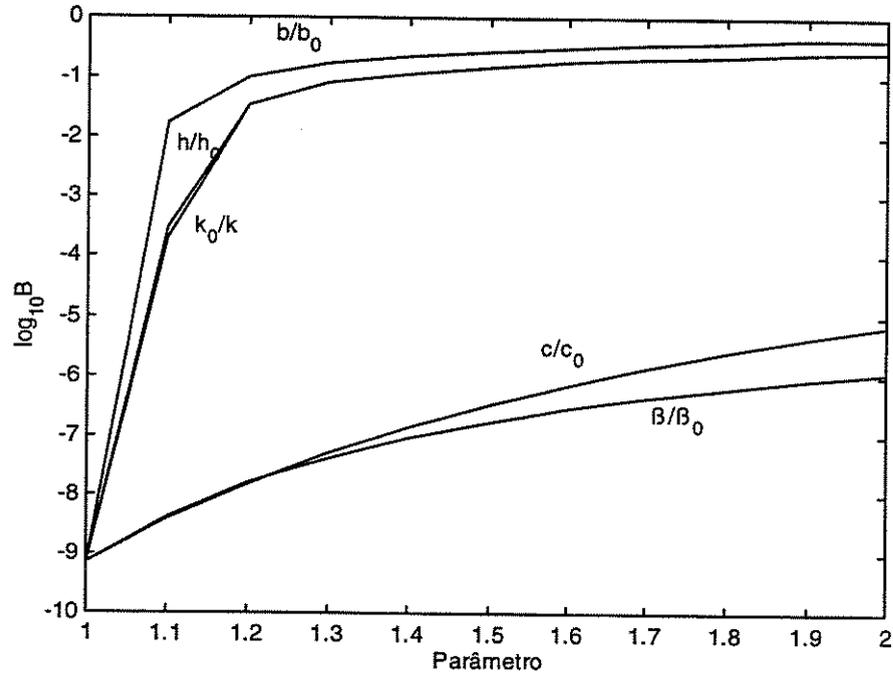


Figura 3.6: Probabilidade de perda B , representando o desempenho do Leaky Bucket na situação II para o serviço “Packet Voice” ($a/b_0 = 0.38$, $M = 1107064$ bits).

Todas estas condições podem ser satisfeitas na região $a \approx m_0$. No caso do serviço “Packet Voice”, impondo-se

$$\begin{aligned}
 B_0 &\approx 10^{-9} && \text{para } b = b_0, h = h_0, k = k_0, \\
 B_1 &\approx 0.1 && \text{para } b_1 = 1.2b_0 \text{ (caso A),} \\
 B_2 &\approx 0.1 && \text{para } h_1 = 1.3h_0 \text{ (caso B),} \\
 B_3 &\approx 0.1 && \text{para } k_1 = k_0/1.3 \text{ (caso C),}
 \end{aligned}$$

obtem-se $a = 0.38b_0 \approx m_0$, $M = 98.25t_0 = 1107064$ bits, pois

$$\begin{aligned}
 t_0 &= b_0 h_0 = 2 \text{ Mbps } 500 \text{ mseg} = 1 \text{ e } 6 \text{ bits}, \\
 M &= 98.25t_0 = 1106688 \Rightarrow N = \frac{M}{538} = 2610.11 \\
 &\Rightarrow M = 2611538 = 1107064 \text{ bits.}
 \end{aligned}$$

É importante observar o grande valor do limiar do contador, o que implica em tempos necessariamente muito grandes para detectar uma violação da taxa de bit média.

Na Figura 3.6 é mostrado o desempenho obtido. A seletividade é boa para os parâmetros controlados b (caso A), h (caso B), k (caso C), mas é ruim para β (caso D) e c (caso E).

3.6. Controle da duração do surto (situação III).

A duração do surto é modificada nos casos B e E da Tabela 3.3.

- Para o caso B, a região efetiva de controle é $a \approx m_0$ e o desempenho é mostrado na Figura 3.6.
- Para o caso E, a região efetiva de controle é $a \approx b_0$, mas a curva é bastante plana (vide Figura 3.5). Isto significa que o Leaky Bucket é pouco sensível à duração do surto quando a fonte incrementa na mesma proporção as durações do surto e de silêncio, mantendo-se a taxa de bit média constante.

3.7. Conclusões.

- Foi analisado o desempenho da técnica de policiamento Leaky Bucket usando-se uma expressão explícita aproximada, obtida com o método de fluxo, para a probabilidade de perda de células, supondo durações de surto e de silêncio com distribuições exponenciais.
- Foi observado que os parâmetros do Leaky Bucket dependem fortemente das características da fonte a ser controlada.
- A taxa de bit de pico pode ser controlada facilmente adotando-se a taxa de drenagem próxima da taxa de pico da fonte.
- O controle da taxa de bit média requer uma taxa de drenagem próxima da taxa média da fonte.
- O comprimento do surto não pode ser efetivamente controlado pelo leaky bucket sempre que a fonte incrementa na mesma proporção as durações do surto e de silêncio.
- No caso do controle da taxa de bit média, é requerido um limiar muito alto no contador do leaky bucket para se obter uma probabilidade de perda aceitável nas condições nominais. Isto implica que são necessários tempos muito grandes para se detectar uma violação da taxa de bit média. Este problema é tanto mais grave quanto maior o grau de surto do serviço (ou seja, com tráfego muito “bursty” como foi observado na Tabela 3.2 para o caso do serviço “Still Picture”).

Capítulo 4

Predição Ótima para o Movimento Browniano Fracionário

4.1. Introdução

Como já relatado anteriormente, as técnicas de controle de tráfego na literatura consideram modelos de tráfego do tipo Markoviano, que representam processos estocásticos com dependência de curto prazo.

Todavia, os maiores usuários das redes ATM deverão utilizar sinais de vídeo a taxa de bit variável (VBR) ou de redes locais. Já foi demonstrado [12] que estes usuários geram tráfego auto-similar, isto é, com dependência de longo prazo, o que deve acarretar sérios problemas de controle de tráfego nessas redes. Infelizmente, nenhum dos modelos comumente utilizados para modelagem de tráfego é capaz de captar esse comportamento auto-similar ou fractal.

Norros [1] propôs modelar o tráfego em redes locais utilizando o chamado tráfego Browniano fracionário, dado por

$$A(t) = mt + \sqrt{am}B_H(t),$$

onde $A(t)$ representa a quantidade de pacotes no intervalo $(0, t]$, m a taxa média, a um coeficiente de variância e $B_H(t)$ o movimento Browniano fracionário (fBm) normalizado (com $\sigma^2 = 1$).

Em [2] é obtido o preditor ótimo do fBm no sentido quadrático médio em tempo contínuo, bem como o correspondente erro quadrático médio do preditor. Os resultados são interessantes e enfatizam algumas propriedades peculiares dos preditores para processos auto-similares. Entretanto, dado que na prática só as amostras do processo são conhecidas, neste capítulo nós desenvolvemos o preditor ótimo do fBm em tempo discreto a N passos e de ordem p , no sentido quadrático médio. O método utilizado difere daquele usado por Norros, pois lida com o processo discreto.

Nossos resultados também verificam certas características distintivas dos preditores mencionadas em [1] e [2], mas para sinais em tempo discreto, de maior utilidade prática.

O movimento Browniano fracionário com parâmetro de Hurst H , $0 < H < 1$, é um processo estocástico não estacionário, Gaussiano, de média zero, denotado por $B_H(t)$, $t \in \mathbb{R}$

, que pode ser definido por uma integral estocástica [5]. O fBm possui algumas propriedades básicas, tais como

1. $B_H(0) = 0$
2. $B_H(t)$ tem caminhos amostrais contínuos.
3. as diferenças $B_H(t+s) - B_H(t)$ têm distribuições Gaussianas de dimensão finita.

Além disso, o carácter não estacionário do fBm é observado na sua função de auto-correlação, dada por

$$E[B_H(t)B_H(s)] = \frac{\sigma^2}{2} [|t|^{2H} + |s|^{2H} - |t-s|^{2H}], \quad (4.1)$$

onde

$$\sigma^2 = \Gamma(1-2H) \frac{\cos(\pi H)}{\pi H}$$

e Γ representa a função Gamma. Pode-se notar que a variância do fBm é

$$\text{Var}[B_H(t)] = \sigma^2 |t|^{2H}. \quad (4.2)$$

Vamos definir o processo $B_H(t)$ amostrado como

$$x(n) = B_H(n),$$

onde o intervalo de amostragem é considerado unitário.

A partir de (4.1) podemos obter a função de auto-correlação de $x(n)$, na forma

$$R_{xx}(n_1, n_2) = E[x(n_1)x(n_2)] = \frac{\sigma^2}{2} [|n_1|^{2H} + |n_2|^{2H} - |n_1 - n_2|^{2H}]. \quad (4.3)$$

Podemos definir o processo incremento N -ário do fBm como a sequência Gaussianiana

$$i_N(n+N) = x(n+N) - x(n). \quad (4.4)$$

Esta é uma sequência estacionária com variância $\sigma^2 N^{2H}$ e função de auto-correlação

$$\begin{aligned} R_{i_N}(k) &= E[i_N(n+N)i_N(n+k+N)] = E[(x(n+N) - x(n)) \\ &\quad (x(n+k+N) - x(n+k))] \\ &= \frac{\sigma^2}{2} (-|k|^{2H} + |k-N|^{2H} + |k+N|^{2H} - |k|^{2H}). \end{aligned}$$

Como um caso especial, para $N = 1$ obtemos a conhecida sequência *incremento unitário* do fBm, chamada ruído Gaussiano fracionário $i(n+1)$, dada por

$$i_1(n+1) = i(n+1) = x(n+1) - x(n). \quad (4.5)$$

Esta é também uma sequência estacionária, com variância σ^2 e função de auto-correlação

$$\begin{aligned} R_i(k) &= E[i(n)i(n+k)] = E[(x(n) - x(n-1))(x(n+k) - x(n+k-1))] \\ &= \frac{\sigma^2}{2} (-|k|^{2H} + |k-1|^{2H} + |k+1|^{2H} - |k|^{2H}). \end{aligned}$$

4.2. Predição ótima do fBm em tempo discreto

Preditor ótimo do fBm de primeira ordem.

Dado que o fBm é um processo Gaussiano, o preditor ótimo global dentre todos os preditores lineares e não lineares, deve ser linear [7]. Todavia, em razão da não estacionariedade do fBm, esse preditor ótimo deve ser *variante no tempo*.

Inicialmente, vamos calcular o preditor ótimo a um passo e de primeira ordem para o fBm. Seja

$$\hat{x}(n + 1/n) = c_n^o(0)x(n). \quad (4.6)$$

O coeficiente ótimo $c_n^o(0)$ é aquele que minimiza o erro quadrático médio σ_e^2 , dado por

$$\sigma_e^2 = E[(x(n + 1) - \hat{x}(n + 1/n))^2] = E[(x(n + 1) - c_n^o(0)x(n))^2]. \quad (4.7)$$

Podemos desenvolver (4.7), usando (4.1) e (4.2), para obtermos a seguinte expressão do erro quadrático médio do preditor

$$\sigma_e^2 = c^2(0)\sigma^2n^{2H} + \sigma^2(n + 1)^{2H} - c(0)\sigma^2[n^{2H} + (n + 1)^{2H} - 1]. \quad (4.8)$$

Como desejamos minimizar σ_e^2 , devemos fazer sua derivada primeira igual a zero, isto é,

$$\frac{\partial \sigma_e^2}{\partial c(0)} = 0 \rightarrow 2\sigma^2n^{2H}c(0) - \sigma^2[n^{2H} + (n + 1)^{2H} - 1] = 0,$$

que nos permite obter o coeficiente ótimo, variante no tempo, do preditor de primeira ordem,

$$c_n^o(0) = \frac{n^{2H} + (n + 1)^{2H} - 1}{2n^{2H}}, \quad (4.9)$$

tendo em vista que sua derivada segunda é positiva, pois

$$\frac{\partial^2 \sigma_e^2}{\partial c(0)^2} = 2\sigma^2n^{2H} > 0.$$

A partir de Eq. (4.9), pode-se ver que existe um estado estacionário para o coeficiente do preditor, pois ele converge para

$$c^o(0) |_{est} = \lim_{n \rightarrow \infty} c_n^o(0) = 1, \quad (4.10)$$

independentemente do valor de H . Da mesma forma, o erro quadrático médio em estado estacionário para o preditor ótimo é independente de H , pois

$$\sigma_e^2 |_{est} = \lim_{n \rightarrow \infty} \sigma_e^2 = \sigma^2. \quad (4.11)$$

Na Figura 4.1 é mostrada a variação de $c_n^o(0)$ em função de n , para $H = 0,8$.

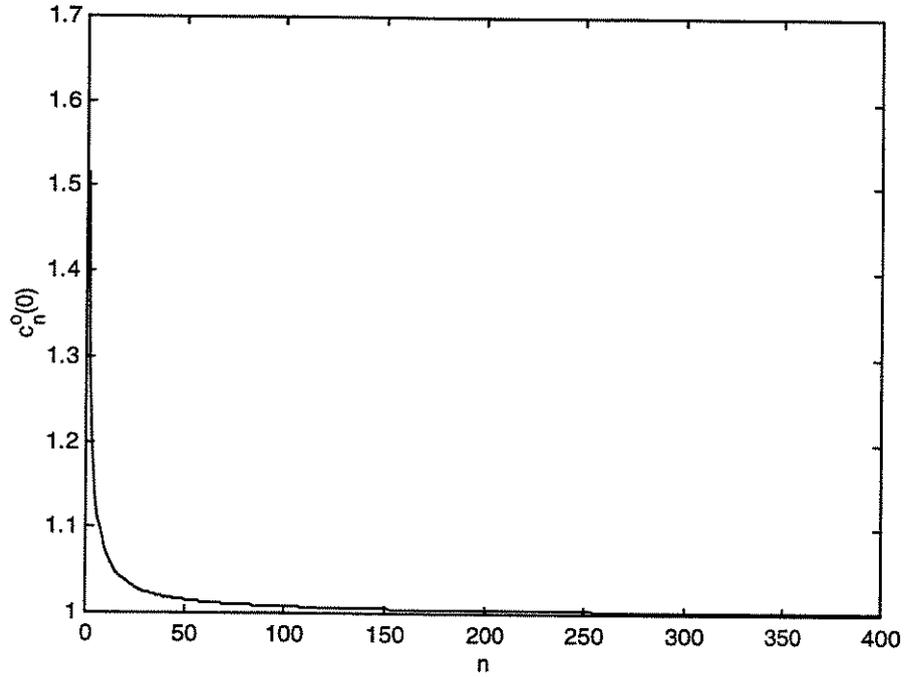


Figura 4.1: Coeficiente ótimo $c_n^o(0)$, variante no tempo, em função de n , para $H = 0,8$.

Para podermos comparar o desempenho dos diferentes preditores, é conveniente trabalhar com o erro quadrático médio normalizado ($NMSE$) dos preditores, definido por

$$NMSE \triangleq \frac{E[(x(n+N) - \hat{x}(n+N/n, n-1, \dots, n-(p-1)))^2]}{E[(x(n+N) - \hat{x}(n+N/n))^2]},$$

que representa o erro quadrático médio relativo ao preditor ótimo a N passos e de ordem p , normalizado pelo preditor ótimo a N passos e de primeira ordem em estado estacionário.

Pode ser demonstrado facilmente que em estado estacionário o preditor ótimo a N passos e de primeira ordem é dado por

$$\hat{x}(n+N/n) = x(n).$$

Assim, utilizando a expressão acima e a equação 4.2, resulta

$$\begin{aligned} NMSE &\triangleq \frac{E[(x(n+N) - \hat{x}(n+N/n, n-1, \dots, n-(p-1)))^2]}{E[(x(n+N) - x(n))^2]} \\ &= \frac{E[(x(n+N) - \hat{x}(n+N/n, n-1, \dots, n-(p-1)))^2]}{\sigma^2 N^{2H}}, \end{aligned} \quad (4.12)$$

Deve-se observar que para o preditor ótimo a N passos e de primeira ordem o $NMSE$ é unitário.

Preditor ótimo do fBm de segunda ordem

Vamos procurar agora o preditor ótimo a um passo e de segunda ordem, da forma

$$\hat{x}(n+1/n, n-1) = c_n^o(0)x(n) + c_n^o(1)x(n-1). \quad (4.13)$$

Para obtermos os coeficientes ótimos $c_n^o(0)$ e $c_n^o(1)$, fazemos iguais a zero as derivadas do erro quadrático médio do preditor, σ_e^2 , em relação aos coeficientes $c(0)$ e $c(1)$, ou seja,

$$\begin{aligned} \frac{\partial \sigma_e^2}{\partial c(0)} &= 2c(0)n^{2H} + c(1)[n^{2H} + (n-1)^{2H} - 1] \\ -[n^{2H} + (n+1)^{2H} - 1] &= 0 \\ \frac{\partial \sigma_e^2}{\partial c(1)} &= 2c(1)(n-1)^{2H} + c(0)[n^{2H} + (n-1)^{2H} - 1] \\ -[(n-1)^{2H} + (n+1)^{2H} - 2^{2H}] &= 0. \end{aligned}$$

A partir dessas equações podemos obter os coeficientes ótimos $c_n^o(0)$ e $c_n^o(1)$ em função do tempo, resultando em

$$\begin{aligned} c_n^o(0) &= \frac{\begin{vmatrix} n^{2H} + (n+1)^{2H} - 1 & n^{2H} + (n-1)^{2H} - 1 \\ (n-1)^{2H} + (n+1)^{2H} - 2^{2H} & 2(n-1)^{2H} \end{vmatrix}}{\begin{vmatrix} 2n^{2H} & n^{2H} + (n-1)^{2H} - 1 \\ n^{2H} + (n-1)^{2H} - 1 & 2(n-1)^{2H} \end{vmatrix}}, \\ c_n^o(1) &= \frac{\begin{vmatrix} 2n^{2H} & n^{2H} + (n+1)^{2H} - 1 \\ n^{2H} + (n-1)^{2H} - 1 & (n-1)^{2H} + (n+1)^{2H} - 2^{2H} \end{vmatrix}}{\begin{vmatrix} 2n^{2H} & n^{2H} + (n-1)^{2H} - 1 \\ n^{2H} + (n-1)^{2H} - 1 & 2(n-1)^{2H} \end{vmatrix}}, \end{aligned}$$

onde $|A|$ representa o determinante da matriz A .

Nas Figuras 4.2 e 4.3 são mostrados os coeficientes $c_n^o(0)$ e $c_n^o(1)$, respectivamente, em função de n , para $H = 0.8$.

Em estado estacionário, observa-se que os coeficientes do preditor satisfazem a

$$c^o(0)|_{est} = \lim_{n \rightarrow \infty} c^o(0) = 1.5157, \quad (4.14)$$

$$c^o(1)|_{est} = \lim_{n \rightarrow \infty} c^o(1) = -0.5157. \quad (4.15)$$

Será mostrado adiante que o erro quadrático médio em estado estacionário, para este preditor, é dado por

$$\sigma_e^2|_{est} = \lim_{n \rightarrow \infty} \sigma_e^2 = 0.734\sigma^2. \quad (4.16)$$

Portanto o erro quadrático médio normalizado assume o valor

$$NMSE = \frac{\sigma_e^2}{\sigma^2} = \frac{0.734\sigma^2}{\sigma^2} = 0.734, \quad (4.17)$$

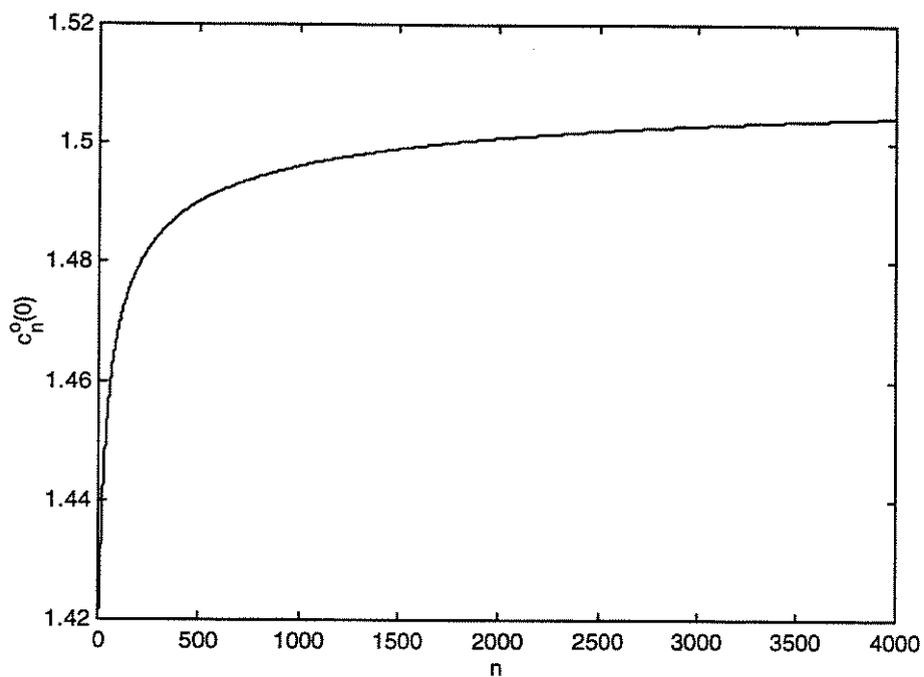


Figura 4.2: Coeficiente ótimo $c_n^o(0)$, variante no tempo, em função de n , para $H = 0,8$.

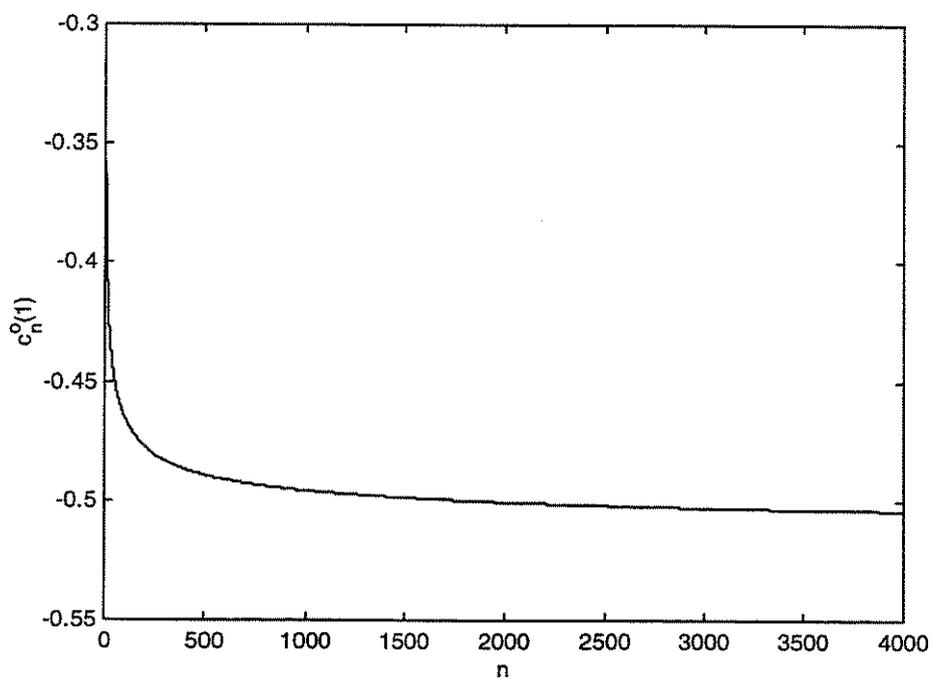


Figura 4.3: Coeficiente ótimo $c_n^o(1)$, variante no tempo, em função de n , para $H = 0,8$.

que é menor do que o *NMSE* do preditor a um passo e de primeira ordem (igual a 1), como é de se esperar.

É importante destacar que apesar do processo a se predizer, o fBm, ser *não estacionário*, o preditor ótimo atinge um *estado estacionário*, o que significa que seus coeficientes e seu erro de predição se tornam *constantes no tempo*. Esta é uma peculiaridade interessante desses preditores, que é válida para o caso geral, como será mostrado mais adiante.

Devido a que os preditores ótimos do fBm são lineares, as equações que permitem calcular os seus coeficientes podem ser obtidas diretamente a partir do *princípio da ortogonalidade*, em vez de se trabalhar com as derivadas. Esta propriedade é expressa pela seguinte proposição.

Proposição 4.2..1 *O preditor ótimo a N passos e de ordem p para o fBm, é da forma*

$$\hat{x}(n + N/n, n - 1, \dots, n - (p - 1)) = \sum_{j=0}^{p-1} c_n^o(j) x(n - j). \quad (4.18)$$

Além disso, os coeficientes ótimos $c_n^o(j)$, $j = 0, \dots, p - 1$, são dados pela equação

$$\mathbf{c}_n^o = \mathbf{F}^{-1} \mathbf{g}, \quad (4.19)$$

com

$$\mathbf{c}_n^o = [c_n(0), c_n(1), \dots, c_n(p - 1)]^t,$$

$$\mathbf{g} = [R_{xx}(n + N, n), R_{xx}(n + N, n - 1), \dots, R_{xx}(n + N, n - (p - 1))]^t,$$

$$\mathbf{F} = \begin{bmatrix} R_{xx}(n, n) & R_{xx}(n - 1, n) & \dots & R_{xx}(n - (p - 1), n) \\ R_{xx}(n, n - 1) & R_{xx}(n - 1, n - 1) & \dots & R_{xx}(n - (p - 1), n - 1) \\ \vdots & \vdots & \vdots & \vdots \\ R_{xx}(n, n - (p - 1)) & R_{xx}(n - 1, n - (p - 1)) & \dots & R_{xx}(n - (p - 1), n - (p - 1)) \end{bmatrix},$$

onde $R_{xx}(\cdot, \cdot)$ é dado por (4.3) e \mathbf{F} é suposta não singular.

O erro quadrático médio de predição é dado por

$$\begin{aligned} \sigma_e^2 &\triangleq E[(x(n + N) - \hat{x}(n + N/n, n - 1, \dots, n - (p - 1)))^2] \\ &= E[(x(n + N))^2] - [\mathbf{c}_n^o]^t \mathbf{g} = \sigma^2(n + N)^{2H} - [\mathbf{c}_n^o]^t \mathbf{g}, \end{aligned} \quad (4.20)$$

onde $\sigma^2(n + N)^{2H}$ é a variância de $x(n + N)$.

Prova: A Eq.(4.18) é direta devido a que o processo $x(n)$ é Gaussiano e então o preditor ótimo do fBm é linear. Como o preditor é linear e a predição é ótima no sentido quadrático médio, o princípio da ortogonalidade [7] deve ser satisfeito, ou seja,

$$\begin{aligned} E [[x(n + N) - \hat{x}(n + N/n, n - 1, \dots, n - (p - 1))] x(n - i)] &= 0, \\ i &= 0, \dots, p - 1; \end{aligned}$$

ou

$$E \left[\left[x(n+N) - \sum_{j=0}^{p-1} c^o(j)x(n-j) \right] x(n-i) \right] = 0, \quad i = 0, \dots, p-1.$$

Esta última equação pode ser desenvolvida usando (4.3) para se obter a Eq. (4.19). Finalmente, como uma consequência também do princípio da ortogonalidade [7] obtemos a Eq. (4.20).

Observações

Para $H = 0.8$ e em estado estacionário, o preditor ótimo a um passo de primeira ordem do fBm é dado por

$$\hat{x}(n+1/n) = x(n),$$

e o preditor ótimo de segunda ordem é

$$\hat{x}(n+1/n, n-1) = 1.5157x(n) + (-0.5157)x(n-1),$$

ou seja, em estado estacionário os coeficientes do preditor ótimo satisfazem

$$c^o(0) = 1, \quad (4.21)$$

$$c^o(0) + c^o(1) = 1, \quad (4.22)$$

para os preditores de primeira e segunda ordem, respectivamente.

É fácil ver que os erros quadráticos médios em estado estacionário, são dados por

$$\sigma_e^2 = E[(x(n+1) - x(n))^2], \quad (4.23)$$

e

$$\begin{aligned} \sigma_e^2 &= E[(x(n+1) - c^o(0)x(n) - c^o(1)x(n-1))^2] \\ &= E[[(x(n+1) - x(n)) + c^o(1)(x(n) - x(n-1))]^2], \end{aligned} \quad (4.24)$$

respectivamente.

Deve-se notar que as expressões de σ_e^2 em estado estacionário são expressões em termos dos incrementos do processo fBm. Lembremos que é justamente o incremento unitário do fBm que define a sequência $i(n)$, chamada ruído Gaussiano fracionário

$$i(n) \triangleq x(n) - x(n-1),$$

onde $x(n)$ é um fBm.

Como é conhecido, $i(n)$ é uma sequência estacionária com variância σ^2 e função de autocorrelação dada por

$$\begin{aligned} R_i(k) &= E[i(n)i(n+k)] = E[(x(n) - x(n-1))(x(n+k) - x(n+k-1))] \\ &= \frac{\sigma^2}{2}(-|k|^{2H} + |k-1|^{2H} + |k+1|^{2H} - |k|^{2H}). \end{aligned} \quad (4.25)$$

Portanto, é claro que devido à estacionariedade da sequência $i(n)$ as expressões dadas por (4.23) e (4.24) são independentes do instante de tempo n . Ou seja, o fato de se poder escrever a expressão de σ_e^2 em estado estacionário em termos de incrementos de $x(n)$ é uma condição suficiente para que o preditor tenha um erro quadrático médio constante em estado estacionário.

Mas, será que esta é também uma condição necessária para que o preditor seja ótimo? Os resultados a seguir demonstram que, mesmo para o caso geral de predição a N passos e de ordem p do fBm, esta condição é realmente necessária.

Proposição 4.2..2 *Seja o preditor ótimo a N passos e de ordem p do fBm, em estado estacionário, dado por*

$$\hat{x}(n + N/n, n - 1, \dots, n - (p - 1)) = \sum_{j=0}^{p-1} c^o(j)x(n - j).$$

Então, em estado estacionário os coeficientes do preditor ótimo devem cumprir a seguinte condição

$$\sum_{j=0}^{p-1} c^o(j) = 1. \quad (4.26)$$

Prova: O princípio da ortogonalidade deve ser satisfeito, ou seja,

$$E \left[\left[x(n + N) - \sum_{j=0}^{p-1} c^o(j)x(n - j) \right] x(n - i) \right] = 0, \quad i = 0, \dots, p - 1.$$

Pode-se desenvolver a expressão acima utilizando-se (4.3) para se obter

$$\begin{aligned} \frac{\sigma^2}{2} [(n + N)^{2H} + (n - i)^{2H} - (N + i)^{2H}] &= \frac{\sigma^2}{2} \sum_{j=0}^{p-1} c^o(j) [(n - j)^{2H} \\ &+ (n - i)^{2H} - | -j + i |^{2H}]; \\ i &= 0, \dots, p - 1. \end{aligned}$$

Portanto, em estado estacionário ($n \rightarrow \infty$) temos

$$\frac{\sigma^2}{2} 2n^{2H} = \frac{\sigma^2}{2} \sum_{j=0}^{p-1} c^o(j) 2n^{2H},$$

ou seja, deve-se cumprir

$$\sum_{j=0}^{p-1} c^o(j) = 1,$$

com a qual fica demonstrada esta importante proposição.

Tendo em vista a Eq. (4.26), que garante que os coeficientes em estado estacionário do preditor ótimo somam 1, vamos observar o que acontece no projeto dos preditores ótimos de terceira e quarta ordens a N passos.

Preditor ótimo do fBm a N passos e de terceira ordem em estado estacionário

O preditor ótimo deve minimizar o erro

$$\sigma_e^2 = E [x(n+N) - c(0)x(n) - c(1)x(n-1) - c(2)x(n-2)]^2.$$

Mas, sem perda de generalidade e aplicando a Eq. (4.26) podemos definir $d(j)$, $j = 0, 1$ tais que

$$\begin{aligned} c^o(0) &= 1 + d(0), \\ c^o(1) &= -d(0) + d(1), \\ c^o(2) &= -d(1). \end{aligned} \tag{4.27}$$

Isto garante que $c^o(0) + c^o(1) + c^o(2) = 1$. Então, a expressão ótima de σ_e^2 fica

$$\begin{aligned} \sigma_e^2 &= E [[x(n+N) - x(n)] - d(0)[x(n) - x(n-1)] - d(1)[x(n-1) - x(n-2)]]^2 \\ &= E [i_N(n+N) - [d(0)i(n) + d(1)i(n-1)]]^2. \end{aligned}$$

Portanto, pode-se considerar que para minimizar σ_e^2 deve-se procurar o preditor ótimo de ordem 2, com coeficientes $d(0)$ e $d(1)$, a partir dos dados $i(n)$ e $i(n-1)$ da sequência Gaussiana e estacionária, correspondente ao incremento N -ário do fBm, definido por $i_N(n+N) \triangleq x(n+N) - x(n)$.

Finalmente, usando a Eq. (4.27) podem-se obter os coeficientes ótimos do preditor a N passos e de terceira ordem do fBm.

Preditor ótimo do fBm a N passos e de quarta ordem em estado estacionário

Analogamente, devemos minimizar o erro

$$\sigma_e^2 = E [x(n+N) - c(0)x(n) - c(1)x(n-1) - c(2)x(n-2) - c(3)x(n-3)]^2.$$

Mas, aplicando a Eq. (4.26) podemos agora definir $d(j)$, $j = 0, \dots, 2$ tais que

$$\begin{aligned} c^o(0) &= 1 + d(0), \\ c^o(1) &= -d(0) + d(1), \\ c^o(2) &= -d(1) + d(2), \\ c^o(3) &= -d(2). \end{aligned} \tag{4.28}$$

Assim, temos que $c^\circ(0) + c^\circ(1) + c^\circ(2) + c^\circ(3) = 1$. A expressão de σ_e^2 é então dada por

$$\begin{aligned}\sigma_e^2 &= E \left[[x(n+N) - x(n)] - d(0)[x(n) - x(n-1)] - d(1)[x(n-1) - x(n-2)] \right. \\ &\quad \left. - d(2)[x(n-2) - x(n-3)] \right]^2 \\ &= E \left[i_N(n+N) - [d(0)i(n) + d(1)i(n-1) + d(2)i(n-2)] \right]^2.\end{aligned}$$

Portanto, novamente pode-se considerar que para minimizar σ_e^2 deve-se procurar o preditor ótimo de ordem 3, com coeficientes $d(0)$, $d(1)$ e $d(2)$, da sequência Gaussiana e estacionária correspondente ao incremento N -ário do fBm $i_N(n+N)$, a partir dos dados $i(n)$, $i(n-1)$ e $i(n-2)$. Usando (4.28) podem-se então obter os coeficientes ótimos do preditor a N passos e de quarta ordem do fBm.

Geralizando, temos o seguinte corolário da **Proposição 4.2..2**.

Corolário 4.2..1 *Projetar o preditor ótimo a N passos e de ordem p (com $p > 1$) do fBm, em estado estacionário, é equivalente a projetar o preditor ótimo de ordem $p-1$ a partir dos dados $i(n-j)$, $j = 0, \dots, p-2$, do incremento N -ário do fBm $i_N(n+N)$, em estado estacionário, ou seja, devem-se procurar os coeficientes $d^\circ(j)$, $j = 0, \dots, p-2$, que minimizam o erro quadrático médio*

$$\sigma_e^2 = E \left[i_N(n+N) - \sum_{j=0}^{p-2} d^\circ(j)i(n-j) \right]^2,$$

e em seguida se obter os coeficientes ótimos $c^\circ(j)$, $j = 0, \dots, p-1$, do preditor do fBm, através da equação

$$\mathbf{c}^\circ = \mathbf{A}\mathbf{d}^\circ + \mathbf{b}, \quad (4.29)$$

sendo

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & -1 & 1 \\ 0 & 0 & \dots & \dots & 0 & -1 \end{bmatrix},$$

$$\mathbf{b} = [1, 0, 0, \dots, 0]^t,$$

$$\mathbf{c}^\circ = [c^\circ(0), c^\circ(1), \dots, c^\circ(p-1)]^t,$$

$$\mathbf{d}^\circ = [d^\circ(0), d^\circ(1), \dots, d^\circ(p-2)]^t.$$

Prova: Para projetar o preditor ótimo a N passos e de ordem p (com $p > 1$) do fBm, deve-se minimizar

$$\sigma_e^2 = E \left[x(n+N) - \sum_{j=0}^{p-1} c^o(j)x(n-j) \right]^2.$$

Definindo $d^o(j)$, $j = -1, \dots, p-2$, tais que

$$\begin{aligned} c^o(j) &= -d^o(j-1) + d^o(j), j = 0, \dots, p-2, \quad d^o(-1) = -1; \\ c^o(p-1) &= -d^o(p-2), \end{aligned} \quad (4.30)$$

a expressão de σ_e^2 pode ser escrita como

$$\sigma_e^2 = E \left[x(n+N) - \left[\sum_{j=0}^{p-2} (-d^o(j-1) + d^o(j))x(n-j) - d^o(p-2)x(n-(p-1)) \right] \right]^2, \quad (4.31)$$

sendo,

$$d^o(-1) = -1,$$

já que é cumprida a condição necessária do preditor ótimo do fBm em estado estacionário,

$$\sum_{j=0}^{p-1} c^o(j) = 1,$$

como pode ser verificado facilmente.

Mas a Equação (4.31) a minimizar pode ser escrita como

$$\sigma_e^2 = E \left[[x(n+N) - x(n)] - \sum_{j=0}^{p-2} d^o(j)[x(n-j) - x(n-(j+1))] \right]^2.$$

Portanto, minimizar σ_e^2 é equivalente a projetar o preditor ótimo de ordem $p-1$ em estado estacionário para o incremento N -ário do fBm. A partir dos coeficientes desse preditor e da Eq. (4.30) pode-se obter os coeficientes ótimos $c^o(j)$, $j = 0, \dots, p-1$, em estado estacionário, do preditor a N passos e de ordem p do fBm.

Com isto fica demonstrado o corolário.

4.3. Projeto do preditor ótimo a N passos e de ordem p do fBm e análise do desempenho

Para $p > 1$ e como consequência do Corolário (4.29), devemos começar procurando o preditor ótimo de ordem $p - 1$, com coeficientes $d^\circ(j)$, $j = 0, \dots, p - 2$, para o incremento N -ario do fBm, isto é,

$$\hat{i}_N(n + N) = \sum_{j=0}^{p-2} d^\circ(j) i(n - j).$$

Como é bem conhecido [7], os coeficientes ótimos $d^\circ(j)$, $j = 0, \dots, p - 2$, no sentido quadrático médio, são dados pela equação de Wiener-Hopf

$$\mathbf{d}^\circ = \mathbf{R}^{-1} \mathbf{r}, \quad (4.32)$$

com

$$\begin{aligned} \mathbf{d}^\circ &= [d(0), d(1), \dots, d(p - 2)]^t \\ \mathbf{r} &= [R_{ii}(0), R_{ii}(1), \dots, R_{ii}(p - 2)]^t \\ \mathbf{R} &= \begin{bmatrix} R_i(0) & R_i(1) & \dots & R_i(p - 2) \\ \vdots & \vdots & \vdots & \vdots \\ R_i(p - 2) & R_i(p - 3) & \dots & R_i(0) \end{bmatrix}, \end{aligned}$$

onde $R_i(\cdot)$ são dadas pela Eq. (4.25) e $R_{ii}(k)$, $k = 0, \dots, p - 2$, são dadas por

$$\begin{aligned} R_{ii}(k) &\triangleq E[i_N(n + N) i(n - k)] = E[(x(n + N) - x(n))(x(n - k) - x(n - (k + 1)))] \\ &= \frac{\sigma^2}{2} [-(N + k)^{2H} + (N + k + 1)^{2H} + k^{2H} - (k + 1)^{2H}] \end{aligned} \quad (4.33)$$

O erro quadrático médio de predição é dado por

$$\begin{aligned} \sigma_e^2 &\triangleq E[(i_N(n + N) - \hat{i}_N(n + N))^2] \\ &= E[(i_N^2(n + N)) - [d^\circ]^t \mathbf{r} = \sigma^2 N^{2H} - [d^\circ]^t \mathbf{r}, \end{aligned}$$

onde $\sigma^2 N^{2H}$ é a variância da sequência estacionária $i_N(n + N)$.

Como consequência do Corolário (4.29) e da Eq. (4.12), o $NMSE$ mínimo na predição do fBm é dado por

$$\begin{aligned} NMSE &= \frac{\sigma_e^2}{\sigma^2 N^{2H}} \\ &= 1 - \frac{[d^\circ]^t \mathbf{r}}{\sigma^2 N^{2H}}, \end{aligned} \quad (4.34)$$

	<i>NMSE</i> teórico
1	1.0000
2	0.4969
3	0.4607
4	0.4398
5	0.4276
6	0.4194
7	0.4134
10	0.4026
50	0.3814

Tabela 4.1: *NMSE* para a predição ótima a 5 passos do fBm com $H = 0.9$ e diferentes ordens p nos preditores.

Os coeficientes ótimos do preditor do fBm podem ser calculados usando-se as Eqs. (4.29) e (4.32).

Na Tabela 4.1 são mostrados os *NMSE* teóricos, dados pela Eq. (4.34), para a predição ótima a 5 passos do fBm, com $H = 0.9$, e diferentes ordens p nos preditores .

Observa-se que para a predição de $x(n + 5)$, faz pouca diferença conhecer $x(n - j)$, $j = 0, 1, \dots, 5$, ou $x(n - j)$, $j = 0, 1, \dots$, ou seja, *a amostra presente e as cinco últimas amostras fornecem tanta informação ao preditor quanto todas as infinitas amostras passadas*. Neste caso, portanto, para todos os efeitos práticos é conveniente usar um preditor de ordem 6. O mesmo comportamento foi observado para outros valores de H , com $0.5 < H < 1$. Este fato está em concordância com a *regra empírica* demonstrada por Norros [1]: deve-se prever a amostra do próximo segundo com as amostras do último segundo, a do próximo minuto com as do último minuto, etc. Este é um resultado interessante e muito peculiar dos processos auto-similares.

Nas Tabelas 4.2, 4.3, 4.4 e 4.5 são mostrados os coeficientes ótimos e o correspondente *NMSE* teórico do preditor ótimo a N passos para diferentes valores de H e N . Como consequência da *regra empírica* mencionada acima, foram usados preditores de ordem $p = N + 1$ em todos os casos. Para efeito de comparação, é também mostrado nessas tabelas o desempenho do preditor ótimo a N passos e de primeira ordem.

Pode-se observar que:

1. Quando H é próximo de 0.5, o preditor ótimo a N passos e de primeira ordem tem quase o mesmo desempenho do preditor ótimo a N passos e de ordem $N + 1$, devido a que a correlação do processo é baixa para esses valores de H .
2. Quando H tende a 1, os preditores ótimos a N passos e de ordem $N + 1$ têm um desempenho muito melhor do que o preditor a N passos e de ordem 1, resultado que

N	p	$NMSE$ teórico	Coeficientes ótimos do preditor c^o						
N	1	1.0000	1.0000						
1	2	0.9779	1.1487	-0.1487					
2	3	0.9754	1.2064	-0.1153	-0.0910				
3	4	0.9743	1.2475	-0.1360	-0.0220	-0.0895			
4	5	0.9737	1.2791	-0.1468	-0.0326	-0.0107	-0.0889		
5	6	0.9733	1.3049	-0.1547	-0.0368	-0.0196	-0.0047	-0.0891	

Tabela 4.2: Coeficientes ótimos e o correspondente $NMSE$ teórico do preditor ótimo a N passos para $H = 0.6$ e diferentes valores de N .

N	p	$NMSE$ teórico	Coeficientes ótimos do preditor c^o						
N	1	1.0000	1.0000						
1	2	0.8979	1.3195	-0.3195					
2	3	0.8895	1.4469	-0.2547	-0.1921				
3	4	0.8854	1.5465	-0.3285	-0.0152	-0.2028			
4	5	0.8831	1.6250	-0.3645	-0.0563	0.0026	-0.2067		
5	6	0.8817	1.6907	-0.3927	-0.0670	-0.0367	0.0173	-0.2116	

Tabela 4.3: Coeficientes ótimos e o correspondente $NMSE$ teórico do preditor ótimo a N passos para $H = 0.7$ e diferentes valores de N .

N	p	$NMSE$ teórico	Coeficientes ótimos do preditor c^o						
N	1	1.0000	1.0000						
1	2	0.7340	1.5157	-0.5157					
2	3	0.7198	1.7271	-0.4228	-0.3043				
3	4	0.7118	1.9082	-0.5993	-0.0374	-0.3463			
4	5	0.7076	2.0533	-0.6798	-0.0650	0.0519	-0.3603		
5	6	0.7049	2.1781	-0.7482	-0.0808	-0.0545	0.0819	-0.3765	

Tabela 4.4: Coeficientes ótimos e o correspondente $NMSE$ teórico do preditor ótimo a N passos para $H = 0.8$ e diferentes valores de N .

N	p	$NMSE$ teórico	Coeficientes ótimos do preditor c^o						
N	1	1.0000	1.0000						
1	2	0.4508	1.7411	-0.7411					
2	3	0.4362	2.0536	-0.6250	-0.4286				
3	4	0.4270	2.3458	-0.9784	0.1602	-0.5276			
4	5	0.4223	2.5824	-1.1280	-0.0501	0.1532	-0.5575		
5	6	0.4194	2.7919	-1.2673	-0.0615	-0.0803	0.2117	-0.5946	

Tabela 4.5: Coeficientes ótimos e o correspondente $NMSE$ teórico do preditor ótimo a N passos para $H = 0.9$ e diferentes valores de N .

decorre da alta correlação do processo. Portanto, podemos usar o passado com somente $p = N + 1 > 1$ coeficientes para reduzir, de forma significativa, o erro quadrático médio de predição.

Nota-se, em concordância com Norros [1], que quanto mais alto o valor de H , maior será a força preditiva do passado. Por exemplo, quando $H = 0.6$ pode ser suficiente usar o preditor ótimo de primeira ordem para qualquer valor de N , mas quando $H = 0.9$ é conveniente usar o preditor ótimo de ordem $N + 1$. Para $N = 5$, isto permite reduzir de 58% o erro quadrático médio em relação ao preditor ótimo de ordem 1.

4.4. Conclusões

A família de processos estocásticos $1/f$, com dependência de longo prazo, constitui uma importante classe de modelos para diferentes aplicações em processamento de sinais, em particular para a modelagem de tráfego auto-similar.

Norros [1] estudou o grau de previsibilidade de curto prazo e forneceu uma regra empírica para a previsão. Neste trabalho nós obtemos o preditor ótimo no sentido quadrático médio, em tempo discreto, a N passos e de ordem p , para o fBm. Demonstramos que, em estado estacionário, este preditor ótimo pode ser projetado a partir de um preditor para os incrementos N -ários do fBm. Também apresentamos valores numéricos dos coeficientes ótimos e dos correspondentes erros quadráticos médios de predição, para diferentes valores dos parâmetros H e N .

Além do interesse teórico no estudo da predição ótima para processos auto-similares, espera-se que em um futuro próximo surgirão numerosas aplicações, especialmente em mecanismos de controle de congestionamento em redes de faixa larga e em esquemas de controle de taxa e alocação de bits para compressão de vídeo. A disponibilidade de preditores simples e eficazes como os desenvolvidos neste trabalho, pode ser de grande importância no projeto e gerenciamento de redes digitais de serviços integrados de alta velocidade.

Capítulo 5

Estimação dos Parâmetros do fBm Usando Transformada de Wavelet

5.1. Introdução

Utilizando o modelo do tráfego Browniano fracionário, o grau de auto-similaridade (ou “burstiness”) do tráfego pode ser medido somente com um parâmetro: o parâmetro de Hurst H do fBm.

Portanto, a estimação de H é de muita importância para caracterizar o grau de auto-similaridade do tráfego e suas outras propriedades estatísticas, bem como para acompanhar as flutuações na dependência de longo prazo do sinal de tráfego.

Já foram desenvolvidos na literatura diferentes estimadores para H , tanto no domínio do tempo quanto no domínio da frequência. Neste capítulo nós destacamos as vantagens da técnica de estimação de H baseada em wavelets [11], [19].

Outro objetivo deste capítulo é analisar também a estimação da constante de variância do fBm usando wavelets.

Outro interessante problema que pode ser muito útil em controle de tráfego, é a obtenção de uma completa predição dos incrementos do fBm em diferentes instantes e escalas de tempo. Nós demonstramos aqui que é possível se obter essas predições fazendo-se uma análise wavelet em tempo-escala do fBm, usando um banco de preditores de Wiener multi-escala [23]. Estas predições podem ser úteis em funções de gerenciamento de tráfego em tempo real, como mecanismos de controle de congestionamento dinâmicos. Por exemplo, um esquema de controle lógico pode escolher a predição na escala adequada de acordo com o nível da fila num certo instante de tempo.

Neste capítulo vamos estudar a análise com transformada de wavelets do fBm, mostrando que é possível resolver os dois problemas mencionados acima de forma eficiente.

5.2. Propriedades da correlação dos coeficientes wavelet do movimento Browniano fracionário

Lembremos que o movimento Browniano fracionário $B_H(t), t \in \mathbb{R}$, com parâmetro H (sendo $0 < H < 1$), é um processo estocástico não estacionário, Gaussiano, com média zero.

A não estacionariedade do fBm é observada na sua função de auto-correlação

$$E[B_H(t)B_H(s)] = \frac{\sigma^2}{2} [|t|^{2H} + |s|^{2H} - |t - s|^{2H}].$$

Dai resulta que a variância do fBm é

$$\text{Var}[B_H(t)] = \sigma^2 |t|^{2H}.$$

Esta família de processos estocásticos é mais geral do que o movimento de Wiener ordinário, obtido como caso particular para $H = 1/2$.

O fBm não tem um espectro definido no sentido usual, mas é possível definir um espectro de potência empírico, da forma [6]

$$S_{B_H}(\omega) = \frac{\sigma^2}{|\omega|^{2H+1}}.$$

Além disso, o processo fBm é auto-similar, isto é, se $B_H(t)$ tem parâmetros H e σ , então o processo $a^{-H}B_H(at - b)$ possui os mesmos parâmetros, para qualquer $a > 0$ e qualquer b .

Pode-se provar que a *dimensão fractal* das funções amostra do processo $B_H(t)$ é igual a $2 - H$ [5].

Em resumo, estes processos estocásticos têm as seguintes características:

1. não estacionariedade (que abrange o tempo)
2. auto-similaridade (que abrange a escala do tempo)

Assim, tendo em vista que a análise com transformada de wavelet é basicamente uma análise *tempo-escala*, ela é uma ferramenta apropriada para a análise, a estimação [24][25] e a predição desses processos.

Como é bem conhecido [3][8][10], o desenvolvimento em wavelets do processo fBm $B_H(t)$, até a escala J , é dado por

$$\begin{aligned} B_H(t) &= 2^{-J/2} \sum_{n=-\infty}^{\infty} a_J[n] \phi(2^{-J}t - n) \\ &+ \sum_{j=-\infty}^J 2^{-j/2} \sum_{n=-\infty}^{\infty} d_j[n] \psi(2^{-j}t - n), \end{aligned}$$

onde $\psi(t)$ é a função wavelet, $\phi(t)$ é a correspondente função de escala e, para uma decomposição ortonormal,

$$\begin{aligned} d_j[n] &= 2^{-j/2} \int_{-\infty}^{\infty} B_H(t) \psi(2^{-j}t - n) dt \\ a_J[n] &= 2^{-J/2} \int_{-\infty}^{\infty} B_H(t) \phi(2^{-J}t - n) dt, \end{aligned}$$

onde $j, n \in Z$.

Nosso objetivo é calcular a correlação dos coeficientes $d_j[n]$ e $a_j[n]$. Pode-se demonstrar que a correlação dos *coeficientes de detalhes* é dada por

$$E(d_j[n]d_k[m]) = \frac{\sigma^2}{2} \left(- \int_{-\infty}^{+\infty} A_\psi(2^{j-k}, \tau - (2^{j-k}n - m)) |\tau|^{2H} d\tau \right) (2^k)^{2H+1}, \quad (5.1)$$

onde

$$A_\psi(\alpha, \tau) = \sqrt{\alpha} \int_{-\infty}^{+\infty} \psi(t)\psi(\alpha t - \tau) dt.$$

Pode-se observar aqui uma importante propriedade. Independentemente da escolha da função wavelet, para uma mesma escala $j = k$, a correlação $E(d_j[n]d_j[m])$ dos coeficientes de detalhes só depende de $n - m$, ou seja, para cada escala j a seqüência $\{d_j[n], n \in Z\}$ é estacionária.

Para poder obter expressões explícitas das funções de auto-correlação envolvidas em nosso estudo, vamos considerar a wavelet de Haar, dada por

$$\psi(t) = \begin{cases} +1 & \text{para } 0 \leq t < 1/2 \\ -1 & \text{para } 1/2 \leq t < 1 \\ 0 & \text{caso contrário,} \end{cases}$$

da qual é obtida a base ortonormal $2^{-j/2}\psi(2^{-j}t - n)$, $n \in Z$, para os subespaços W_j da análise de multiresolução [8].

Usando então a Eq. (5.1) com a wavelet de Haar, obtemos a função de auto-correlação da seqüência dos coeficientes de detalhes na escala j , para $n \neq 0$ [11], dada por

$$\begin{aligned} R_j(n) &= E[d_j[m+n]d_j[m]] \\ &= \frac{\sigma^2}{2(2H+1)(2H+2)} \xi[n](2^j)^{2H+1}, \end{aligned} \quad (5.2)$$

com

$$\begin{aligned} \xi[n] &= [|n| - 1]^{2H+2} - 4[|n| - (1/2)]^{2H+2} + 6|n|^{2H+2} + \\ &+ [|n| + 1]^{2H+2} - 4[|n| + (1/2)]^{2H+2}. \end{aligned} \quad (5.3)$$

A variância dos coeficientes de detalhes é dada por

$$\begin{aligned} R_j(0) &= \text{Var}(d_j[n]) \\ &= \frac{\sigma^2}{2} V_\psi(H)(2^j)^{2H+1}, \end{aligned} \quad (5.4)$$

com

$$V_\psi(H) = \frac{1 - 2^{-2H}}{(H+1)(2H+1)}. \quad (5.5)$$

A variância dos *coeficientes de aproximação* da transformada, $a_J[n]$, é dada por [11]

$$\text{Var}(a_J[n]) = \frac{\sigma^2}{2} \left[- \int_{-\infty}^{\infty} (\gamma_\phi(\tau) - 2\phi(\tau - n)) |\tau|^{2H} d\tau \right] (2^J)^{2H+1}, \quad (5.6)$$

onde

$$\gamma_\phi(\tau) = \int_{-\infty}^{\infty} \phi(t)\phi(t - \tau)dt.$$

A seguir vamos calcular $\text{Var}(a_J[n])$ para a wavelet de Haar. Da equação anterior temos que

$$\text{Var}(a_J[n]) = (2^J)^{2H+1} \frac{\sigma^2}{2} \left[- \int_{-\infty}^{\infty} \gamma_\phi(\tau) |\tau|^{2H} d\tau + \int_{-\infty}^{\infty} 2\phi(\tau - n) |\tau|^{2H} d\tau \right].$$

A primeira integral é

$$\begin{aligned} I &= -2 \int_0^1 (1 - \tau) \tau^{2H} d\tau \\ &= -2 \left(\frac{1}{2H+1} - \frac{1}{2H+2} \right) = -\frac{2}{(2H+1)(2H+2)}. \end{aligned}$$

A segunda integral é dada por

$$\begin{aligned} II &= 2 \int_n^{n+1} \tau^{2H} d\tau \\ &= 2 \left[\frac{(n+1)^{2H+1} - n^{2H+1}}{2H+1} \right]. \end{aligned}$$

Portanto,

$$\text{Var}(a_J[n]) = (2^J)^{2H+1} \sigma^2 \left[-\frac{1}{(2H+1)(2H+2)} + \frac{(n+1)^{2H+1} - n^{2H+1}}{2H+1} \right].$$

Pode-se observar que a sequência de coeficientes de aproximação $a_J[n]$ tem uma variância que aumenta com n , ou seja, ela é a responsável pela componente não estacionária do fBm. Note que as sequências $d_j[n]$ representam os incrementos do fBm em diferentes escalas de tempo j . Em razão da estacionariedade de $d_j[n]$, nós estaremos interessados aqui somente na predição a N passos destas sequências, que pode ser obtida com um preditor de Wiener de ordem p em cada escala.

5.3. Estimação do parâmetro H e da variância σ^2 de uma série auto-similar

Através de uma decomposição em wavelets de um fBm, pode-se calcular a variância dos coeficientes de detalhes $d_j[n]$, que resulta em

$$\text{Var}(d_j[n]) = \frac{\sigma^2}{2} V_\psi(H) (2^j)^{2H+1},$$

onde $V_\psi(H)$ é dependente da wavelet ψ usada na decomposição. Para o caso da wavelet de Haar, pode-se demonstrar que [11]

$$V_\psi(H) = \frac{1 - 2^{-2H}}{(H + 1)(2H + 1)}.$$

Portanto, aplicando o \log_2 à $\text{Var}(d_j[n])$, obtemos uma reta em função da escala j , dada por

$$\log_2[\text{Var}(d_j[n])] = (2H + 1)j + [\log_2 \sigma^2 + \log_2(\frac{V_\psi(H)}{2})],$$

ou seja,

$$\log_2[\text{Var}(d_j[n])] = aj + b,$$

sendo a a declividade da reta e b a sua ordenada em relação à origem.

Fazendo-se uma regressão linear de mínimos quadrados, é possível:

1. Estimar a declividade da reta e a partir dela obter uma estimação \hat{H} do parâmetro H .
2. Com a estimação \hat{H} calcular $V_\psi(\hat{H})$.
3. Com $V_\psi(\hat{H})$ e a estimação da ordenada à origem \hat{b} , estimar a variância σ^2 .

As vantagens deste método para a estimação de H são as seguintes [19]:

1. É um estimador não polarizado de H .
2. É um estimador eficiente. Devido à menor correlação nos coeficientes wavelets que no processo original, a convergência é mais rápida. Já foi demonstrado que sua variância é a menor possível, atingindo o limite de Cramer-Rao para um dado J .
3. Os estimadores tradicionais no domínio do tempo têm desempenho ruim (uma alta polarização e uma variância sub-ótima) devido às dependências de longo prazo do processo.
4. Os estimadores tradicionais no domínio da frequência se baseiam no comportamento espectral próximo da frequência zero, estimando o H através da média de periodogramas. Esses estimadores têm alta polarização devido a que a análise com $\Delta f = \text{constante}$ não está adaptada ao sinal $1/f$. Por outro lado, o estimador baseado em wavelets pode ser considerado como uma estimação espectral com $f/\Delta f = \text{constante}$, que se adapta perfeitamente ao processo auto-similar $1/f$.
5. O estimador de Whittle é uma aproximação do estimador de máxima verossimilhança (MLE). Tem a desvantagem de ser um método paramétrico.

5.3.1. Resultados das simulações

A fim de avaliar esses estimadores, foram realizadas neste trabalho distintas simulações com sequências de fBm com $H = 0.8$ e $\sigma^2 = 1.19$, e comprimento de 16384 amostras cada uma, geradas usando o “random midpoint displacement method” [4]. Fazendo uma decomposição em wavelets de Haar, foi calculada a variância dos coeficientes de detalhes $d_j[n]$ em função da escala j e, através de uma regressão linear de mínimos quadrados, foram estimados os parâmetros H e σ^2 . Na Tabela 5.1 são mostrados os valores obtidos de \hat{H} , bem como os de \hat{b} e $\hat{\sigma}^2$ para as distintas simulações. Também são mostrados a média, o desvio padrão e o erro relativo (desvio padrão)/média nas Tabelas 5.2, 5.3 e 5.4, respectivamente.

\hat{H}	\hat{b}	$\hat{\sigma}^2$	escalas usadas
0.8033	-3.4516	1.2794	4-11
0.8371	-3.6243	1.1604	4-11
0.7581	-2.8514	1.8848	4-11
0.7967	-3.2728	1.4421	4-11
0.8079	-3.3751	1.3531	4-11

Tabela 5.1: Valores obtidos de \hat{H} , ordenada à origem (\hat{b}) e $\hat{\sigma}^2$ para distintas simulações.

\hat{H}	\hat{b}	$\hat{\sigma}^2$	escalas usadas
0.8006	-3.3150	1.4240	4-11

Tabela 5.2: Médias.

\hat{H}	\hat{b}	$\hat{\sigma}^2$	escalas usadas
0.0283	0.2892	0.2775	4-11

Tabela 5.3: Desvios padrões.

\hat{H}	\hat{b}	$\hat{\sigma}^2$	escalas usadas
0.0354	-0.0873	0.1949	4-11

Tabela 5.4: Desvio padrão/Média.

Pode-se observar que os erros relativos nas estimações do H e da ordenada à origem são muito pequenos (3 % e 8 %, respectivamente).

O erro relativo na estimação da variância σ^2 é maior (20 %). Isto pode ser justificado da maneira a seguir. A estimação de σ^2 é obtida como

$$\hat{\sigma}^2 = 2^{\hat{b}} \frac{2}{V_{\psi}(\hat{H})}.$$

$$\frac{d\hat{\sigma}^2}{d\hat{b}} = 2^{|b|} \big|_{b=-3.553} \ln 2 \frac{2}{V_{\psi}(H)|_{H=0.8}} = 0.82,$$

já que os valores teóricos são $H = 0.8$ e $b = -3.553$. Ou seja,

$$\Delta\hat{\sigma}^2 \approx \Delta\hat{b},$$

como foi observado nas simulações, já que os desvios padrões foram comparáveis. Mas, como

$$\sigma^2 = 1.19 \ll |b| = 3.553,$$

então resulta

$$\frac{\Delta\hat{\sigma}^2}{\sigma^2} \gg \frac{\Delta\hat{b}}{b}.$$

Portanto, a convergência na estimação de σ^2 é mais lenta do que a convergência na estimação do H .

Foram realizados os cálculos, para as mesmas séries acima, usando a wavelet de Daubechies de ordem 4, mas não foram obtidos melhores resultados nas estimações de H e σ^2 do que usando a wavelet de Haar. Isto está de acordo com os resultados da literatura, mas é possível melhorar as estimações com a wavelet de Haar, incluindo pesos na regressão linear de mínimos quadrados [13], como é mostrado a seguir.

Estimação de H e σ^2 usando pesos

Quando o nível j aumenta, menos amostras são disponíveis e portanto a variância na estimação de $Var(d_j[n])$ aumenta. É possível obter um melhor estimador fazendo uma regressão linear ponderada de mínimos quadrados usando os pesos [13]

$$S_j = \frac{(N_0 \ln^2 2)}{2^{j+1}}.$$

Foram realizadas distintas simulações com as mesmas sequências de fBm acima com $H = 0.8$ e $\sigma^2 = 1.19$, e comprimento de 16384 amostras cada uma, geradas usando o “random midpoint displacement method”. Na Tabela 5.5 são mostrados os valores obtidos de \hat{H} , bem como os de \hat{b} e $\hat{\sigma}^2$ para as distintas simulações. Também são mostrados a média, o desvio padrão e o erro relativo (desvio padrão)/média nas Tabelas 5.6, 5.7 e 5.8, respectivamente.

\hat{H}	\hat{b}	$\hat{\sigma}^2$	escalas usadas
0.8097	-3.5174	1.2274	4-11
0.8338	-3.5836	1.1910	4-11
0.8089	-3.4750	1.2634	4-11
0.8221	-3.5794	1.1853	4-11
0.8112	-3.4180	1.3163	4-11

Tabela 5.5: Valores obtidos de \hat{H} , ordenada à origem (\hat{b}) e $\hat{\sigma}^2$, para distintas simulações. Foi utilizada uma regressão linear ponderada.

\hat{H}	\hat{b}	$\hat{\sigma}^2$	escalas usadas
0.8171	-3.5147	1.2367	4-11

Tabela 5.6: Médias.

\hat{H}	\hat{b}	$\hat{\sigma}^2$	escalas usadas
0.0107	0.0705	0.0545	4-11

Tabela 5.7: Desvios padrões.

\hat{H}	\hat{b}	$\hat{\sigma}^2$	escalas usadas
0.0131	-0.0201	0.0440	4-11

Tabela 5.8: Desvio padrão/Média.

Pode-se observar que, incluindo os pesos, os erros relativos nas estimações do H e da ordenada à origem, como também na estimação da variância, são muito pequenos (1 % , 2 % e 4 %, respectivamente).

5.4. Banco de preditores de Wiener multi-escala usando wavelets

O outro problema interessante, mencionado acima, é a obtenção da predição a N passos das seqüências estacionárias $d_j[n]$. Conforme justificado, usaremos preditores de Wiener com p coeficientes, isto é,

$$\hat{d}_j[n + N/n] = \sum_{l=0}^{p-1} h_j^o(l) d_j[n - l], \quad j = 1, \dots, J.$$

Os coeficientes ótimos dos preditores de Wiener são dados por

$$\mathbf{h}_j^o = \mathbf{R}_{d_j}^{-1} \mathbf{r}_{d_j}, \quad (5.7)$$

onde

$$\mathbf{h}_j^o = [h_j(0), h_j(1), \dots, h_j(p-1)]^t$$

$$\mathbf{r}_{d_j} = [R_j(N), R_j(N+1), \dots, R_j(N+p-1)]^t$$

$$\mathbf{R}_{d_j} = \begin{bmatrix} R_j(0) & R_j(1) & \dots & R_j(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_j(p-1) & R_j(p-2) & \dots & R_j(0) \end{bmatrix},$$

H	$NMSE$ teórico	$NMSE$ simulado
0.50	1.0000	1.0000
0.55	0.9997	0.9978
0.60	0.9982	0.9945
0.65	0.9937	0.9894
0.70	0.9829	0.9806
0.75	0.9599	0.9638
0.80	0.9149	0.9291
0.85	0.8313	0.8553
0.90	0.6826	0.7033
0.95	0.4270	0.4207

Tabela 5.9: Valores teóricos do $NMSE$ e os obtidos nas simulações para predição a 5 passos de $d_j[n]$ para diferentes valores do parâmetro H . Foi usado $p = 6$.

e $R_j(n)$ é a função de auto-correlação das sequências de coeficientes de detalhes do fBm $B_H(k)$, dada por (5.2), (5.3), (5.4) e (5.5).

O mínimo erro quadrático médio de predição na escala j é dado por

$$\sigma_j^2 \triangleq E[(d_j[n + N] - \hat{d}_j[n + N/n])^2] = R_j(0) - [h_j^0]^t r_{d_j}, \quad j = 1, \dots, J,$$

onde $R_j(0)$ é dado por (5.4) e (5.5).

Devido ao fato das sequências $d_j[n]$ serem estacionárias, vamos usar neste caso o seguinte $NMSE$

$$NMSE = \frac{\sigma_j^2}{R_j(0)} = 1 - \frac{[h_j^0]^t r_{d_j}}{R_j(0)}.$$

Como consequência da auto-similaridade do processo, o $NMSE$ da predição é o mesmo em todas as escalas.

Na Tabela 5.9 são mostrados os valores teóricos do $NMSE$ e os obtidos nas simulações com 2048 amostras de fBm geradas com o “random midpoint displacement method”, para diferentes valores do parâmetro H .

Portanto, fica assim demonstrado que o esquema desenvolvido permite a obtenção de uma predição completa e em paralelo dos incrementos do fBm em diferentes instantes e escalas de tempo. Foi observada uma grande coincidência entre os valores teóricos e os obtidos em simulações para o $NMSE$.

Em resumo, a análise wavelet:

1. Fornece uma estimaco do parâmetro H do fBm, que caracteriza o grau de auto-similaridade do tráfego.
2. Fornece uma estimaco da variânca do fBm, que caracteriza o grau de variabilidade do tráfego.
3. Permite as prediões do tráfego no espao tempo-escala, que podem ser úteis em funões de gerenciamento de tráfego em tempo real como mecanismos de controle de congestionamento dinâmicos.

Capítulo 6

Policciamento de Tráfego com Filtro de Kalman

6.1. Introdução

Foi demonstrado no Capítulo 3 que o algoritmo proposto de controle de tráfego “leaky bucket” apresenta os seguintes problemas para tráfego com dependência de curto prazo (SRD):

- Os parâmetros do “leaky bucket” dependem fortemente das características da fonte a ser controlada.
- No caso do controle da taxa de bit média, é requerido um limiar muito alto no contador do “leaky bucket” para se obter uma probabilidade de perda aceitável em condições nominais. Isto implica que são necessários tempos muito grandes para se detectar uma violação da taxa de bit média. Este problema é tanto mais grave quanto maior o grau de surto do serviço (ou seja com tráfego muito “bursty”).
- O comprimento do surto não pode ser controlado pelo “leaky bucket” sempre que a fonte incrementar na mesma proporção as durações do surto e de silêncio.

Como é de se esperar, para tráfego real auto-similar, com dependência de longo prazo (LRD), os problemas do algoritmo “leaky bucket” aumentam significativamente. Além disso, com tráfego auto-similar as estimações mais elementares dos parâmetros estatísticos do tráfego devem ser revisadas devido à presença de LRD [14]. Isto decorre da forte correlação entre amostras distantes do processo, que produz uma lenta convergência das médias temporais, acarretando uma lenta detecção da violação dos parâmetros negociados.

Portanto, deve-se procurar uma técnica de controle de tráfego com a simplicidade do “leaky bucket”, mas adequada para controlar tráfego LRD.

Neste trabalho consideramos o modelo de tráfego Browniano fracionário proposto por Norros [1]

$$A(t) = mt + \sqrt{am}B_H(t);$$

onde $B_H(t)$ é um fBm normalizado (com $\sigma^2 = 1$).

Lembremos que o processo $A(t)$ possui três parâmetros básicos, H , m e a , com as seguintes interpretações: $m > 0$ é a taxa média de entrada, $a > 0$ é um coeficiente de variância e $0.5 \leq H < 1$ é o parâmetro de Hurst de $B_H(t)$.

Na prática, é possível medir com a função de controle de tráfego na UNI, a quantidade de tráfego recebida, por exemplo, em células, durante um certo intervalo ΔT , quantidade esta que é dada por

$$A(t + \Delta T) - A(t) = m\Delta T + \sqrt{am}(B_H(t + \Delta T) - B_H(t)),$$

ou, na forma discreta se $t = n\Delta T$ e com ΔT subentendido,

$$A(n + 1) - A(n) = m\Delta T + \sqrt{am}(B_H(n + 1) - B_H(n)).$$

Chamando

$$K_1 = m\Delta T,$$

$$K_2 = \sqrt{am},$$

$$Y(n) = A(n) - A(n - 1),$$

$$i_H(n) = B_H(n) - B_H(n - 1),$$

temos que o tráfego medido é dado por

$$Y(n) = K_1 + K_2 i_H(n). \tag{6.1}$$

Portanto, seria desejável atingir o seguinte:

1. A partir dos valores medidos de $Y(n)$, estimar os três parâmetros que caracterizam o tráfego oferecido pela fonte, isto é, K_1 , K_2 e H .
2. Obter uma predição da quantidade de tráfego $\hat{Y}(n + 1/n)$, a ser recebida durante o próximo intervalo ΔT , para que, baseada nos valores medidos e na predição, a função de controle de tráfego possa decidir o que fazer com as células daquela fonte.

Deve-se enfatizar que o estudo de Norros [1], de predição de tráfego, considera que os parâmetros a , m e H são conhecidos. Neste caso, o problema da predição do tráfego é reduzido ao problema da predição do fBm estudado no Capítulo 4.

Na verdade, estes parâmetros também devem ser estimados e comparados com os valores declarados pelo usuário no contrato de tráfego. Portanto, o objetivo deste capítulo é analisar a possibilidade de se usar um filtro de Kalman para realizar a função de controle de tráfego da rede, estimando o tráfego a ser recebido e os seus outros parâmetros.

A escolha de um filtro de Kalman, em lugar de outro algoritmo como por exemplo o RLS (“recursive least squares”), é devido ao fato de que o filtro de Kalman tem a vantagem de poder aceitar conhecimento “a priori” das variáveis, o que leva a um melhor desempenho e uma maior flexibilidade em relação ao algoritmo RLS. Em nosso caso, o conhecimento “a priori” seria os *valores dos parâmetros declarados pelo usuário no contrato de tráfego*.

6.2. Estrutura da correlação do processo $i(n)$

A partir da função de auto-correlação do fBm normalizado, $B_H(t)$, dada por

$$E[B_H(t)B_H(s)] = \frac{1}{2}[|t|^{2H} + |s|^{2H} - |t - s|^{2H}],$$

é possível se obter a função de auto-correlação do processo $i(n)$, dada por

$$R_i(k) = E[i(n)i(n+k)] = \frac{(\Delta T)^{2H}}{2}[-|k|^{2H} + |k-1|^{2H} + |k+1|^{2H} - |k|^{2H}].$$

A variância do processo $i(n)$ é

$$\sigma_i^2 = R_i(0) = (\Delta T)^{2H}.$$

O fato do parâmetro de Hurst H somente aparecer na estrutura de correlação de $i(n)$, implica que ele não pode ser estimado pelo filtro de Kalman. Mas, como foi analisado no Capítulo 5, existem técnicas eficientes, baseadas em wavelets [11], para estimar o parâmetro H .

Vamos então analisar a possibilidade de se usar um filtro de Kalman para estimar K_1 , K_2 e $i(n)$.

Para viabilizar o uso de um filtro de Kalman, é necessário um modelo em variáveis de estado dos processos envolvidos. Pode-se mostrar [21] que os processos auto-similares $1/f$ não admitem uma representação exata em variáveis de estado. Assim, uma solução possível sub-ótima é aproximar o processo $i(n)$ por um modelo auto-regressivo de ordem p , $AR(p)$, ou seja,

$$i(n) = \sum_{j=1}^p \phi_{p,j} i(n-j) + e(n), \quad (6.2)$$

onde $e(n)$ é um processo de ruído branco com variância σ_N^2 .

A partir de (6.2), pode-se mostrar que a variância do processo $i(n)$ é dada por

$$\sigma_i^2 = \sum_{j=1}^p \phi_{p,j} R_i(j) + \sigma_N^2. \quad (6.3)$$

Por outro lado, podemos definir a seqüência de coeficientes de autocorrelação como

$$r_i(n) = \frac{R_i(n)}{\sigma_i^2}.$$

Para obtermos os parâmetros $\phi_{p,j}$ do modelo AR proposto, devemos resolver as equações de Yule-Walker

$$r_i(n) = \sum_{j=1}^p \phi_{p,j} r_i(n-j), \quad n \geq 1,$$

ou seja,

$$\mathbf{r}_i = \mathbf{R}\boldsymbol{\phi},$$

onde

$$\begin{aligned} \mathbf{r}_i &= [r_i(1), r_i(2), \dots, r_i(p)]^t \\ \mathbf{R} &= \begin{bmatrix} 1 & r_i(1) & \dots & r_i(p-1) \\ \vdots & \vdots & \vdots & \vdots \\ r_i(p-1) & r_i(p-2) & \dots & 1 \end{bmatrix} \\ \boldsymbol{\phi} &= [\phi_{p,1}, \phi_{p,2}, \dots, \phi_{p,p}]^t. \end{aligned}$$

Em alguns exemplos a seguir vamos supor, sem perda de generalidade, que $p = 4$ no modelo AR. Esta suposição nos permite escrever as equações de forma mais simples, mas sua generalização é imediata.

Vamos analisar a seguir três casos de estimação.

6.3. Caso I: estimação de K_1 , K_2 e $i(n)$ com um preditor de Kalman. O parâmetro H é suposto já estimado com a técnica de wavelets

Esta técnica seria, em princípio, a mais razoável. Vamos analisar o que acontece. A observação do preditor é dada por (6.1)

$$Y(n) = K_1 + K_2 i_H(n).$$

Devido à relação não linear entre as variáveis K_2 e $i(n)$, é necessário utilizar um filtro de Kalman estendido (EKF) [18].

Vamos definir o vetor de estados como

$$\mathbf{X}(n) = [i(n), i(n-1), \dots, i(n-p+1), K_1(n), K_2(n)]^t.$$

Baseado na definição de $\mathbf{X}(n)$, o seguinte modelo de estados pode ser obtido

$$\mathbf{X}(n) = \mathbf{A}\mathbf{X}(n-1) + \mathbf{B}V(n-1)$$

$$Y(n) = \mathbf{H}^T(n)\mathbf{X}(n) + u(n),$$

onde

$$\mathbf{B} = [1, 0, \dots, 0]^t$$

$$\mathbf{A} = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_p & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$

$$V(n) = [e(n)]$$

$$\begin{aligned} \mathbf{H}^T(n) &= \left. \frac{\partial h_n(\mathbf{X})}{\partial \mathbf{X}} \right|_{\mathbf{X}=\hat{\mathbf{X}}(n/n-1)} = [x_6(n), 0, \dots, 1, x_1(n)] \Big|_{\mathbf{X}=\hat{\mathbf{X}}(n/n-1)} \\ &= [\hat{x}_6(n/n-1), 0, \dots, 1, \hat{x}_1(n/n-1)] \end{aligned}$$

$$u(n) = h_n(\hat{\mathbf{X}}(n/n-1)) - \mathbf{H}^T(n)\hat{\mathbf{X}}(n/n-1)$$

$$h_n(X) = x_5(n) + x_6(n)x_1(n).$$

As equações do filtro de Kalman estendido são as seguintes [18]:

Ganho de Kalman:

$$\mathbf{L}(n) = \mathbf{P}(n/n-1)\mathbf{H}(n)(\mathbf{H}^T(n)\mathbf{P}(n/n-1)\mathbf{H}(n) + R(n))^{-1}$$

Estimação a priori:

$$\hat{\mathbf{X}}(n+1/n) = \mathbf{A}(\hat{\mathbf{X}}(n/n-1) + \mathbf{L}(n)[Y(n) - h_n(\hat{\mathbf{X}}(n/n-1))])$$

Matriz de covariância do erro da estimação atualizada:

$$\mathbf{P}(n/n) = \mathbf{P}(n/n-1) - \mathbf{L}(n)\mathbf{H}^T(n)\mathbf{P}(n/n-1)$$

Matriz de covariância do erro da estimação a priori:

$$\mathbf{P}(n+1/n) = \mathbf{A}\mathbf{P}(n/n)\mathbf{A}^T + \mathbf{B}Q(n)\mathbf{B}^T,$$

onde, em nosso caso, a matriz de covariância do ruído de observação é

$$R(n) = [0],$$

pois não existe ruído no modelo de observação, e a matriz de covariância do ruído no modelo em variáveis de estado de $i(n)$ é

$$Q(n) = [\sigma_N^2].$$

Portanto, podem-se obter as predições de K_1 , K_2 e $i(n)$ a partir do vetor de estados, como

$$\hat{i}(n+1/n) = \hat{x}_1(n+1/n);$$

$$\hat{K}_1(n+1/n) = \hat{x}_5(n+1/n);$$

$$\hat{K}_2(n+1/n) = \hat{x}_6(n+1/n).$$

Mas, para que as estimativas converjam para os valores corretos, é necessário que o sistema linearizado seja *completamente observável*. A seguir fazemos uma análise do problema da observabilidade

Aplicando o critério de observabilidade dado em [18], temos que o sistema linearizado definido por $[\mathbf{A}, \mathbf{H}]$, é completamente observável se, e somente se, a matriz de observabilidade \mathbf{O} tem Posto completo, isto é,

$$\text{Posto } [\mathbf{O}] = \text{Posto } [\mathbf{H} \ \mathbf{A}^T\mathbf{H} \ (\mathbf{A}^T)^2\mathbf{H} \ \dots \ (\mathbf{A}^T)^5\mathbf{H}] = 6,$$

onde, em nosso caso,

$$\mathbf{H} = [\hat{x}_6(n/n-1), 0, \dots, 1, \hat{x}_1(n/n-1)]^T,$$

$$\mathbf{A}^T = \begin{bmatrix} \phi_1 & 1 & 0 & \dots & 0 & 0 \\ \phi_2 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_p & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}.$$

Mas a matriz de observabilidade \mathbf{O} tem a forma

$$\mathbf{O} = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & 1 & 1 & 1 \\ \hat{x}_1(n/n-1) & \hat{x}_1(n/n-1) & \hat{x}_1(n/n-1) & \hat{x}_1(n/n-1) & \hat{x}_1(n/n-1) & \hat{x}_1(n/n-1) \end{bmatrix}.$$

Dado que as últimas duas linhas da matriz \mathbf{O} são iguais, temos que

$$\text{Posto}[\mathbf{O}] < 6$$

e, portanto, o sistema não é totalmente observável, ou seja, não é possível estimar K_1 , K_2 e $i(n)$ a partir da única observação $Y(n)$.

Portanto, uma possível solução é estimar com um *preditor linear* de Kalman somente K_1 e $i(n)$, já que neste caso o sistema é completamente observável, como pode ser verificado facilmente. Os parâmetros H e K_2 podem ser estimados com a técnica de wavelets, como foi analisado no Capítulo 5. Assim, propomos esta técnica como o seguinte caso.

6.4. Caso II: estimação de K_1 e $i(n)$ com um preditor de Kalman. Os parâmetros H e K_2 são supostos já estimados com a técnica de wavelets

A observação do preditor é dada por (6.1)

$$Y(n) = K_1 + K_2 i_H(n),$$

onde neste caso K_2 é conhecido.

Devido a que agora a relação entre as variáveis K_1 e $i(n)$ é linear, pode-se usar um filtro de Kalman ordinário (linear).

Vamos definir o vetor de estados como

$$\mathbf{X}(n) = [i(n), i(n-1), \dots, i(n-p+1), K_1(n)]^t.$$

Baseado na definição de $\mathbf{X}(n)$, o seguinte modelo de estados pode ser obtido

$$\mathbf{X}(n) = \mathbf{A}\mathbf{X}(n-1) + \mathbf{B}V(n-1)$$

$$Y(n) = \mathbf{H}^T\mathbf{X}(n),$$

onde

$$\mathbf{B} = [1, 0, \dots, 0]^t,$$

$$\mathbf{A} = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_p & 0 \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix},$$

$$V(n) = [e(n)],$$

$$\mathbf{H}^T = [K_2, 0, \dots, 1].$$

As equações do filtro de Kalman são então dadas por

$$\mathbf{L}(n) = \mathbf{P}(n/n-1)\mathbf{H}(n)(\mathbf{H}^T\mathbf{P}(n/n-1)\mathbf{H} + R(n))^{-1},$$

$$\hat{\mathbf{X}}(n+1/n) = \mathbf{A}(\hat{\mathbf{X}}(n/n-1) + \mathbf{L}(n)[Y(n) - \mathbf{H}^T\hat{\mathbf{X}}(n/n-1)]),$$

$$\mathbf{P}(n/n) = \mathbf{P}(n/n-1) - \mathbf{L}(n)\mathbf{H}^T\mathbf{P}(n/n-1),$$

$$\mathbf{P}(n+1/n) = \mathbf{A}\mathbf{P}(n/n)\mathbf{A}^T + \mathbf{B}Q(n)\mathbf{B}^T,$$

onde, em nosso caso,

$$R(n) = [0]$$

$$Q(n) = [\sigma_N^2].$$

Portanto, podem-se obter as predições de K_1 e $i(n)$ a partir do vetor de estados, como

$$\hat{i}(n+1/n) = \hat{x}_1(n+1/n),$$

$$\hat{K}_1(n+1/n) = \hat{x}_5(n+1/n).$$

6.5. Caso III: estimação de $Y(n)$ com um preditor de Kalman. Todos os parâmetros H , K_1 e K_2 são supostos desconhecidos.

A correlação do processo de tráfego, dado por Eq. (6.1)

$$Y(n) = K_1 + K_2 i_H(n),$$

pode ser estimada a partir de N medições de $Y(n)$, como

$$\hat{R}_Y(k) = \frac{1}{N} \sum_{n=1}^{N-k} Y(n)Y(n+k), \quad k = 0, 1, \dots, p. \quad (6.4)$$

O objetivo é usar um filtro de Kalman para prever o processo tráfego $Y(n)$. Analogamente, é necessário desenvolver um modelo em variáveis de estado para o processo de tráfego $Y(n)$.

A diferença em relação aos casos I e II, onde foi usado um modelo AR para aproximar só o termo $i(n)$, é que aqui usamos um modelo AR para aproximar o processo tráfego $Y(n)$ completo, dado que não são conhecidos os parâmetros K_1 e K_2 .

Vamos aproximar $Y(n)$ por um modelo auto-regressivo de ordem p , $AR(p)$, ou seja,

$$Y(n) = \sum_{j=1}^p c_{p,j} Y(n-j) + e(n) \quad (6.5)$$

onde $e(n)$ é um processo de ruído branco com variância σ_N^2 .

A partir de (6.5) pode-se mostrar que a variância do processo $Y(n)$ é dada por

$$\sigma_Y^2 = \sum_{j=1}^p c_{p,j} R_Y(j) + \sigma_N^2. \quad (6.6)$$

Podemos definir a sequência de coeficientes de autocorrelação como

$$r_Y(n) = \frac{R_Y(n)}{\sigma_Y^2},$$

onde $R_Y(n)$ são calculados usando (6.4).

Para obtermos os parâmetros $c_{p,j}$ do modelo AR proposto, devemos resolver as equações de Yule-Walker

$$r_Y(n) = \sum_{j=1}^p c_{p,j} r_Y(n-j), \quad n \geq 1,$$

ou seja,

$$\mathbf{r}_Y = \mathbf{R}\mathbf{c},$$

onde

$$\mathbf{r}_Y = [r_Y(1), r_Y(2), \dots, r_Y(p)]^t,$$

$$\mathbf{R} = \begin{bmatrix} 1 & r_Y(1) & \dots & r_Y(p-1) \\ \vdots & \vdots & \vdots & \vdots \\ r_Y(p-1) & r_Y(p-2) & \dots & 1 \end{bmatrix},$$

$$\mathbf{c} = [c_{p,1}, c_{p,2}, \dots, c_{p,p}]^t.$$

Vamos definir o vetor de estados como

$$\mathbf{X}(n) = [Y(n), Y(n-1), \dots, Y(n-p+1)]^t.$$

Baseado na definição de $\mathbf{X}(n)$, o seguinte modelo de estados pode ser obtido

$$\mathbf{X}(n) = \mathbf{A}\mathbf{X}(n-1) + \mathbf{B}V(n-1),$$

$$Y(n) = \mathbf{H}^T \mathbf{X}(n),$$

onde

$$\mathbf{B} = [1, 0, \dots, 0]^t,$$

$$\mathbf{A} = \begin{bmatrix} c_{p,1} & c_{p,2} & \dots & c_{p,p} \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix},$$

$$V(n) = [e(n)],$$

$$\mathbf{H}^T = [1, 0, \dots, 0].$$

As equações do filtro de Kalman são as seguintes

$$\mathbf{L}(n) = \mathbf{P}(n/n-1)\mathbf{H}(n)(\mathbf{H}^T\mathbf{P}(n/n-1)\mathbf{H} + R(n))^{-1},$$

$$\hat{\mathbf{X}}(n+1/n) = \mathbf{A}(\hat{\mathbf{X}}(n/n-1) + \mathbf{L}(n)[Y(n) - \mathbf{H}^T\hat{\mathbf{X}}(n/n-1)]),$$

$$\mathbf{P}(n/n) = \mathbf{P}(n/n-1) - \mathbf{L}(n)\mathbf{H}^T\mathbf{P}(n/n-1),$$

$$\mathbf{P}(n+1/n) = \mathbf{A}\mathbf{P}(n/n)\mathbf{A}^T + \mathbf{B}Q(n)\mathbf{B}^T,$$

onde, em nosso caso,

$$R(n) = [0],$$

$$Q(n) = [\sigma_N^2].$$

Portanto, pode-se obter a predição do tráfego $Y(n)$, a partir do vetor de estados, como

$$\hat{Y}(n+1/n) = \hat{x}_1(n+1/n).$$

6.6. Resultados das simulações

A fim de testar os modelos obtidos, foram realizadas simulações para os casos II e III acima. Foi usado $p = 4$ em todos os modelos AR , pois este valor fornece uma boa aproximação para a dependência de longo prazo dos processos auto-similares encontrados na prática.

Para o caso II (em que os parâmetros H e K_2 são conhecidos ou bem estimados com outra técnica):

Foi gerada uma série auto-similar $i_H(n)$ com o algoritmo “random midpoint displacement method”, com $H = 0.8$ e 2048 amostras. A partir de $i_H(n)$ foram obtidas duas séries de tráfego $Y(n)$:

Série 1 com os parâmetros $m = 50$, $a = 2$ e $\Delta T = 1$, ou seja,

$$Y(n) = 50 + 10 i_H(n).$$

Série 2 com os parâmetros $m = 50$, $a = 15$ e $\Delta T = 1$, ou seja,

$$Y(n) = 50 + 27.386 i_H(n).$$

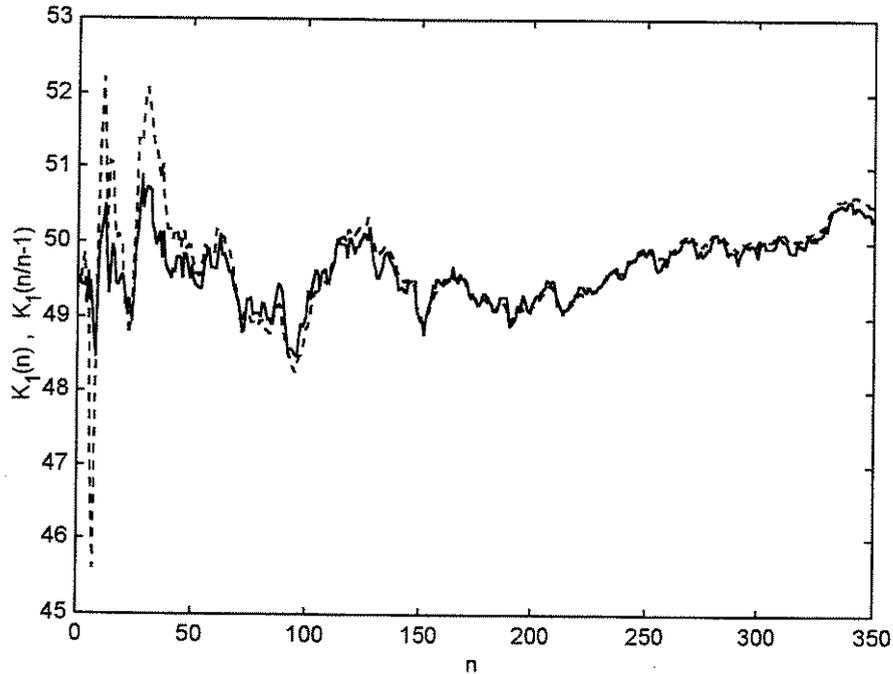


Figura 6.1: Estimação da taxa média para a série 1 com o filtro de Kalman (—) e com o leaky bucket (....).

Simulações com a série 1:

Nas Figuras 6.1 e 6.2 são mostradas a estimação da taxa média m (tanto a obtida com o filtro de Kalman quanto a dada pelo leaky bucket), e o processo $Y(n)$ e sua predição a um passo, respectivamente.

Simulações com a série 2:

Nas Figuras 6.3 e 6.4 são mostradas a estimação da taxa média m (obtida com o filtro de Kalman e com o leaky bucket), e o processo $Y(n)$ e sua predição a um passo, respectivamente.

Na Tabela 6.1 temos a comparação dos valores obtidos nas simulações para as séries 1 e 2 para o valor médio da série, $E[Y(n)]$, a variância da série, $E[(Y(n) - E[Y(n)])^2]$, a variância do erro de predição do tráfego a um passo, $E[(Y(n) - \hat{Y}(n))^2]$, bem para o erro de predição do tráfego normalizado, dado por

$$\frac{E[(Y(n) - \hat{Y}(n))^2]}{E[(Y(n) - E[Y(n)])^2]}$$

Pode-se observar das Figuras 6.1 e 6.3 que a predição da taxa média do tráfego converge mais rápido ao valor verdadeiro se a variância do tráfego é menor, como é razoável. No entanto, na predição a um passo os erros de predição normalizados são os mesmos, como deve ser, já que para as duas séries o H toma o mesmo valor ($H = 0.8$). A pequena diferença é devida aos erros inerentes das simulações.

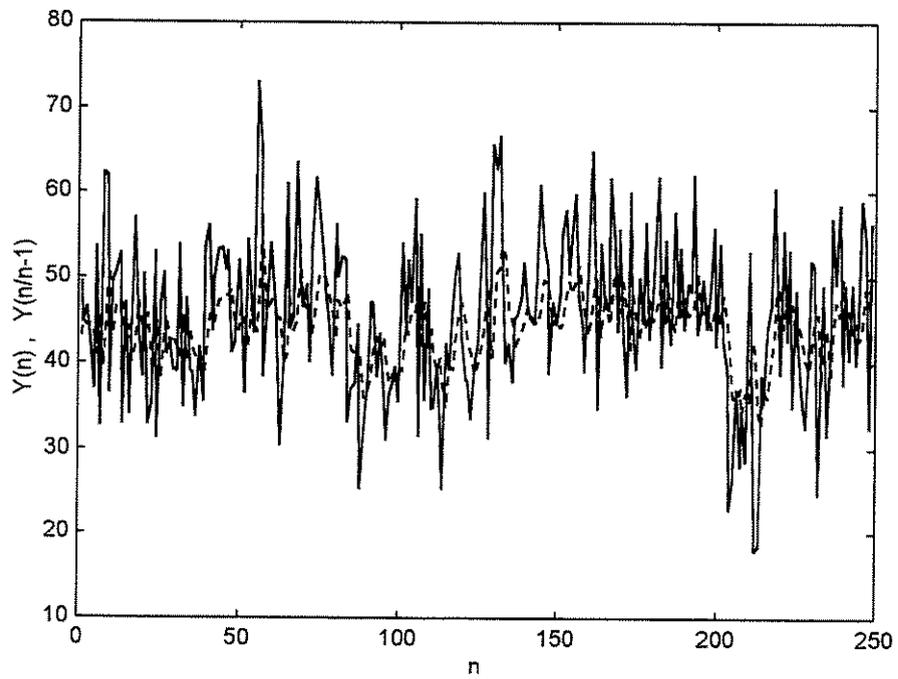


Figura 6.2: Processo $Y(n)$ (—) e sua previsão a um passo (...), para a série 1.

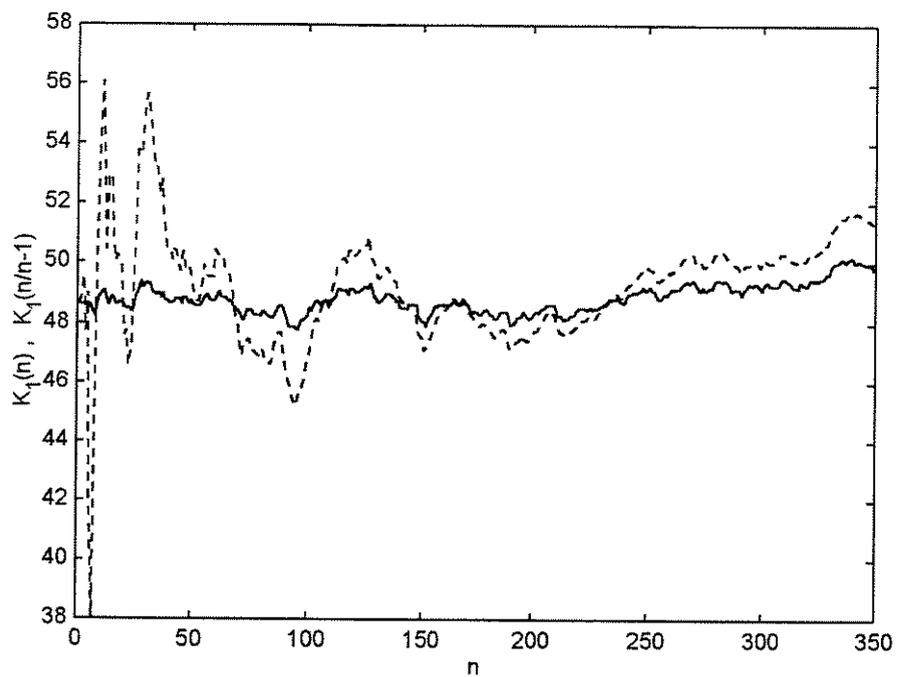


Figura 6.3: Estimação da taxa média para a série 2 com o filtro de Kalman (—) e com o leaky bucket (...).

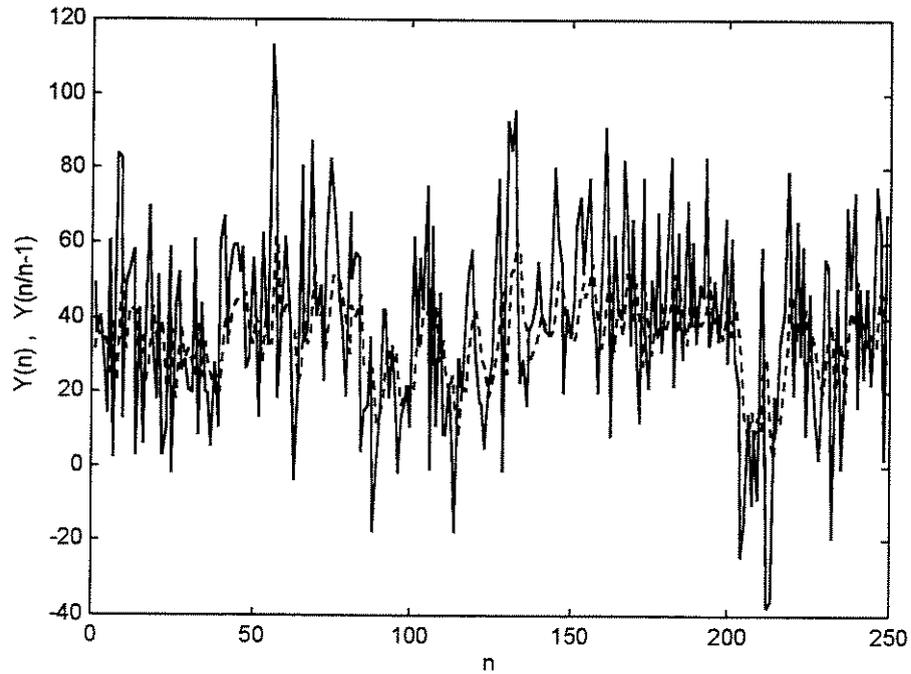


Figura 6.4: Processo $Y(n)$ (—) e sua predição a um passo (...), para a série 2.

série	$E[Y(n)]$	$E[(Y(n) - E[Y(n)])^2]$	$E[(Y(n) - \hat{Y}(n))^2]$	$\frac{E[(Y(n) - \hat{Y}(n))^2]}{E[(Y(n) - E[Y(n)])^2]}$
série 1	52.8970	92.9906	79.0367	0.8499
série 2	57.9337	697.4291	577.7971	0.8284

Tabela 6.1: Comparação dos valores obtidos nas simulações para as séries 1 e 2 no caso II.

série	$E[Y(n)]$	$E[(Y(n) - E[Y(n)])^2]$	$E[(Y(n) - \hat{Y}(n))^2]$	$\frac{E[(Y(n) - \hat{Y}(n))^2]}{E[(Y(n) - E[Y(n)])^2]}$
série 1	52.8970	92.9906	84.7556	0.9114
série 2	57.9337	697.4291	629.5988	0.9027

Tabela 6.2: Comparação dos valores obtidos nas simulações para as séries 1 e 2 no caso III.

Observa-se na predição da taxa média do tráfego que a técnica proposta neste trabalho tem um desempenho superior ao do leaky bucket.

Pode-se observar também o bom desempenho do preditor, ainda que a aproximação $AR(p)$ do processo $i(n)$ tenha sido utilizada. Assim, neste caso é possível predizer com o filtro de Kalman a taxa média de entrada do tráfego e a quantidade de células a ser gerada durante o próximo intervalo ΔT , com a ressalva de que os parâmetros H e K_2 devem ser previamente estimados com a técnica baseada em wavelets.

Para o caso III (em que os parâmetros H , K_1 e K_2 são desconhecidos):

Aqui também foram realizadas simulações com as séries 1 e 2 acima.

Neste caso foram obtidos os valores da Tabela 6.2.

Comparando as Tabelas 6.1 e 6.2, observa-se que o erro de predição normalizado para o caso III é maior do que para o caso II, o que decorre do fato de que dispomos de menos informação no caso III, já que aqui todos os parâmetros do tráfego são desconhecidos.

Outras diversas simulações foram também realizadas com as séries auto-similares denominadas h05.dat, h06.dat, h07.dat, h08.dat e h09.dat, com diferentes valores do H , geradas com o método de Paxson [15]. Na Tabela 6.3 são mostrados os valores obtidos para a variância da série, $E[(Y(n) - E[Y(n)])^2]$, a variância do erro da predição do tráfego a um passo, $E[(Y(n) - \hat{Y}(n))^2]$, o erro de predição do tráfego normalizado pela variância da série $Y(n)$,

$$\frac{E[(Y(n) - \hat{Y}(n))^2]}{E[(Y(n) - E[Y(n)])^2]},$$

bem como para o erro de predição normalizado pelo valor quadrático médio da série $Y(n)$,

$$\frac{E[(Y(n) - \hat{Y}(n))^2]}{E[(Y(n))^2]}.$$

Observa-se que à medida que o parâmetro H da série auto-similar aumenta, o erro de predição diminui, o que é de se esperar já que a série é mais fácil de predizer. Pode-se observar um bom comportamento do preditor quando H é maior do que 0.7, que é justamente o intervalo de valores de H encontrados nas séries de tráfego auto-similar práticas [12]. Foram realizadas simulações com as séries reais de tráfego auto-similar BC-pOct89 e BC-Oct89Ext obtidas no Bellcore e que estão disponíveis na Internet no endereço <http://ita.ee.lbl.gov/html/contrib/BC.html>. Na Tabela 6.4 são mostrados os valores obtidos para a variância da série, $E[(Y(n) - E[Y(n)])^2]$, a variância do erro da predição do tráfego a

série	$E[(Y(n) - E[Y(n)])^2]$	$E[(Y(n) - \hat{Y}(n))^2]$	$\frac{E[(Y(n) - \hat{Y}(n))^2]}{E[(Y(n) - E[Y(n)])^2]}$	$\frac{E[(Y(n) - \hat{Y}(n))^2]}{E[(Y(n))^2]}$
h05	1.5758e5	1.8946e5	1.2023	0.5594
h06	1.5744e5	1.7914e5	1.1378	0.5298
h07	1.5592e5	1.5614e5	1.0014	0.4647
h08	1.5418e5	1.2279e5	0.7964	0.3667
h09	1.6542e5	0.8186e5	0.4949	0.2343

Tabela 6.3: Comparação dos valores obtidos nas simulações para as séries *h05*, *h06*, *h07*, *h08* e *h09* no caso III.

série	$E[(Y(n) - E[Y(n)])^2]$	$E[(Y(n) - \hat{Y}(n))^2]$	$\frac{E[(Y(n) - \hat{Y}(n))^2]}{E[(Y(n) - E[Y(n)])^2]}$
BC-pOct89	3.9201 e8	1.9515 e8	0.4978
BC-Oct89Ext	8.0436 e6	3.4900 e6	0.4339

Tabela 6.4: Valores obtidos nas simulações para as séries BC-pOct89 e BC-Oct89Ext no caso III.

um passo, $E[(Y(n) - \hat{Y}(n))^2]$, e o erro de predição do tráfego normalizado pela variância da série $Y(n)$. Para a série BC-pOct89 foram utilizados intervalos de agregação de 0.1 seg, enquanto que para a série BC-Oct89Ext foi de 1 seg. Na Figura 6.5 é mostrado o processo $Y(n)$ e sua predição a um passo para as primeiras 350 amostras da série BC-Oct89Ext agregada em intervalos de 1 seg.

É observado um desempenho muito bom dos preditores para estas séries reais de tráfego, principalmente se for considerado que todos os parâmetros do tráfego (H , K_1 e K_2) foram considerados desconhecidos nas simulações.

Conclusões

1. Este capítulo utilizou com sucesso uma característica importante do modelo de tráfego Browniano fracionário de Norros, que permite caracterizar o processo de tráfego de forma parcimoniosa, com somente três parâmetros, m , a e H , usando o processo fBm. O objetivo do controle de tráfego consiste em estimar se os valores destes parâmetros do tráfego medido coincidem com os valores declarados pelo usuário no contrato de tráfego.
2. Na literatura, Norros estudou o problema da predição do tráfego Browniano fracionário, mas supondo que os valores dos três parâmetros são conhecidos, caso em que o problema se reduz a prever o fBm. Na prática, todavia, os valores desses três parâmetros devem também ser estimados, para se possa caracterizar totalmente o tráfego do usuário e analisar se ele cumpre ou não o contrato de tráfego.
3. Este capítulo tratou do estudo da possibilidade de se usar um filtro de Kalman para estimar os parâmetros do tráfego. A conclusão final é que se deve usar uma técnica baseada em wavelets para estimar os parâmetros H e K_2 , e então usar um preditor de Kalman para estimar conjuntamente tanto o processo incremento de tráfego $i(n)$ como o parâmetro de taxa média do tráfego. Observou-se também que, no caso III onde todos os parâmetros do tráfego foram considerados desconhecidos, foi possível se obter boas

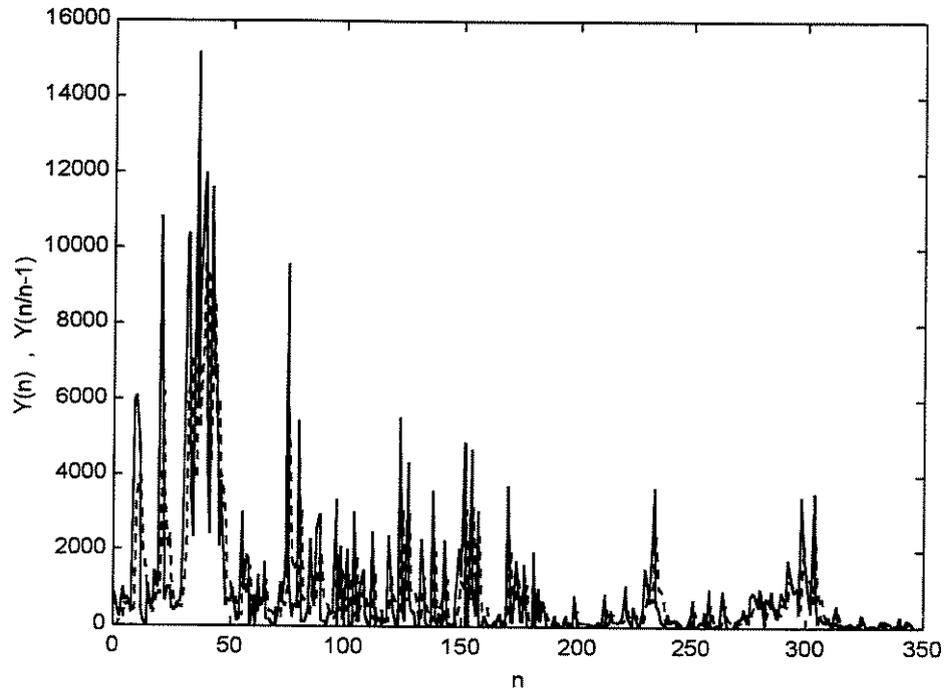


Figura 6.5: Processo $Y(n)$ (—) e sua previsão a um passo (...), para as primeiras 350 amostras da série BC-Oct89Ext agregada em intervalos de 1 seg.

previsões do processo de tráfego $Y(n)$ usando previsão de Kalman, sobretudo quando o valor de H é maior que 0.7, que é justamente o intervalo de valores de H encontrados nas séries de tráfego auto-similar práticas [12]. Isto foi verificado para as séries reais de tráfego auto-similar BC-pOct89 e BC-Oct89Ext do Bellcore.

Capítulo 7

Estimação Sequencial com Transformada de Wavelet de Baixa Complexidade

7.1. Introdução

Como é conhecido, a DFT (Discrete Fourier Transform), com suas importantes aplicações em análise espectral, tem complexidade de implementação de $O(N^2)$, onde N é quantidade de amostras. Foi com o descobrimento do algoritmo FFT, de implementação da DFT, com complexidade $O(N \log N)$ que a implementação da DFT foi estendida amplamente. Analogamente, hoje a decomposição em wavelets com transformadas rápidas, é muito utilizada nas áreas de processamento de sinais, telecomunicações, medicina, etc.

Em muitas aplicações torna-se necessaria fazer uma decomposição em wavelets na forma sequencial, processando em tempo real blocos de N amostras, onde N é, em geral, um número grande. Como é conhecido [20], a DFWT (Discrete Fast Wavelet Transform) tem uma complexidade de apenas $O(N)$. Neste capítulo demonstra-se que para se fazer uma decomposição sequencial em wavelets, existe um número ótimo de amostras K^{opt} que devem ser substituídas no bloco anterior, e que usando esse número ótimo, é possível se obter uma complexidade de apenas $O(\sqrt{N})$ a partir do segundo bloco. Mais precisamente, para o primeiro bloco a complexidade é $O(N)$ e para os demais é $O(\sqrt{N})$. Além disso, demonstra-se a robustez da complexidade com relação ao valor de K , ou seja, para um valor de N fixo é possível escolher K em um intervalo amplo de, no mínimo, quatro potências de dois e manter uma complexidade menor que $10\sqrt{N}$, o que permite adaptar o valor de K^{opt} aos requisitos de uma certa aplicação. Todos estes resultados são válidos para o caso de $L = 2$ e para o caso assintótico de $\frac{\sqrt{N}}{L} \rightarrow \infty$, onde L é o tamanho dos filtros usados na decomposição wavelets. Além disso, para o caso geral de qualquer valor de N e L , demonstra-se uma condição suficiente para K para se ter uma complexidade menor que $10\sqrt{N}$.

K	N_e
2	$\frac{N}{2} - 1$
4	$\frac{N}{2} - 2 + \frac{N}{4} - 1 = \frac{3}{4}N - 3$
8	$\frac{N}{2} - 4 + \frac{N}{4} - 2 + \frac{N}{8} - 1 = \frac{7}{8}N - 7$
....	...

Tabela 7.1: Quantidade de coeficientes de detalhamento (igual à quantidade de coeficientes de aproximação) que não deve ser calculada novamente N_e .

7.2. Complexidade da decomposição em wavelets sequencial usando wavelets de Haar

Vamos considerar, em primeiro lugar, um desenvolvimento em wavelets usando Haar, ou seja, com filtros de comprimento $L = 2$.

Se a quantidade inicial de amostras é N , em cada nível i vamos ter uma quantidade de amostras N_i que é a metade de amostras do nível imediatamente superior, o que acarreta

$$N_i = \frac{N}{2^i}.$$

Seja K a quantidade de novas amostras que chegam e antigas amostras que saem em cada bloco. Se $K = 1$ todos os coeficientes da transformada são alterados e não temos uma redução em complexidade, mas se K é uma potência de 2, então temos uma redução de complexidade. Se $K = 2$, então no primeiro nível do desenvolvimento só um coeficiente muda de valor e, portanto, precisa ser reavaliado. Os demais, todavia, são os mesmos anteriores deslocados. Se $K = 4$, só dois coeficientes do primeiro nível e um coeficiente do segundo nível precisam ser reavaliados, e assim sucessivamente. A quantidade de coeficientes de detalhamento (igual à quantidade de coeficientes de aproximação) que não devem ser calculados novamente, N_e , pois vão ser os coeficientes anteriores deslocados, é mostrada na Tabela 7.1, como pode ser verificado facilmente.

Portanto, temos que em geral a quantidade de coeficientes que não devem ser calculados, é dada por

$$N_e = (K - 1) \frac{N}{K} - (K - 1) = (K - 1) \left(\frac{N}{K} - 1 \right).$$

Neste capítulo vamos medir a complexidade em *quantidade de coeficientes que devem ser calculados*. A quantidade total de coeficientes de aproximação e de detalhamento que devem ser calculados no desenvolvimento em wavelets “in batch” é $2(N - 1)$. Usando o algoritmo sequencial proposto neste trabalho, a quantidade de coeficientes que devem ser calculados é a diferença entre o total de coeficientes $2(N - 1)$ e a quantidade total de coeficientes (detalhamento + aproximação) que não se alteram, $2N_e$, isto é,

$$Complexidade_{seq} = 2(N - 1) - 2(K - 1) \left(\frac{N}{K} - 1 \right). \quad (7.1)$$

Aqui vamos supor que N e K são potências de 2, ou seja,

$$\begin{aligned} N &= 2^r, \\ K &= 2^s. \end{aligned}$$

Portanto, a equação da complexidade (7.1) fica

$$Complexidade_{seq} = 2(2^r - 1) - 2(2^s - 1)(2^{r-s} - 1) = 2^{s+1} + 2^{r-s+1} - 4 \quad (7.2)$$

O objetivo é procurar o valor ótimo de $K = 2^s$, quantidade de amostras que chegam e que saem, para minimizar a complexidade, ou seja, minimizar o número de coeficientes que devem ser re-calculados. Então, tomando a derivada parcial da complexidade, dada por Eq. (7.2), em relação a s , obtemos

$$\begin{aligned} 2^{s+1} \ln 2 - 2^{r-s+1} \ln 2 &= 0, \\ s &= \frac{r}{2}. \end{aligned}$$

Mas s deve ser um número inteiro, o que implica em

- Se r é par:

$$\begin{aligned} s^{opt} &= \frac{r}{2}, \\ K^{opt} &= \sqrt{N}, \end{aligned} \quad (7.3)$$

$$\begin{aligned} Complexidade_{seq} &= 2^{\frac{r}{2}+1} + 2^{\frac{r}{2}+1} - 4 \\ &= 4(\sqrt{N} - 1). \end{aligned} \quad (7.4)$$

- Se r não é par:

- Usando o inteiro superior a $\frac{r}{2}$:

$$\begin{aligned} s^{opt} &= \frac{r+1}{2}, \\ Complexidade_{seq} &= 2^{\frac{r+1}{2}+1} + 2^{\frac{r-1}{2}+1} - 4 \\ &= 2\sqrt{2}\sqrt{N} + 2\frac{1}{\sqrt{2}}\sqrt{N} - 4. \end{aligned}$$

- Usando o inteiro inferior a $\frac{r}{2}$:

$$\begin{aligned} s^{opt} &= \frac{r-1}{2}, \\ Complexidade_{seq} &= 2^{\frac{r-1}{2}+1} + 2^{\frac{r+1}{2}+1} - 4 \\ &= 2\frac{1}{\sqrt{2}}\sqrt{N} + 2\sqrt{2}\sqrt{N} - 4. \end{aligned}$$

Como pode ser observado, é obtida a mesma complexidade em qualquer dos dois casos de aproximação de s a inteiro. Portanto, se r não é par:

$$K^{opt} = \sqrt{N/2} \text{ ou } K^{opt} = \sqrt{2N}, \quad (7.5)$$

$$\begin{aligned} \text{Complexidade}_{seq} &= 2^{\frac{r-1}{2}+1} + 2^{\frac{r+1}{2}+1} - 4 \\ &= (2\sqrt{2} + 2\frac{1}{\sqrt{2}})\sqrt{N} - 4. \end{aligned} \quad (7.6)$$

Com isso obtemos qual é o número ótimo de amostras que devem ingressar e sair para minimizar a complexidade, em função de N : se r é par \sqrt{N} , e se r é impar $\sqrt{N/2}$ ou $\sqrt{2N}$. Assim, fica provado que com este número de amostras a complexidade é $O(\sqrt{N})$ em relação à quantidade total de amostras N .

É interessante demonstrar que se $K = \alpha\sqrt{N}$, onde α é uma constante, também resulta que a complexidade é $O(\sqrt{N})$, pois neste caso

$$\begin{aligned} \text{Complexidade}_{seq} &= 2(N-1) - 2(K-1)\left(\frac{N}{K} - 1\right)|_{K=\alpha\sqrt{N}} \\ &= (2\alpha + \frac{2}{\alpha})\sqrt{N} - 4. \end{aligned}$$

Para um valor fixo de N , vamos procurar o intervalo de valores de α tal que a complexidade obtida com $K = \alpha\sqrt{N}$ é da mesma ordem de magnitude que a complexidade obtida com o K^{opt} dada pelas equações (7.4) e (7.6). Vamos arbitrar, por exemplo, que $\text{Complexidade}_{seq} < 10\sqrt{N}$. Portanto, deve-se ter

$$(2\alpha + \frac{2}{\alpha}) < 10, \quad (7.7)$$

o que implica em

$$0.21\sqrt{N} < K < 4.79\sqrt{N}. \quad (7.8)$$

Portanto, para que $\text{Complexidade}_{seq} < 10\sqrt{N}$, temos para K cinco potências de dois possíveis se r é par ($0.25\sqrt{N}, 0.5\sqrt{N}, \sqrt{N}, 2\sqrt{N}, 4\sqrt{N}$) e quatro potências de dois possíveis se r é impar ($0.5\sqrt{N/2}, \sqrt{N/2}, \sqrt{2N}, 2\sqrt{2N}$), ou seja, existe um intervalo amplo para escolher o valor do K de acordo com as necessidades de uma aplicação específica.

Tudo isto foi provado para um desenvolvimento em wavelets usando Haar, ou seja, com filtros de comprimento $L = 2$. Na próxima seção vamos analisar o caso de L geral.

7.3. Complexidade da decomposição em wavelets sequencial usando wavelets com qualquer L

Consideremos um desenvolvimento em wavelets usando filtros de comprimento L , com L par e $L \geq 2$. Seja K a quantidade de novas amostras que chegam e que saem.

Pode ser verificado facilmente que a quantidade de coeficientes de detalhamento (igual à quantidade de coeficientes de aproximação) que não devem ser calculados novamente, N_e , porque vão ser os coeficientes anteriores deslocados, satisfaz

– Se $K = 2$:

$$N_e = \frac{N}{2} - \left(1 + \frac{L-2}{2}\right).$$

– Se $K = 4$:

$$N_e = \frac{N}{2} - \left(2 + \frac{L-2}{2}\right) + \frac{N}{4} - \left(\left\lceil \frac{2 + \frac{L-2}{2}}{2} \right\rceil + \frac{L-2}{2}\right).$$

– Se $K = 8$:

$$N_e = \frac{N}{2} - \left(4 + \frac{L-2}{2}\right) + \frac{N}{4} - \left(\left\lceil \frac{4 + \frac{L-2}{2}}{2} \right\rceil + \frac{L-2}{2}\right) + \frac{N}{8} - \left[\left[\left\lceil \frac{\left\lceil \frac{4 + \frac{L-2}{2}}{2} \right\rceil + \frac{L-2}{2}}{2} \right\rceil + \frac{L-2}{2}\right].\right.$$

.....

(7.9)

e assim sucessivamente. Foi considerado que o número de coeficientes em todos os níveis i , $i = 1, \dots, \log_2 K$, é maior ou igual do que a quantidade de coeficientes que mudam. Para K próximo a N isto não é satisfeito, e deve-se substituir o termo correspondente ao nível i ,

$$\frac{N}{2^i} - (\dots), \quad (7.10)$$

pelo zero. Devido a este fato e às partes inteiras que aparecem, a expressão geral de N_e em função de K fica muito complexa.

Já vimos que a expressão da complexidade é dada por

$$Complexidade_{seq} = 2(N-1) - 2N_e. \quad (7.11)$$

Vamos, então, procurar um limitante superior e um limitante inferior da $Complexidade_{seq}$ que nos vão permitir obter resultados úteis.

Para obtermos um limitante superior para $Complexidade_{seq}$, procuramos um limitante inferior de N_e . Para isso, em primeiro lugar observamos que em N_e as partes inteiras que aparecem têm a seguinte propriedade

$$\lceil \frac{x}{2} \rceil = \frac{x}{2},$$

se x é par, ou

$$\lceil \frac{x}{2} \rceil = \frac{x}{2} + 0.5,$$

se x é ímpar.

Então, é possível tirar as partes inteiras e ao termo seguinte somar 0.5, resultando em

$$\left(\lceil \frac{x}{2} \rceil + \frac{L-2}{2} \right) \leq \left(\frac{x}{2} + \frac{L-1}{2} \right).$$

Em segundo lugar, é possível aplicar a seguinte desigualdade para os termos $L-2$, que resultam

$$\left(\dots + \frac{L-2}{2} \right) \leq \left(\dots + \frac{L-1}{2} \right).$$

Finalmente, observamos que se para algum termo $\frac{N}{2^i} - (\dots) < 0$, este deveria ser substituído pelo zero, mas se deixamos esse termo, obtemos portanto um limitante inferior de N_e .

Então, a seguinte expressão para o limitante superior de $Complexidade_{seq}$ é obtida

$$\begin{aligned} Complexidade_{seq} &= 2(N-1) - 2N_e \\ &\leq 2(N-1) - 2(K-1)\frac{N}{K} + 2(K-1) + \frac{2(L-1)}{K} \sum_{i=1}^{\log_2 K} i2^{i-1} \\ &= 2(N-1) - 2(K-1)\left(\frac{N}{K} - 1\right) \\ &\quad + \frac{2(L-1)}{K} [K(\log_2 K - 1) + 1] \end{aligned} \tag{7.12}$$

Para obtermos um limitante inferior de $Complexidade_{seq}$, podemos escolher $L=2$, já que como é observado em (7.9), vamos obter um limitante superior de N_e . Então, usando o resultado (7.1) da seção anterior para $L=2$, resulta

$$\begin{aligned} Complexidade_{seq} &= 2(N-1) - 2N_e \\ &\geq 2(N-1) - 2(K-1)\left(\frac{N}{K} - 1\right) \end{aligned} \tag{7.13}$$

Com $K = \alpha\sqrt{N}$, onde α é constante, usando (7.12) e (7.13) obtemos

$$\begin{aligned} \left(2\alpha + \frac{2}{\alpha}\right)\sqrt{N} - 4 &\leq Complexidade_{seq} \\ &\leq \left(2\alpha + \frac{2}{\alpha}\right)\sqrt{N} - 4 + \frac{2(L-1)}{\alpha\sqrt{N}} \\ &\quad + 2(L-1)(\log_2(\alpha\sqrt{N}) - 1). \end{aligned} \tag{7.14}$$

$L; N$	2048	4096	8192	16384	32768	65536
2	16 – 128	16 – 256	32 – 256	32 – 512	64 – 512	64 – 1024
4	16 – 128	16 – 256	32 – 256	32 – 512	64 – 512	64 – 1024
8	16 – 128	16 – 256	32 – 256	32 – 512	64 – 512	64 – 1024
10	16 – 128	16 – 128	32 – 256	32 – 512	64 – 512	64 – 1024
12	16 – 128	16 – 128	32 – 256	32 – 512	64 – 512	64 – 1024
14	16 – 128	16 – 128	32 – 256	32 – 256	64 – 512	64 – 1024
16	16 – 64	16 – 128	32 – 256	32 – 256	64 – 512	64 – 1024
32	32	32 – 64	32 – 128	64 – 256	64 – 512	64 – 512
64	–	–	–	64 – 128	64 – 256	128 – 512
128	–	–	–	–	–	–

Tabela 7.2: Valores do K que resultam em $Complexidade_{seq} < 10\sqrt{N}$ para diferentes valores de N e L .

Para o caso assintótico

$$\frac{\sqrt{N}}{L} \rightarrow \infty, \quad (7.15)$$

os dois limitantes coincidem e, portanto, neste caso

$$Complexidade_{seq} = \left(2\alpha + \frac{2}{\alpha}\right) \sqrt{N},$$

ou seja, com $K = \alpha\sqrt{N}$ a complexidade assintótica é $O(\sqrt{N})$.

Para um dado valor de N que satisfaz (7.15), para que $Complexidade_{seq} < 10\sqrt{N}$ obtemos a mesma condição (7.7) que para o caso de $L = 2$. Ou seja, para o caso assintótico de (7.15) são válidos os mesmos resultados que para o caso de $L = 2$. Ou seja, temos para K cinco potências de dois possíveis se r é par ($0.25\sqrt{N}, 0.5\sqrt{N}, \sqrt{N}, 2\sqrt{N}, 4\sqrt{N}$) e quatro potências de dois possíveis se r é ímpar ($0.5\sqrt{N/2}, \sqrt{N/2}, \sqrt{2N}, 2\sqrt{2N}$).

Para o caso prático de valores de L e N finitos, podemos obter, a partir da desigualdade à direita de (7.14), uma condição suficiente que α deve satisfazer para que $Complexidade_{seq} < 10\sqrt{N}$, qual seja,

$$L - 1 \leq \frac{(10 - 2\alpha - \frac{2}{\alpha}) \sqrt{N} + 4}{2 \left(\log_2(\alpha\sqrt{N}) - 1 + \frac{1}{\alpha\sqrt{N}} \right)}. \quad (7.16)$$

Usando (7.9), (7.10) e (7.11), foram calculados numericamente, na forma exata, para diferentes valores de N e L , os valores de K que resultam em $Complexidade_{seq} < 10\sqrt{N}$. Os resultados são mostrados na Tabela 7.2, onde – significa que para esses valores de N e L não existem valores de K que resultem em $Complexidade_{seq} < 10\sqrt{N}$, e $a - b$ significa o intervalo de potências de dois a partir de a e até b .

É importante enfatizar as seguintes conclusões:

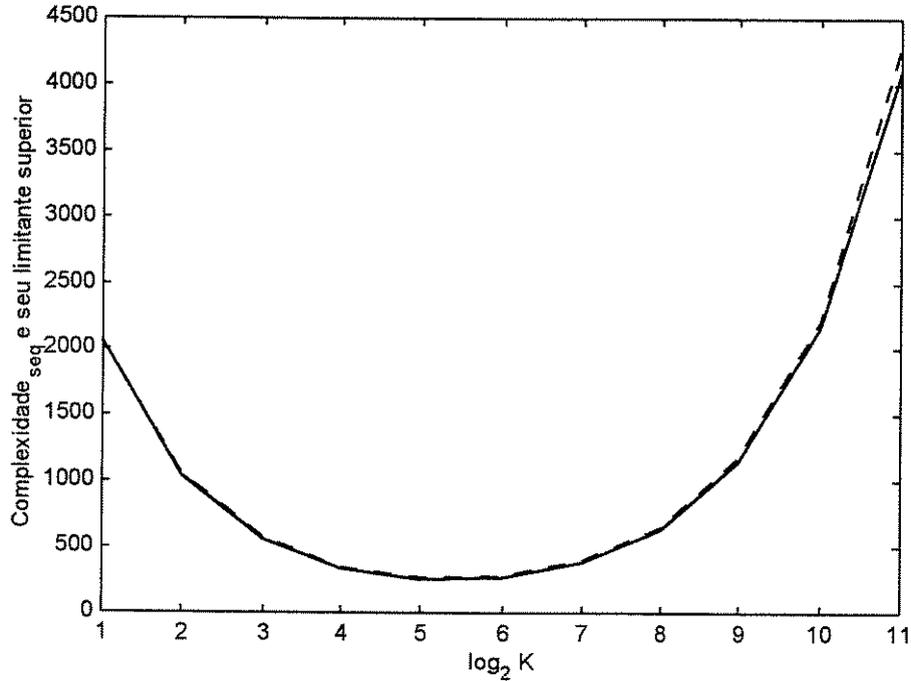


Figura 7.1: $Complexidade_{seq}$ e seu limitante superior para $N = 2048$ e $L = 10$, em função de K .

1. Se $L \ll \sqrt{N}$, o intervalo de valores de K é o mesmo que para o caso de $L = 2$, ou seja, existem cinco potências de dois possíveis para K se r é par ($0.25\sqrt{N}$, $0.5\sqrt{N}$, \sqrt{N} , $2\sqrt{N}$, $4\sqrt{N}$) e quatro potências de dois possíveis se r é ímpar ($0.5\sqrt{N/2}$, $\sqrt{N/2}$, $\sqrt{2N}$, $2\sqrt{2N}$).
2. Mantendo N constante à medida que L aumenta, a $Complexidade_{seq}$ aumenta e o intervalo de valores do K diminui, mas sempre que a condição (7.16) é satisfeita tem-se $Complexidade_{seq} < 10\sqrt{N}$.
3. Em todos os casos da Tabela 7.2 foi observado que (7.16) foi também uma condição necessária para $Complexidade_{seq} < 10\sqrt{N}$. Isto é devido a que (7.12) é um limitante superior, mas muito apertado, para $Complexidade_{seq}$, como é observado na Figura 7.1, onde é mostrada a $Complexidade_{seq}$ e seu limitante superior para $N = 2048$ e $L = 10$, em função de K . O mesmo comportamento acontece para os outros valores de N e L . Para o caso da Figura 7.1, a condição suficiente (7.16) é satisfeita para quatro potências de dois, $K = 16, 32, 64, 128$, com as quais a complexidade é menor que $10\sqrt{N}$, ou seja, $Complexidade_{seq} < 450$, como é observado na Figura 7.1. Isto oferece flexibilidade para escolher o valor de K de acordo com os requisitos de uma certa aplicação e ao mesmo tempo trabalhar com uma baixa complexidade (menor que $10\sqrt{N}$).

7.4. Conclusões

Neste capítulo foi demonstrado que é possível se obter a decomposição em wavelets na forma sequencial, com uma complexidade de $O(\sqrt{N})$, ao invés de $O(N)$, se um número adequado, K , de amostras no bloco anterior for substituído por igual número no bloco seguinte.

Para um valor de N fixo e para o caso de $L = 2$, ou para o caso assintótico de $\frac{\sqrt{N}}{L} \rightarrow \infty$, existem cinco ou quatro potências de dois para o valor de K , correspondendo a $\log_2 K$ par ou ímpar, respectivamente, com os quais a complexidade é menor que $10\sqrt{N}$. Para o caso geral de quaisquer valores de N e L finitos, foi dada uma condição suficiente para α para que, com $K = \alpha\sqrt{N}$, a complexidade seja menor que $10\sqrt{N}$. Foram apresentados cálculos numéricos que confirmaram todos estes resultados. Para os valores de N e L de interesse prático, o intervalo possível de valores de K é amplo, oferecendo flexibilidade para se escolher o valor de K de acordo com os requisitos de uma certa aplicação e ao mesmo tempo trabalhar com uma baixa complexidade (menor que $10\sqrt{N}$).

Capítulo 8

Conclusões

As contribuições originais deste trabalho são:

1. Desenvolvemos o preditor ótimo do fBm em tempo discreto a N passos e de ordem p , de utilidade prática. São obtidos os valores numéricos dos coeficientes e do correspondente erro quadrático médio em estado estacionário do preditor ótimo, para diferentes valores dos parâmetros H , N e p .
2. Propomos usar a análise baseada em wavelets para:
 - Estimar não somente o parâmetro H do fBm mas também a constante de variância do fBm.
 - Obter uma completa predição dos incrementos do fBm em diferentes instantes e escalas de tempo, usando um banco de preditores de Wiener multi-escala.
3. Usamos a técnica baseada em wavelets para estimar o parâmetro de Hurst H e o parâmetro de variância K_2 da série, e então usamos um preditor de Kalman linear para estimar conjuntamente tanto o processo incremento de tráfego $i(n)$ como o parâmetro de taxa média do tráfego. Para isso, são apresentadas várias simulações.
 - Demonstra-se que a estimação da taxa média do tráfego obtida com esta técnica proposta é superior à dada pelo leaky bucket.
 - Foi observado um bom desempenho do preditor, ainda que a aproximação $AR(p)$ do processo $i(n)$ tenha sido utilizada.
 - Observou-se também que quando todos os parâmetros do tráfego foram considerados desconhecidos, foi possível se obter boas predições do processo de tráfego $Y(n)$ usando predição de Kalman, sobretudo quando o valor de H é maior que 0.7, que é justamente o intervalo de valores de H encontrados nas séries de tráfego auto-similar práticas [12]. Foi verificado um bom desempenho do preditor para as séries reais de tráfego auto-similar BC-pOct89 e BC-Oct89Ext do Bellcore.
4. Finalmente, no Capítulo 7 desenvolveu-se uma técnica que permite fazer a decomposição em wavelets na forma sequencial, para blocos de N amostras, com uma complexidade de apenas $O(\sqrt{N})$ ao invés de $O(N)$, se um número adequado K de amostras no bloco anterior é substituído por outro número de K amostras no bloco seguinte.

- Demonstra-se que para os valores de N e L de interesse prático, o intervalo possível de valores de K é bastante amplo, o que oferece flexibilidade para escolher o valor de K de acordo com os requisitos de uma certa aplicação e ao mesmo tempo trabalhar com uma baixa complexidade $O(\sqrt{N})$.
- Esta técnica, combinada com a estimação baseada em wavelets do parâmetro de Hurst e da variância do fBm mencionada acima, constitui-se numa poderosa ferramenta para, em tempo real, acompanhar as alterações na dependência de longo prazo do sinal de tráfego.

A principal contribuição deste trabalho consiste na originalidade das importantes técnicas desenvolvidas nos Capítulos 4, 5, 6 e 7, que são de utilidade não somente para análise e predição de tráfego auto-similar, mas também para outras muitas aplicações com processos auto-similares.

Uma futura linha de pesquisa é aplicar as técnicas propostas neste trabalho sobre um número maior de séries reais de tráfego auto-similar, bem como utilizá-las para obter esquemas de controle inteligente para as redes de faixa larga.

Bibliografia

- [1] Norros I. On the use of fractional Brownian motion in the theory of connectionless networks, *IEEE Journal on Selected Areas in Communications*, Vol.13, No.6., August 1995, pp. 953-962.
- [2] Gripenberg G. and Norros I. On the prediction of fractional Brownian motion, *J. Appl. Prob.*, 33, 1996, pp. 400-410.
- [3] Wornell G. W. *Signal Processing with Fractals: A Wavelet Basis Approach*, Prentice Hall PTR, New Jersey, 1996.
- [4] Barnsley M. F. et. al. *The science of fractal images*, Springer-Verlag, New York, 1988.
- [5] Mandelbrot B. B. and van Ness J. W. Fractional brownian motions, fractional noises and applications, *SIAM Rev.*, Vol. 40, No. 4, 1968, pp. 422-437.
- [6] Keshner M. S. 1/f Noise, *Proc. IEEE*, Vol. 70, No. 3, 1982.
- [7] Shanmugan, K. S. and Breipohl A. M. *Random Signals: Detection, Estimation and Data Analysis*, John Wiley and Sons, Inc., New York, 1988.
- [8] Meyer Y. *Ondelettes*, Hermann, Paris, 1990.
- [9] Butto M. , Cavallero E. and Tonietti A. Effectiveness of the Leaky Bucket policing mechanism in ATM networks, *IEEE Journal on Selected Areas in Communications*, Vol.9, N0.3, April 1991, pp. 335-342.
- [10] Chui C. K. *An Introduction to Wavelets*, Academic Press, San Diego, 1992.
- [11] Flandrin P. Wavelet analysis and synthesis of fractional brownian motion, *IEEE Trans. Inf. Theory*, Vol. IT-38, No. 2, March 1992, pp. 910-917.
- [12] Leland W. E., Taqqu M., Willinger W. and Wilson D. V. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, 1994, pp. 1-14.
- [13] Abry P., Gonçalves P. and Flandrin P. Wavelets, spectrum estimation, 1/f processes, wavelets and statistics. *Lectures Note in Statistics*, Vol. 105, 1995, pp. 15-30.
- [14] Beran, J. *Statistics for Long-Memory Processes*, Chapman and Hall, 1994.
- [15] Paxson, V. *Fast approximation of self-similar network traffic*, Lawrence Berkeley Laboratories Technical Report LBL-36750, 1995.
- [16] Onvural R. O. *Asynchronous Transfer Mode Networks: Performance Issues*, second ed., Artech House, 1995.

- [17] Roberts J. et. al. (eds.) *Broadband Network Teletraffic: Final Report of Action - Cost 242*, Springer-Verlag, Berlin, 1996.
- [18] Anderson, B. D. O. and Moore, J. B. *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, N. J., 1979.
- [19] Abry P. and Veitch D. Wavelet analysis of long-range dependent traffic, *IEEE Transactions on Information Theory*, Vol. 44, No. 1, January 1998, pp. 2-15.
- [20] Burrus C. S., Gopinath R. A. and Guo H. *Introduction to wavelets and wavelet transforms*, Prentice-Hall, Inc; 1998.
- [21] Gustavo A. Hirchoren, "Otimização de Sincronismo em Redes Digitais Síncronas com Preditor de Kalman", *Tese de Mestrado, DECOM-FEEC-UNICAMP*, 3 de Setembro de 1996.
- [22] Gustavo A. Hirchoren e Dalton S. Arantes, "Optimal Phase-Locked Loop Design with Kalman Predictors for Synchronous Networks", *International Conference on Acoustics, Speech and Signal Processing - ICASSP '97*, Vol. 3, pp. 1917-1921, Munique, Alemanha, 21 a 24 de Abril, 1997.
- [23] Gustavo A. Hirchoren e Dalton S. Arantes, "Predictors for the Discrete Time Fractional Gaussian Processes", *1998 IEEE/SBT International Telecommunications Symposium*, pp. 49-53, São Paulo, Agosto de 1998.
- [24] Gustavo A. Hirchoren e C. E. D'Attellis, "On the optimal number of scales in estimation of fractal signals using wavelets and filter banks", *Signal Processing*, 63, 1997, pp. 55-63.
- [25] Gustavo A. Hirchoren e C. E. D'Attellis, "Estimation of fractal signals using wavelets and filter banks", *IEEE Trans. on Signal Processing*, Vol. 46, No. 6, June 1998, pp. 1624-1630.