Self-Similarity of A Reverse-Filter Traffic Generator

Prepared by Jin-Yuan Chen

Advisory by Prof. Po-Ning Chen

In Partial Fulfillment of the Requirements

For the Degree of
Master of Science

Department of Communications Engineering

National Chiao Tung University

Hsinchu, Taiwan 30056, Republic of China

E-mail: alcon.cm90g@nctu.edu.tw

June, 2003

Abstract

In early days, Poisson processes were commonly used as network traffic models. It was done under the premise that the network traffic behavior is similar to that in telephony system. Recent studies on network traffic argue that network traffic processes are much more faithfully modelled by using statistical models of *long-range dependence* or *self-similarity*, rather than the short range dependent Poisson. A nature need due to this trend is the synthesization of a self-similar network traffic for simulation of network.

In this thesis, we present a reverse-filter-based approach for synthesizing self-similar network traffic. The reverse filter is an IIR (infinite impulse response) filter, as contrary to the FIR (finite impulse response) filter proposed by Hua [2]. Both approaches have the same computational complexity. Comparison between our reverse filter approach and Hua's forward filter approach in degree of synthesis self-similarity is subsequently performed. Surprisingly, the improvement of our reverse filter approach over the forward filter approach is quite limited.

Acknowledgements

I am deeply grateful to my advisor, Prof. Po-Ning Chen, for his continuous encouragement, support and enthusiastic discussion throughout this research. This work would not been possible without his advice and commitment.

I would like to thank Chien Yao for introducing me into the field of self-similar processes and his suggestions and kind help. His valuable suggestions helped me succeed in this work.

I would like to thank National Chiao-Tung University for providing such a wonderful environment and such many resources for me in my graduate life.

Special thank to my dear friend, Chi-Ming, Hsu-Huei, and Chia-Long, for their great care and help throughout my time in school. I would like to thank my friends and the members of Network Technology Laboratory.

This thesis is dedicated to my mother, for her love and encouragement.

Contents

Abstract **i**

Acknowledgements **i**

List of Tables **vii**

List of Figures **x**

1 Introduction **1**

1.1 Preliminaries 1

1.2 Problem Description 3

1.3 Organization of the Thesis 4

2 Self-Similar Process and Analysis Tests **5**

2.1 Long-Range Dependence 5

2.2 Self-Similarity 6

2.2.1 Slowly Decay Variance 9

2.2.2 Hurst parameter 10

2.2.3 Statistical Test for Self-Similarity 11

3 Schemes of Self-Similar Traffic Generator **12**

3.1 Fast Fourier Transform Traffic Synthesizer 12

3.2 A Linear Approximation to FFT Traffic Synthesizer 14

3.3 Forward-Filter-Based Self-Similar Traffic Generator 16

3.4 Reverse-Filter-Based Self-Similar Traffic Generator 18

3.5 Analysis Of Truncation Effect 21

3.6 Integerization 30

4 Simulations of Synthetic Traces **31**

4.1 Original Result of the Two method Filter-Based Generator 31

4.2 Integerization Effect 50

5 Conclusions and Further Work **60**

5.1 Conclusions 60

5.2 Further Work 61

Bibliography **62**

List of Tables

3.1 Execution time in seconds for synthesizing a self-similar sequence [4]. 15

3.2 The self-similar parameters for the synthetic traffics. The best-fit lines are calculated for $1 \leq m \leq W$. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. Deviations are defined as $D_f = (\hat{H}_f - H)/H$ and $D_r = (\hat{H}_r - H)/H$, where H is the targeted self-similar parameter. 28

3.3 The self-similar parameters for the synthetic traffics. The best-fit lines are calculated for $W \leq m \leq 10^{0.6}W$. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. Deviations are defined as $D_f = (\hat{H}_f - H)/H$ and $D_r = (\hat{H}_r - H)/H$, where H is the targeted self-similar parameter. 29

4.1 The self-similar parameters for the synthetic traffics in Fig. 4.1. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter. 35

4.2 The self-similar parameters for the synthetic traffics in Fig. 4.2. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter. 36

4.3 The self-similar parameters for the synthetic traffics in Fig. 4.3. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter. 37

4.4 The self-similar parameters for the synthetic traffics in Fig. 4.4. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter. 41

4.5 The self-similar parameters for the synthetic traffics in Fig. 4.5. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter. 42

4.6 The self-similar parameters for the synthetic traffics in Fig. 4.6. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter. 43

4.7 The self-similar parameters for the synthetic traffics in Fig. 4.7. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter. 47

4.8 The self-similar parameters for the synthetic traffics in Fig. 4.8. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter. 48

4.9 The self-similar parameters for the synthetic traffics in Fig. 4.9. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter. 49

List of Figures

1.1 Illustration of actual Ethernet traffic and synthetic Ethernet traffic of Poisson mode and self-similar mode. (Source: Fig. 9.5 in [9]) 2

2.1 Comparison of self-similar process and non-self-similar processes. (Source: Fig. 8.3 in [9]) 7

3.1 Relation between the power spectral densities of the input and output random processes through a filter. Let $S_y(w)$ denote the power spectrum of the discrete random process Y_n obtained by passing the random process X_n with power spectrum $S_x(w)$ through a filter with transfer function $H(w)$. 16

3.2 The intuitive expectation of the improvement of reverse filter approach over forward filter approach. 20

3.3 The variance-equivalent m-averaged process. 21

3.4 The *variance-equivalent m-averaged process* of the filter output process. . . 22

3.5 Variance-time analysis ($\log_{10}$ scale) for the truncated reverse and truncated forward filter outputs with truncation window $W = 10^2$. The solid (blue) line is the forward filter result, and the dotted (green) line is the reverse filter result. 24

3.6 Variance-time analysis ($\log_{10}$ scale) for the truncated reverse and truncated forward filter outputs with truncation window $W = 10^3$. The solid (blue) line is the forward filter result, and the dotted (green) line is the reverse filter result. 25

3.7 Variance-time analysis ($\log_{10}$ scale) for the truncated reverse and truncated forward filter outputs with truncation window $W = 10^4$. The solid (blue) line is the forward filter result, and the dotted (green) line is the reverse filter result. 26

4.1 Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^2 and mean rate 1. 32

4.2 Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^2 and mean rate 10. 33

4.3 Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^2 and mean rate 100. 34

4.4 Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^3 and mean rate 1. 38

4.5 Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^3 and mean rate 10. 39

4.6 Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^3 and mean rate 100. 40

4.7 Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^4 and mean rate 1. 44

4.8 Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^4 and mean rate 10. 45

4.9 Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^4 and mean rate 100. 46

4.10 Variance-time plots ($\log_{10}$ scale) for the two filter-based integerized synthetic arrivals with truncation window 10^2 and mean rate 1. 51

4.11 Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^2 and mean rate 10. 52

4.12 Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^2 and mean rate 100. 53

4.13 Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^3 and mean rate 1. 54

4.14 Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^3 and mean rate 10. 55

4.15 Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^3 and mean rate 100. 56

4.16 Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^4 and mean rate 1. 57

4.17 Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^4 and mean rate 10. 58

4.18 Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^4 and mean rate 100. 59

Chapter 1

Introduction

1.1 Preliminaries

With the mature of network technology, the Internet expanded rapidly in recent years. People use the Internet to exchange data, browse data, and get new information in their everyday life. The Internet can even provide services like e-commerce and distant-phone. These data aggregated from various sources with diverse characteristics complicate the work of traffic modelling for Internet.

One of the main components in a network system is the data traffic. Traffic model is therefore of crucial importance for accessing the performance of a network system [11]. A widely used category of traffic models in network system engineering and performance analysis is that of the stochastic models. Among them, the most popular one in early days is perhaps is the Poisson process. The usage of Poisson processes has been justified by the circuit-switch telephony system; however, its application to Internet traffic is somewhat arguable due to the best-effort packet-switch nature of Internet traffic.

Recent measurement suggested that a good network traffic model should emulate the *long-range dependence* or *self-similarity* nature of real network traffic [9]. As an example, the three plots at the first column of Fig. 1.1, which were obtained by observing true Ethernet

traffic, look similar to each other in a distributional sense. This result indicates that the statistical behavior of true Ethernet traffic is almost independent of the observation scale. This is dramatically different from what is synthesized from conventional Poisson model as illustrated at the second column of Fig. 1.1 — at high resolution (time unit = 0.1 seconds), the traffic is quite bursty, yet at low resolution (time unit = 100 seconds), the traffic rate becomes smooth. Willinger et al [12] also generated a sample traffic in terms of a self-similar traffic model with self-similar parameter $H = 0.9$, which is illustrated at the right column of Fig. 1.1. This self-similar synthetic traffic, as anticipated, has the same distributional behavior as the first column.

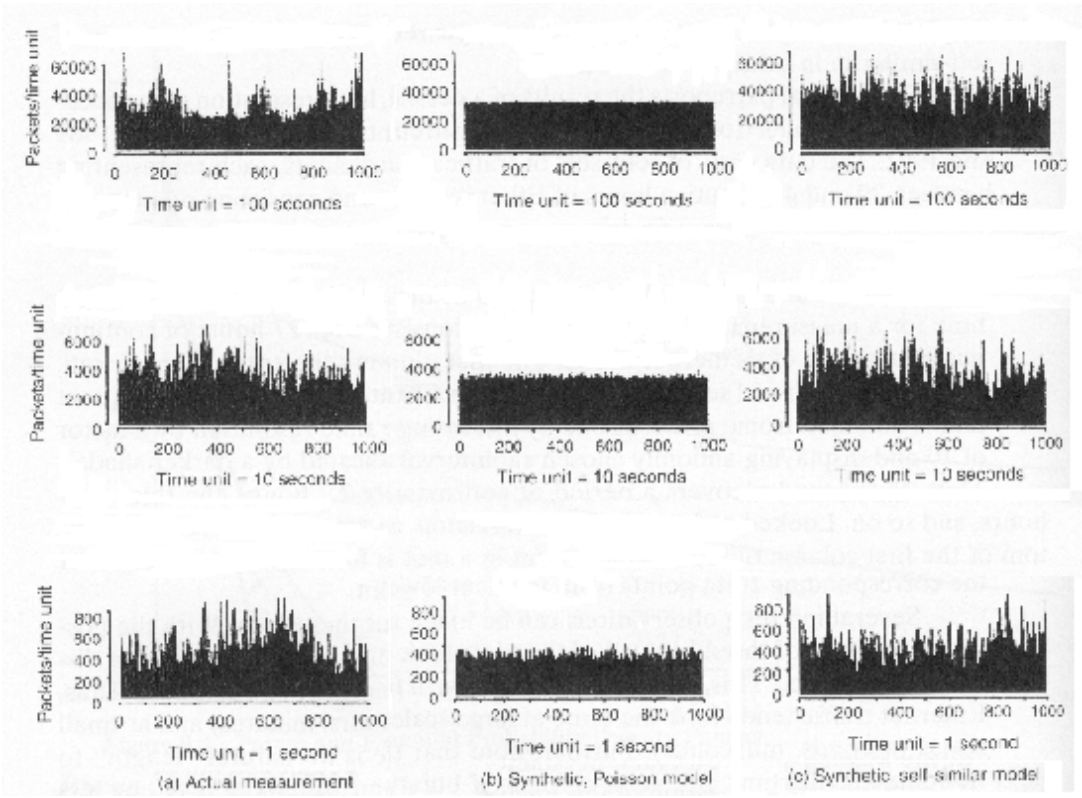

Figure 1.1: Illustration of actual Ethernet traffic and synthetic Ethernet traffic of Poisson mode and self-similar mode. (Source: Fig. 9.5 in [9])

1.2 Problem Description

In order to assure the feasibility, a network system is often tested before its deployment. A properly conducted test can provide a precise guide for the true performance of the deployed network system. Apparently, a basic requirement for a good test is a synthetic traffic that is close to the true traffic. A fallacy may be obtained if a system, before its deployment, is tested or analyzed based on an ideal network traffic that is by no means close to the true traffic. Therefore, how to synthesize a traffic that well approximates the true traffic characteristic becomes an essential research subject.

There have been quite a few publications in self-similar traffic synthesization. In [3], Lau et al proposed a so-called random midpoint displacement algorithm to generate a self-similar network trace. In [8], Paxson synthesized self-similar traffic by means of traffic spectrum fitting to fractional Gaussian noise. A fast algorithm that generates self-similar network trace that improves the above work has recently been proposed by Ledesma and Liu [4]. To relax the requirement of pre-specifying the trace length before the traffic generation in the above work, Hua and Chen proposed a (forward-)filter-based self-similar traffic generator [2].

In this thesis, we follow the work of Hua and Chen by replacing the forward filter by an equivalent reverse filter. The key difference between forward- and reverse-filter traffic generators is that the forward method uses an FIR (finite impulse response) filter, while the reverse method employs an IIR (infinite impulse response) filter. It was expected that with an IIR filter that has infinite memory in its output, the synthetic traffic can achieve the same degree of long-term dependence (as the forward method) with less algorithmic complexity, which is indeed true from our analysis and simulations. However, such improvement is quite limited to our surprise. Details will be provided in subsequent chapters.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 gives the background of second-order self-similar processes, and also the variance-time analysis that is used in our evaluation of the degree of traffic self-similarity. Chapter 3 briefs existing self-similar traffic synthetic algorithms, and closes by the introduction of our reverse-filter-based self-similar traffic generation scheme. Chapter 4 illustrates numerical results in comparison of the performance between forward and reverse filters. Chapter 5 concludes our work and points out possible future work.

Chapter 2

Self-Similar Process and Analysis Tests

In this chapter, we will introduce the basic concepts of long-range dependence, self-similarity, slowly decay variance, Hurst effect, and the variance-time analysis.

2.1 Long-Range Dependence

Long-range dependence is defined in terms of the behavior of the autocovariance function $B(k)$ of a stationary process. For most processes that are of interest in practical application, the autocovariance function decays to zero with k. For example, the Poisson increment process[1] $\{X_t, t \in \Re\}$ with increment L and mean λ has autocovariance function

$$B(k) = E[X_t X_{t+k}] - \lambda^2 = \lambda^2 - \lambda^2 = 0 \quad \text{for } k > L. \tag{2.1}$$

According to the rate of convergence to zero of $B(k)$, a stationary process can be classified into either *long-range dependent* or *short-range dependent*.

[1] **Definition (Poisson counting process)** A Poisson counting process with parameter λ is a process $\{N_t, t \geq 0\}$ with $N_0 = 0$ and stationary independent increments, satisfying that for $0 < t_1 < t_2$, $(N_{t_2} - N_{t_1})$ is Poisson distributed with mean $\lambda(t_1 - t_2)$.

The mean and variance of N_t are both λt, which is a function of time t; hence, it is not stationary. A stationary process related to the Poisson counting process is defined in the following.

Definition (Poisson increment process) A Poisson increment process $\{X_t, t > 0\}$ with parameters λ and L is defined as $X_t = (N_{t+L} - N_t)/L$, where $\{N_t, t > 0\}$ is a Poisson counting process with parameter λ.

Definition 2.1 *[1, eq. (1.20)] A discrete-time second-order-stationary process $X_1, X_2, \ldots$ is said to be* long-range dependent*, if*

$$\sum_{k=-\infty}^{\infty} |B(k)| = \infty. \tag{2.2}$$

Definition 2.2 *[1, eq. (1.24)] A discrete-time second-order-stationary process $X_1, X_2, \ldots$ is said to be* short-range dependent*, if*

$$\sum_{k=-\infty}^{\infty} |B(k)| < \infty. \tag{2.3}$$

Definition 2.1 indicates that the correlations decay to zero so slowly that they are not summable. More specifically, (2.2) holds if

$$B(k) \sim k^{-\beta} \quad \text{as } k \to \infty,$$

for $0 < \beta < 1$.[2] This leads to the following more restricted definition of *long-range dependence*.

Definition 2.3 *[9, pp. 191] A discrete-time second-order-stationary process $X_1, X_2, \ldots$ is said to be* long-range dependent*, if*

$$B(k)/B(0) \sim k^{-\beta} \quad \text{as } k \to \infty, \tag{2.4}$$

where $0 < \beta < 1$.

2.2 Self-Similarity

The self-similar process was first introduced by Mandelbrot in 1968 [5, 6, 7]. As shown in Fig. 2.1, the self-similar process waves at different time scales resemble each other, while the non-self-similar processes at high levels of magnification become more irregular and

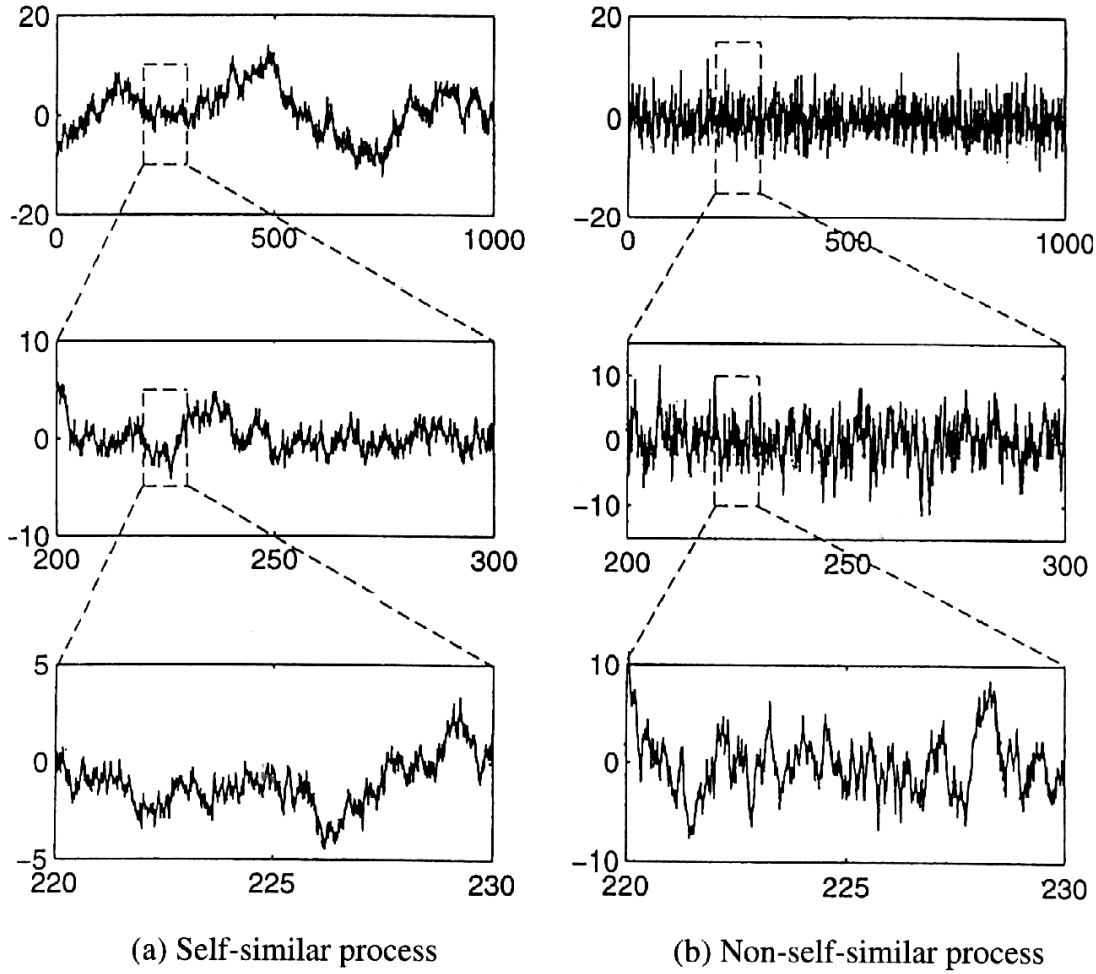

Figure 2.1: Comparison of self-similar process and non-self-similar processes. (Source: Fig. 8.3 in [9])

choppier. Hence, the definitions of self-similarity shall somehow retain the same characteristic in different time scale.

In this section, we briefs the definitions of strictly self-similar, and exactly and asymptotically second-order self-similar processes.

Define the m-averaged process $\boldsymbol{X}^{(m)} = (X_1^{(m)}, X_2^{(m)}, \ldots)$ of a discrete-time stationary

[2]Here and below, $f(x) \sim h(x)$ as $x \to \infty$ means that $[f(x)/h(x)] \to 1$ as $x \to \infty$ [10, pp. 1715].

parent process $X_1, X_2, \ldots$ as:

$$X_j^{(m)} = \frac{1}{m} \sum_{i=1}^{m} X_{mj-m+i}, \tag{2.5}$$

where m and j are positive integers. Then the autocovariance function $B_m(k)$ and variance $B_m(0)$ of $\boldsymbol{X}^{(m)}$ are given by:

$$B_m(k) = \mathrm{Cov}\{X_j^{(m)}, X_{j+k}^{(m)}\} \quad \text{and} \quad B_m(0) = \mathrm{Var}\{X_j^{(m)}\}.$$

Denote the autocorrelation coefficient function of $\boldsymbol{X}^{(m)}$ as $R_m(k) = B_m(k)/B_m(0)$. We then proceed to give various definitions of self-similarity.

Definition 2.4 *[10, Def. C] A discrete-time stationary process* $\boldsymbol{X} = (X_1, X_2, X_3, \ldots)$ *is called* strictly self-similar *with parameter* $H = 1 - (\beta/2)$*, where* $0 < \beta < 1$*, if*

$$m^{1-H} \boldsymbol{X}^{(m)} \stackrel{dis}{=} \boldsymbol{X} \quad \text{for } m = 1, 2, 3, \ldots, \tag{2.6}$$

where "$\stackrel{dis}{=}$*" means equality in the sense of finite-dimensional distributions.*

Definition 2.5 *[10, Def. A] A discrete-time second-order-stationary process* $\boldsymbol{X} = (X_1, X_2, X_3, \ldots)$ *is called* exactly second-order self-similar *with parameter* $H = 1 - (\beta/2)$*, where* $0 < \beta < 1$*, if its autocorrelation coefficient function is*

$$R_1(k) = \frac{1}{2}[|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}] \text{ for } k = 1, 2, \ldots. \tag{2.7}$$

Theorem 2.1 *[10, Thm. 1] For a second-order-stationary process* $\boldsymbol{X} = (X_1, X_2, X_3, \ldots)$ *and* $0 < \beta < 1$*, the following statements are equivalent:*

1. $R_1(k) = (1/2)[|k+1|^{2-\beta} - 2|k|^{2-\beta} + |k-1|^{2-\beta}]$ *for* $k = 1, 2, \ldots$;
2. $B_m(0) = B_1(0) m^{-\beta}$ *for* $m = 1, 2, \ldots$;
3. $B_m(k) = B_1(k) m^{-\beta}$ *for* $m = 1, 2, \ldots$ *and* $k = 0, 1, 2, \ldots$.

Theorem 2.1 indicates that the exactly second-order self-similarity can be equivalently defined by using any of three statements. Definition 2.5 just conveniently takes the first statement in its definition.

Using Taylor expansion, we can obtain that for $0.5 < H < 1$,

$$\begin{aligned} R_1(k) &= \frac{1}{2}\left[|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}\right] \\ &= \frac{1}{2}k^{2H}\left[\left|1+k^{-1}\right|^{2H} - 2 + \left|1-k^{-1}\right|^{2H}\right] \\ &= \frac{1}{2}k^{2H}\big[\left(1 + 2Hk^{-1} + H(2H-1)k^{-2} + O(k^{-3})\right) \\ &\quad -2 + \left(1 - 2Hk^{-1} + H(2H-1)k^{-2} + O(k^{-3})\right)\big] \text{ as } k \to \infty \\ &= H(2H-1)k^{2H-2} + O(k^{2H-3}) \text{ as } k \to \infty. \end{aligned}$$

Therefore, an *exactly second-order self-similar* process is long-range dependent in the sense of Definition 2.3.

Next, we introduce the definition of asymptotic second-order self-similarity.

Definition 2.6 *[10, Def. D] A discrete-time second-order-stationary process* $\boldsymbol{X} = (X_1, X_2, X_3, \ldots)$ *is called* asymptotically second-order self-similar *with parameter* $H = 1 - (\beta/2)$*, where* $0 < \beta < 1$*, if*

$$\lim_{m\to\infty} R_m(k) = \frac{1}{2}[|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}] \text{ for } k = 1, 2, 3, \ldots. \tag{2.8}$$

Some equivalent definitions to Definition 2.6 can again be found in [10].

2.2.1 Slowly Decay Variance

From Theorem 2.1 and $H = 1 - (\beta/2)$,

$$\mathrm{Var}\{X^{(m)}\} = B_m(0) = B_1(0)m^{-\beta} = \frac{\mathrm{Var}\{X\}}{m^{2-2H}}. \tag{2.9}$$

As a result, the variance of m-averaged process $\boldsymbol{X}^{(m)}$ decreases more slowly than the reciprocal of the average size m for an exactly second-order self-similar process $\boldsymbol{X}$ with $0.5 < H < 1$. In fact, $\mathrm{Var}\{X^{(m)}\}$ decreases as a slope of $(2H - 2)$ in log-log plot against m.

2.2.2 Hurst parameter

The parameter H in the previous session is called the *Hurst parameter*, which is named after H. E. Hurst who spent a lifetime studying the Nile and other rivers and the problems related to water storage.

Hurst presented the problem of designing an ideal reservoir to regulate the flow of the Nile based on the given record of observed flow. In his system setting, $X_1, X_2, X_3, \ldots$ represents the inflow during year $j = 1, 2, 3, \ldots$. Then he used a re-scaled range (or the R/S statistics) to characterize the degree of self-similarity of the Nile, i.e., the Hurst parameter H. Specifically, the range $R(n)$ and sample variance $S^2(n)$ are defined by:

$$R(n) = \max_{1 \le t \le n} L_j(n) - \min_{1 \le t \le n} L_j(n) \quad \text{and} \quad S^2(n) = \frac{1}{n} \sum_{j=1}^{n} \left(X_j - \bar{X}_n \right)^2, \tag{2.10}$$

where

$$L_j(n) = \sum_{k=1}^{j} X_k - \bar{X}_n \quad \text{and} \quad \bar{X}_n = \frac{1}{n} \sum_{j=1}^{n} X_j.$$

The R/S statistics, as its name revealed, is then given by $R(n)/S(n)$.

Hurst, after examining a number of different natural phenomena, found that R/S statistics is well described by $(n/2)^H$ as n large for $H > 0.5$. Some researchers therefore refer to this result as Hurst Effect. Additionally, Mandelbrot and Van Ness [6] showed that if the observation sequences are short-range dependent, then $R(n)/S(n)$ will be of the order $n^{0.5}$ as n large.

2.2.3 Statistical Test for Self-Similarity

In this section, we introduce a commonly used method by which the degree of self-similarity of a tested sequence is determined. The method is called *Variance-Time Analysis*.

By Theorem 2.1, the variances of an exactly second-order self-similar parent process and its m-averaged process satisfy:

$$\mathrm{Var}\{X^{(m)}\} = \frac{\mathrm{Var}\{X\}}{m^{2-2H}}.$$

Equivalently, the above equation can be re-written as:

$$\log\left(\frac{\mathrm{Var}\{X^{(m)}\}}{\mathrm{Var}\{X\}}\right) = (2H-2)\,\log(m).$$

Thus, if we plot the curve of $\log(\mathrm{Var}\{X^{(m)}\}/\mathrm{Var}\{X\})$ against $\log(m)$, we will get a straight line with slope $(2H-2)$, which can be used to determine H.

Chapter 3

Schemes of Self-Similar Traffic Generator

Schemes of self-similar network traffic synthesizer are summarized in this chapter. The first one introduced is the fast Fourier transform (FFT) method that synthesizes self-similar sample paths for one type of self-similar process — fractional Gaussian noise [8]. The second one uses a linear approximation to reduce the complexity of the above method [4]. Thereafter, we introduce the forward-filter-based self-similar traffic generator [2]. Then a reverse-filter-based self-similar traffic generator is proposed.

3.1 Fast Fourier Transform Traffic Synthesizer

The method uses the Discrete-Time Fourier Transform (DTFT) to synthesize the fractional Gaussian noise (FGN). Specifically, by sampling the power spectrum of the FGN, and taking inverse-DTFT of these samples, the time-domain self-similar arrivals are readily obtained.

Let the power spectrum of the FGN process be denoted by $S_y(w)$. Then for $0 < H < 1$ and $-\pi \leq w < \pi$,

$$S_y(w) = 2\,\sin(\pi H)\,\Gamma(2H+1)\,\left[1-\cos(w)\right]\,\left[\,|w|^{-2H-1} + A(w)\right], \tag{3.1}$$

where $A(w) = \sum_{k=1}^{\infty}[(2\pi k + w)^{-2H-1} + (2\pi k - w)^{-2H-1}]$.

The main difficulty of using eq. 3.1 to obtain the samples of the power spectrum is the vexing infinite number of summands in $A(w)$. Paxson resolved the problem by approximating $A(w)$ as:

$$\begin{aligned} A(w) &\approx \sum_{k=1}^{3}\left[(2\pi k + w)^{-2H-1} + (2\pi k - w)^{-2H-1}\right] \\ &\quad + \frac{1}{8H\pi}\sum_{k=3}^{4}\left[(2\pi k + w)^{-2H-1} + (2\pi k - w)^{-2H-1}\right], \end{aligned} \tag{3.2}$$

which gives an approximate power spectrum

$$\begin{aligned} \hat{S}_y(w) &= 2\sin(\pi H)\,\Gamma(2H+1)\,[1-\cos(w)]\Bigg(|w|^{-2H-1} \\ &\quad + \sum_{k=1}^{3}\left[(2\pi k + w)^{-2H-1} + (2\pi k - w)^{-2H-1}\right] \\ &\quad + \frac{1}{8H\pi}\sum_{k=3}^{4}\left[(2\pi k + w)^{-2H-1} + (2\pi k - w)^{-2H-1}\right]\Bigg). \end{aligned} \tag{3.3}$$

Notably, the samples of the above approximate spectrum is by no means a set of true FGN samples. Paxson referred to this approach as the "quacks like a duck" approach — from the adage that if an object looks like a duck, walks like a duck, and quacks like a duck, we might as well call it a duck. He then showed that the above approximate is good enough to pass the "quacks like a duck" criterion.

The inputs to Paxson's method include self-similar parameter H and the number of observations in synthesized sample path, n. Paxson's method proceeds as follows:

- Construct a sequence of values S_1,..... $S_{n/2}$, where $S_j = \hat{S}_y(2\pi j/n)$.
- Multiply each sample by an independent exponentially distributed random variable with mean 1.

- Take the square root of all samples. Multiply each by an independent phase that is uniformly distributed over $[0, 2\pi)$.
- Make copies of the current samples to the other half of the spectrum (i.e., from $-\pi$ to 0) to ensure *symmetry*.
- Take Inverse Fast Fourier Transform to the $2n$ spectrum samples to obtain the approximate FGN sample path.

The Paxson's method can synthesize a self-similar process path with a moderate computational complexity. But the traffic sequence cannot be generated on the fly, and the resultant H may be slightly different from targeted H.

3.2 A Linear Approximation to FFT Traffic Synthesizer

In 2000, Ledesma and Liu proposed to use linear approximation to FFT Traffic Synthesizer of Paxson [4]. To be specific, they demonstrated that a linear approximation can be used to determine the power spectrum of the FGN, and this linear approximation reduces the complexity of the computation without compromising the accuracy in synthesizing the power spectrum of the FGN.

From the above section, the power spectrum of the FGN is formulated by:

$$S_y(w) = 2\sin(\pi H)\,\Gamma(2H+1)\,[1-\cos(w)]\,[|w|^{-2H-1} + A(w)] \tag{3.4}$$

for $0 < H < 1$ and $-\pi \le w < \pi$, where $A(w) = \sum_{k=1}^{\infty}[(2\pi k + w)^{-2H-1} + (2\pi k - w)^{-2H-1}]$. We can re-write $A(w)$ as:

$$\sum_{k=1}^{2}[(2\pi k + w)^{-2H-1} + (2\pi k - w)^{-2H-1}] + A_{3:\infty}$$

where $A_{3:\infty} = \sum_{k=3}^{\infty}[(2\pi k + w)^{-2H-1} + (2\pi k - w)^{-2H-1}]$. Using a linear function $D(w) = p \cdot w + q$ to approximate $A_{3:\infty}$, and determining the optimal p and q according to the mean-squared-error criterion, we obtain:

$$\begin{cases} p^* &= -\frac{6}{\pi}F + \frac{12}{\pi^3}G \\ q^* &= \frac{4}{\pi}F - \frac{6}{\pi^2}G, \end{cases} \tag{3.5}$$

where

$$F = \sum_{k=3}^{\infty}\left[\frac{(2\pi k - \pi)^{-2H} - (2\pi k + \pi)^{-2H}}{2H}\right] \tag{3.6}$$

and

$$\begin{aligned} G &= \sum_{k=3}^{\infty}\Bigg(\frac{(2\pi k)[(2\pi k + \pi)^{-2H} + (2\pi k - \pi)^{-2H} - 2(2\pi k)^{-2H}]}{2H} \\ &\quad - \frac{[(2\pi k + \pi)^{-2H+1} + (2\pi k - \pi)^{-2H+1} - 2(2\pi k)^{-2H+1}]}{2H - 1}\Bigg). \end{aligned} \tag{3.7}$$

Table 3.1: Execution time in seconds for synthesizing a self-similar sequence [4].

Length	Ledesma and Liu	Paxson
65,536	5	7
131,072	10	12
262,144	22	24
524,288	45	48
1,048,576	94	260
2,097,152	259	659

From Tab. 3.1, we can see that the Paxson's method requires more computation time than the linear approximation method. Similar to Paxson's method, the linear approximation method requires the knowledge of the sequence size before its execution, and cannot generate self-similar traffic on the fly.

3.3 Forward-Filter-Based Self-Similar Traffic Generator

In this section, we brief the forward-filter-based self-similar traffic generator proposed by Hua and Chen [2]. It is also designed based on power-spectrum fitting. Details of the method can be found in [2].

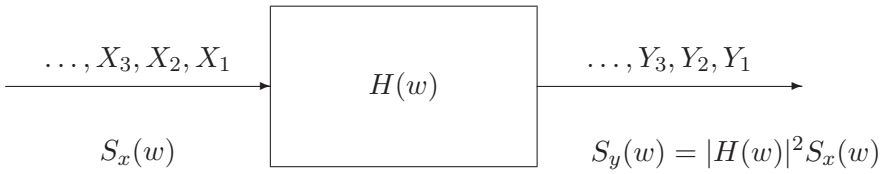

Figure 3.1: Relation between the power spectral densities of the input and output random processes through a filter. Let $S_y(w)$ denote the power spectrum of the discrete random process Y_n obtained by passing the random process X_n with power spectrum $S_x(w)$ through a filter with transfer function $H(w)$.

From Definition 2.4, the autocorrelation coefficient function of an exactly second-order self-similar process with parameter H equals

$$R_1(k) = \frac{1}{2}[|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}] \quad \text{for } k \in I_1,$$

which gives a power spectrum

$$S_y(w) = \sin(\pi H) \cdot \Gamma(2H+1) \cdot |1 - e^{-jw}|^2 \sum_{k=-\infty}^{\infty} |w + 2\pi k|^{-1-2H} \quad \text{for } -\pi \leq w < \pi.$$

Similar to [8], the forward-filter-based self-similar traffic generator approximates the above infinite summation by a finite sum. Specifically, Hua and Chen proposed to only take the main term $k = 0$, and yields that

$$S_y(w) \approx \sin(\pi H) \cdot \Gamma(2H+1) \cdot |1 - e^{-jw}|^2 \cdot |w|^{-1-2H} \quad \text{for } -\pi \leq w < \pi.$$

By observing that the degree of self-similarity is mostly determined by the spectrum behavior close to origin, they further approximate $|1 - e^{-jw}|$ by $|w|$. In particular,

$$\begin{aligned}
|1 - e^{-jw}| &= |1 - \cos(w) + j\sin(w)| \\
&= \sqrt{[1 - \cos(w)]^2 + \sin^2(w)} \\
&= \sqrt{1 - 2\cos(w) + \cos^2(w) + \sin^2(w)} \\
&= \sqrt{2[1 - \cos(w)]} \\
&= \sqrt{2\{[\cos^2(w/2) + \sin^2(w/2)] - [\cos^2(w/2) - \sin^2(w/2)]\}} \\
&= \sqrt{2[2\sin^2(w/2)]} \\
&= 2|\sin(w/2)| \\
&\cong 2|w/2| = |w|, \quad \text{as } |w| \text{ close to } 0.
\end{aligned}$$

As a consequence of the new simplification, the approximate spectrum becomes

$$\tilde{S}_y(w) = |1 - e^{-jw}|^{1-2H} \quad \text{for } -\pi \leq w < \pi \tag{3.8}$$

where the coefficients, $\sin(\pi H) \cdot \Gamma(2H + 1)$, is removed by normalization for simplicity. It remains to design a filter whose output spectrum due to an i.i.d. input equals $\tilde{S}_y(w)$, or specifically, $|H(w)|^2 = |1 - e^{-jw}|^{1-2H}$. By Taylor's expansion, he obtain:

$$\begin{aligned}
(1 - z^{-1})^{-a} &= 1 + \frac{a}{1!}z^{-1} + \frac{a(a+1)}{2!}z^{-2} + \cdots + \frac{a(a+1)\cdots(a+n-1)}{n!}z^{-n} + \cdots \\
&= \sum_{n=0}^{\infty} \frac{\Gamma(n+a)}{\Gamma(n+1)\Gamma(a)} z^{-n},
\end{aligned} \tag{3.9}$$

where $\Gamma(\cdot)$ represents the Euler gamma function defined as $\Gamma(n) = \int_0^\infty t^{n-1}e^{-t}dt$. Replacing z with e^{jw}, and a with $(2H - 1)/2$ in (3.9), gives that

$$(1 - e^{-jw})^{(1-2H)/2} = \sum_{n=0}^{\infty} \frac{\Gamma(n + H - 0.5)}{\Gamma(n+1)\Gamma(H - 0.5)} e^{-jwn}.$$

A term-wise comparison with z-transform expression of $H(w) = \sum_{n=0}^{\infty} h_f[n]e^{-jwn}$ concludes that

$$h_f[n] = \begin{cases} \dfrac{\Gamma(n+H-0.5)}{\Gamma(n+1)\Gamma(H-0.5)}, & \text{for } n \geq 0 \\ \\ 0, & \text{otherwise,} \end{cases}$$

and the relation between input sequence $x[n]$ and output sequence $y[n]$ is equal to:

$$y[n] = \sum_{k=0}^{\infty} h_f[k] \cdot x[n-k].$$

3.4 Reverse-Filter-Based Self-Similar Traffic Generator

From the design of the forward-filter-based self-similar traffic generator, we learn that the z-transforms of input $X(z)$ and output $Y(z)$ can be characterized by:

$$H(z) = (1-z^{-1})^{(1-2H)/2} = \frac{1}{(1-z^{-1})^{(2H-1)/2}} = \frac{Y(z)}{X(z)}, \tag{3.10}$$

which implies that

$$(1-z^{-1})^a Y(z) = X(z), \tag{3.11}$$

where $a = (2H-1)/2 \in (0, 0.5)$. The binomial expansion of $(1-z^{-1})^a$ is equal to:

$$\begin{aligned} (1-z^{-1})^a &= 1 + \frac{-a}{1!}z^{-1} + \frac{-a(1-a)}{2!}z^{-2} + \cdots + \frac{-a(1-a)\cdots(n+1-a)}{n!}z^{-n} + \cdots \\ &= 1 - a\sum_{n=1}^{\infty} \frac{\Gamma(n-a)}{\Gamma(n+1)\Gamma(1-a)}z^{-n}. \end{aligned} \tag{3.12}$$

where $\Gamma(\cdot)$ represents the Euler gamma function defined as $\Gamma(n) = \int_0^{\infty} t^{n-1}e^{-t}dt$. This new expression shows that the outputs $y[1], y[2], y[3] \ldots$ can also be obtained through:

$$\begin{aligned} y[n] &= x[n] + a\sum_{k=1}^{\infty} \frac{\Gamma(k-a)}{\Gamma(k+1)\Gamma(1-a)}y[n-k] \\ &= x[n] + \sum_{k=1}^{\infty} h_r[k] \cdot y[n-k], \end{aligned}$$

where

$$
\begin{aligned}
h_r[k] &= \begin{cases} \dfrac{a \cdot \Gamma(k-a)}{\Gamma(k+1)\Gamma(1-a)}, & \text{for } k \geq 1 \\ 0, & \text{otherwise,} \end{cases} \\
&= \begin{cases} \dfrac{(H-0.5) \cdot \Gamma(k-H+0.5)}{\Gamma(1.5-H)\Gamma(k+1)}, & \text{for } k \geq 1 \\ 0, & \text{otherwise,} \end{cases}
\end{aligned}
$$

Unlike the forward filter model, the above formula gives an infinite impulse response filter (IIR) even if a finite truncation on $h_r[\cdot]$ is applied. Notably, without truncation on the filter, the reverse filter model is indeed equivalent to the forward filter model used by Hua [2].

The traffic generator based on the above model is called *reverse filter self-similar traffic generator* is because it has a feedback or reverse path. In other words, the present output is directly dependent on the past outputs and the present input. Hence, the output should have infinite memory. It is based on this reason that we conjectured the reverse filter approach with truncation should be more self-similar than the forward filter approach with truncation (cf. Fig. 3.2).

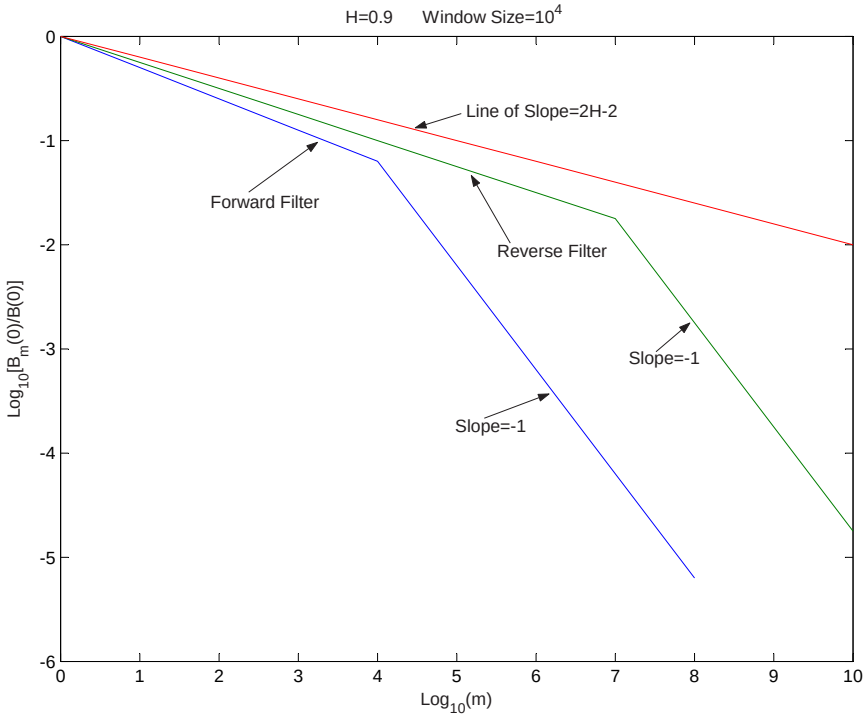

Figure 3.2: The intuitive expectation of the improvement of reverse filter approach over forward filter approach.

3.5 Analysis Of Truncation Effect

Since the reverse filter $\{h_r[n]\}_{n\geq 0}$ has infeasibly infinite length, a truncation on its length is necessary for practical implementation. Define the truncated reverse filter with truncation window W as:

$$h_r[n;W] \;=\; \begin{cases} \dfrac{(H-0.5)\cdot\Gamma(n-H+0.5)}{\Gamma(n+1)\Gamma(1.5-H)}, & \text{for } 1\leq n<W; \\ \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

Also, define the *variance-equivalent m-average process* $\bar{Y}_1^{(m)}, \bar{Y}_2^{(m)}, \bar{Y}_3^{(m)}, \ldots$ of a random process $Y_1, Y_2, Y_3, \ldots$ to be the output due to input $Y_1, Y_2, Y_3, \ldots$ and filter $g[n;m]$, where $g[n;m]$ is defined as:

$$g[n;m] = \begin{cases} \dfrac{1}{m}, & \text{for } 0\leq n<m; \\ \\ 0, & \text{otherwise.} \end{cases}$$

For clarity, the generation of *variance-equivalent m-averaged process* is depicted in Fig. 3.3. Process $\bar{Y}_1^{(m)}, \bar{Y}_2^{(m)}, \bar{Y}_3^{(m)}, \ldots$ is named the *variance-equivalent m-averaged process* because it has the same marginal variance as the m-average process defined in Chapter 2 (cf. Eq. 2.5), provided that the parent process $Y_1, Y_2, Y_3, \ldots$ is wide-sense stationary.

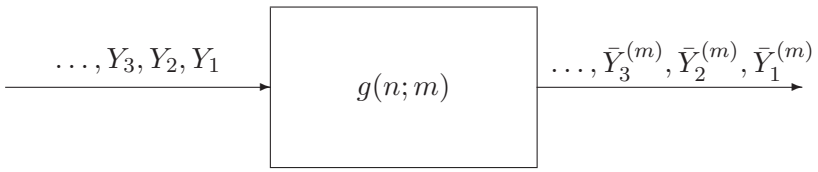

Figure 3.3: The variance-equivalent m-averaged process.

We then perform similar analysis as that in [2] to examine the variance-time plot of the output $Y_1, Y_2, Y_3, \ldots$ due to input $X_1, X_2, X_3, \ldots$ and the truncated reverse filter $h_r[n;W]$ in

term of $g[n;m]$, as shown in Fig. 3.4. Through the analysis, we can obtain the expression of the marginal variance of $\bar{Y}_n^{(m)}$, denoted as $B_m^{(r)}(0;W)$, for the reverse filter.

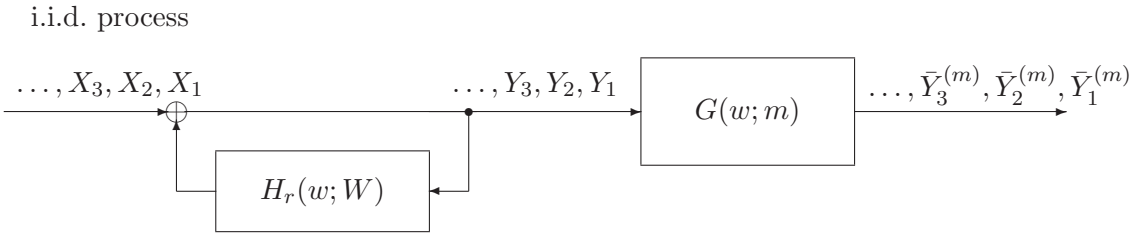

Figure 3.4: The *variance-equivalent m-averaged process* of the filter output process.

Let $L(w)$ be the impulse response of the combined filter of $h_r[n;W]$ and $g[n;m]$. Then

$$L(w) = \frac{G(w;m)}{1 - H_r(w;W)}.$$

By assuming that the power spectrum of the i.i.d. input $X_1, X_2, X_3, \ldots$ is equal to 1, we derive:

$$\begin{aligned} B_m^{(r)}(0;W) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |L(w)|^2 dw \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{|G(w;m)|^2}{|1 - H_r(w;W)|^2} dw \\ &= \frac{1}{2\pi m^2}\int_{-\pi}^{\pi} \frac{1}{|1 - H_r(w;W)|^2} \frac{\sin^2(mw/2)}{\sin^2(w/2)} dw. \end{aligned}$$

For comparison, we quote [2] the marginal variance $B_m^{(f)}(0;W)$ of the m-averaged process

of the output through forward filter in the following.

$$B_m^{(f)}(0;W) = \begin{cases} \dfrac{1}{m^2}\left\{\displaystyle\sum_{l=0}^{W-m}\left(\sum_{n=l}^{l+m-1} h_f[n]\right)^2 + \sum_{l=0}^{m-2}\left[\left(\sum_{n=0}^{l} h_f[n]\right)^2 + \left(\sum_{n=W-1-l}^{W-1} h_f[n]\right)^2\right]\right\}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } m \le W; \\ \dfrac{1}{m^2}\left\{(m-W+1)\left(\displaystyle\sum_{n=0}^{W-1} h_f[n]\right)^2 + \sum_{l=0}^{W-2}\left[\left(\sum_{n=0}^{l} h_f[n]\right)^2 + \left(\sum_{n=l+1}^{W-1} h_f[n]\right)^2\right]\right\}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } m > W, \end{cases}$$

where

$$h_f[n] = \frac{\Gamma(n+H-0.5)}{\Gamma(n+1)\Gamma(H-0.5)} \text{ for } n \ge 0.$$

We can then use the above expressions of $B_m^{(f)}(0;W)$ and $B_m^{(r)}(0;W)$ to obtain the variance-time plots in terms of

$$\log_{10}[B_m^{(f)}(0;W)] \text{ and } \log_{10}[B_m^{(r)}(0;W)] \quad \text{for } 0 \le \log_{10}(m) \le 5.$$

The follow figures are the results for different truncation windows sizes.

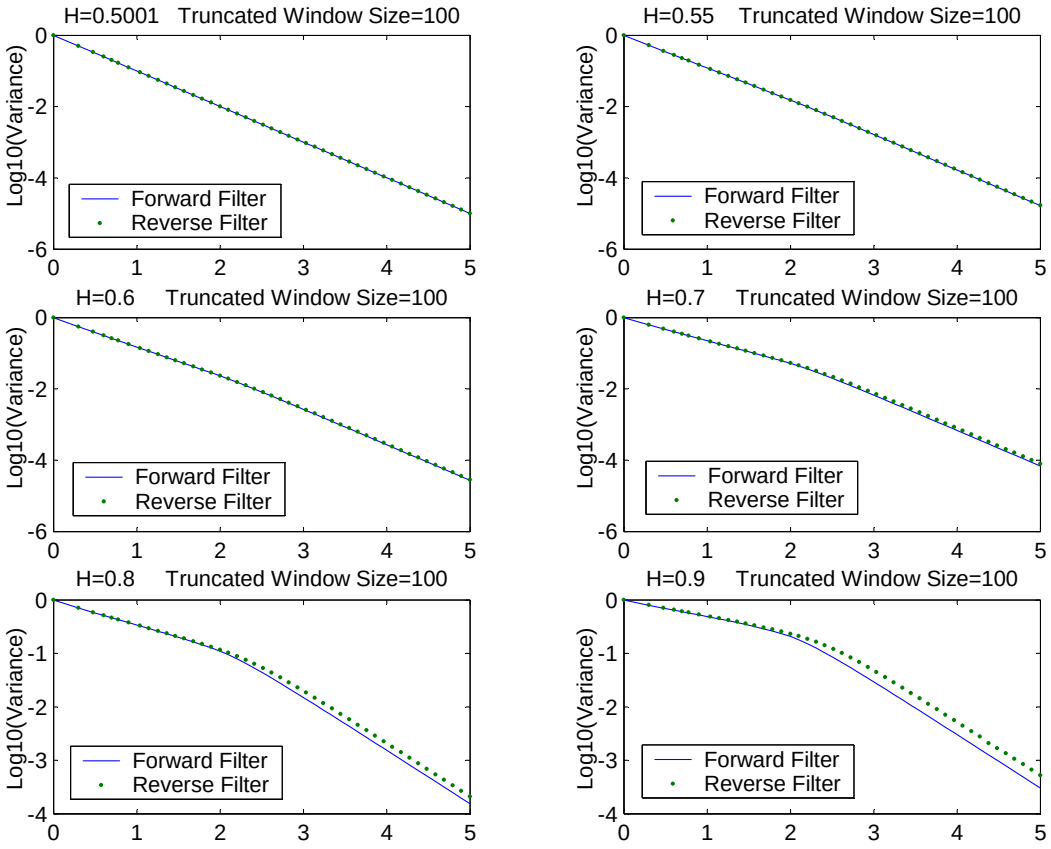

Figure 3.5: Variance-time analysis ($\log_{10}$ scale) for the truncated reverse and truncated forward filter outputs with truncation window $W = 10^2$. The solid (blue) line is the forward filter result, and the dotted (green) line is the reverse filter result.

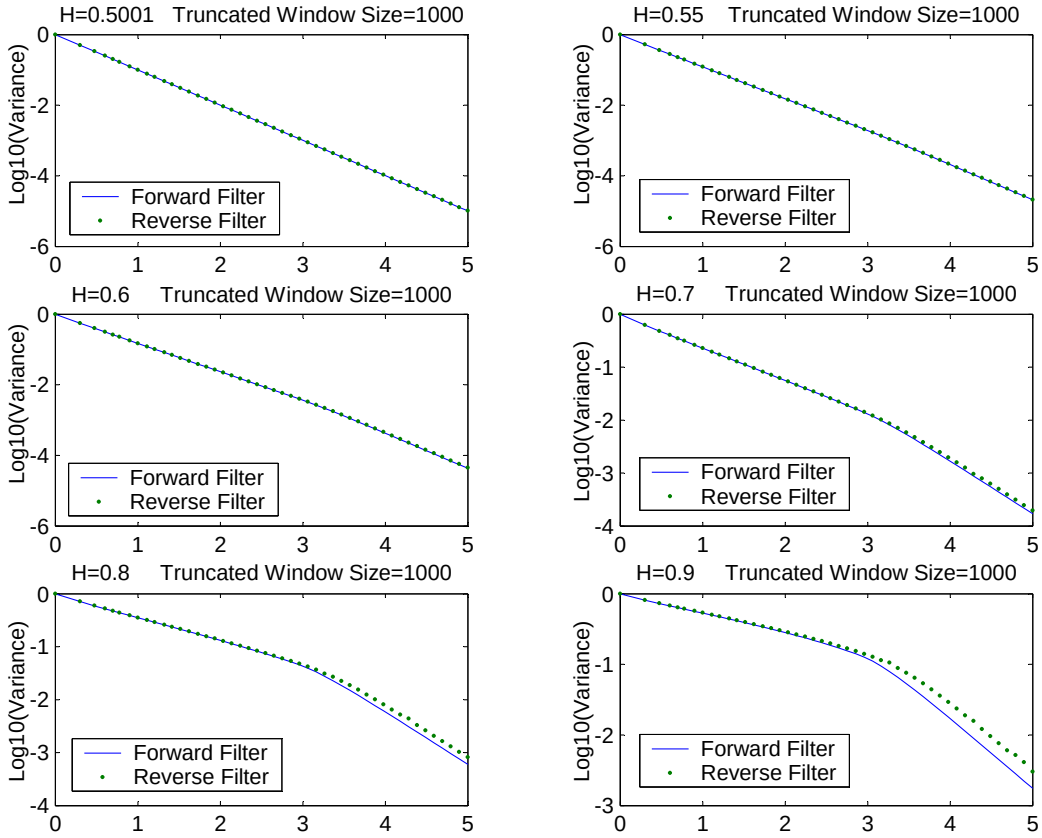

Figure 3.6: Variance-time analysis ($\log_{10}$ scale) for the truncated reverse and truncated forward filter outputs with truncation window $W = 10^3$. The solid (blue) line is the forward filter result, and the dotted (green) line is the reverse filter result.

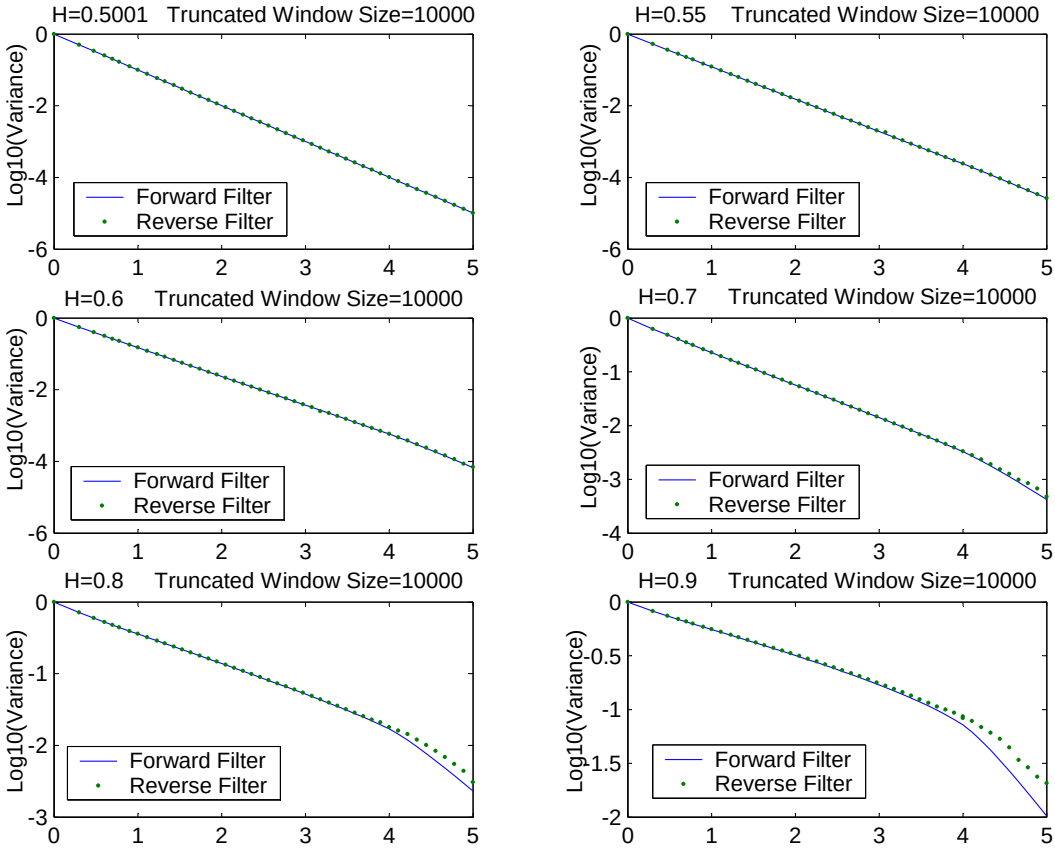

Figure 3.7: Variance-time analysis ($\log_{10}$ scale) for the truncated reverse and truncated forward filter outputs with truncation window $W = 10^4$. The solid (blue) line is the forward filter result, and the dotted (green) line is the reverse filter result.

From Figs. 3.5–3.7, we observed that the curve for reverse filter approach almost coincides with the curve for forward filter approach, when H is equal to $0.5001, 0.55, 0.6$, and 0.7. As H reaches 0.8 and 0.9, the reverse filter approach does perform a little better in the sense of higher self-similarity.

To double-confirm our observations, the minimal square error criterion is adopted to find the lines that best-fit[1] the points obtained in Figs. 3.5–3.7. By equating the slopes of these best-fit lines to $(2H - 2)$, the corresponding H's are summarized in Tabs. 3.2 and 3.3.

[1]The best-fit line is defined as the line which satisfies the minimal square error criterion. Specifically, for points (x_1, y_1), (x_2, y_2), ..., (x_n, y_n), the best fit line $y = -\beta x + c$ satisfies

$$-\beta = \frac{(1/n)\sum_{i=1}^{n} x_i y_i - \bar{x}\bar{y}}{(1/n)\sum_{i=1}^{n} x_i^2 - \bar{x}^2},$$

where $\bar{x} = (x_1 + x_2 + \cdots + x_n)/n$ and $\bar{y} = (y_1 + y_2 + \cdots + y_n)/n$. Then we can get the resultant self-similar parameter $\hat{H} = 1 - \beta/2$.

Table 3.2: The self-similar parameters for the synthetic traffics. The best-fit lines are calculated for $1 \leq m \leq W$. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. Deviations are defined as $D_f = (\hat{H}_f - H)/H$ and $D_r = (\hat{H}_r - H)/H$, where H is the targeted self-similar parameter.

Window Size=10^2				
Targeted H	$\hat{H}_f$	$\hat{H}_r$	D_f	D_r
0.501	0.500909	0.500912	-0.91e-004	-0.88e-004
0.55	0.545929	0.545952	-0.004071	-0.004048
0.6	0.591944	0.592084	-0.008056	-0.007916
0.7	0.682097	0.683299	-0.017903	-0.016701
0.8	0.765413	0.769475	-0.034587	-0.030525
0.9	0.836101	0.845290	-0.063899	-0.054710

Window Size=10^3				
Targeted H	$\hat{H}_f$	$\hat{H}_r$	D_f	D_r
0.501	0.500956	0.500951	-0.44e-004	-0.49e-004
0.55	0.547706	0.547689	-0.002294	-0.002311
0.6	0.595532	0.595554	-0.004680	-0.004554
0.7	0.689892	0.690462	-0.010108	-0.009538
0.8	0.778615	0.780930	-0.021385	-0.019070
0.9	0.855109	0.860802	-0.044891	-0.039198

Window Size=10^4				
Targeted H	$\hat{H}_f$	$\hat{H}_r$	D_f	D_r
0.501	0.500970	0.500977	-0.3e-004	-0.23e-004
0.55	0.548579	0.548932	-0.001421	-0.001068
0.6	0.597209	0.596992	-0.002791	-0.003008
0.7	0.693410	0.693665	-0.006590	-0.006350
0.8	0.784956	0.786866	-0.015044	-0.013134
0.9	0.865082	0.870323	-0.034918	-0.029677

Table 3.3: The self-similar parameters for the synthetic traffics. The best-fit lines are calculated for $W \leq m \leq 10^{0.6}W$. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. Deviations are defined as $D_f = (\hat{H}_f - H)/H$ and $D_r = (\hat{H}_r - H)/H$, where H is the targeted self-similar parameter.

Window Size=10^2				
Targeted H	$\hat{H}_f$	$\hat{H}_r$	D_f	D_r
0.5001	0.5000643	0.5000700	-0.000713	-0.000598
0.55	0.526311	0.528772	-0.043071	-0.038596
0.6	0.548175	0.557430	-0.086375	-0.070949
0.7	0.580904	0.614373	-0.170137	-0.122323
0.8	0.602104	0.671024	-0.247370	-0.161220
0.9	0.615083	0.726966	-0.316574	-0.192259

Window Size=10^3				
Targeted H	$\hat{H}_f$	$\hat{H}_r$	D_f	D_r
0.5001	0.5000484	0.5000519	-0.001030	-0.000961
0.55	0.522934	0.525323	-0.049210	-0.044868
0.6	0.541746	0.550603	-0.097090	-0.082328
0.7	0.569467	0.602010	-0.186476	-0.139985
0.8	0.587001	0.656429	-0.266248	-0.179463
0.9	0.597748	0.706871	-0.335836	-0.214587

Window Size=10^4				
Targeted H	$\hat{H}_f$	$\hat{H}_r$	D_f	D_r
0.5001	0.5000465	0.5000419	-0.001068	-0.001159
0.55	0.525465	0.527560	-0.044609	-0.040800
0.6	0.546588	0.555105	-0.089020	-0.074825
0.7	0.578034	0.615440	-0.174238	-0.120800
0.8	0.598354	0.700691	-0.252058	-0.124136
0.9	0.610636	0.753034	-0.321515	-0.163295

3.6 Integerization

As the output $y[n]$ of the (either forward or reverse) filter is in general not an integer (even if we can make them non-negative by feeding in the filter with non-negative i.i.d. inputs), they cannot be used as a network arrival sequence if no certain "rounding" mechanism is provided.

Hua and Chen proposed to use "rounding $y[n]$ to the nearest non-negative integer" mechanism for the sake of simplicity; however, such a mechanism may destroy the self-similar structure of the output sequence, if the mean traffic rate of the filter input is too small.

To amend this problem, we propose to round the accumulative function of $y[n]$ instead of $y[n]$ itself. Specifically, we let

$$z[n] = \big[F_Y[n] - F_Y[n-1]\big]^+,$$

where $[x]^+ = \max\{x, 0\}$, and

$$F_Y[n] = \left\lfloor \sum_{k=1}^{n} y[k] \right\rfloor.$$

Notably, the operator $[\cdot]^+$ can be removed if $y[1], y[2], y[3], \ldots$ are non-negative. We will examine the effect of the new mechanism by simulation.

Chapter 4

Simulations of Synthetic Traces

This chapter examines the degree of self-similarity of the traffics synthesized by the forward- and reverse-filter-based self-similar traffic synthesizers described in the previous chapter. The variance-time analysis will be used in our examination. In addition, rounding impact on degree of output self-similarity will also be empirically studied.

4.1 Original Result of the Two method Filter-Based Generator

Figures 4.1–4.9 illustrate the synthetic arrival traffics of length 10^6 in terms of the forward filter and the reverse filter. The truncation windows taken are 100, 1000 and 10000, respectively. The mean rates of the filter input process used in our simulations are 1, 10, and 100, respectively. As aforementioned, these figures surprisingly show that the improvement of the reverse filter synthesizer over the forward filter synthesizer in increasing the degree of self-similarity is quite limited.

A more detailed observation gives that a larger window size, or a larger input mean rate, or a higher filter parameter H gives more improvement. This can be confirmed by taking a close look at the simulation result corresponding to window size $10,000$, input mean rate

100 and filter parameter $H = 0.9$ in Fig. 4.9. The self-similar parameters obtained from Figs. 4.1–4.9 are listed in Tabs. 4.1–4.9.

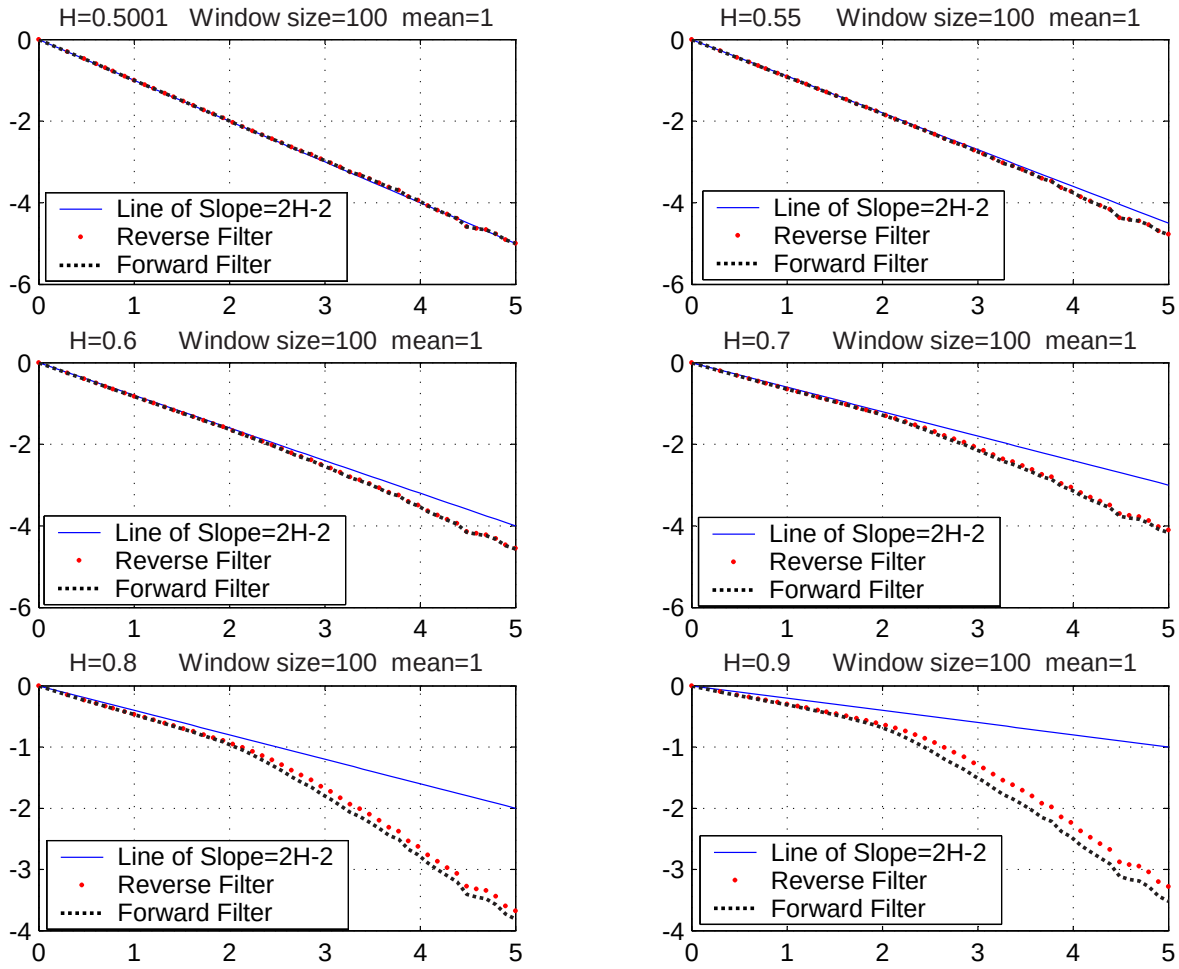

Figure 4.1: Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^2 and mean rate 1.

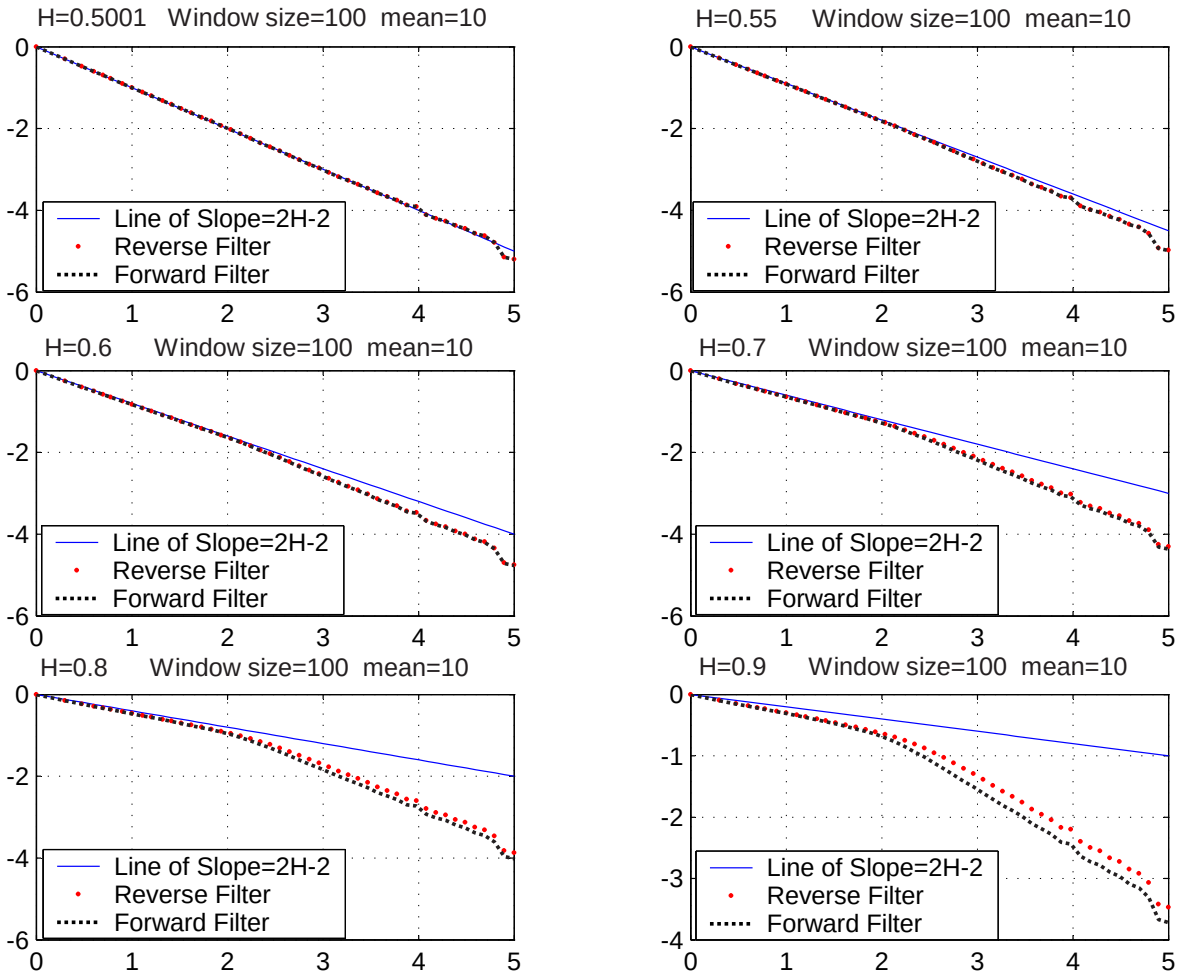

Figure 4.2: Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^2 and mean rate 10.

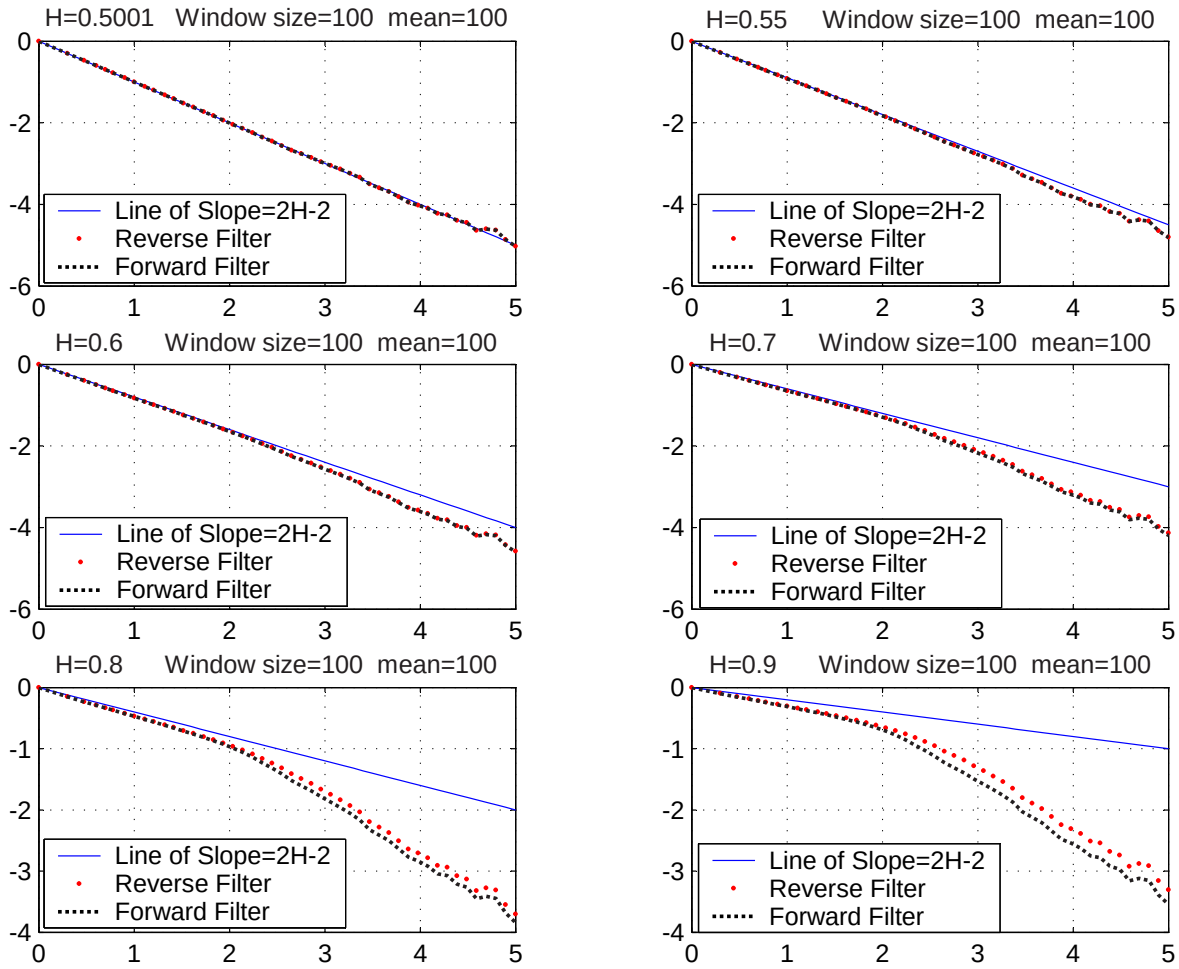

Figure 4.3: Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^2 and mean rate 100.

Table 4.1: The self-similar parameters for the synthetic traffics in Fig. 4.1. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter.

Window Size= 10^2 and mean rate= 1				
Targeted H	$\hat{H}_f(m \leq W)$	$\hat{H}_r(m \leq W)$	$\hat{H}_f(W \leq m \leq 10^{0.6}W)$	$\hat{H}_r(W \leq m \leq 10^{0.6}W)$
0.5001	0.501260971	0.501260975	0.515888448	0.516808234
0.55	0.546097230	0.546115922	0.541163694	0.542083290
0.6	0.592068671	0.592204877	0.558707330	0.567850033
0.7	0.682493542	0.683633132	0.586579041	0.620015754
0.8	0.766404866	0.770393771	0.604143124	0.673417427
0.9	0.837857103	0.846922931	0.614666028	0.727600348
Targeted H	$\hat{H}_f(m > W)$	$\hat{H}_r(m > W)$	$\hat{H}_f$(for all m)	$\hat{H}_r$(for all m)
0.5001	0.496599819	0.496601023	0.501348013	0.501349501
0.55	0.500299782	0.500737237	0.523510278	0.524070384
0.6	0.503273069	0.504885843	0.544856931	0.546974512
0.7	0.507426563	0.513397521	0.583449169	0.591667778
0.8	0.509938942	0.522827208	0.615993557	0.634239056
0.9	0.511408850	0.533920803	0.641784779	0.673746092

Table 4.2: The self-similar parameters for the synthetic traffics in Fig. 4.2. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter.

Window Size= 10^2 and mean rate= 10				
Targeted H	$\hat{H}_f(m \leq W)$	$\hat{H}_r(m \leq W)$	$\hat{H}_f(W \leq m \leq 10^{0.6}W)$	$\hat{H}_r(W \leq m \leq 10^{0.6}W)$
0.5001	0.501717795	0.501717774	0.492440995	0.492447176
0.55	0.546540772	0.546546646	0.515732820	0.518217386
0.6	0.592490989	0.592573017	0.535442685	0.544620197
0.7	0.682855306	0.683746685	0.564670221	0.598169587
0.8	0.766689515	0.770073962	0.583534293	0.652801085
0.9	0.838057451	0.846073763	0.595181259	0.708065106
Targeted H	$\hat{H}_f(m > W)$	$\hat{H}_r(m > W)$	$\hat{H}_f$(for all m)	$\hat{H}_r$(for all m)
0.5001	0.491181591	0.491182192	0.496573196	0.496574400
0.55	0.494988342	0.495452565	0.518751181	0.519317128
0.6	0.498071690	0.499792428	0.540115463	0.542257434
0.7	0.502435031	0.508855616	0.578742716	0.587074931
0.8	0.505131423	0.519062084	0.611317188	0.629868173
0.9	0.506754322	0.531165293	0.637131572	0.669758662

Table 4.3: The self-similar parameters for the synthetic traffics in Fig. 4.3. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter.

Window Size= 10^2 and mean rate= 100				
Targeted H	$\hat{H}_f(m \leq W)$	$\hat{H}_r(m \leq W)$	$\hat{H}_f(W \leq m \leq 10^{0.6}W)$	$\hat{H}_r(W \leq m \leq 10^{0.6}W)$
0.5001	0.500707672	0.500707704	0.491663801	0.491669797
0.55	0.545387889	0.545396173	0.515628322	0.518149462
0.6	0.591212765	0.591301630	0.535969642	0.545297640
0.7	0.681407493	0.682312048	0.566144477	0.600293031
0.8	0.765216136	0.768615685	0.585473918	0.656326259
0.9	0.836730200	0.844787560	0.597160944	0.712978035
Targeted H	$\hat{H}_f(m > W)$	$\hat{H}_r(m > W)$	$\hat{H}_f$(for all m)	$\hat{H}_r$(for all m)
0.5001	0.505548601	0.505550811	0.501073470	0.501075382
0.55	0.509544823	0.510001070	0.523277326	0.523838606
0.6	0.512778613	0.514460090	0.544668887	0.546792673
0.7	0.517335392	0.523547349	0.583359335	0.591613353
0.8	0.520120262	0.533491188	0.616017091	0.634381383
0.9	0.521764358	0.545033086	0.641937659	0.674205128

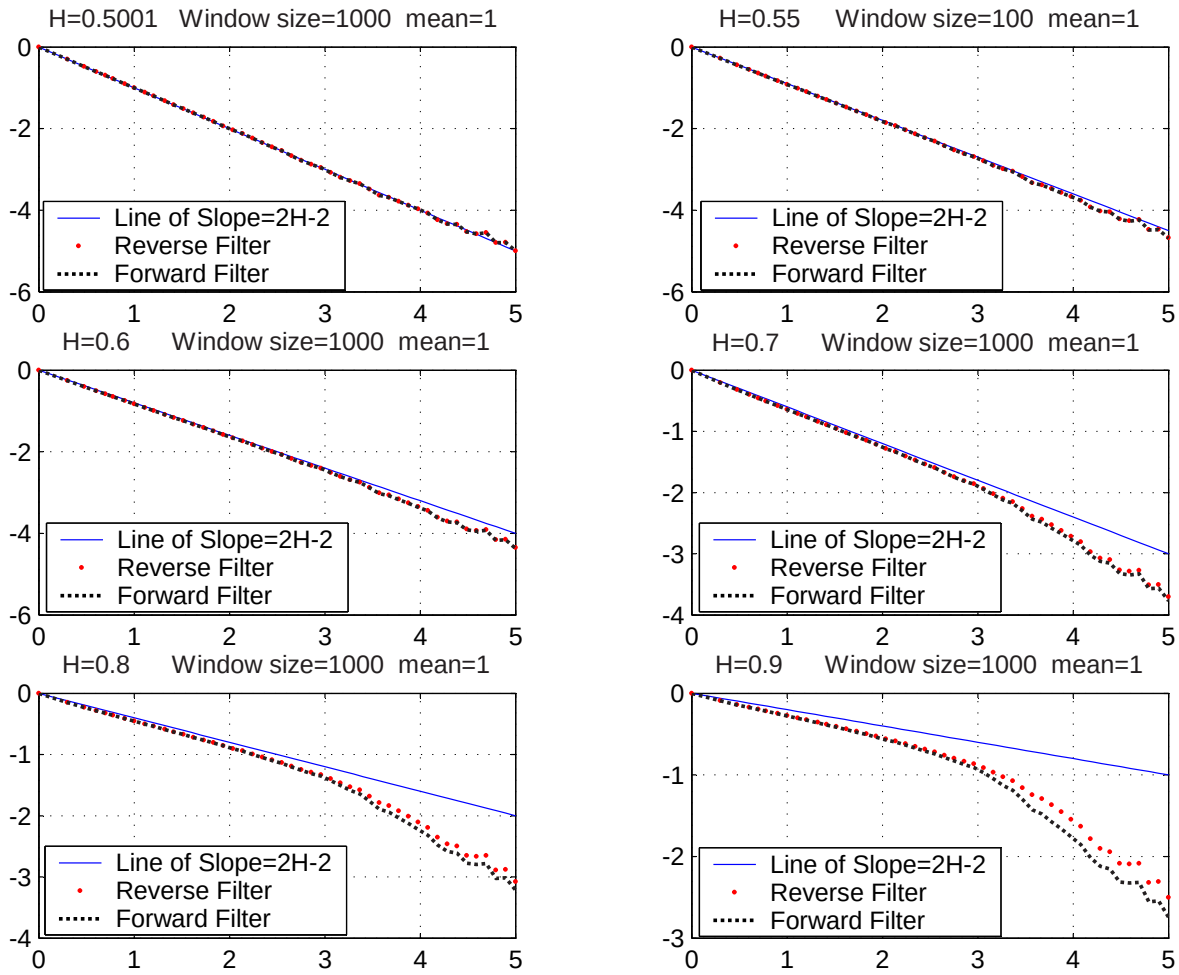

Figure 4.4: Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^3 and mean rate 1.

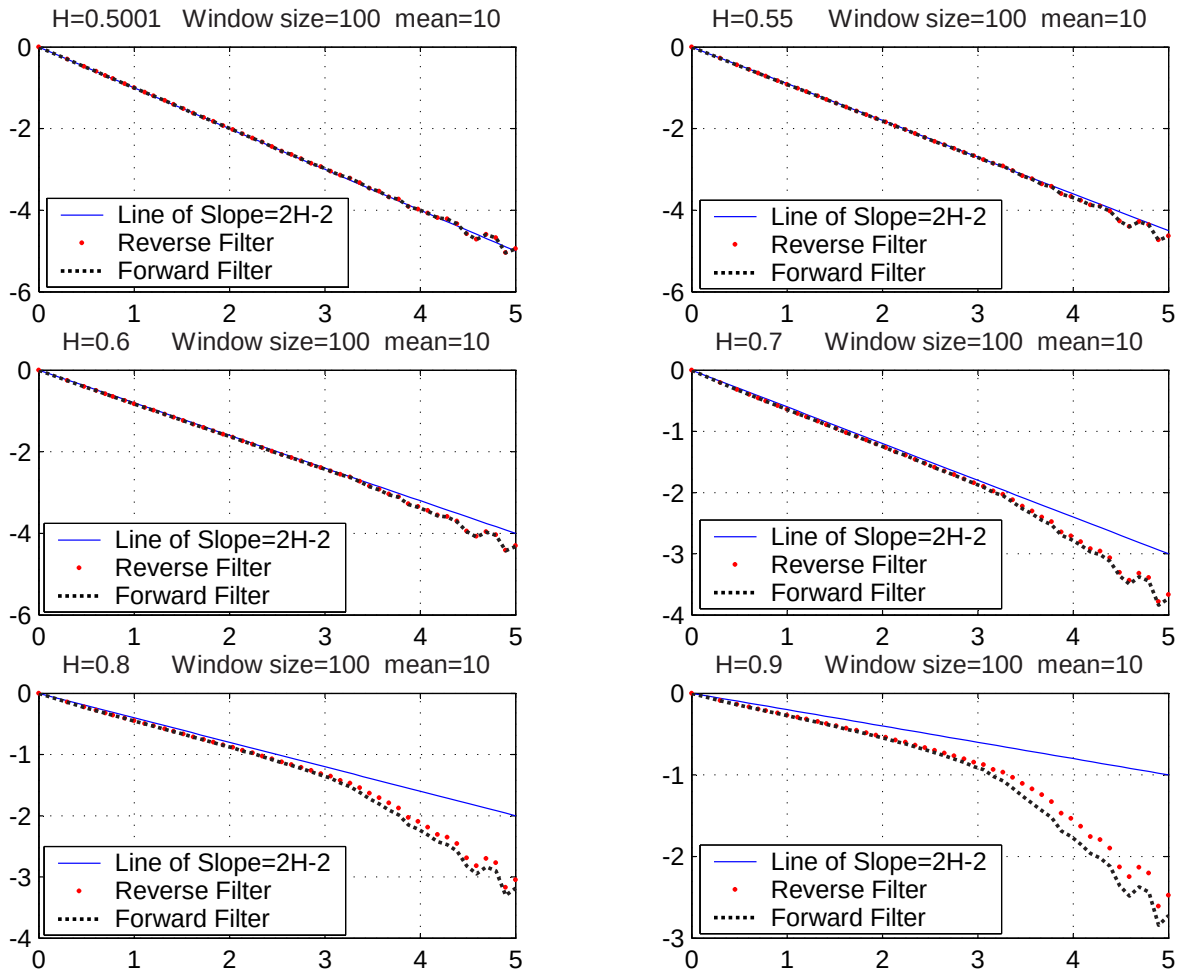

Figure 4.5: Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^3 and mean rate 10.

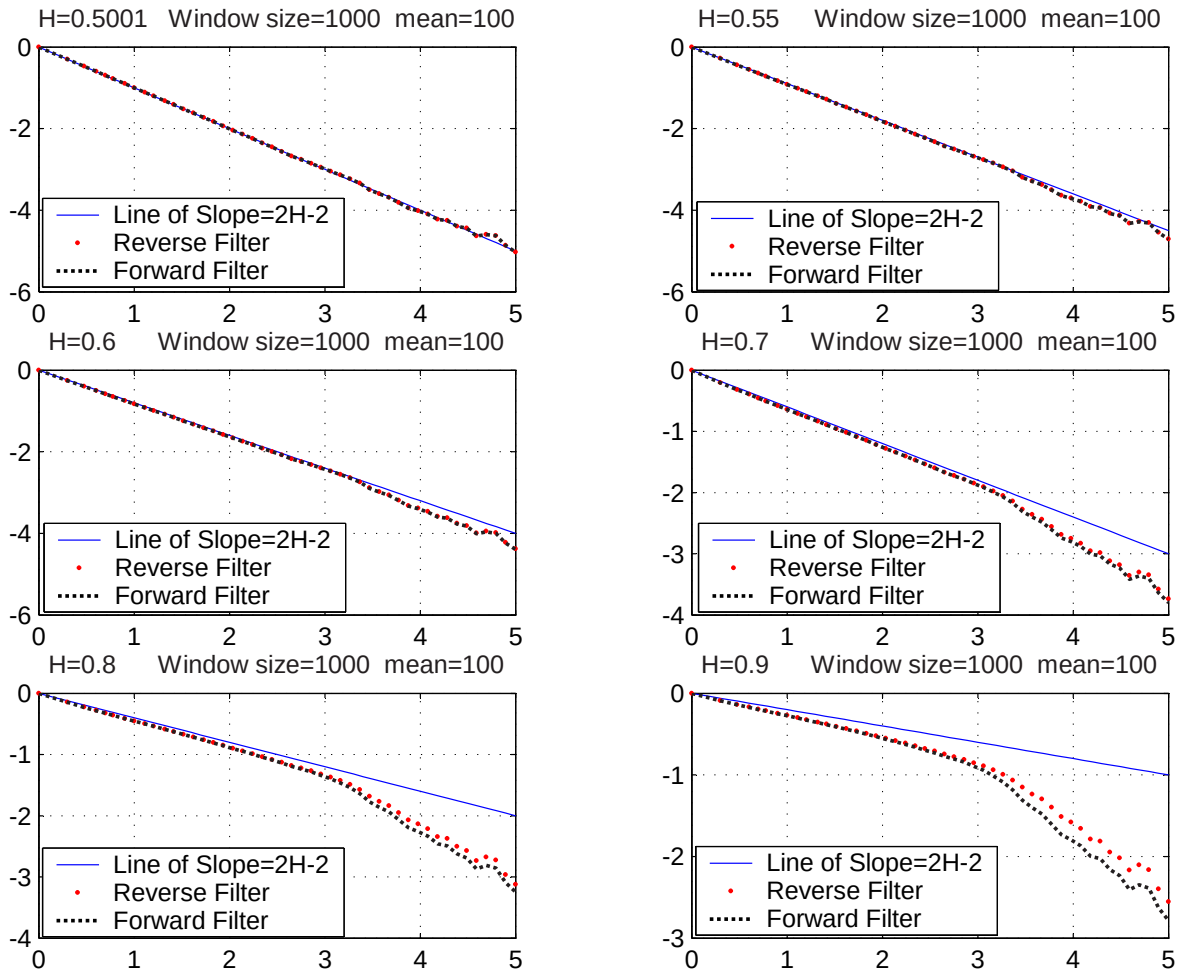

Figure 4.6: Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^3 and mean rate 100.

Table 4.4: The self-similar parameters for the synthetic traffics in Fig. 4.4. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter.

Window Size= 10^3 and mean rate= 1				
Targeted H	$\hat{H}_f(m \leq W)$	$\hat{H}_r(m \leq W)$	$\hat{H}_f(W \leq m \leq 10^{0.6}W)$	$\hat{H}_r(W \leq m \leq 10^{0.6}W)$
0.5001	0.498954604	0.498954576	0.478525976	0.478527232
0.55	0.545590738	0.545598865	0.500509826	0.502602652
0.6	0.593268447	0.593332835	0.518321516	0.526495677
0.7	0.687280370	0.687870085	0.543178024	0.574518351
0.8	0.775720702	0.777951755	0.557968425	0.625613037
0.9	0.852199226	0.857655140	0.566363453	0.681486674
Targeted H	$\hat{H}_f(m > W)$	$\hat{H}_r(m > W)$	$\hat{H}_f$(for all m)	$\hat{H}_r$(for all m)
0.5001	0.518747190	0.518746804	0.500920574	0.500920620
0.55	0.525569713	0.526309565	0.536311487	0.536736510
0.6	0.531094180	0.533958460	0.571247273	0.572955153
0.7	0.538934659	0.549870753	0.636851387	0.643810334
0.8	0.543839017	0.567678880	0.695110535	0.711055666
0.9	0.546870991	0.588444618	0.743354416	0.771975380

Table 4.5: The self-similar parameters for the synthetic traffics in Fig. 4.5. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter.

Window Size= 10^3 and mean rate= 10				
Targeted H	$\hat{H}_f(m \leq W)$	$\hat{H}_r(m \leq W)$	$\hat{H}_f(W \leq m \leq 10^{0.6}W)$	$\hat{H}_r(W \leq m \leq 10^{0.6}W)$
0.5001	0.502022764	0.502022735	0.504580767	0.504581169
0.55	0.548758912	0.548772242	0.524862802	0.527203785
0.6	0.596560397	0.596650465	0.541984015	0.551058440
0.7	0.690857663	0.691593635	0.567596626	0.601536517
0.8	0.779493096	0.782145021	0.584688534	0.655307404
0.9	0.855816829	0.862064300	0.595934355	0.711047831
Targeted H	$\hat{H}_f(m > W)$	$\hat{H}_r(m > W)$	$\hat{H}_f$(for all m)	$\hat{H}_r$(for all m)
0.5001	0.491919890	0.491920678	0.500487360	0.500487604
0.55	0.497627154	0.498322413	0.535845128	0.536272321
0.6	0.502197715	0.504904149	0.570728389	0.572445213
0.7	0.508544643	0.518987825	0.636152336	0.643146662
0.8	0.512337609	0.535332771	0.694081245	0.710078572
0.9	0.514502781	0.555080831	0.741772136	0.770366287

Table 4.6: The self-similar parameters for the synthetic traffics in Fig. 4.6. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter.

Window Size= 10^3 and mean rate= 100				
Targeted H	$\hat{H}_f(m \leq W)$	$\hat{H}_r(m \leq W)$	$\hat{H}_f(W \leq m \leq 10^{0.6}W)$	$\hat{H}_r(W \leq m \leq 10^{0.6}W)$
0.5001	0.498786062	0.498786136	0.448318471	0.448319907
0.55	0.545667383	0.545669359	0.469875293	0.472046238
0.6	0.593706762	0.593747221	0.487915016	0.496406404
0.7	0.688673810	0.689171999	0.514443554	0.546892233
0.8	0.778115182	0.780141748	0.531556384	0.601076016
0.9	0.855235338	0.860290147	0.542242573	0.659653600
Targeted H	$\hat{H}_f(m > W)$	$\hat{H}_r(m > W)$	$\hat{H}_f$(for all m)	$\hat{H}_r$(for all m)
0.5001	0.512064537	0.512063485	0.501342956	0.501343166
0.55	0.519304017	0.520059567	0.536811554	0.537233878
0.6	0.525188436	0.528100127	0.571806381	0.573501601
0.7	0.533575508	0.544535530	0.637451880	0.644335203
0.8	0.538818682	0.562389566	0.695592699	0.711314039
0.9	0.542021966	0.582771662	0.743467215	0.771610729

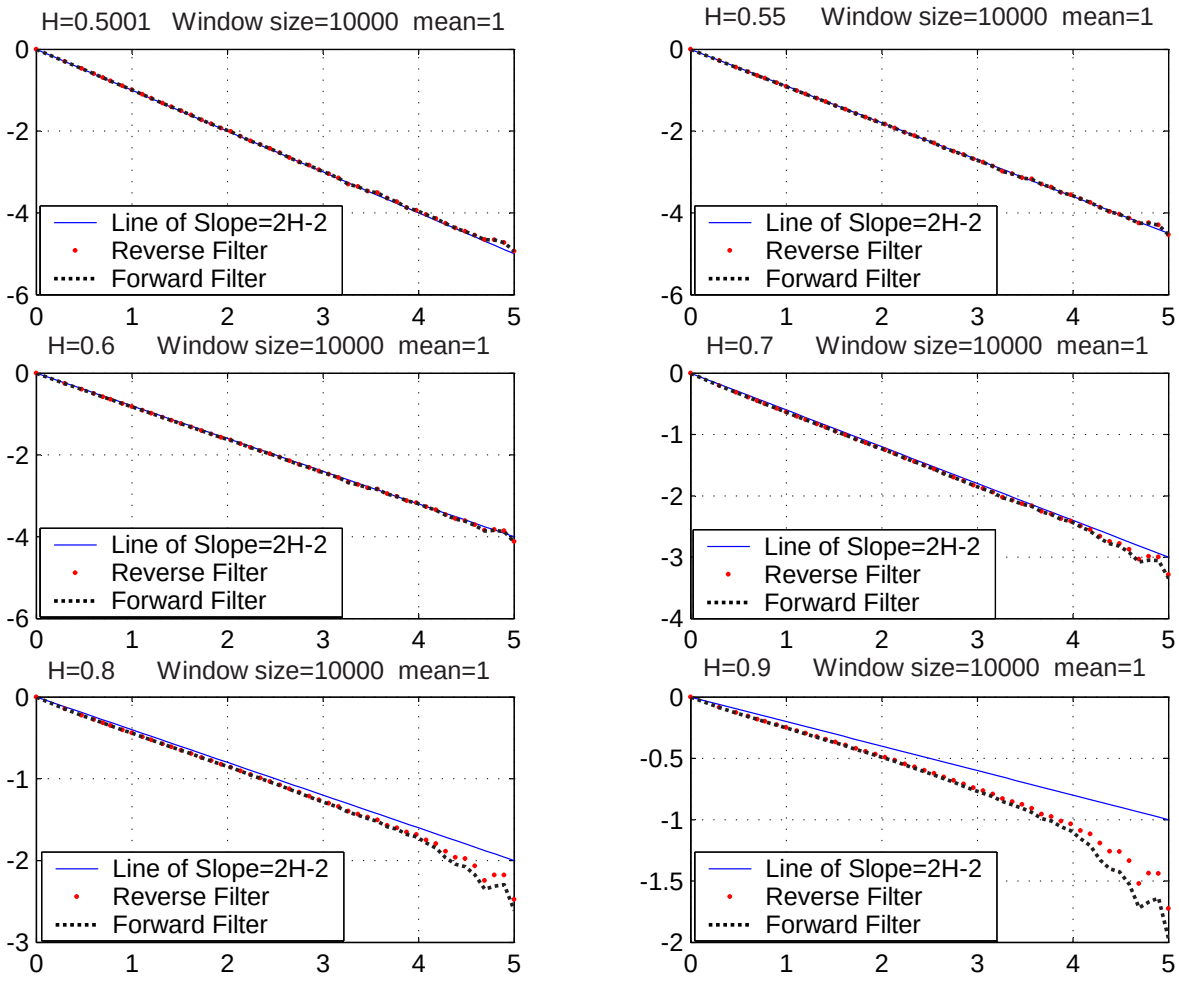

Figure 4.7: Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^4 and mean rate 1.

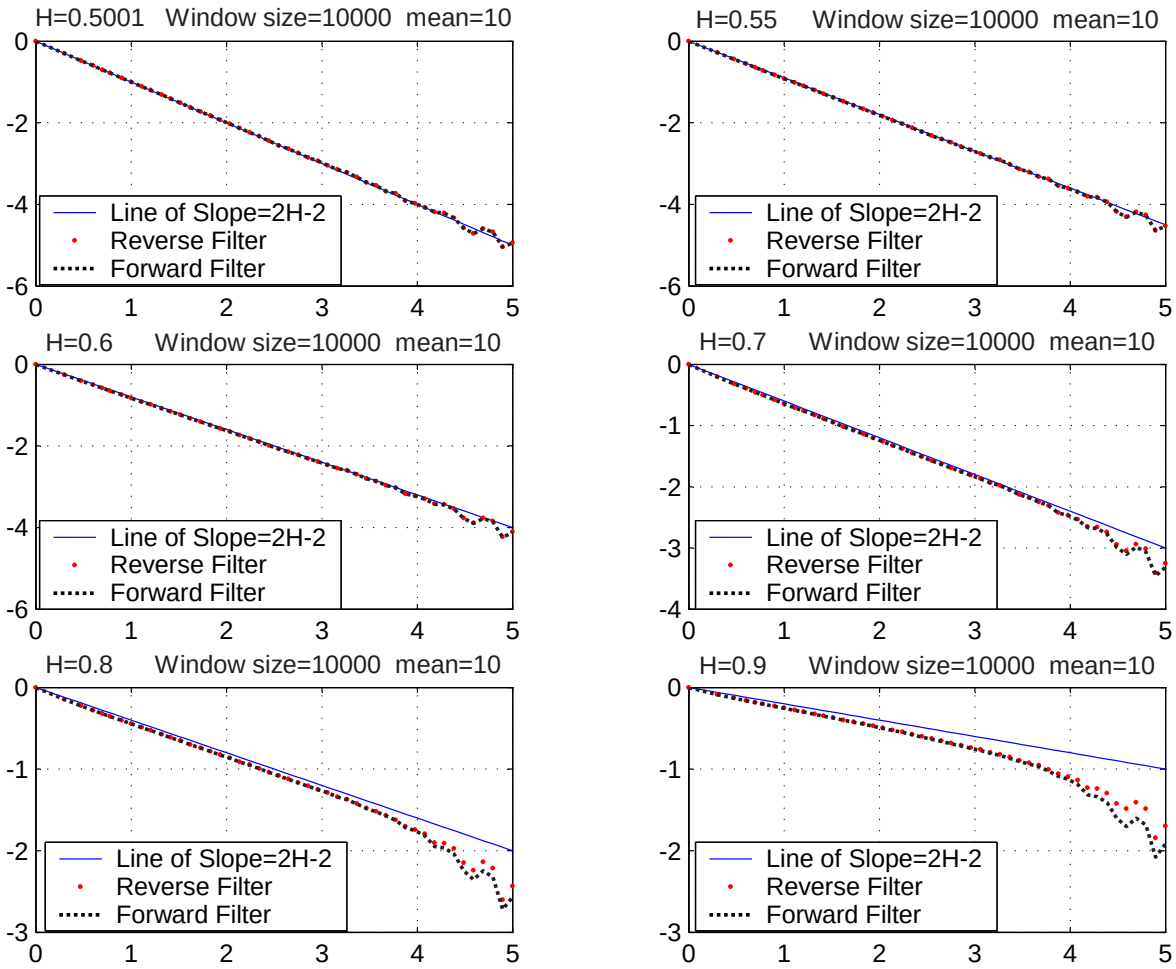

Figure 4.8: Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^4 and mean rate 10.

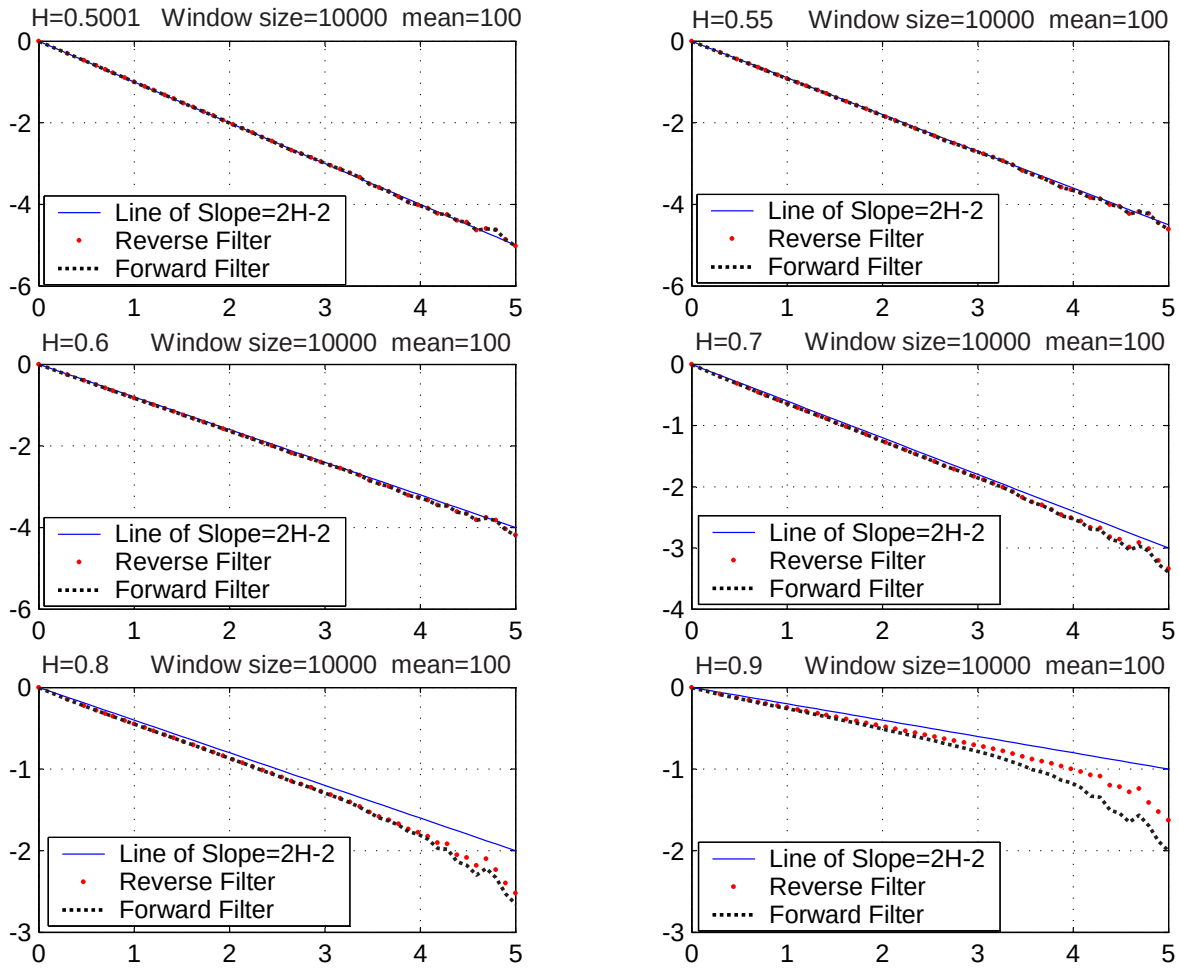

Figure 4.9: Variance-time plots ($\log_{10}$ scale) for the two filter-based synthetic arrivals with truncation window 10^4 and mean rate 100.

Table 4.7: The self-similar parameters for the synthetic traffics in Fig. 4.7. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter.

Window Size= 10^4 and mean rate= 1				
Targeted H	$\hat{H}_f(m \leq W)$	$\hat{H}_r(m \leq W)$	$\hat{H}_f(W \leq m \leq 10^{0.6}W)$	$\hat{H}_r(W \leq m \leq 10^{0.6}W)$
0.5001	0.502663565	0.502663591	0.499656574	0.499657533
0.55	0.550436362	0.550458536	0.529486644	0.531872041
0.6	0.599215607	0.599329526	0.554802737	0.563995995
0.7	0.695678616	0.696397994	0.593413833	0.627446413
0.8	0.787441046	0.789737858	0.620335402	0.691490788
0.9	0.867678988	0.872544713	0.639188033	0.758955409
Targeted H	$\hat{H}_f(m > W)$	$\hat{H}_r(m > W)$	$\hat{H}_f$(for all m)	$\hat{H}_r$(for all m)
0.5001	0.552278927	0.552279392	0.506475740	0.506475853
0.55	0.566694932	0.568272430	0.551946678	0.552185760
0.6	0.578331075	0.584440386	0.597617025	0.598598318
0.7	0.594684707	0.617913286	0.685729718	0.689869001
0.8	0.604707659	0.655533075	0.766961756	0.776695185
0.9	0.610720742	0.700961849	0.836245417	0.853715090

Table 4.8: The self-similar parameters for the synthetic traffics in Fig. 4.8. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter.

Window Size= 10^4 and mean rate= 10				
Targeted H	$\hat{H}_f(m \leq W)$	$\hat{H}_r(m \leq W)$	$\hat{H}_f(W \leq m \leq 10^{0.6}W)$	$\hat{H}_r(W \leq m \leq 10^{0.6}W)$
0.5001	0.502071229	0.502071188	0.369091988	0.369092158
0.55	0.549947402	0.549938034	0.396368318	0.399037077
0.6	0.598857151	0.598836571	0.420805580	0.431325303
0.7	0.695576474	0.695684350	0.460111180	0.500602906
0.8	0.787477330	0.788168744	0.488704018	0.577046316
0.9	0.867614568	0.870113146	0.509177667	0.667264948
Targeted H	$\hat{H}_f(m > W)$	$\hat{H}_r(m > W)$	$\hat{H}_f$(for all m)	$\hat{H}_r$(for all m)
0.5001	0.493786355	0.493785923	0.500678600	0.500678580
0.55	0.511942270	0.513691703	0.545710003	0.545921730
0.6	0.527349180	0.534175325	0.591022843	0.591895027
0.7	0.527349180	0.576389347	0.678637982	0.682326966
0.8	0.565683305	0.621857514	0.759549886	0.768129352
0.9	0.575360873	0.674978596	0.828557137	0.844703731

Table 4.9: The self-similar parameters for the synthetic traffics in Fig. 4.9. $\hat{H}_f$ and $\hat{H}_r$ represent the obtained values for forward filter approach and reverse filter approach, respectively. H is filter parameter used in the filter synthesizers, which can be viewed as the targeted self-similar parameter.

Window Size= 10^4 and mean rate= 100				
Targeted H	$\hat{H}_f(m \leq W)$	$\hat{H}_r(m \leq W)$	$\hat{H}_f(W \leq m \leq 10^{0.6}W)$	$\hat{H}_r(W \leq m \leq 10^{0.6}W)$
0.5001	0.497246198	0.497246120	0.511465019	0.511466758
0.55	0.544548086	0.544563855	0.531073160	0.532938024
0.6	0.592923733	0.593015851	0.547128319	0.554365159
0.7	0.688771672	0.689408190	0.570131197	0.598184174
0.8	0.780205286	0.782078295	0.584353974	0.653841934
0.9	0.860654818	0.878830584	0.592574436	0.748523990
Targeted H	$\hat{H}_f(m > W)$	$\hat{H}_r(m > W)$	$\hat{H}_f$(for all m)	$\hat{H}_r$(for all m)
0.5001	0.539545377	0.539543479	0.501524117	0.501524117
0.55	0.557843286	0.559306594	0.546162490	0.546380935
0.6	0.572982555	0.578642629	0.591086999	0.591998771
0.7	0.594811150	0.616269697	0.678042970	0.681997106
0.8	0.608365390	0.657394621	0.758625797	0.768168948
0.9	0.616268404	0.693442538	0.827941415	0.861618275

4.2 Integerization Effect

Hua and Chen [2] proposed to use "rounding $y[n]$ to the nearest non-negative integer" mechanism for the sake of simplicity; however, such a mechanism may destroy the self-similar structure of the output sequence, if the mean traffic rate of the filter input is too small. Can our proposed "integerization method" amend this problem? As it is hard to examine the influence of integerization operation by analysis, simulation-based examination is performed instead.

From Figs. 4.10–4.18, we observe that the simulation results are almost identical to those obtained in the previous section. This indicates that our "rounding on accumulative arrival" can truly preserve the self-similarity of the real-valued synthetic arrivals.

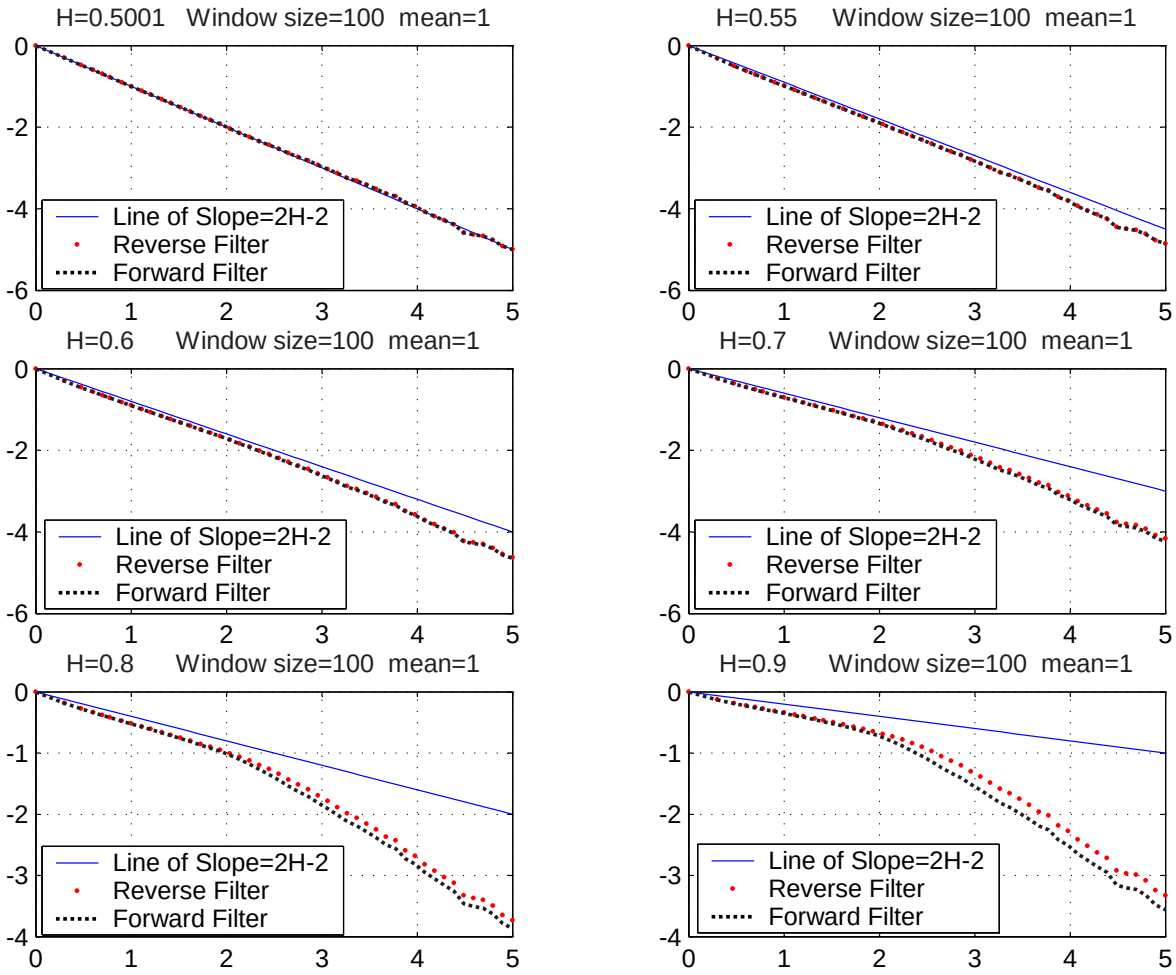

Figure 4.10: Variance-time plots ($\log_{10}$ scale) for the two filter-based integerized synthetic arrivals with truncation window 10^2 and mean rate 1.

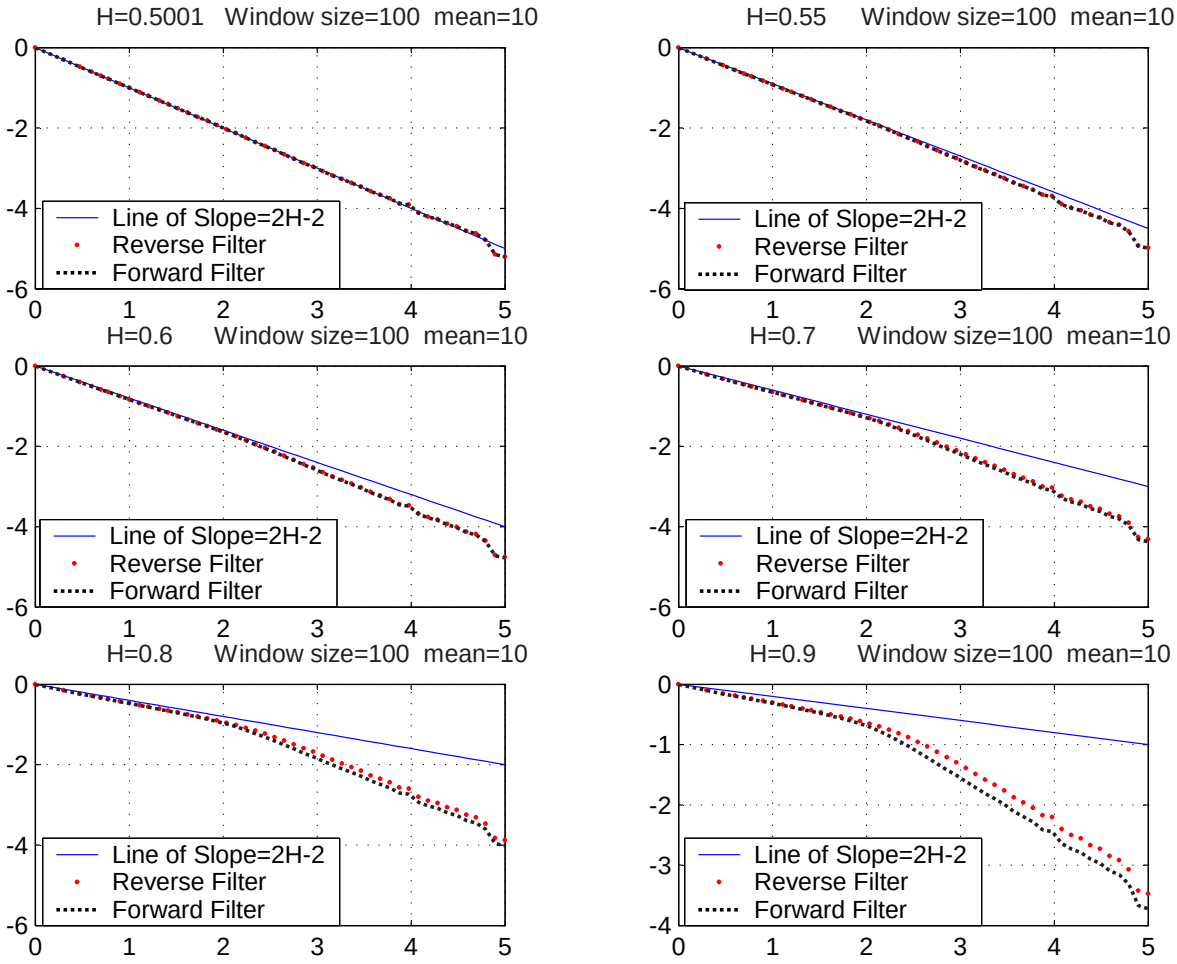

Figure 4.11: Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^2 and mean rate 10.

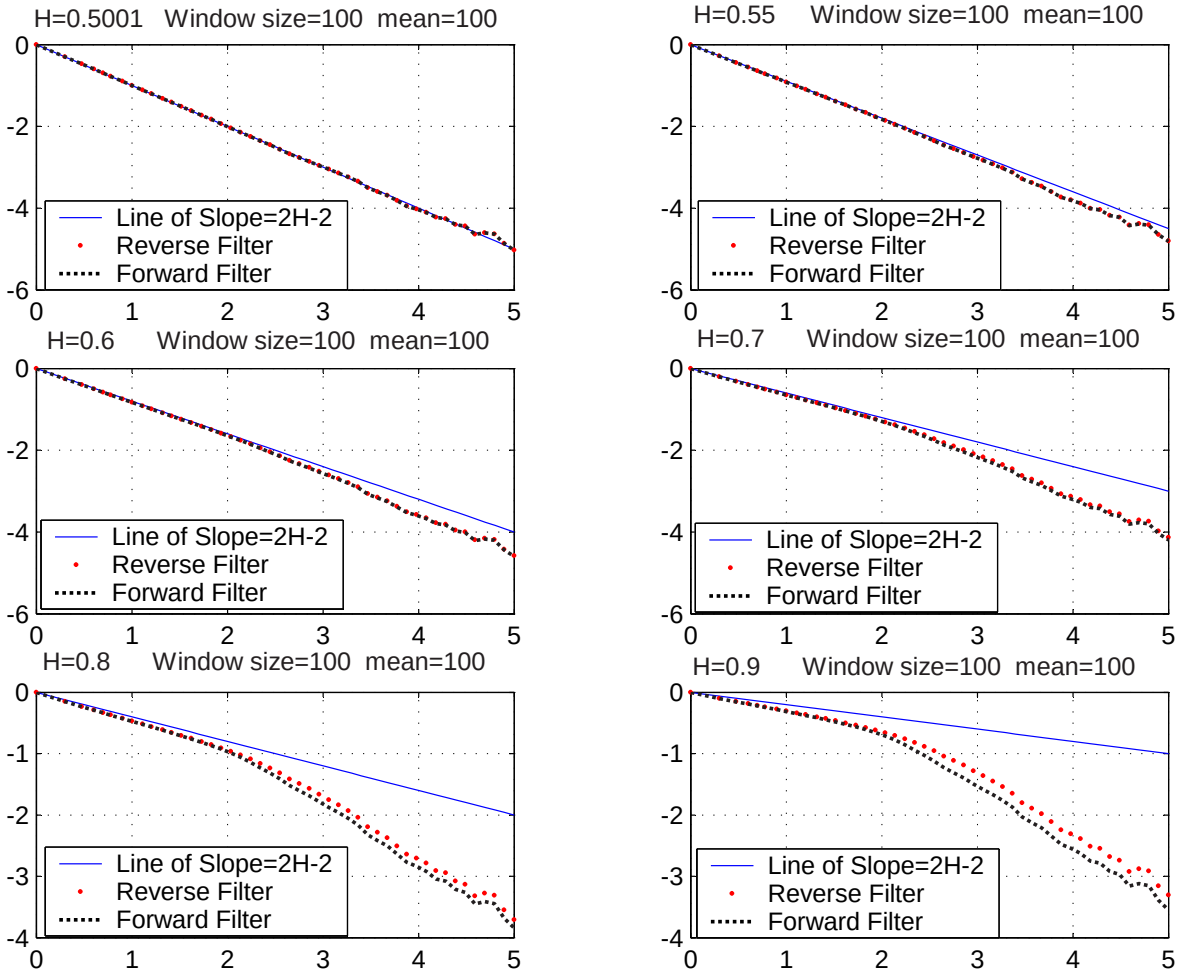

Figure 4.12: Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^2 and mean rate 100.

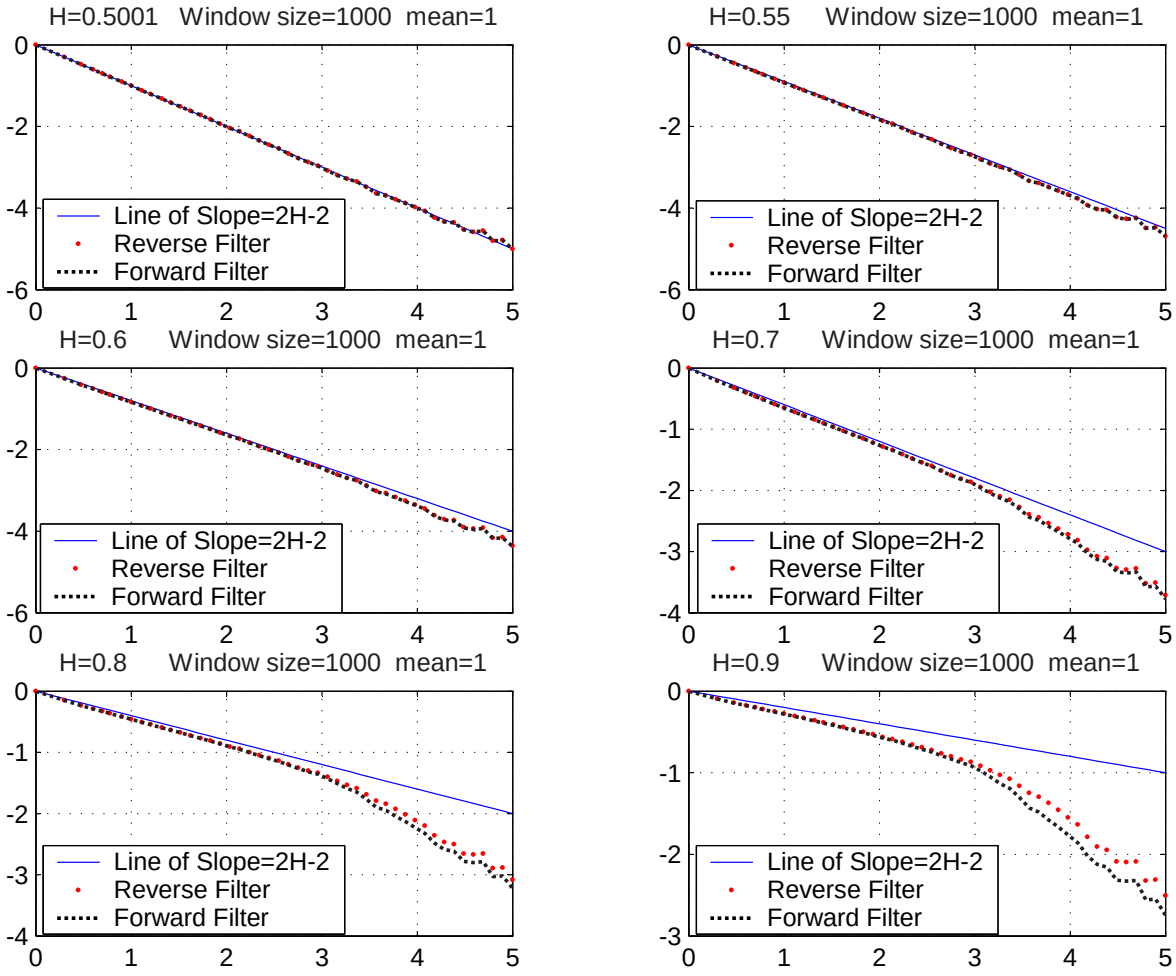

Figure 4.13: Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^3 and mean rate 1.

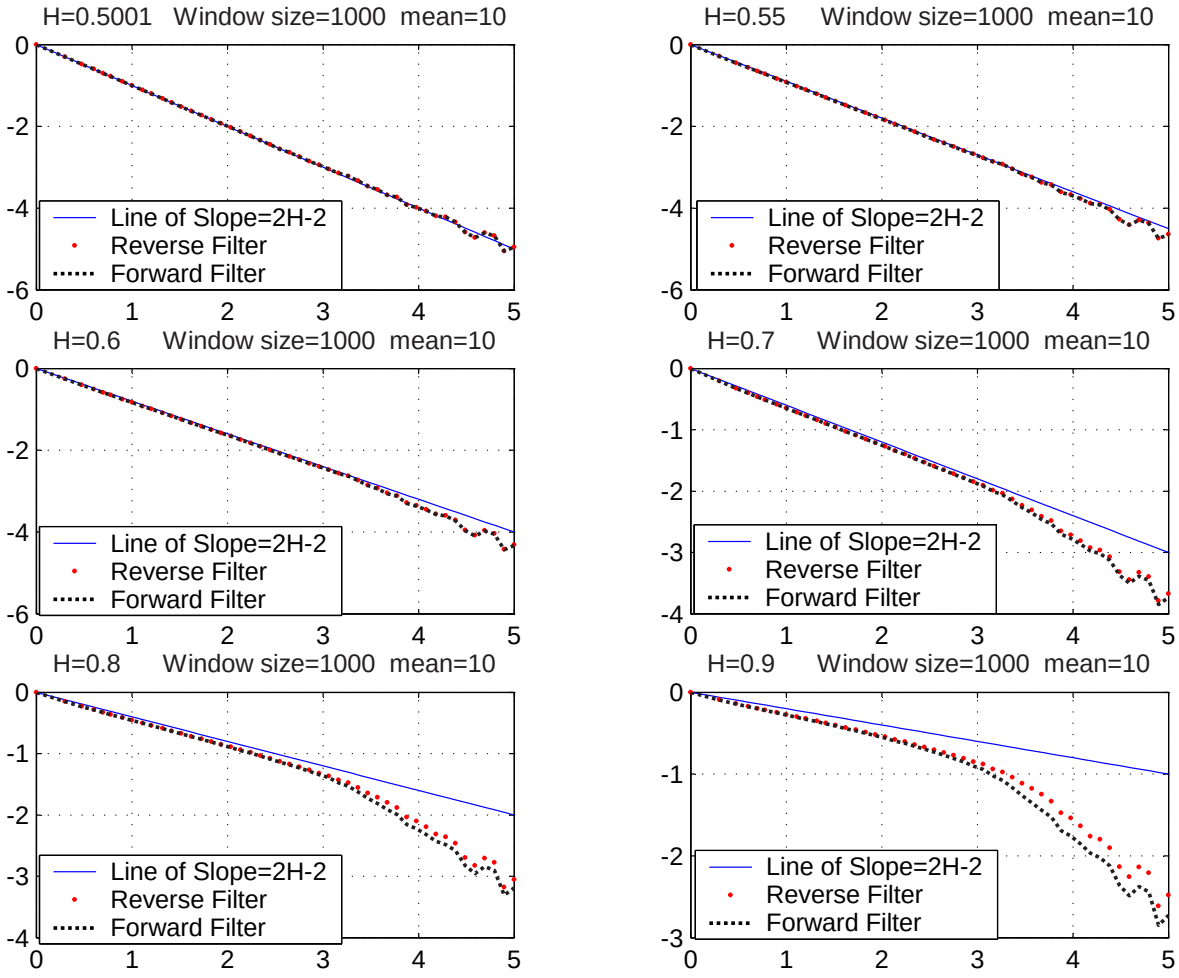

Figure 4.14: Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^3 and mean rate 10.

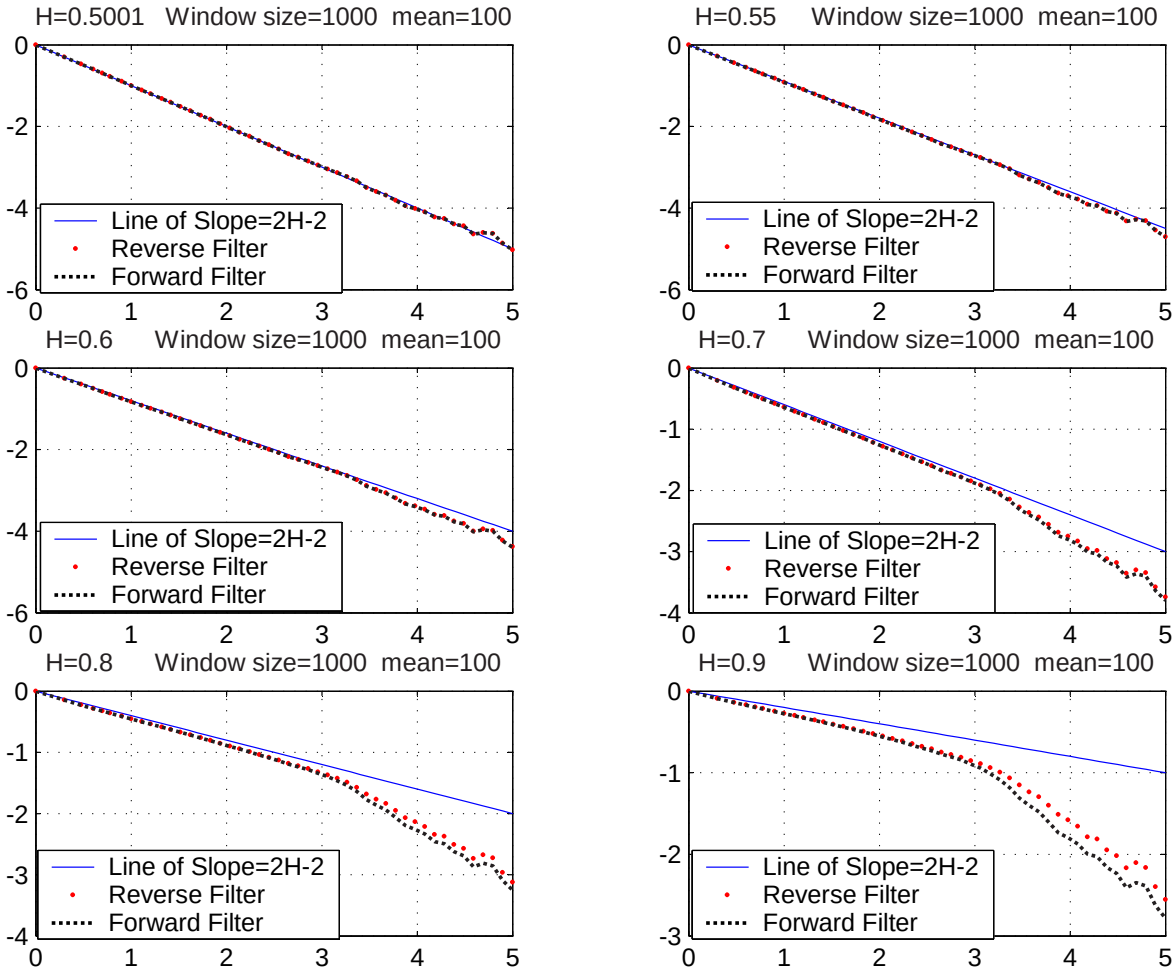

Figure 4.15: Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^3 and mean rate 100.

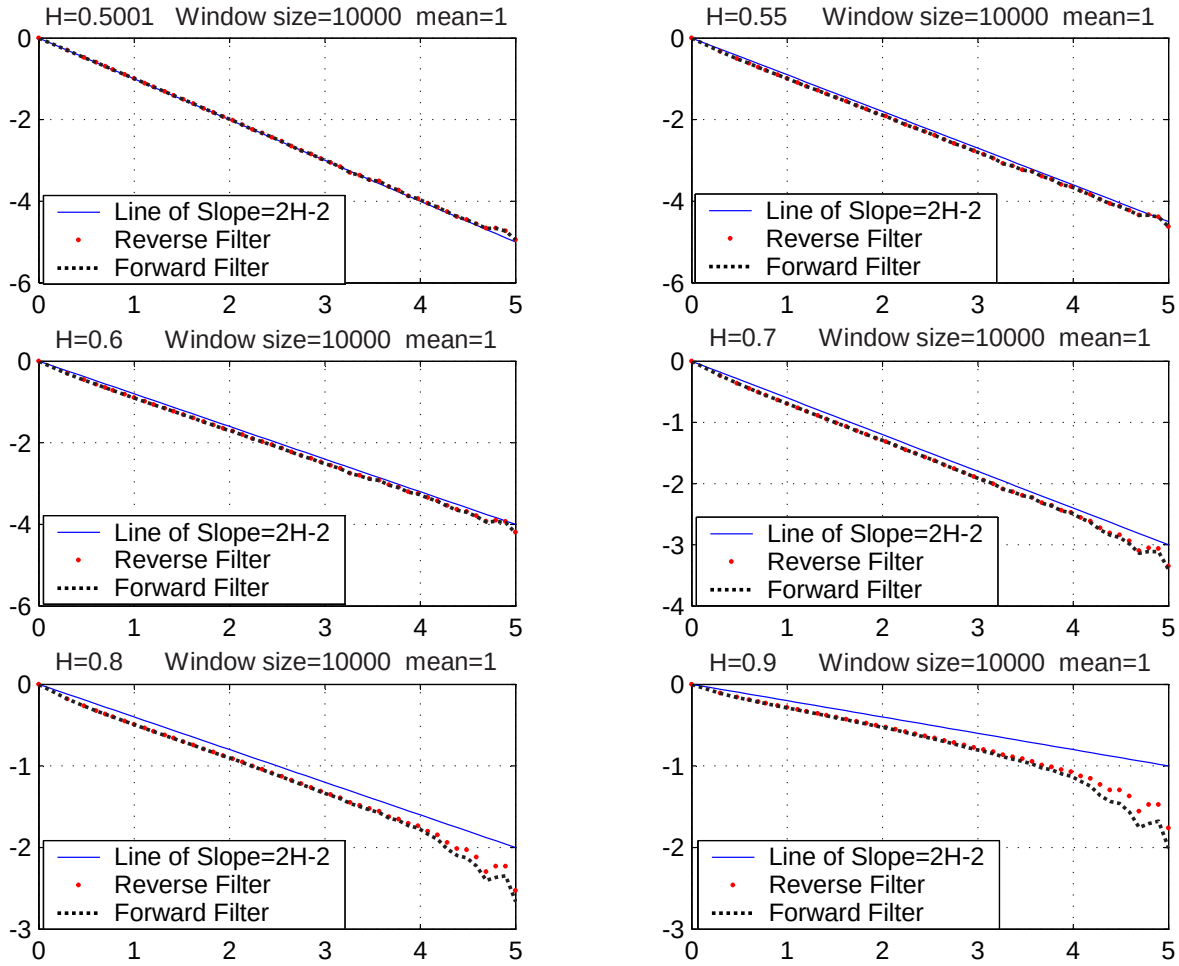

Figure 4.16: Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^4 and mean rate 1.

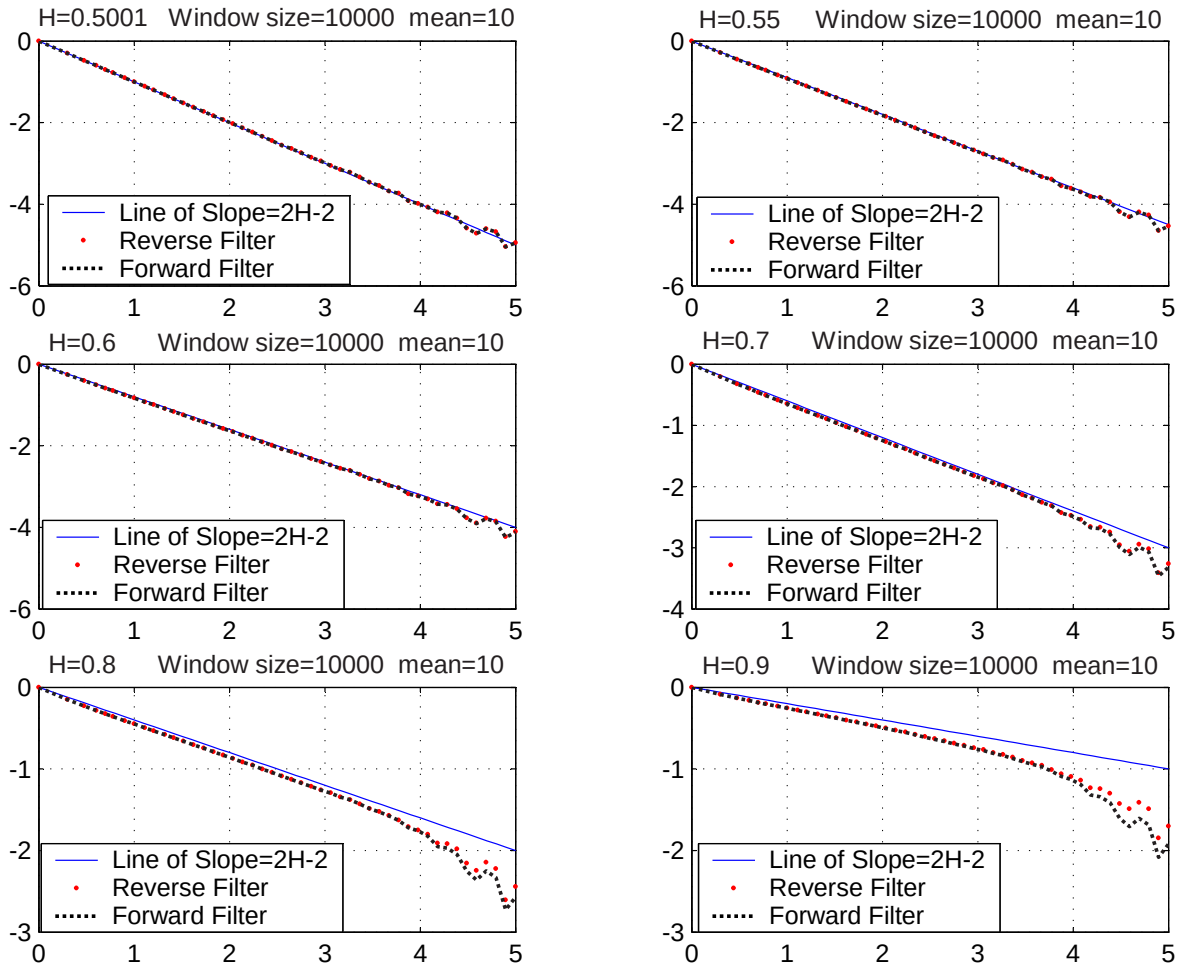

Figure 4.17: Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^4 and mean rate 10.

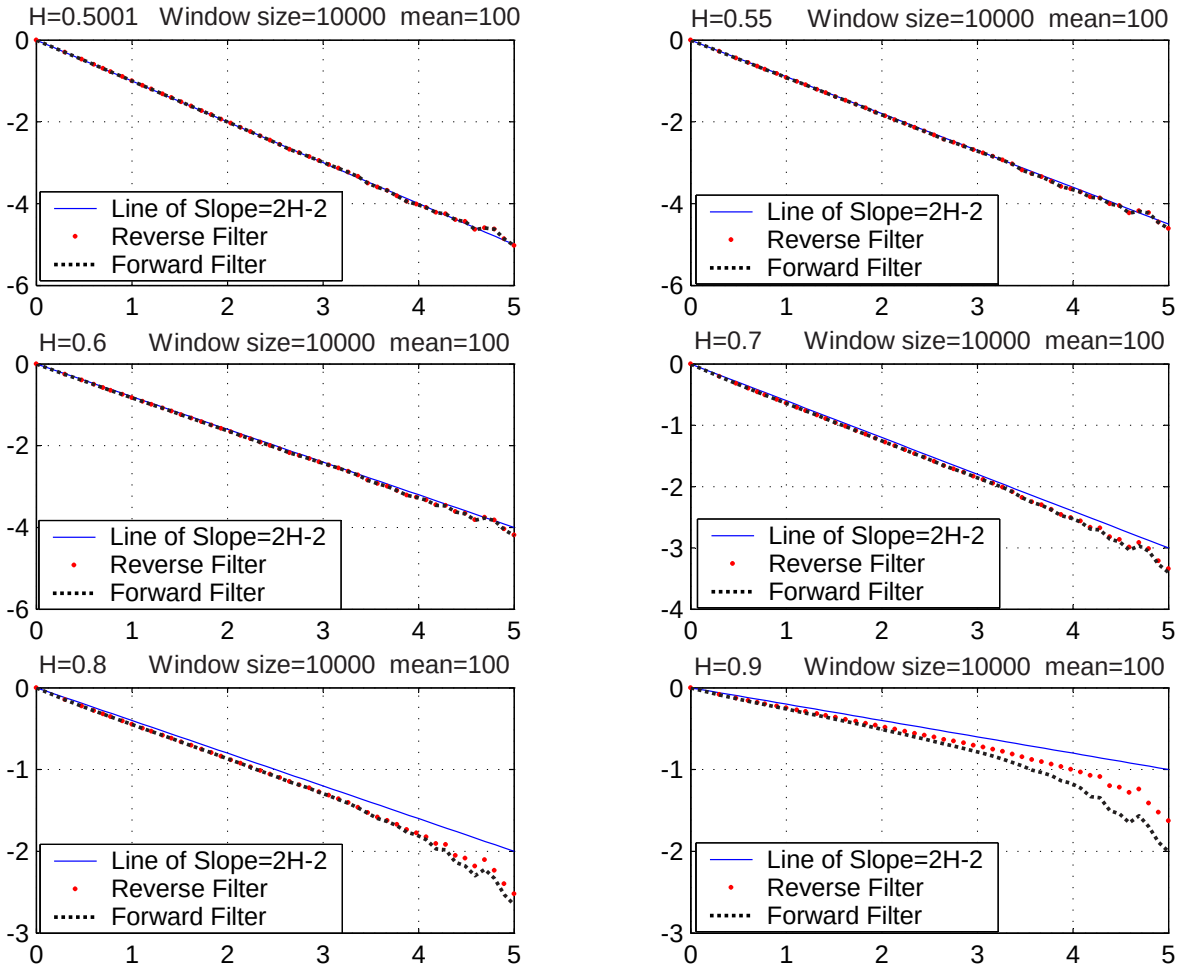

Figure 4.18: Variance-time plots ($\log_{10}$ scale) for the two filter-based intergerized synthetic arrivals with truncation window 10^4 and mean rate 100.

Chapter 5

Conclusions and Further Work

5.1 Conclusions

In this thesis, a reverse filter technique is proposed for generating the self-similar network traffics. In general, a filter can be equivalently implemented in two structures: forward filter and reverse filter. These two structures become different when a finite truncation is applied to them.

From our simulations, we obtain that the reverse filter traffic synthesizer can give a more self-similar traffic, especially at larger H close to 1. It is worth mentioning that the reverse filter approach has exactly the same complexity as the forward filter approach; hence, a reverse filter traffic synthesizer is indeed a preferable structure even if its improvement over the forward filter approach is quite limited.

To summarize and conclude the research in our lab on filter-based traffic synthesizers, both forward- and reverse-filter approaches can generate long-range dependent traffic with adjustable levels of burstiness and correlation. Such a model is parsimonious in number of model parameters. Specifically, only three parameters are required, which are the target self-similar parameter H (that controls the burstiness and correlation of the synthetic traffic), the input mean λ (that defines the mean of the synthetic traffic), and the filter truncation

window W (that determines not only the length of the filter but also the valid aggregation size of preserved self-similar nature).

When they are compared with two existing self-similar traffic generators—the Paxson IFFT approach and the Ledesma/Liu FFT approach, filter-based approaches provides additional advantage that the synthetic traffic can be generated on the fly, and is always non-negative.

Both the Paxson IFFT approach and Ledesma/Liu FFT approach require $(n/2)(\log_2(n)+2)$ complex multiplications, where n is the length of the generated trace. This complexity basis is different from that of filter-based synthesizer, which requires $n \times W$ complex multiplications, where W represents the truncation window size. As W is usually fixed through the trace synthesization, we can say that the complexity of filter-based approaches is linear with respect to the trace size.

5.2 Further Work

The wavelet filter technique is popular in communications. It is our feeling that by introducing a wavelet filter, a better performance (in the sense of synthesizing a more self-similar trace) may be able to be achieved, especially a feasible assumption on finite filter length is applied. It could be an interesting future work along this research line.

Bibliography

[1] J. Beran, *Statistics for Long-Memory Processes*, Chapman & Hall: New York, 1994.

[2] K. L. Hua, *On Generator of Network Arrivals with Self-Similar Nature*, Master Thesis, Department of Communications Engineering, National Chiao Tung Unversity, June 1997.

[3] W. C. Lau, A. Erramilli, J. Wang and W. Willinger. "Self-similar traffic generation: The Random midpoint eisplacement algorithm and its properties," *IEEE Proc. ICC'95*, pp. 466–472, Seattle, WA, 1995.

[4] S. Ledesma and D. Liu, "A fast method for generation self-similar network traffic," *Communication Technology Proceedings*, 2000 WCC-ICCT, 2000.

[5] B. B. Mandelbrot, "A fast fractional Gaussian noise generator," *Water Resources Research*, vol. 7, pp. 543–553, 1971.

[6] B. B. Mandelbrot and J. W. Van Ness, "Fractional Brownian motions, fractional noises and applications," *SIAM Rev.*, vol. 10, pp. 422–437, 1968.

[7] B. B. Mandelbrot and J. R. Wallis, "Computer experiments with fractional Gaussian noises," *Water Resources Research*, vol. 5, pp. 228–267, 1969.

[8] V. Paxson, "Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic," *Computer Communication Review*, vol. 27, no. 5, pp. 5–18, Oct. 1997.

[9] W. Stallings, *High-Speed Networks and Internets: Performance and Quality of Service*, Prentice-Hall International, Inc, pp. 220–247, 2002.

[10] B. Tsybakov and N. D. Georganas, "Self-similar processes in communications networks," *IEEE Trans. Inform. Theory*, vol. 44, no. 5, pp. 1713–1725, Sept. 1998.

[11] W. Willinger, "Traffic modeling for high-speed network: Theory versus practice," In *Stochastic Networks*, vol. 71, pp. 395-409, 1995.

[12] W. Willinger, M. Taqqu, R. Sherman and D. Wilson, "Self-similar through high variability: Statistical analysis of Ethernet LAN traffic at the source level," *IEEE/ACM Trans. Networking*, Feb. 1997.