On the Chaotic Behavior of Variable-Bit-Rate Video and Its Application in Traffic Modeling*

Marwan Krunz† and Ahmad Alkhatib
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721

Last Revised: August 5, 2001

Abstract

Nonlinear chaos-based modeling offers an alternative approach to stochastic (typically, linear) approaches, with the advantages of lower dimensionality and more determinism. In this paper, we investigate the presence of chaos in variable-bit-rate (VBR) video traffic and explore its application in traffic synthesis and forecasting. We provide statistical evidence that points to the potential chaoticity of VBR video time series. Our evidence is based on the sensitivity of the trajectories to initial conditions, the correlation coefficient between the transformed video sequence (after filtering out any apparent autocorrelations) and a predicted version of it, and the estimated value of the maximum Lyapunov exponent. Accurate forecasting of the future values of a presumably chaotic time series requires good estimation of the *embedding dimension*. We present a novel approach for estimating the embedding dimension of a suspectedly chaotic time that is modeled according to the nonlinear functional relationship of Farmer and Sidorowich [10]. This approach indicates that the minimum embedding dimension in video sequences is seven. Using this estimate along with a modified forecasting approach of the one in [10], we generate synthetic video sequences and show that they exhibit chaotic behavior.

1 Introduction

Emerging network services are being designed to support a wide range of applications with diverse traffic characteristics and quality-of-service (QoS) requirements. Accurate modeling of the transported traffic is essential to achieving efficient resource allocation subject to QoS constraints and to designing online admission control strategies. In this paper, our focus is on modeling variable-bite-rate (VBR)

*An abridged version of this paper was presented at the *IEEE ICC 2000 Conference*, New Orleans, June 2000. This work was supported in part by the National Science Foundation under Career Award ANI 9733143.

†Correspondence author. Phone: (520) 621-8731, email: krunz@ece.arizona.edu

video traffic at the source, just before entering the network. Video traffic, in general, is expected to constitute a significant fraction of the traffic volume in future high-speed networks. To display constant-quality video, the encoder generates compressed frames whose sizes (in bits) vary depending on the innovation in the scene activity. Typically, frames are generated at a constant frame rate (e.g., 30 frames/second). Thus, characterizing the variability of the bit rate amounts to modeling the frame-size sequence. This VBR sequence is known to exhibit a sophisticated structure with multiple-time-scale variations and persistence correlations.

Classical approaches to traffic modeling (see, for example, the surveys in [1, 11]) often assume that the observed time series have been generated by a linear system with possibly random perturbations (i.e., noise). ARMA and ARIMA processes are examples of such approaches. On the one hand, linear models enjoy attractive simplicity and several systematic techniques for constructing and validating them are already available [5]. On the other hand, such models fail to capture the rich dynamics and apparent nonlinearity exhibited by several observed sequences. This deficiency has been a major motivation behind the inception of nonlinear chaos modeling.

The main goal of this paper is to explore the viability of applying chaos theory in modeling, characterizing, and forecasting time series produced by VBR video encoders. Chaos is an inherent property of *some* nonlinear dynamic systems. It manifests itself in the sensitivity of solutions (or trajectories) to initial conditions; a property that is advantageous in generating versatile synthetic sequences. Chaotic models provide a deterministic alternative to the probabilistic (mostly, linear) models, where in the former case the rich dynamics of the modeled data are captured by slightly perturbing the initial conditions every time a new trace is to be synthesized. In essence, a chaotic system contains an infinite number of unstable periodic orbits, which if modeled linearly would require an infinite-dimensional model to cover the broadband (infinite) frequency spectrum present in chaotic sequences. Consider, for example, the *logistic map*, which is one of the simplest chaotic systems. The logistic map is defined on the interval $[0, 1]$ according to the following relation:

$$x_n = 4\lambda x_{n-1}(1 - x_{n-1}), \quad n = 1, 2, \ldots \tag{1}$$

where $\lambda \in [0, 1]$ is a fixed parameter. Depending on the value of λ, this nonlinear map can exhibit different behaviors [27]. For $\lambda \leq 0.75$ the iterates of this system converge onto a limit point (an attractor) provided that the initial point x_0 is in the interval (0,1). As λ exceeds 0.75, the single-point attractor bifurcates into two fixed points; a phenomenon known as period doubling (as $n \to \infty$, x_n will alternate between the two fixed points). As we continue to increase λ, the period doubling bifurcates, giving solutions of periods 4, 8, 16, ..., until λ is about 0.892. For $\lambda > 0.892$, both chaotic orbits and odd-period limit cycles (period 3) start to appear. In the chaotic state, the resulting time series contains an infinite number of unstable periodic orbits [23], which is considered an advantage in characterization, especially during the reconstruction of the attractor in the phase space.

Chaos theory and its applications have been the subjects of many books and research articles.

Introductory material to the subject can be found, for example, in [16]. More theoretical treatment can be found in [22] and [26]. Practical methods for applying chaos theory in nonlinear time series modeling are discussed in [15]. These methods were later implemented in the public-domain TISEAN package [13], which is also used in our work. Survey articles on chaos-based nonlinear time series modeling are found in [18, 20]. From the standpoint of time series analysis, the interest in chaotic models is largely related to their low dimensionality (note, however, high-dimensional chaotic systems exist as well). So it is not surprising that estimating the dimension of chaotic time series has been the focus of many research papers (see, for example, [17, 19, 3, 2, 7, 6], and the references therein). As discussed in Section 3, several measures of dimensionality have been studied in the literature. However, such measures are often difficult to estimate from empirical data with an unknown underlying structure. Instead, the empirical data are typically *embedded* in some state space that preserves the invariant characteristics of the unknown structure (e.g., fractality, shape of the attractor, etc.). This so-called *delay coordinates* (or time-delay) embedding approach was first introduced by Packard et al. [24] and later studied more rigorously by Takens [28]. A critical aspect of this approach is the estimation of the *embedding dimension*, which for practical modeling and forecasting problems reflects the parsimony (and hence, complexity) of the model.

Recently, there has been growing interest in investigating the chaotic nature of network traffic and applying it in network-related queueing analysis [8, 4, 29]. Inspired by the observed self-similarity of network traffic and its relation to the fractal trajectories of chaotic systems, the authors in [8] investigated the feasibility of modeling the packet generation process by deterministic chaotic maps. Simple nonlinear maps with few parameters were used, which capture the fractal properties of the traffic. Furthermore, the authors outlined a purely deterministic approach for analyzing the queueing performance of a FIFO server with a chaotic input. The feasibility of queueing analysis under chaotic inputs was also discussed in [4], where large deviation methods were suggested as a possible means of pursuing such analysis. In [29] the authors investigated the chaotic behavior of TCP congestion control. Several tests were conducted to verify that for certain parameter values, TCP congestion control indeed exhibits the essential characteristics of a chaotic system (e.g., sensitivity to initial conditions, complex orbit periods, non-integer dimensionality, strange attractors, positive rates of expansion). The authors went further to suggest that the observed self-similarity in TCP traffic may be attributed to the chaoticity of the congestion control mechanism, rather than the heavy tailedness of the distributions of the burst periods.

The rest of the paper is structured as follows. In Section 2 we explore the presence of chaos in video traffic. Section 3 presents our approach for estimating the embedding dimension and the results of applying this approach to video traffic. Reconstructing the attractor of the chaos-based video model is discussed in Section 4. Finally, the paper is concluded in Section 5.

2 Exploring Chaos in Video Traffic

In this section, we explore the chaotic behavior of VBR video traffic. Testing for the chaoticity of time series has been in the center of many previous studies (see [21] and the references therein). These studies all point out to the difficulty of establishing the boundaries between deterministic chaos and stochasticity, especially when the time series is "contaminated" by noise. Therefore, our main interest in chaos stems from its potential advantage over linear approaches in modeling and forecasting video traffic. We are less interested in verifying whether a given time series is truly chaotic or not, which may not be an appropriate question in the first place. Nonetheless, the applicability of chaos-based techniques is clearly dependent on the *chaotic tendency* of the underlying time series. For this purpose, it would be useful to first investigate how much chaos (if any) is present in a video time series, which would motivate subsequent application of chaos-based techniques.

Our data consist of a sequence of 171,000 points, each representing the size (in bytes) of a compressed video frame. This sequence was generated by compressing the *Star Wars* movie using a motion-JPEG encoder [12] (see the reference for details of the encoding procedure). We use three techniques to search for chaos in the underlying time series. The first technique is subjective while the last two are more objective.

2.1 Sensitivity to Initial Conditions

One reason to suspect the chaoticity of a time series is if small perturbations in the initial conditions lead to divergent trajectories. Consider a time series $\{x_n : n = 1, 2, \ldots, N\}$. If this series was generated by a purely deterministic dynamical system, then its evolution can be described by a nonlinear difference equation of the following form [10]:

$$x_{n+1} = F(X_n) \tag{2}$$

where F is a nonlinear mapping from $\mathbb{R}^d$ onto $\mathbb{R}$, $X_n = [x_n \;\; x_{n-1} \;\; x_{n-2} \cdots x_{n-d+1}]$ is a d-dimensional vector of previous values known as the *state vector*, and d is the *embedding dimension* of the system. As we show in Section 3, the *Star Wars* time series can be sufficiently embedded in a 7D state space, i.e., there are no two identical 7D state vectors that also have the same image in the range of F. So we can get a preliminary indication of a chaotic behavior if we can show that small perturbations in the 7D state vectors of this time series cause the rest of the trajectory to diverge from its original path. Such perturbations can be emulated by embedding the trace in a 6D state space and looking for pairs of 6D state vectors for which the two vectors constituting the pair are identical in value but have different images in the range of F. Table 1 depicts six such pairs taken from different locations within the trace. When the vectors in each pair are concatenated with their images (third column in the table), they appear as if they are slightly perturbed versions of one another in the 7D space. The last column indicates the amount of perturbation.

Pair Index	Value	Next Data Point	Perturbation
1	(464,460,456,459,462,457)	454/449	6
2	(511,512,514,512,511,510)	510/517	7
3	(540,536,535,535,538,535)	536/537	1
4	(555,552,552,556,553,552)	556/555	1
5	(562,553,555,548,561,564)	563/558	5
6	(639,638,638,636,637,633)	635/640	5

Table 1: Pairs of identical 6D state vectors in the *Star Wars* sequence with different images.

Each 7-dimensional state vector in Table 1 can be thought of as a set of initial conditions for a dynamic system whose trajectory is provided by the embedded points that follow the "initial" state vector. Figure 1 depicts the subsequent 600 data points in the video trace that follow each pair of initial state vectors. It is clear that when the system is initiated by state vectors that are very close in value but are not identical, the resulting trajectories diverge from each other, suggesting the presence of chaos in the video trace (i.e., the underlying dynamics is sensitive to initial conditions). Note that since $d = 7$, the trajectory that follows any 7D state vector must be unique.

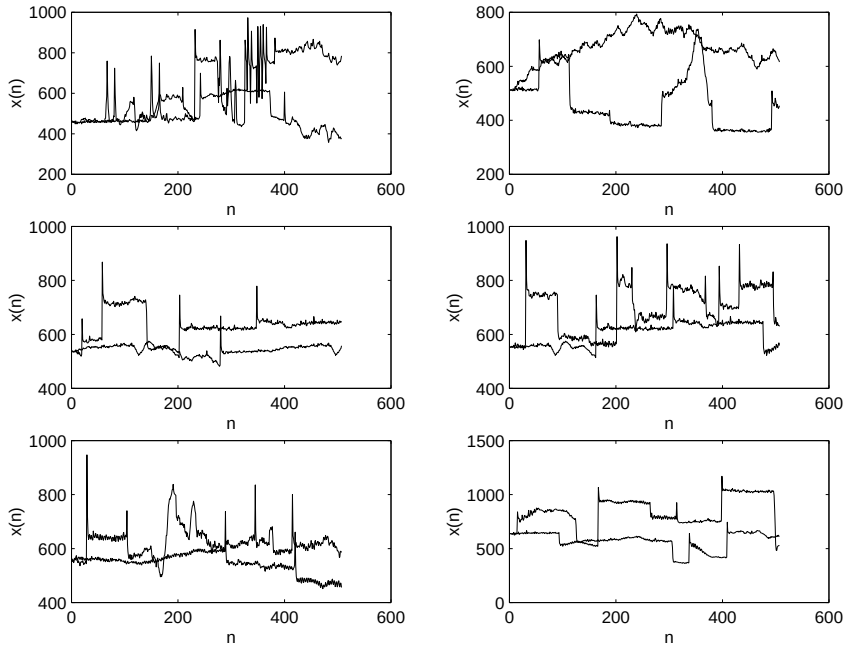


Figure 1: Preliminary evidence of chaos in video traffic (sensitivity to initial conditions).

2.2 Short-Term Forecasting Capability

Several statistical tools have been previously proposed for discriminating between deterministic chaos and stochasticity (see [21] and the references therein). Naturally, these tools are most efficient in discriminating between pure chaoticity (i.e., in the absence of measurement noise) and white noise. In

[21] Marzocchi et al. presented a technique for detecting low-dimensional chaos in noise-contaminated time series. The intuition behind this technique is that, due to their determinism, purely chaotic systems have better *short-term* prediction capability than purely random processes. This superior capability, however, deteriorates quickly as the forecasting window becomes larger. To put their intuition into work, Marzocchi et al. suggested transforming the original time series into one that has almost the same spectrum as white noise. This is achieved by subtracting from the original series an appropriately fitted autoregressive of order k component, where $k \geq 1$. Then, the correlation coefficient (ρ) between the transformed time series and a "predicted" version of it (see below) is plotted as a function of the embedding dimension (d). For chaotic series, ρ increases with d up to some point, say d^*, beyond which the trend is reversed and the correlation coefficient starts decreasing due to the increase in the number of degrees of freedom. In contrast, for pure white noise, the correlation coefficient is almost independent of the value of d. Figure 2 depicts the correlation coefficient as a function of d for a sample trajectory of the logistic map (with $\lambda = 1$ and $x_0 = 0.7$) and for a sample path of Gaussian white noise. The sudden drop in ρ for $d > 10$ is a clear indication of the chaoticity of the logistic map. In practice, the time series of interest are neither purely chaotic nor purely stochastic, and therefore the trends in the correlation coefficient are more difficult to interpret.

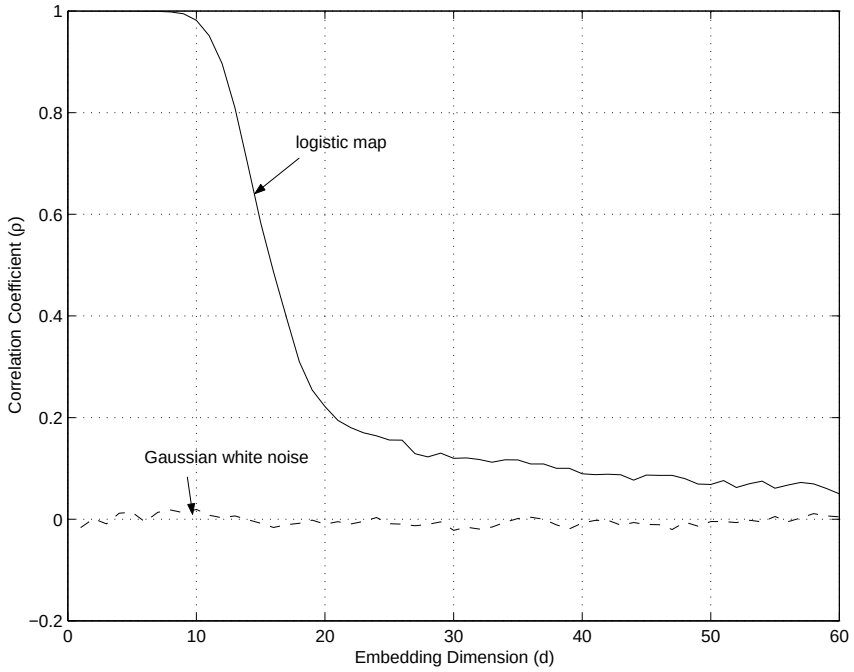


Figure 2: Correlation coefficient between actual and predicted time series versus embedding dimension.

In order to compute ρ, one needs to obtain a predicted version of the time series under study. This is often done using the nearest-neighbor approach (also known as the *analogue method*), which is briefly described as follows. First, the examined time series is embedded using the time-delay method in a d-dimensional state space, for some d. Let $X_i = [x_i \;\; x_{i-1} \;\; x_{i-2} \cdots x_{i-d+1}]$ be the most recent state

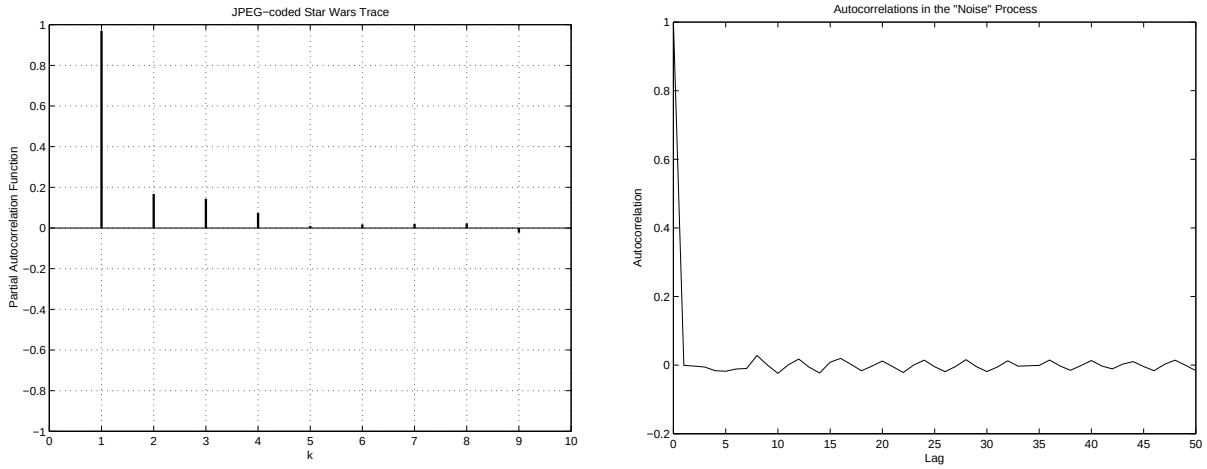
vector. To predict x_{i+p}, $p \geq 1$, we first look within previous state vectors for the nearest neighbor to X_i, say X_j where $j < i$. Then, we take x_{j+p} as a prediction of x_{i+p}. It is also possible to employ more sophisticated nearest-neighbor approaches (e.g., finding the k-nearest neighbors and using the average of their mappings as the predictor [20]), at the expense of added computational cost.

We now apply the above approach to the *Star Wars* sequence. A plot of the normalized autocorrelation function (ACF) of this sequence (not shown) indicates the presence of strong autocorrelations. To filter out most of these autocorrelations, we first fit an AR(k) model to the *Star Wars* sequence. The value of k can be determined by computing the *partial autocorrelation function* (PACF) of the time series [5, Chapter 3]. Figure 3(a) depicts the PACF for the *Star Wars* sequence. It indicates nonnegligible one through four autoregressive modes. Hence, we use an AR(4) model. Let $\{x_n : n = 1, 2, \ldots\}$ be the original sequence. Then, the transformed sequence (after subtracting the AR(4) component) is given by:

$$\acute{x}_n = x_n - a_1 x_{n-1} - a_2 x_{n-2} - a_3 x_{n-3} - a_4 x_{n-4}, \quad n = 5, 6, \ldots \tag{3}$$

$$= x_n - 0.7727 x_{n-1} - 0.0478 x_{n-2} - 0.0854 x_{n-3} - 0.07 x_{n-4} \tag{4}$$

where the coefficients $a_1, \ldots, a_4$ are determined using standard autoregression analysis [5]. Figure 3(b) depicts the ACF of the process $\{\acute{x}_n : n = 1, 2, \ldots\}$. Clear, the transformed process is almost uncorrelated, reflecting the goodness of the AR(4) fit.

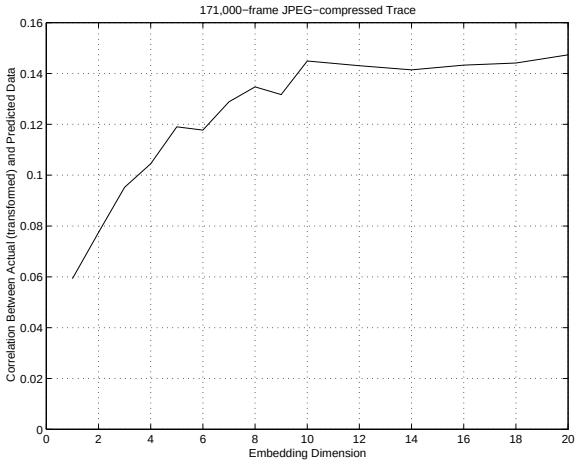


(a) Partial autocorrelation function (b) ACF of the residual ("noise") process

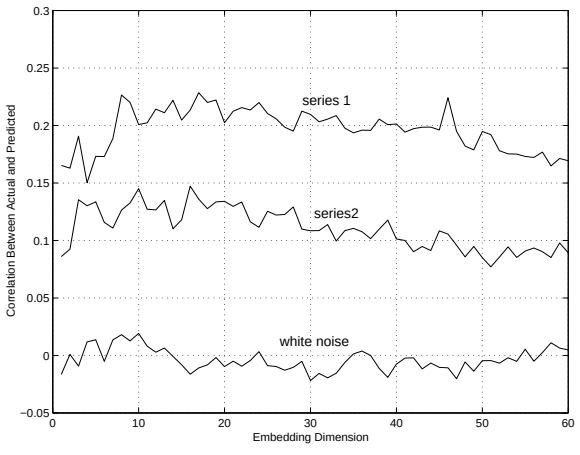
Figure 3: Fitting the *Star Wars* sequence using an AR(4) model.

Now that most of the correlations in the video time series are filtered out, we can apply Marzocchi's test to the transformed sequence $\{\acute{x}_n : n = 5, 6, \ldots\}$. Figure 4(a) depicts the correlation coefficient between this sequence and its predicted version as a function of d. As d increases, ρ increases as well, clearly deviating from the behavior of white noise. Unfortunately, due to the excessive time associated with the computation of ρ for large values of d, we were able to perform such computation only for

$d \leq 20$ (it took about four days to obtain ρ for $d = 20$ on an Ultra-Sparc workstation). To compute ρ for $d > 20$, we apply the test to 10,000-long segments of the *Star Wars* trace (the computational time is linearly proportional to both the size of the time series and the embedding dimension). Figure 4(b) depicts the results for two such segments (indicated by Series 1 and 2) for d from 1 to 60. Series 1 consists of data points 50,001 through 60,000, while Series 2 consists of data points 80,001 through 90,000 in the *Star Wars* trace. Despite the fluctuations in ρ, some chaotic trends can be observed. Consider Series 1, for example. As d goes from 1 to 8, ρ tends to increase. The trend is subtle when d goes from 8 to 18. However, for $d > 18$, there is a clear decline in the value of ρ as d increases. Similar remarks can be made about the behavior of ρ in Series 2. As mentioned earlier, one should not expect a pure chaotic behavior in real time series. Trends that are more indicative of chaoticity than of white noise should be sufficient.



(a) Complete *Star Wars* sequence



(b) 10,000-long segments of *Star Wars* sequence

Figure 4: Correlation coefficient between the transformed *Star Wars* sequence and a predicted version of it (using the nearest-neighbor approach), as a function of the embedding dimension.

2.3 Maximum Lyapunov Exponent

Lyapunov exponents are invariant measures of bounded dynamical systems (chaotic or non-chaotic). A dynamical system can have as many Lyapunov exponents as the true physical dimension of its attractor in the state space. Each exponent reflects the rate of expansion (positive exponent) or contraction (negative exponent) of the attractor in one of the dimensions. A rate of expansion is an indication of how fast on average two trajectories with slightly different initial conditions diverge from one another in a given direction (and vice versa for a rate of contraction). A dynamical system is chaotic if at least one of its Lyapunov exponents is positive.

In practice, the true dimension of the system is not known, and the spectrum of Lyapunov exponents can only be obtained in some embedding space. The number of obtained exponents in this case is given by the embedding dimension. If the embedding dimension is larger than the true (unknown)

dimension of the system, then some of the obtained Lyapunov exponents are *spurious*. Whether to identify these spurious exponents or not is still an issue of debate [15]. In any case, since a single positive Lyapunov exponent is sufficient to render a time series chaotic, one can focus on estimating the value of the largest Lyapunov exponent; if this value is positive, the time series is likely to be chaotic. A robust method for estimating the maximum Lyapunov exponent of a time series was proposed by Rosenstein et al. [25] and independently by Kantz [14], and was later implemented in the TISEAN package [13]. The idea behind this method is to directly test for any exponential divergence of nearby trajectories. More specifically, consider a time series $\{y_k : k = 1, 2, \ldots\}$. Let $\{Y(k) : k = d, d+1, \ldots\}$ be a d-dimensional embedding of this series; $Y_k = [y_k \; y_{k-1} \cdots y_{k-d+1}]$ for all k. Let Y_n and Y_{n^*} be two embedded state vectors with $||Y_n - Y_{n^*}|| = \Delta \ll 1$, where $||\cdot||$ is some distance metric defined on $\mathbb{R}^d$. One can view Y_n and Y_{n^*} as the initial points of two trajectories, and inspect the evolution of these trajectories in time. In particular, if $\Delta_m \stackrel{\triangle}{=} ||Y_{n+m} - Y_{n^*+m}|| \approx \Delta e^{\lambda m}$ for a reasonable range of m values, then λ is the maximum Lyapunov exponent. In theory, λ is an invariant measure, so its value should not vary with d, m, and n. However, due to the boundedness of the attractor, an exponential growth, if any, can be sustained only for a finite time period, after which a saturation effect takes place. Moreover, λ is a global quantity, so it represents the *average* behavior of the system. The attractor may locally expand (and even contract) at different rates. So some averaging over the data points is needed to ensure that λ is *robustly* estimated. In [25] and [14], the estimation of λ proceeds as follows. First, the time series is embedded in a state space with some dimension d. An embedded point Y_{n_0} is selected, and all the embedded points that are within ϵ distance from Y_{n_0} are found. These points, along with Y_{n_0}, represent a neighborhood of Y_{n_0} of radius ϵ, and are indicated by the set $\mathcal{N}_\epsilon(n_0)$. For a given time $n_0 + m$, $m > 0$, one computes the average distance between y_{n_0+m} and the set of data points resulting from advancing each embedded point in $\mathcal{N}_\epsilon(n_0)$ by m time units along the trajectory. This average distance, which is known as the *stretching factor*, gives an indication about the expansion in the size of the neighborhood $\mathcal{N}_\epsilon(n_0)$ after m time units. By taking its logarithm and averaging over several initial points Y_{n_0}, one can obtain an unbiased estimator of λ. Formally, one computes

$$S(\epsilon, m, n_0, d) \stackrel{\triangle}{=} \frac{1}{N} \sum_{n_0=1}^{N} \ln \left(\frac{1}{|\mathcal{N}_\epsilon(n_0)|} \sum_{Y_j \in \mathcal{N}_\epsilon(n_0)} | \, y_{n_0+m} - y_{j+m} \, | \right) \tag{5}$$

where N is the number of initial points. Then, $S(\epsilon, m, n_0, d)$ is plotted as a function of m for several values of ϵ and d. The slopes of these plots, which are expected to be relatively close in value, constitute the estimated maximum Lyapunov exponent.

The above estimation procedure was implemented in the routine `lyap_k` of the TISEAN package, which we applied to Series 1 and 2 of the *Star Wars* time series. Representative plots of $S(\epsilon, m, n_0, d)$ for various values of d, ϵ, and N are shown in Figure 5 (Parts (a) and (b) are for Series 1, while Parts (c) and (d) are for Series 2). Note that ϵ needs to be as small as possible to accurately capture the exponential divergence, if any; yet it has to be large enough to include a sufficient number of

neighbors. The larger the embedding dimension, the smaller is the number of points that can be found in a neighborhood, which impacts the goodness of the estimation (notice how the plots become less smooth as d increases).

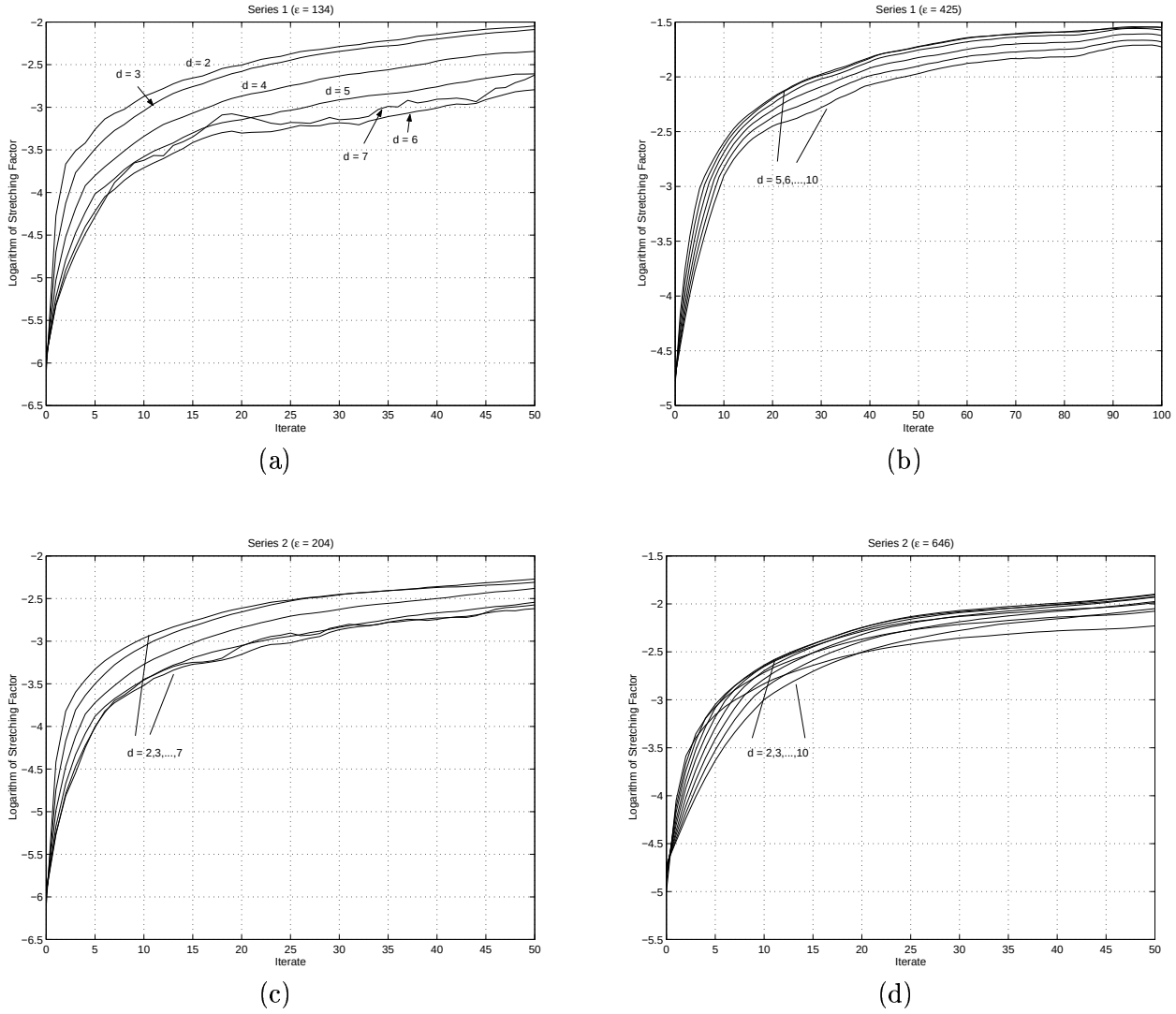


Figure 5: Logarithm of $S(\epsilon, m, n_0, d)$ versus m for Series 1 and 2 of the *Star Wars* trace.

All plots in Figure 5 have positive slopes, indicating a positive value for λ, i.e., presence of chaoticity. We use simple linear fitting of $\ln S(\epsilon, m, n_0, d)$ over the values of m (the iterates) from 15 to 45 to obtain estimates of λ. For Series 2 these estimates are given in Table 2 for four different values of ϵ. For $\epsilon = 115$ and $\epsilon = 204$, some embedding dimensions were ignored (these are indicated by dashes), as they produced neighborhoods with too few points. All the estimates of λ fall in the interval [0.01,0.023].

Embedding Dimension (d)	Estimated Maximum Lyapunov Exponent			
	$\epsilon = 115$	$\epsilon = 204$	$\epsilon = 363$	$\epsilon = 646$
2	0.0153	0.0136	0.0131	0.0126
3	0.0197	0.0165	0.0140	0.0130
4	0.0229	0.0190	0.0157	0.0131
5	0.0224	0.0200	0.0172	0.0139
6	—	0.0214	0.0188	0.0148
7	—	0.0188	0.0199	0.0155
8	—	—	0.0195	0.0167
9	—	—	0.0202	0.0181
10	—	—	0.0198	0.0197

Table 2: Estimated values of the maximum Lyapunov exponent in Series 2 (obtained through mean-square fitting over iterates from 15 to 45).

3 Estimating the Embedding Dimension

The dimension of a dynamic system is an indication of the number of degrees of freedom of that system. In other words, it is the number of initial conditions that are required to specify a solution. Several measures of dimensionality have been used in the literature (see [18] and the references therein), including the topological dimension, the fractal dimension, the information dimension, the correlation dimension, and the phase-space dimension. Unfortunately, it is difficult to *robustly* estimate these measures if the underlying structure of the time series is unknown, which is typically the case. From a practical standpoint, one is more interested in the *embedding dimension* (i.e., the dimension of the embedding space), since it is the one that mostly impacts the computational complexity of forecasting and controlling chaotic time series. Farmer [9] defines an embedding as a smooth diffeomorphism map $f : X \to Y$, where Y is a smooth submanifold of the smooth manifold X. The embedding dimension of an attractor is then defined to be the minimum dimension of a smooth manifold in which an attractor can be embedded. In this section, we first present an approach for estimating the embedding dimension of any time series, and then apply this approach to video traffic.

3.1 Basic Approach

To estimate the embedding dimension (d), we consider time series that are generated by nonlinear dynamic systems of the form in (2), which may or may not be chaotic. We then proceed in two steps. The first step focuses on finding the minimum number of consecutive delays D_u (i.e., length of the embedded vectors) that will ensure a functional relationship between the domain and range of the function F in (2). The resulting D_u is a conservative estimate of d since certain delays between the first and last delays may not be needed (e.g., x_n may be a function of only x_{n-1}, x_{n-2}, and x_{n-6}). In contrast to linear regression analysis where the insignificant delays are easily recognized by the negligible values of their coefficients (obtained directly from fitting), this is not the case in nonlinear systems. Therefore, to check if all the D_u delays are needed, we have to test the validity

of the functional relationship for all combinations of k (not necessarily successive) delays, starting with $k = D_u - 1$ and going down. Accordingly, the second step deals with searching for the optimal delay coordinates within a prescribed window, potentially reducing the estimate of d. In fact, using a smaller value of d than the estimated one would result in violating the functional relationship.

Step one of the above estimation procedure is executed as follows. First, the time series under examination is embedded into a 1-dimensional state space with points in the domain and corresponding points in the range. The relation between x_n and $V = [x_{n-1}]$ is then tested. If this relationship is functional, i.e., $x_n = f(x_{n-1})$ for some function f, then the time series can be embedded in a dimension-one state space. Otherwise, the embedding dimension is set to two, and the relationship between the embedded vectors in the 2D domain and the corresponding points in the range are tested. If the functional relationship assumption is satisfied, then the system generating the time series is of dimension two. Otherwise, the process is repeated with higher-order embeddings. Once the minimum number of consecutive delays (D_u) that satisfy a functional relationship is found, we start by testing all combinations of $k = D_u - 1$ delays within a predefined window.

To illustrate the first step, consider the sequence 1,2,3,1,3,2,3,1. In this case, the relationship between x_{n-1} and x_n is not functional since, for example, the value 1 in the domain corresponds to two different values in the range (2 and 3), as shown in Table 3(a). On the other hand, a functional

Domain	Range
1	2*
2	3
3	1
1	3*
3	2
2	3
3	1

(a)

Domain	Range
(1,2)	3
(2,3)	1
(3,1)	3
(1,3)	2
(3,2)	3
(2,3)	1

(b)

Table 3: Time Series $(1, 2, 3, 1, 3, 2, 3, 1)$ embedded in: (a) 1-dimensional state space, and (b) 2-dimensional state space.

relationship between x_n and $V = [x_{n-1}, x_{n-2}]$ can be found, as shown in Table 3(b).

After estimating D_u in step 1, we try in step 2 to refine this estimate by searching for the optimal delay coordinates of size D_l within a window of D_m delays, where $D_l < D_u$, $D_m = \alpha D_u$, and α depends on the size and information content of the empirical data. For example, if $D_u = 3$ then step 2 proceeds by searching for a 2-dimensional embedded vector $V = [x_{n-j}, x_{n-k}]$ that satisfies the functional relationship, where j and k are in the range $1, 2 \ldots, D_u$ with $j \neq k$ and with the case $(j = 1, k = 2)$ bypassed.

To quantify how far a mapping is from satisfying a functional relationship, we use the following

measure:

$$R(d) = \frac{P_{domain}(d) - P_{range}(d)}{N(d)}, \tag{6}$$

where $N(d)$ is the number of d-dimensional data points in the domain, $P_{domain}(d)$ is the number of repetitions of points in the domain (number of times the attractor passes through the same state in dimension d), and $P_{range}(d)$ is the number of repetitions in the range in dimension d that correspond to repetitions in the domain. For example, every repetition of the pattern (i, j, k) in the empirical series (beyond the first occurrence) will increment the value of $P_{domain}(3)$ by one. If any of these repetitions has the same image of a previous occurrence of (i, j, k), then the value of $P_{range}(3)$ will be incremented by one. Every violation of the functional relationship will increase the difference between $P_{domain}(d)$ and $P_{range}(d)$ by one. When $R(d) = 0$, d is the estimated embedding dimension ($d = D_u$).

So far, we have been requiring that the estimated d results in a strictly functional relationship between the inputs and the output of the model. This works fine in the absence of noisy data. To incorporate the effect of noise, we can relax the strict equality that is used to test the functional relationship in the empirical data. More specifically, let $V_1 = V_2$ be two identical state vectors in the domain with corresponding images p_1 and p_2. Then, we consider the functional relationship to be satisfied if $|p_1 - p_2| \leq \epsilon$ for some predefined $\epsilon \geq 0$. The larger the value of ϵ, the larger is the noise contribution in the model. To illustrate, consider the sequence $1.20, 1.90, 3.40, 1.20, 1.90, 3.39, 1.20, 1.50$, embedded in a two-dimensional state space. As seen in Table 4, for $\epsilon < 0.01$ this sequence cannot be represented as a function of two variables, since the functional relationship is violated for the embedded point $V = [1.20\ 1.90]$. If we take $\epsilon \geq 0.01$, this relationship will be satisfied.

Domain	Range
(1.20,1.90)	3.40*
(1.90,3.40)	1.20
(3.40,1.20)	1.90
(1.20,1.90)	3.39*
(1.90,3.39)	2.1
(3.39,2.1)	1.5

Table 4: Embedding the time series $(1.20, 1.90, 3.40, 1.20, 1.90, 3.39, 1.20, 1.50)$ in a 2-D state space.

We can go one step further and relax the strict equality on the embedded points in the domain (along with relaxing the inequality in the range), as follows. Let V_1 and V_2 be two embedded points in the domain, and let p_1 and p_2 be their corresponding images in the range. If $||V_1 - V_2|| \leq \epsilon_1$ *and* $|p_1 - p_2| \leq \epsilon_2$ for some ϵ_1 and ϵ_2, then we consider the functional relationship to be satisfied. The parameters ϵ_1 and ϵ_2 depend on the amount of noise in the data.

3.2 Application to Video Traffic

We now apply the previously presented estimation approach to video data. The results are summarized in Table 5. As these results indicate, a functional relationship cannot be found for $d \leq 6$. For $d = 7$, $R(d) = 0$. Thus, $d = 7$ is our starting estimate of the embedding dimension. In addition, the procedure of eliminating delays was applied, but we still found that seven consecutive delays are needed to produce a functional relationship. We verify the adequacy of our estimate by contrasting it with the estimates provided by the *false nearest neighbors* (FNN) method [17]. Figure 6 depicts the fraction of false nearest neighbors as a function of d for Series 1 and 2 of the *Star Wars* trace (we used the routine `false_nearest` of the TISEAN package). It is clear that as d increases from 1 to 7, the fraction of FNN is reduced significantly, indicating an unfolding of the attractor. For $d \geq 7$ this fraction barely changes, suggesting that $d = 7$ is the minimum acceptable embedding dimension.

d	$P_{domain}(d)$	$P_{rang}(d)$	$R(d)$
1	166941	6311	.96
2	132102	5020	.75644
3	29430	1311	.17
4	1762	84	.01
5	89	6	.000494
6	6	0	.0000357
7	0	0	0.0

Table 5: Estimating the embedding dimension in the *Star Wars* trace.

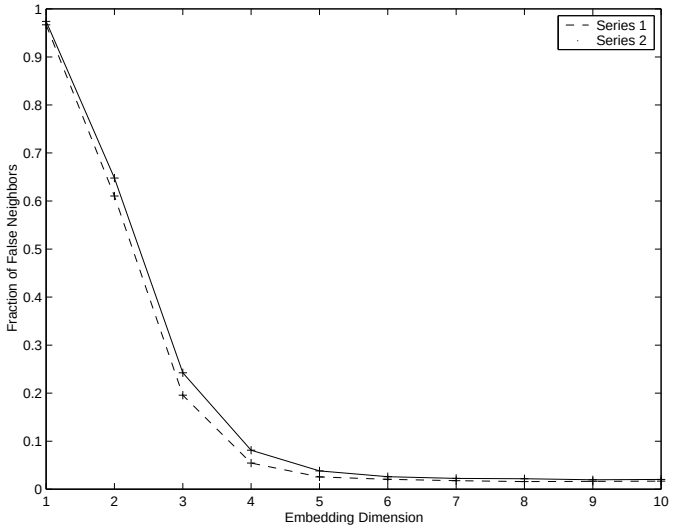


Figure 6: Fraction of false nearest neighbors versus the embedding dimension for the *Star Wars* trace.

4 Reconstructing the Attractor

Now that we established the presence of chaos in video traffic and estimated the embedding dimension, the next step is to reconstruct the attractor of the chaotic model and use it to generate synthetic video traces. Reconstructing the attractor amounts to finding the approximate form of F in (2), which can be done using local or global techniques [20]. We choose to work with local techniques since they require fewer data points at a time and are more adaptive to nonstationarities in the traffic. For a given d, we approximate F by the best composite fit $\hat{F}$, which consists of multiple "local" functions of simple forms (e.g., linear, quadratic). As suggested in [10], these local functions can be obtained by embedding the empirical time series in a d-dimensional state space, dividing this space into local regions, and then finding a good fit over each local region. The local regions in the domain of F are found by searching the embedded time series for the closest q neighbors to the current state vector (i.e., the one composed of the most recently predicted d values). The prediction accuracy and the final form of the attractor depend on the choice of q. If q is small, then the domain of the local region is too small to accurately describe the behavior of the function (i.e., the fitted hyperplane would not be stable). On the other hand, if q is large, then the domain of the local region is too large to get the benefit of localization. Subsequently, $\hat{F}$ would fail to capture discontinuities in F. One way to estimate q would be to search for the optimal q that minimizes the short-term prediction error within a given window, or to introduce a criteria that would force $\hat{F}$ to be continuous. We propose to vary the value of q in each prediction step. We speculate that this process would smooth edges in $\hat{F}$ and fill gaps in local regions with scarce data. In short, the method to predict the next value of the time series begins with finding a local region in the domain close to the current state vector using the variable parameter q. Then, a linear local approximate function for each region in the domain is constructed and used to predict the next point. Consequently, the next state is known and the process is iterated. Finally, we box the attractor to prevent it from escaping the predefined limits.

Next, we apply the above scheme to the *Star Wars* trace. In our initial attempts to reconstruct the attractor (figures not shown for brevity), we fixed the value of q for all local regions. But we found that in this case the reconstruction technique is extremely sensitive to the value of q. This is attributed to the fact that F contains edges and regions of discontinuities. So we tried a different approach in which the value of q varies deterministically in a periodic manner according to $q(n) = \hat{q} + a(n) mod(b)$. A fixed $\hat{q}$ is sufficient for the success of the model for any continuous map F. Varying q is effective when F contains gaps, jumps and discontinuities. In our case, this resulted in a better model than the one produced under a fixed q. The final model consists of a piece-wise linear segments, which (combined) dynamically approximate the function F. Since each of these segments is a hyperplane in the 7-D dimension, it is not possible to display them pictorially. To verify that the model is chaotic, we show in Figure 7 ten synthetic traces generated using slightly different initial conditions (which are given in Table 6). The sets of initial conditions that were used to generate these traces are identical

in six of the seven coordinates. The perturbation of these sets from the first one (Trace A) is given in the last column of the table. Clearly, the resulting trajectories differ significantly, indicating the presence of chaos in the synthetic data.

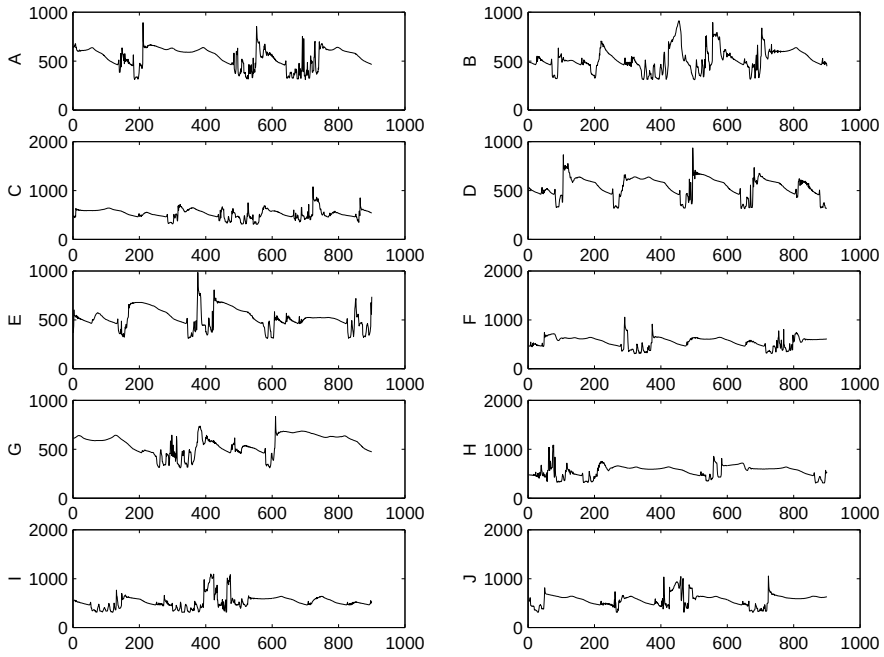


Figure 7: Ten synthetic traces with perturbed initial conditions.

Trace	Initial Condition	Perturb.
A	(496,496,498,499,505,497,499)	0
B	(496,496,498,499,505,497,495)	-4
C	(496,496,498,499,505,497,496)	-3
D	(496,496,498,499,505,497,497)	-2
E	(496,496,498,499,505,497,498)	-1
F	(496,496,498,499,505,497,500)	1
G	(496,496,498,499,505,497,501)	2
H	(496,496,498,499,505,497,502)	3
I	(496,496,498,499,505,497,503)	4
J	(496,496,498,499,505,497,504)	5

Table 6: Initial conditions for the synthetic traces in Figure 7.

5 Conclusions

In this paper, we provided statistical evidence that points to the presence of chaos in VBR video traffic. Our evidence was based on the sensitivity of the trajectories to initial conditions, the correlation coefficient between the transformed video sequence (after filtering out any apparent autocorrelations)

and a predicted version of it, and the estimated value of the maximum Lyapunov exponent. We presented a method for estimating the embedding dimension of a time series that is modeled according to the nonlinear representation of Farmer and Sidorowich [10]. When applied to the *Star Wars* sequence, this method yielded an embedding dimension of seven. By embedding the original *Star Wars* sequence in a 7D state space and using a nearest-neighbors-based technique (modified from [10]), we were able to produce synthetic video sequences that exhibit chaotic dynamics.

References

[1] A. Adas. Traffic models in broadband networks. *IEEE Communications*, 35(7):82–89, July 1997.

[2] A. M. Albano. Using higher-order correlations to define an embedding window. *Physica D*, 54:85–91, 1991.

[3] Z. Aleksic. Estimating the embedding dimension. *Physica D*, 52:362–366, 1991.

[4] V. Anantharam. Queueing systems with traffic models based on deterministic dynamical systems. In *Proceedings of the 35th Allerton Conference on Communication, Control and Computing*, pages 233–241, 1997.

[5] G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control.* Holden-Day, Inc., 1976. (Revised Edition).

[6] D. S. Broomhead and G. P. King. Extracting qualitative dynamics from experimental data. *Physica D*, 20:217–226, 1986.

[7] T. Buzug and G. Pfister. Optimal delay time and embedding for delay-time coordinates by analysis of the global static and total dynamical behavior of strange attractors. *Physical Review A*, 45(10):7073–7084, 1992.

[8] A. Erramilli and R. P. Singh. An application of deterministic chaotic maps to model packet traffic. *Queueing Systems*, 20:171–205, 1995.

[9] J. D. Farmer. Chaotic attractors of an infinite-dimensional system. *Physica (Amsterdam) 4D*, pages 366–346, 1982.

[10] J. D. Farmer and J. J. Sidorowich. Predicting chaotic time series. *Phys. Rev. Lett.*, 59(8):845–848, 1987.

[11] V. S. Frost and B. Melamed. Traffic modeling for telecommunications networks. *IEEE Communications Magazine*, 32(3):70–81, Mar. 1994.

[12] M. W. Garrett and W. Willinger. Analysis, modeling, and generation of self-similar VBR video traffic. In *Proc. of the SIGCOMM '94 Conference*, pages 269–280, Sept. 1994.

[13] R. Hegger, H. Kantz, and T. Schreiber. Practical implementation of nonlinear time series methods: The TISEAN package. *Chaos*, 9:413–440, 1999.

[14] H. Kantz. A robust method to estimate the maximal Lyapunov exponent of a time series. *Physical Letters A*, 185:77, 1994.

[15] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis.* Cambridge University Press, Cambridge, 1997.

[16] D. Kaplan and L. Glass. *Understanding Nonlinear Dynamics.* Springer, New York, 1995.

[17] M. B. Kennel, R. Brown, and H. D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45(6):3403–3411, 1992.

[18] D. Kugiumtzis, B. Lillekjendlie, and N. Christophersen. Chaotic time series. part I: Estimation of some invariant properties in state space. *Modeling, Identification and Control*, 15(4):205–224, Feb. 1994.

[19] W. Liebert, K. Pawelzik, and H. Schuster. Optimal embeddings of chaotic attractors from topological considerations. *Europhys. Lett.*, 14:521–526, 1991.

[20] B. Lillekjendlie, D. Kugiumtzis, and N. Christophersen. Chaotic time series. part II: System identification and prediction. *Modeling, Identification and Control*, 15(4):225–243, Feb. 1994.

[21] W. Marzocchi, F. Mulargia, and G. Gonzato. Detecting low-dimensional chaos in time series of finite length generated from discrete parameter processes. *Physica D*, 90:31–39, 1996.

[22] E. Ott. *Chaos in Dynamical Systems.* Cambridge University Press, Cambridge, 1993.

[23] E. Ott, C. Grebogi, , and J. A. York. Controlling chaos. *Phys. Review Letters*, 64:1196–1202, Feb. 1990.

[24] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Physical Review Letters*, 45(9):712–716, 1980.

[25] M. T. Rosenstein, J. J. Collins, and C. J. De Luca. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D*, 65:117, 1993.

[26] H.-G. Schuster. *Deterministic Chaos: An Introduction.* Physik Verlag, Weinheim, 1988.

[27] M. Tabor. *Chaos and Integrability in Nonlinear Dynamics: An Introduction.* Wiley Interscience, 1989.

[28] F. Takens. Detecting strange attractors in turbulence. In D. A. Rand and L. S. Young, editors, *Dynamical Systems and Turbulence*, pages 366–381. Springer-Verlag, Berlin, 1981.

[29] A. Veres and M. Boda. The chaotic nature of TCP congestion control. In *Proceedings of the IEEE INFOCOM 2000 Conference*, pages 1715–1723, 2000.