MULTI-STEP-AHEAD PREDICTION OF MPEG-CODED VIDEO SOURCE TRAFFIC USING EMPIRICAL MODELING TECHNIQUES

A Thesis

by

DEEPANKER GUPTA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Mechanical Engineering

MULTI-STEP-AHEAD PREDICTION OF MPEG-CODED VIDEO SOURCE TRAFFIC USING EMPIRICAL MODELING TECHNIQUES

A Thesis

by

DEEPANKER GUPTA

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Alexander G Parlos
(Chair of Committee)

Suhada Jayasuriya
(Member)

Aniruddha Datta
(Member)

Dennis O'Neal
(Head of Department)

December 2004

Major Subject: Mechanical Engineering

ABSTRACT

Multi-Step-Ahead Prediction of MPEG-Coded Video Source
Traffic Using Empirical Modeling Techniques. (December 2004)

Deepanker Gupta, B. Tech., Indian Institute of Technology, Kharagpur, India
Chair of Advisory Committee: Dr. Alexander G. Parlos

In the near future, multimedia will form the majority of Internet traffic and the most popular standard used to transport and view video is MPEG. The MPEG media content data is in the form of a time-series representing frame/VOP sizes. This time-series is extremely noisy and analysis shows that it has very long-range time dependency making it even harder to predict than any typical time-series. This work is an effort to develop multi-step-ahead predictors for the moving averages of frame/VOP sizes in MPEG-coded video streams.

In this work, both linear and non-linear system identification tools are used to solve the prediction problem, and their performance is compared. Linear modeling is done using Auto-Regressive Exogenous (ARX) models and for non linear modeling, Artificial Neural Networks (ANN) are employed. The different ANN architectures used in this work are Feed-forward Multi-Layer Perceptron (FMLP) and Recurrent Multi-Layer Perceptron (RMLP).

Recent researches by Adas (October 1998), Yoo (March 2002) and Bhattacharya et al. (August 2003) have shown that the multi-step-ahead prediction of individual frames is very inaccurate. Therefore, for this work, we predict the moving average of the frame/VOP sizes instead of individual frame/VOPs. Several multi-step-ahead predictors are developed using the aforementioned linear and non-linear tools for two/four/six/ten-step-ahead predictions of the moving average of the frame/VOP size time-series of MPEG coded video source traffic.

The capability to predict future frame/VOP sizes and hence the bit rates will enable more effective bandwidth allocation mechanism, assisting in the development of advanced source control schemes needed to control multimedia traffic over wide area networks, such as the Internet.

To *my parents*

ACKNOWLEDGMENTS

It is with deep sense of gratitude that I express my sincere thanks to Dr. Alexander G. Parlos for his valuable guidance, kind cooperation and timely suggestions during my stay at Texas A&M University. I would like to thank my parents Ravinder and Kiran, and sister Sonali for always being there for me. Thanks are also due to my friends Aninda, Dan, Lin, Tolis, Vivek and Ram for helping me understand things and encouraging me to do better throughout my MS degree. Last, but not the least, I am grateful to the faculty members of Mechanical Engineering Department at Texas A&M for introducing me to new depths of knowledge.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

In the present age the Internet has emerged as the primary source for information exchange around the world. It has come a long way since the time of its introduction as a network connecting different sections of the United States Department of Defense. Today, nearly all organizations, irrespective of whether they are academic, business or any other, use the Internet to send and receive information. Along with that, the Internet has also become a major source of entertainment. It is because of this fact that in the near future the audio-visual media will form the majority of the traffic transferred over the internet.

A video is a set of images shown at a certain specified speed. The encoder encodes these images into a bit stream. The server transmits these bits over the Internet to the client. The client's side decoder decodes the bits and regenerates the images. The images are then displayed to the user at a predetermined rate [1]. The encoding of audio is done in a similar fashion.

A. Motivation

The motivation of this research comes from the problem of dynamic bandwidth allocation of encoded real-time multimedia streams over wide area networks. The importance of the problem of dynamic allocation of network resources has attracted a large number of researchers. The need of dynamic bandwidth allocation arises because not all users of the network require the same bandwidth. If the available bandwidth is allocated equally among the end users, irrespective of the utilization statistics, much

The journal model is *IEEE Transactions on Automatic Control.*

of the bandwidth goes unused and hence gets wasted. Since all network users do not send and receive data over the network at the same time, it does not make sense to provide then with equal resources at a given time. Thus a regulated dynamic bandwidth allocation scheme will increase the efficiency of network resource utilization. This scheme will also ease network congestion. In order to regulate the bandwidth allocation among network end users dynamically, it is necessary to predict the traffic generated by end users. Since multimedia traffic will form a majority of the total network traffic, a scheme for the prediction of traffic generated by source of multimedia streams will significantly aid in design of an efficient dynamic bandwidth allocation.

An important fact to keep in mind is that most of the multimedia applications over the network are time sensitive. When a real-time multimedia application sends time dependent information over the network, if some of the packets are delayed or get lost in the network, it could lead to a significant deterioration in the quality of the video as seen by the end user at the receiver side. This decline in quality might not be acceptable for the application. The quality of the multimedia stream as perceived by the end user is very sensitive to network congestion. One popular method to improve the quality of real time multimedia applications over the Internet is to have a receiver side buffer. The decoder at the receiver side does not start the playback until the receiver buffer is filled up to a certain specified level. This scheme tries to provide uninterrupted multimedia packets to the decoder in the correct sequence. It also tries to maintain the quality of the video. But this scheme fails to provide acceptable quality for a live broadcast over the Internet.

Lately, a lot of work is being done to develop a controller on the source side that can control the flow of multimedia packets over the Internet. This controller along with the existing schemes at the receiver side will try to provide better video quality to the end user. One of the important inputs to this controller will be the prediction of the traffic generated by multimedia sources. This predicted traffic combined with other inputs will assist the controller to vary the send-rate and packet sizes of the multimedia stream over the networks. Figure 1 shows the block diagram of the scheme. The scope of this work is limited to the design of the predictor.

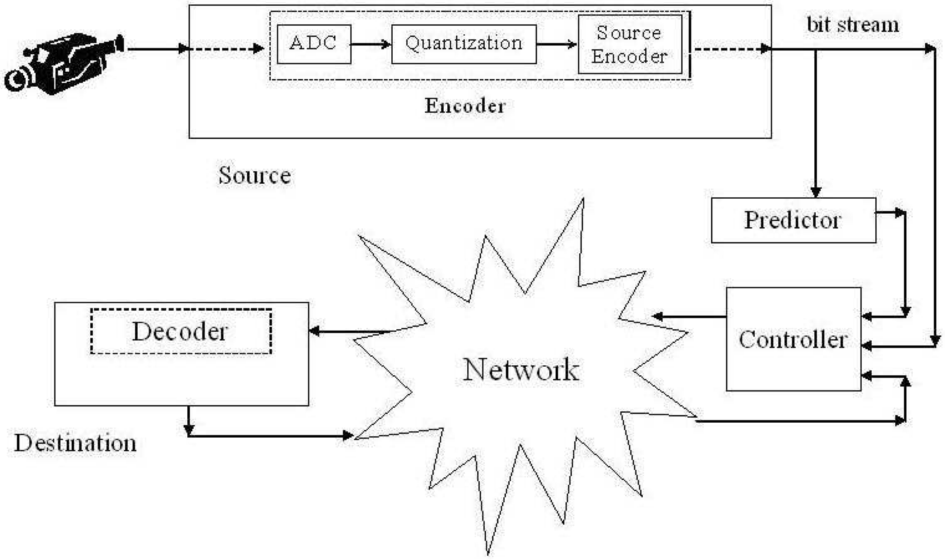

Fig. 1. Block diagram of the predictive control scheme.

B. MPEG Standards

The most widely used video format is the MPEG, which stands for Motion Pictures Experts Group. MPEG has its own standards which dictate the encoding and decoding of audio-visual/multimedia data. Many video standards have been released since the time MPEG was first introduced. The first of these standards MPEG-1, was a protocol to encode/decode the media which could be transferred over the T1 lines and could be played on a CD–ROM. This standard was designed for a bit rate of about 1.5 Mbits/sec. MPEG dictates the rules which are used to encode the media files. These rules define the syntax in which the media bit streams must be encoded in order to be compliant with the decoder [2]. The encoder for MPEG-1 is much more complex than the decoder. Therefore it can be used with a large number of decoders. Next MPEG-2 was developed which unlike MPEG-1 had tools for interlace, scalable syntax and a range of profiles to include a wide range of applications.

MPEG-4 is the next in the series of MPEG standards. MPEG-1 and MPEG-2 mainly address compression of audio and video, focusing much less on communication and transport issues. MPEG-4, in addition to improved audio and video compression, also enables interactive presentations. MPEG still continues to upgrade its standards with the newer versions being MPEG-7 which addresses content description and indexing, and MPEG-21 which focuses on content management, protection and transactions [3]. The most widely used format at present is the MPEG-4, which is the reason for focusing the present research on this standard.

C. Literature Review

A significant amount of literature has been devoted for the analysis of video source traffic in the past. Most of the work is related to the derivation of the statistical source models. These statistical models are primarily autoregressive models. They are tested by matching model parameters against the real data statistics. The paper by Bae and Suda [4] is a good survey of such video models.

Frost and Melamed [5] review a few commonly used traffic models for the purpose of simulation. The traffic models which they discuss are generalized in nature and in some cases fail to represent the dynamics of multimedia streams over high-speed transport networks like ATM. A survey of models based on Poisson and Bernoulli processes along with fluid, autoregressive, and self-similar traffic models is also presented in [5].

Krunz and Hughes [6] describe the statistical behavior of the VBR streams using histograms and autocorrelation functions. Their algorithm determines the instants of a scene change based on the changes in the size of successive $I-$ frames. The length of the scene is modeled by a geometric distribution. Lognormal distributions with varying μ and σ were used to model the size of $I-$, $P-$, and $B-$frames separately in a video stream coded using MPEG. The parameters of the proposed models were obtained by matching some statistical characteristics of an actual video sequence and the model under consideration.

Heyman, and Lakshman [7] propose a model for video conferences, based on three parameters – mean, variance of the bit rate and the correlation of the number of bits in adjacent frames. They follow a modeling strategy that first identifies scene changes and then constructs different models for changing lengths of the scenes and number of cells in a scene change frame. Finally, models for the number of cells per frame for frames within scenes are developed. The decision to identify scene changes are taken after it is found that scene changes precipitate spikes in the bit rate and these spikes are the dominant cause of cell losses. Scaled second order difference of the frame sizes is used to identify the scene changes. Gamma, Weibull and Generalized Pareto distributions are selected as candidates for describing scene lengths and scene-change frames.

Wang, Jung, and Meditch [8] propose a VBR broadcast video traffic modeling technique based on wavelet decomposition approach. Their method decomposes video traffic into two parts via wavelet transformation, and models each part separately. In the first part, they attempt to capture the long-term trend of the traffic. This part is modeled by an *Auto-Regressive* (AR) process. In the second part, classified by vector quantization, they address the short-term behavior of the video traffic. This part preserves the periodic coding structure in traffic data and provides an unified approach for frame and slice level traffic modeling.

Rožić and Vojnović [9] present an extensive statistical analysis of a 2 hour long MPEG empirical trace. The model proposed in this paper is based on a statistical decomposition procedure. Each of the components are modeled by an *Autoregressive Integrated Moving Average* (ARIMA) process. This model also incorporates the effect of scene changes. Due to strong evidence of seasonal and trend components, the video stream is decomposed into three components: seasonal, trend, and stochastic. The seasonal component is included into overall model as a deterministic process with a period of length 12.

Krunz and Tripathi [10] present a comprehensive model for variable-bit-rate video streams. The model proposed by them captures long-term variations by incorporating scene changes, detected by the fluctuations of $I-$frames. The size of $I-$frames is modeled as the sum of two random components: a scene-related component and an AR component that accounts for the fluctuations in the scene. Two other random processes are used to model the sizes of $P-$ and $B-$frames. Liu, Sara, and Sun [11] extend the work of Krunz and Tripathi to come up with a model that captures the autocorrelation structure between the sizes of frames by using two second order *Auto-Regressive* (AR) processes nested within each other.

To estimate a model of video traffic in a more efficient manner, Krunz and Makowski [12] suggest a compromise between Markovian and *Long Range Dependent* (LRD) models. They argue that the autocorrelation function $\rho(k)(k = 0, 1, \ldots)$ of a compressed video sequence is better captured by $\rho(k) = e^{-\beta\sqrt{k}}$ than by $\rho(k) = k^{-\beta} = e^{-\beta logk}$ (long range dependence) or $\rho(k) = e^{-\beta k}$ (Markovian). The authors introduce a video model with the above mentioned correlation structure. This model is based on M/G/∞ input process. By varying G, many forms of time dependence can be displayed. For video traffic, appropriate G is derived according the correlation function $e^{-\beta\sqrt{k}}$. This model exhibits short-range dependence.

Feng and Lo [13] present their work on a hierarchical model for MPEG video. This model represents the bit-rate variations of MPEG video at two different levels: scene level and frame level. It is believed that scene lengths in MPEG video streams are uncorrelated. The authors advocate the use of Gamma distribution to represent the scene length. They concur with other researchers about the distribution of the scene lengths in MPEG video streams. They endorse the fact that the scene length is dependent on the motion content of the video scene. In this paper, the authors present three autoregressive models, one each for $I-$, $P-$,and $B-$frames.

Chiruvolu et al. [14] derive a modulating Markov chain model. Each state in the model represents the $I-$, $B-$, $P-$frames of a *Group of Pictures* (GOP). Based on the average number of bits generated during the scenes, they can be classified into high- and low-activity scenes. An auxiliary Markov chain models the scene activity. Transitions of the auxiliary Markov chain represent scene changes of a video sequence. The bit generation during low-activity scene is modeled by independent AR processes each for $I-$, $P-$, $B-$frames.

Liu, Ansari, and Shi [15] propose Markov modulated self-similar processes to model MPEG video sequences. These processes are able to capture *Long Range Dependency* (LRD) characteristics of video *Auto-Correlation Function* (ACF). Like previous research papers dealing with the same modeling processes, they decompose an MPEG compressed video sequence into three parts according to different motion/change complexity and model them individually by a self-similar process.

Manzoni, Cremonesi, and Serazzi [16] model the $I-$frame sequence only. They argue that $I-$frame sequence captures the dynamic characteristics of video traffic at higher granularity and the decisions based on $I-$frame size values are also valid for $P-$

and $B-$frames. They characterize a *Frame Size Sequence* (FSS) by a tuple of eight variables. They use multivariate analysis to identify the parameters corresponding to the eight variables. This technique is used for static characterization of VBR streams. Dynamic characterization is also performed on the video streams. The research paper also describes some of the forecasting models developed after performing the required analysis.

A survey of traffic characteristics of multiplexed VBR MPEG-1 encoded video sources transmitted over ATM B-ISDN networks is done by Doulamis et al. [17]. After analyzing the results obtained from the survey, they propose some traffic models. Their work concentrates on three layers: the frame layer, GOP layer, and Intermediate layer. In the first layer, the aggregate MPEG-1 stream, which constitutes the frame-layer signal is examined, and a correlated AR model of high order is introduced to estimate the network resources. Reduction of the required parameter is achieved by analyzing the MPEG-1 video sources at a higher layer (GOP layer). In order to maintain accurate approximation of the MPEG-1 traffic behavior and simultaneously reduce the number of required parameters, an intermediate layer, which efficiently combines properties of other layers is introduced.

Chandra and Reibman [18] explain how a traffic model can incorporate features and changes in the encoding process and correspondingly influence the network performance while simultaneously matching the statistical descriptors of the measured data. Markov chain model discussed in [18] is accurate enough to model video of varying activity levels. The model derived in this paper concentrates on $I-$ and $P-$frames only. From the state space representation of video frame rates, frames can be divided into three regions: $PP-$frames that are self-similar in nature, $PI-$frames that manage the periodic transition from $P-$frames to the next $I-$frame, and $IP-$frames that manage the transition from $I-$frame to the subsequent $P-$frames. Frames in the PP region of the state-space are segmented into K states for better characterization of *Short Range Dependency* (SRD) among them. This segmentation is carried out using K-means cluster detection algorithm [19]. A Markov chain with K states is used to model the transitions between the states. A first order autoregressive model is used to represent the observed correlation structure. The single layer model is adapted in a suitable manner to model a two-layer video source.

He et al. [20] study the statistical model of MPEG video over wireless networks. A real-time MPEG video traffic workload was generated and transmitted over a wireless network. The model parameters of $I-$, $P-$ and $B-$frames were measured before and after wireless transmission. The parameters associated with the model remained virtually unchanged irrespective of the kind of errors that were introduced within the network.

Adas [21] proposes the use of an adaptive linear predictor for the GOP size prediction. The paper claims that the order of the adaptive linear predictor is small (12 or less) and that it does not increase with the size of variable bit rate (VBR) video traffic trace. A study of the performance of dynamically allocating bandwidth based on linear prediction using Renegotiated Constant Bit Rate (RCBR) service model is presented. The author advocates the use of normalized least mean square (NLMS) algorithm instead of least mean square (LMS) for adaptation. A $k-$step–ahead linear predictor is given by

$$\widehat{x}(n+k) = \sum_{l=0}^{p-1} \omega(l) x(n-l) \tag{1.1}$$

where $\omega(l)$, for $l = 0, 1, ..., p-1$ are prediction filter coefficients. Adas proposes the following NLMS update equation:

$$\omega(n+1) = \omega(n) + \frac{\mu e(n) x(n)}{||x(n)||^2} \tag{1.2}$$

The use of NLMS makes the adaptation less sensitive to the step size μ. These modifications are used to develop a scheme to predict the next GOP size.

Yoo [22] develops an adaptive traffic prediction scheme for VBR MPEG videos that includes an analysis of the effects of scene changes and traffic variations on the prediction errors. The research paper only provides a methodology to predict the sizes of $I-$ and $P-$frames and ignores the prediction of $B-$frames. The author contends that a dynamic resource allocation method for networks is required. A dynamic approach uses a real-time measurement to adaptively determine the bandwidth share for each user. With dynamic resource allocation, a high network utilization can be achieved without any prior knowledge of the video traffic. Yoo selects an adaptive time domain prediction using the LMS algorithm. This scheme does not require any prior knowledge of the video statistics. It also does not assume the contents to be stationary. A pth-order, *1*-step linear predictor is described by equation 1.3. It uses the linear combination of the current and previous values of $X(n)$. W_n is the time varying coefficient vector that minimizes the mean square error.

$$\widehat{x}(n+1) = \sum_{l=0}^{p} \omega_n(l)x(n-l) = W_n^T X(n) \tag{1.3}$$

The author also proposes algorithms for the prediction of the $I-$ and $P-$frames in his work. The research paper also contains a prediction scheme for multiplexed video streams. Multiplexed video streams have completely different statistical properties in comparison to single stream.

Chodorek and Chodorek [23] develop a linear predictor of the MPEG video traffic based on partitioning of the phase space into sub-regions. Phase-space trajectory describes the temporal evolution of the state of the analyzed system. They can be treated as geometrical illustration of the dynamic information contained in the measured signal. For many systems, points plotted in the phase space are collected in clusters. The geometrical properties of the clusters provide information about the global state of the system. The central task of the phase space analysis of the MPEG video traffic is estimation of embedding parameters. These parameters define the trajectories of the series in an m-dimensional phase space. In the research paper [24], Chodorek proposes the use of six different linear regressive models for predicting the frame size time-series. These models are implemented one at a time depending on the type of the frame(I, B or P).

While a majority of the literature on modeling of MPEG video source traffic advocate the use stochastic models based on autoregressive modeling techniques, researchers have also used non-linear modeling tools such as neural networks to model MPEG video source traffic. Though the simplicity of autoregressive models make them the obvious choice, the thrust behind the use of neural networks is their capability to model inherent non-linear dynamics of the complex systems. Tarraf et al. [25] propose an approach to characterize and model multimedia traffic in ATM networks using Neural Networks (NNs). They make use of a multilayer perceptron neural network to predict the statistical variations of i-th packet arrival process resulting from the superposition of N packetized video and M packetized audio sources. They show that NNs can be successfully utilized to characterize various aspects of multimedia traffic prediction.

Moh et al. [26] adopt a neural network methodology to predict VBR traffic represented by a continuous autoregressive (AR) Markov model. They suggest a $1-5-1$ perceptron neural network model for prediction of VBR traffic. Based on the predictions obtained from the neural network, they also propose a dynamic bandwidth allocation scheme for ATM networks.

Chang and Hu [27] investigate the application of a *Pipelined Recurrent Neural Network* (PRNN) to the adaptive traffic prediction of MPEG video signal over dynamic ATM networks. A general *Nonlinear Autoregressive Moving Average* (NARMA) process characterizes the traffic signal of each picture type (I, P, and B) of MPEG video. A mean-squared error predictor based on NARMA model is developed to provide the best prediction for the video traffic signal.

Two research papers presented by Doulamis, Doulamis, and Kollias [28] and [29] investigate neural networks for non-linear online traffic prediction of VBR MPEG coded video sources. In these papers [28] and [29], Doulamis et al. develop three *Non-linear Autoregressive Models* (NAR) to model the aggregate MPEG-2 video sequence, each of which corresponds to one of the three types of frames ($I-$, $P-$, and $B-$frames). The optimal mean-squared error predictor of the NAR model is implemented using a feed-forward neural network with a *Tapped Delay Line* (TDL) filter. A new algorithm handles the significant effect of correlation among $I-$, $P-$ and $B-$frames on the estimation of network resources. A new mechanism is proposed to improve the modeling accuracy based on a generalized regression neural network.

Bhattacharya et al. [30] proposes a novel approach to model $I-$, $B-$ and $P-$frames separately using neural networks. In the past researchers have developed predictors for the frame sizes. In this paper, the authors also develop predictors for the moving average or mean bit rate. In their research paper, Bhattacharya et al. [30] go a step further and develop multi-step-ahead predictors for the mean bit rate. They develop separate predictor for single, two and four-step-ahead predictions using feed-forward multilayer perceptron (FMLP) and recurrent multilayer preceptron (RMLP) neural networks. An important concept that they introduce in this paper is the use of indicators as external inputs. Instead of predicting the mean bit rate as a time-series, they use indicators, for example the size of the $I-$frame, which are derived from the time-series as external inputs. The authors advocate that the use of these indicators improves the accuracy of the predictors. A $11-22-1$ structure

FMLP model is suggested for single-step-ahead prediction of the mean bit rate. For two-step-ahead prediction, the authors suggest an RMLP neural network model with $11-22-1$ structure. For four-step-ahead prediction, the authors present a $14-22-1$ structure RMLP model. These predictors are tested on a number of MPEG-4 video traces and are shown to perform consistently for varying video qualities.

The research papers which are most relevant to this research are the ones by Adas [21], Yoo [22], Doulamis, Doulamis, and Kollias [28] and [29], Chang and Hu [27] and Bhattacharya et al. [30]. Out of these papers, the first two use linear techniques to model MPEG-coded video source traffic. The others use neural networks to design the predictors. All these papers, with the exception of [30], propose single-step-ahead prediction (SSP) schemes. In the papers [28] and [29], the authors do not show the prediction errors while performing SSP of different traces. Chang and Hu [27] do not show results of models being tested on a wide array of video data traces. Adas also shows the prediction errors for only a few video traces. Yoo presents a comprehensive prediction of $I-$ and $P-$frames of a number of video traces. These traces can be obtained from a public archive [31]. A detailed information about the prediction errors is presented by Yoo. The paper by Bhattacharya et al. [30] is by far the most suited reference for this research. In this paper, the authors present SSP for $I-$, $B-$ and $P-$frames separately. They also propose models for prediction of mean bit rates of MPEG-coded video data traces for up to four-steps-ahead. The MPEG-coded video traces used in this work can be found at http://www-tkn.ee.tu-berlin.de/research/trace/trace.html.

The prediction of MPEG-coded video source traffic, as seen from the literature, has not been very good. This is because the time-series to be predicted has a long-range time dependency and is extremely noisy and this leads to high prediction errors. Additionally, most of the models proposed by the researchers involve designing of a simulator which generates synthetic MPEG data that statistically matches the real data. These models can not be implemented online to predict this traffic. In the past, empirical models have been used to model complex dynamic systems. In this research, both linear and non-linear techniques of empirical modeling are used to model the mean bit rate. A comparison of these models based on certain standard performance metrics is also presented.

D. Research Objectives and Proposed Approach

1. Research Objectives

The objective of this work is to design several multi-step predictors which can be used to predict the moving average of frame/VOP sizes of an MPEG-coded video trace for a specified time horizon, using different methods and compare the results. In order to predict the source traffic of any encoded movie irrespective of the quantization parameters, the designed predictor needs to be robust and capture the dynamics of the MPEG encoding process as much as possible. If predicted accurately, this information can be used by a source controller for dynamic bandwidth allocation and congestion control. This controller would regulate the rate at which source traffic is entering the network. The objective of this controller will be to maintain the QoS, irrespective of the delay variations over the network. This will be done in conjunction with admission control. For a connection on the Internet, the round trip time (RTT) between the source and the destination can be up to a few seconds. Thus it may take some time for the controller to receive network information in order to implement the control. In order to compensate for the dead-time of the controller, one must predict the source traffic a few seconds ahead in time. Thus the need for multi-step-ahead prediction. This work focuses only on the prediction of the MPEG-coded video source. The design of the relevant source controller is not within the scope of this research.

2. Proposed Approach

The MPEG-coded video stream data used for the present research is encoded using the MPEG-4 standard. The encoded data traces have fixed-size GOVs with deterministic, regular GOV patterns. The temporal dependency of the frame/VOP sizes of the in an encoded MPEG stream cannot be modeled effectively using statistical techniques because they fail to predict the source traffic in real-time.

For multi-step-ahead prediction using linear models, the ARX models give best results. Hence ARX models are used extensively for linear modeling in this work. For multi-step-ahead prediction using non-linear models, FMLP, RMLP and ESN models are employed. In case of FMLP and RMLP, the algorithm developed by Parlos, Rais and Atiya [32] is used. This algorithm has been proven to be very robust for multi-step-ahead prediction by Bhattacharya et al. [30]. For ESN, the algorithm developed by Jaeger [33] is used. This algorithm has been successfully used for other time-series predictions.

E. Contribution

The contributions of the current research work are:

1. Development of multi-step-ahead prediction schemes for predicting MPEG-coded video source traffic using empirical modeling techniques for up to ten-steps-ahead.

2. Performance comparison of different linear and non-linear empirical modeling techniques developed to predict MPEG-coded video source traffic.

F. Organization of the Thesis

This thesis is divided into six chapters. In Chapter II, a brief introduction to MPEG-4 video coding standards is presented. The concept of $I-, B-$ and $P-$VOPs is also explained in this chapter. Chapter III introduces the linear and non-linear empirical modeling techniques used in this research. The training algorithms for the non-linear modeling neural networks are also presented. In Chapter IV, a detailed analysis of the VOP size time-series is presented. This chapter also explains the need to predict the moving average of the VOP sizes and presents the mathematical formula used to calculate the moving average time-series. In Chapter V, the different prediction models developed in this research are presented. This chapter also presents the results obtained in the prediction of I-VOPs. The results are tabulated in terms of the performance metrics which are introduced at the beginning of this chapter. In Chapter VI, the models developed in this research and the ones published in literature are compared and conclusions are made based on this comparison. Finally some recommendations for future work are made.

CHAPTER II

OVERVIEW OF MPEG-4

MPEG stands for Moving Pictures Experts Group. This group was formed in 1988 under the Joint Committee of ISO (International Standards Organization) and IEC (International Electrotechnique Commission). Of all the existing MPEG standards, MPEG-4 is the most widely used standard. This chapter provides a brief overview of MPEG-4. The description of MPEG-4 which follows is extracted from [1] and [34].

MPEG coding utilizes both the spatial and the temporal redundancy of the video stream. The concept of video objects and their temporal instances, *Video Object Planes* (VOPs), is central to MPEG-4 video. The main objective for the formulation of MPEG-4 was encoding video and audio at very low rates. Another objective was to increase error resilience to packet losses. MPEG-4 architecture has made it possible for the generation of many new types of applications which were not possible in the earlier versions of MPEG. MPEG-4 was optimized for three bit rate ranges -

1. Below 64 kbits/s
2. 64 kbits/s to 384 kbits/s
3. 384 kbits/s to 4 Mbits/s

Introduction of objects is one of the significant contribution of this standard. Different parts of the final scene can be coded and transmitted separately as *video objects* and *audio objects*, to be brought together by the decoder. Separation of objects allows interaction with the objects. This feature is very useful in games and educational software. The three key characteristics of MPEG-4 streams are:

1. A picture can be separated into different objects. These objects can be encoded using different techniques. These object can again be joined to form the

composite picture by the decoder.

2. The objects of a picture can be either an integral part of the picture or they might be synthetic like computer generated shapes and captions.
3. Instructions encoded in the bit stream enables several different kinds of presentations of the same bit stream.

A. MPEG-4 Video Hierarchy

The first level of hierarchy in MPEG-4 is based on the objects in the image. Each of the object in the image is a *Visual Object Plane* (VOP). Therefore, each object in the scene is represented by a series of VOPs in time. This is not true for the static objects in the video images. The static objects can be represented by a single VOP. A VOP contains texture and shape data associated with the respective object. VOPs are analogous to frames in the earlier versions of MPEG standards and can be coded using I-frames or motion compensation techniques.

Next level of hierarchy in the MPEG-4 standard is *Group of Video Object Planes* (GOV). They provide points in the bit stream where the VOPs are coded independently from each other. *Video Object Layer* (VOL) is used for providing the facility of scalable coding of a sequence of VOPs or even GOVs. Scaling might be spatial or temporal. Multiple VOLs correspond to multiple scaling of a sequence. The decision on the various aspects of scaling are decided by the availability of resources like bandwidth and computational power. *Video Object* includes all aspects of an object within its definition. That means it includes all the VOLs associated with an object. *Video Session* (VS) is the top video level of MPEG-4 standard. It comprises of all video objects irrespective of their nature of origin in the scene. As in the earlier versions of the MPEG standard, encoder syntax must support many coding possibilities. The blocks of the images can be coded as either $I-$, $P-$ or $B-$VOP.

B. VOP Types in MPEG-4

The VOPs are encoded in three different ways.

1. Intra-coded ($I-$VOPs): An *intra-coded VOP* or *$I-$VOP* is encoded using only information from within that VOP. These VOPs are independent of all other VOPs preceding or succeeding them. The texture values in each macroblock are Discrete Cosine Transform (DCT) coded. The DCT coefficients are then quantized and variable-length-coded.

2. Non-intra VOPs ($P-$VOPs and $B-$VOPs): Non-intra VOPs use information from outside the VOP, i.e., from the VOPs that have already been encoded. In non-intra VOPs, motion-compensated information is used for a macroblock. This results in less data than directly coding the macroblock. There are two types of non-intra VOPs - *predicted VOPs ($P-$VOPs)* and *bidirectional VOPs ($B-$VOPs).* The first step is to encode the first VOP comprising the clip as an $I-$VOP without borrowing any information from any other VOP. This VOP is sent to the respective decoder. The decoder decodes the VOP and stores the reconstructed object in the memory. A copy of the encoded VOP is decoded by the encoder itself and preserved in the memory of the encoder. Technically the encoded VOP and the actual VOP should be the same but they are not as the compression involved during encoding is lossy in nature. This $I-$VOP is going to be used as a reference for getting $B-$ and $P-$VOPs. To encode a $P-$VOP, each macroblock in the $P-$VOP will search for a matching macroblock in the $I-$VOP decoded by the encoder. In forward predicted VOPs, each macroblock is predicted from the closest match in the precedent $I-$ (or $P-$) VOP using

motion vectors. Residuals of the macroblocks must be considered if they are not identical. Thus, $P-$VOPs are predicted from the preceding $I-$VOPs and $P-$VOPs. $B-$VOPs are encoded by using the information from $I-$VOPs as well as $P-$VOPs. This is known as *bidirectional encoding*. In $B-$VOPs each macroblock is predicted from the preceding $I-$ (or $P-$) VOP and the succeeding $P-$ (or $I-$) VOP. The prediction errors are DCT coded, quantized and variable-length-coded.

C. Scalability in MPEG-4

MPEG-4 provides both spatial and temporal scalability at the object level. In both cases this technique is used to generate a base layer, representing the lowest quality to be supported by the bit stream, and one or more enhancement layers. A single coding operation may produce all the layers of enhancement. The decision to scale and send information can be implemented in two ways. When the application in question knows the bandwidth limitations, versions of the bit stream may be sent across the network that include only the base layer along with some lower order enhancement layers. The other way is to send all the layers and let the decoder decide which layer it wants to accept for display. This decision will depend on the resolution of the display device or the computational resources of the client machine.

Figure 2 depicts an encoder that implements spatial scalability. The input VOP is down-converted to a lower resolution that forms the base layer. This layer is encoded. A decoder constructs the base later VOP as it will appear at the decoder. This decoded VOP is up-converted to the same resolution as input. A subtraction process yields the differences from the original image. These residuals are separately encoded in an enhancement layer encoder. Each stream of encoded VOPs form a video object layer. The base layer VOL uses both intra- and intercoding. However, the enhancement layer uses only predictive coding. The base layer VOPs are used as references. Temporal scalability is far easier to understand when compared to spatial scalability. The stream of incoming VOPs is split. The required number of VOPs is sent to the base layer encoder and the remainder are sent to one or more enhancement layers.

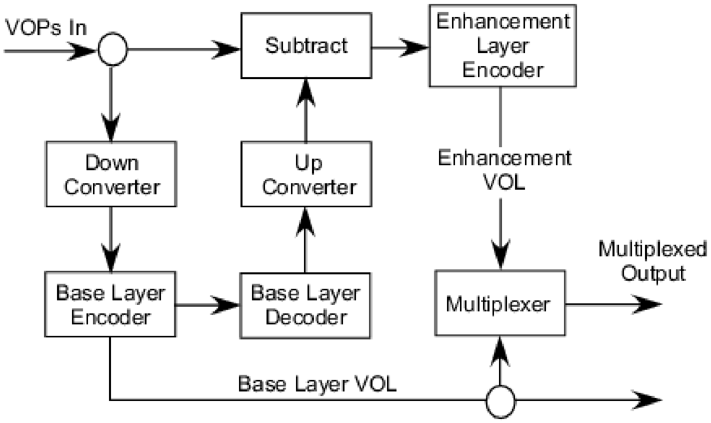

Fig. 2. Spatially scalable encoder for a single enhancement layer [Ref. 2, p. 208].

D. Visual Profiles of MPEG-4

MPEG-4 provides a structure of profiles and levels for the coding of natural, synthetic, and synthetic/natural hybrid visual content.

1. Profiles for Natural Video Content

1. Simple Visual Profile - It is used for coding of rectangular video objects, suitable for applications on mobile networks. This type of coding is very efficient and

error-resilient.

2. Simple Scalable Visual Profile - This profile supports coding of temporal and spatial scalable objects to Simple Visual Profile. This is useful for applications that provide services at more than one level of quality due to bit-rate or decoder resource limitations.

3. Core Visual Profile - When arbitrary-shaped and temporally scalable objects need to be coded to Simple Visual Profile, the Core Visual Profile steps in with the required support. Applications like Internet multimedia applications that provide simple content interactivity have found this profile very useful.

4. Main Visual Profile - It adds support for coding of interlaced, semi-transparent, and sprite objects to the Core Visual Profile. This profile is used by applications that are responsible for entertainment-quality broadcast and DVD applications.

5. N-Bit Visual Profile - Many applications require their video objects to have pixel depths ranging from 4 to 12 bits. These objects need to be coded to the Core Visual profile. $N-$Bit Visual Profile provides the necessary support to accomplish this.

6. Advanced Real-Time Simple Profile - Real time coding applications like videophone, teleconferencing, and remote observation need far more error resilient coding techniques of rectangular video objects. The coded objects should have better temporal resolution stability with low buffering delay. This profile provides the support for achieving the above mentioned objectives.

7. Core Scalable Profile - It adds support for coding of temporal and spatial scalable arbitrarily shaped objects to the Core Profile. The main functionality of

this profile is object based SNR and spatial/temporal scalability for regions or objects of interest.

8. Advanced Coding Efficiency - This profile improves the coding efficiency for both rectangular and arbitrary shaped objects. It is suitable for applications such as mobile broadcast reception, acquisition of image sequences (camcorders), and other applications where high coding efficiency is requested.

2. Profiles for Synthetic and Synthetic/Natural Video Content

1. Simple Facial Animation Visual Profile - It helps to animate a facial model.

2. Scalable Texture Visual Profile - Some applications need multiple scalability levels. This profile provides spatial scalable coding of still image objects useful for applications like mapping texture onto objects in games.

3. Basic Animated $2 - D$ Texture Visual Profile - This profile provides spatial scalability, SNR scalability, and mesh-based animation for still image objects.

4. Hybrid Visual Profile - This profile combines the ability to decode arbitrary-shaped and temporally scalable natural video objects with the ability to decode several synthetic and hybrid objects, including simple face and animated still image objects. It is suitable for various content rich multimedia applications.

5. Advanced Scalable Texture Profile - Applications that require fast random access along with multiple scalability levels and arbitrary shaped coding of still objects require many extra features. Decoding of arbitrary shaped texture and still images including shape coding, wavelet tiling, and error resilience are few of these features. This profile helps to support all these extra features.

6. Advanced Core Profile - It combines the ability to decode arbitrary shaped video objects with the ability to decode arbitrary shaped scalable still image objects. It is suitable for various content-rich multimedia applications.

7. Simple Face and Body Animation Profile - This profile is a superset of the Simple Face Animation profile with body animation.

E. Scene Modeling and Interactivity

MPEG-4 can handle multiple video objects. These objects can be transmitted independently. The can be arranged together in the prescribed fashion at the decoder. MPEG-4 can also handle audio objects similarly. *BInary Format for Scenes* (BIFS) is the language that describes how objects should be brought together at the decoder to form a complete scene. BIFS is based on *Virtual Reality Modeling Language* (VRML) with extensions that provide simpler constructs for $2-D$ objects and coordinate spaces. BIFS provides a hierarchical or tree structure where objects may be combined into groups. Individual objects may be Manipulated within a group. If necessary, a whole group can be manipulated. Nodes of the tree may be added or removed any time.

Each object and each group has a local coordinate that describes its location spatially as well as temporally. All these local coordinates are related to a set of global coordinates describing the scene. BIFS data is sent in a separate stream or multiplexed with video or audio sessions.

MPEG-4 is designed for use in applications where the end-user may interact with the scenes being played by the multimedia application. The application must have the ability to change the presentation based upon user requests. For interactivity, any interactive device like mice, remote control, and keyboard is required to produce

BIFs commands. These commands are combined with the BIFS content of the master scene to change the order of objects that are to be decoded by the decoder at the client side. The decoder need not be aware of the origin of BIFS commands.

F. Chapter Summary

This chapter provides a brief description of MPEG-4 video standards. MPEG-4 is expected to be the most widely used format for video encoding in the future. The video traces used in this work are all coded in accordance with the MPEG-4 standards. As discussed earlier in this chapter, MPEG-4 introduces the concept of object-oriented encoding of video into VOPs, followed by GOVs, VOLs, and VS subsequently in its hierarchy. This enables further compression of the video without compromising the quality. The next chapter provides an overview of the different empirical modeling techniques that are used in this research.

CHAPTER III

EMPIRICAL MODELING TECHNIQUES

A. Introduction

The research problem address in this work is the multi-step-ahead prediction of moving average of VOP size time-series of MPEG-coded video source traffic. In order to predict future VOP sizes, one needs to model the system which generates these VOPs. There are two types of models used in system modeling, physical models and empirical models. In physical models, mathematical expressions are derived which describe the relation between the input system variables and the output system variables. On the other hand, empirical models are derived from the observed data of the system. Empirical models are also called ’black-box’ models. The system models developed in this research are empirical models.

Empirical models are developed from the observed data of the system under consideration. This type of modeling is necessary in cases where the dynamics of the system becomes increasingly complex and uncertainties in data limit further mathematical analysis. The modeling of the system is crucial for the predictors to give accurate results. In general, the empirical modeling techniques to be considered in the research can be categorized as linear techniques and non-linear techniques.

B. Linear Techniques

These techniques assume that the system at hand can be represented using a linear model. Linear models can be classified into two categories, the input-output models and state-space models. The input-output models are used to represent the relation between the system inputs and outputs in the form of a linear regression. The state-space models use intermediate states to represent the system. Some commonly used linear models are Auto Regressive Exogenous (ARX) models, Auto Regressive Moving Average Exogenous (ARMAX) models, Output Error (OE) models, state-space models, etc.

In this work, we use input-output models for linear modeling. This is because the MPEG-coded video bit rate data is in the form of a time-series. Furthermore, out of the commonly used input-output models, ARX models give best results for the problem at hand. The following section provides a deeper insight of modeling using ARX modeling technique.

1. Auto-Regressive Exogenous (ARX) Model Structure

ARX is the simplest of all linear system modeling techniques. The AR in ARX model denotes the auto-regressive part and X denotes the extra input called exogenous variable. The general Single-Input Single-Output (SISO) ARX model can be represented by the following equation:

$$y(t) = a_1 y(t-1) + ... + a_{n_y} y(t-n_y) + b_1 u(t-n_k) + ... + b_{n_u} u(t-n_k-n_u+1) \quad (3.1)$$

where *y(t)* is the output of the ARX model, n_y is the number of past outputs also called as the lag terms of the model, *u(t)* is the input to the ARX model, n_u is the number of past input lags used in the model and n_k is the pure time delay (the dead time) in the system. The coefficients $a_1, ..., a_{n_y}$ and $b_1, ..., b_{n_u}$ are assumed to be known.

From the SISO ARX model represented in equation 3.1, the single-step-ahead predictor (SSP), $\widehat{y}(t+1/t)$, can be represented by the following equation:

$$\widehat{y}(t+1/t) = a_1 y(t) + ... + a_{n_y} y(t-n_y+1) + b_1 u(t-n_k+1) + ... + b_{n_u} u(t-n_k-n_u+2) \quad (3.2)$$

Similarly, the multi-step-ahead predictor (MSP) can be represented by:

$$\widehat{y}(t+1/t-p+1) = a_1\widehat{y}(t/t-p+1) + ... + a_{n_y}\widehat{y}(t-n_y+1/t-p+1) + \\ b_1u(t-n_k+1) + ... + b_{n_u}u(t-n_k-n_u+2) \tag{3.3}$$

The above equation represents a p-step-ahead predictor.

2. Auto-Regressive Exogenous Parameter Estimation

The predictor form of the ARX model is discussed in the previous section. The parameters of the ARX model, $a_1, ..., a_{n_y}$ and $b_1, ..., b_{n_u}$ must be determined from the measurement data, $y(t)$ and $u(t)$. The estimation of these parameters is discussed in this section. In the matrix form, the ARX predictor can be written as

$$\widehat{y}(t+1/t) = \varphi^T(t+1)\theta \tag{3.4}$$

where, $\varphi(t+1) = [y(t), ..., y(t-n_y+1), u(t-n_k+1), ..., u(t-n_k-n_u+2)]^T$ and $\theta = [a_1, ..., a_{n_y}, b_1, ..., b_{n_u}]^T$.

Here, the ARX predictor form is written as a scalar product between the data vector $\varphi(t+1)$ and the parameter vector θ. This is in the form of a linear regression with the parameter vector θ as the regression vector and hence the least square method can be used to solve for θ.

In order to solve for the parameters of the ARX predictor using least-squares algorithm, the mean-square of the prediction error, $V_N(\theta, Z^N)$ is defined as

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{t=1}^{N}[y(t) - \widehat{y}(t/t-1;\theta)]^2 \tag{3.5}$$

where Z^N is the data set of N input-output samples $u(t)$ and $y(t)$ for $t = 1, ..., N$. The above equation is also be interpreted as the objective function of the least-squares problem. The objective function, $V_N(\theta, Z^N)$, is minimized with respect to θ. The solution to this least-squares problem is the value of $\widehat{\theta}_N$ that minimizes $V_N(\theta, Z^N)$. This is given by

$$\widehat{\theta}_N = [\sum_{t=1}^{N} \varphi(t)\varphi^T(t)]^{-1} \sum_{t=1}^{N} \varphi(t)y(t) \tag{3.6}$$

In case of a time-series there is no exogenous part. Hence in this case the model becomes an Auto-Regressive (AR) model.

C. Non-linear Techniques

One of the most frequently used tools for non-linear system modeling is a multilayer perceptron (MLP) neural network. By selecting properly the training sets, which represent most of the operating conditions of the system, neural networks (NNs) can be used to model the disturbances as well. The robustness to the uncertainties of the plant is due to the fact that these are empirical models that do not require specific system parameters. The use of NNs for modeling therefore helps in generating predictions of outputs, that are insensitive to disturbances. The description of the NNs which follows is extracted from [34].

A NN has multiple interconnected processing elements grouped into layers. Each layer has several nodes. The inputs to one node in a layer are the outputs of all other nodes in the previous layer. The nodes algebraically sum these weighted signals and pass them through a nonlinear squashing function to produce a net output. The function is usually a sigmoid function or a hyperbolic tangent. Based on the structure of the network or the way the nodes are interconnected, there are two broad categories of NNs; the feed-forward multi-layer perceptron (FMLP) and the recurrent multi-layer perceptron (RMLP). FMLP is different from RMLP in the sense that there is no cross talk between the nodes of a given layer. Figure 3 and Figure 4 show the structure of an

FMLP and an RMLP network respectively. Each layer in a multilayer neural network has its own specific function. The input layer accepts input signals from the outside world and redistributes these signals to all neurons in the hidden layer. The input layer rarely includes computing neurons, and thus does not process input patterns. The output layer accepts output signals, a stimulus pattern, from the hidden layer and establishes the output pattern of the entire network. Any continuous function can be expressed with one hidden layer. Two hidden layers can predict discontinuous functions too.

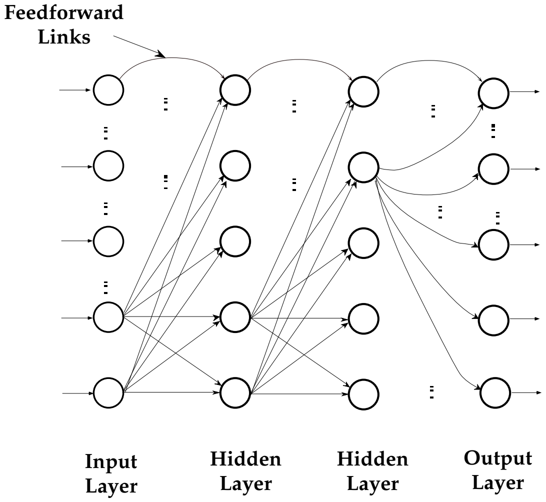

Fig. 3. Schematic diagram of the FMLP network.

1. FMLP Networks

FMLP is a static NN, and in the NN literature such networks are called feed-forward or non-recurrent. The network is composed of an input layer, a series of hidden layers and an output layer. In this network, the signals from each node are transmitted to all the nodes in the next layer, and only the hidden layers have a sigmoid-type

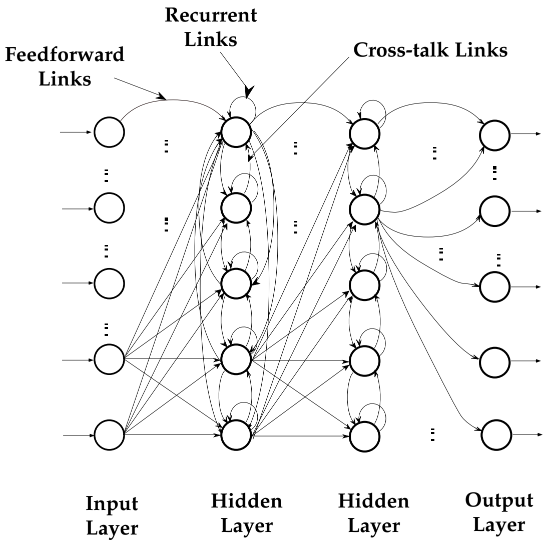

Fig. 4. Schematic diagram of the RMLP network.

discriminatory function. In this work, a hyperbolic tangent has been used as the discriminatory function. The input and the output layers have linear discriminatory functions and the input layer has no biases. FMLPs with appropriate signals in the input layer are good at approximating static nonlinearities, i.e. memory-less nonlinear functions. Each of the processing elements of an FMLP network is governed by the following equation:

$$x_{[l,i]} = \sigma_{[l,i]}\Big(\sum_{j=1}^{N_{[l-1]}} w_{[l-1,j][l,i]}\, x_{[l-1,j]} + b_{[l,i]} \Big), \tag{3.7}$$

for $i = 1, \ldots, N_{[l]}$ (the node index), and $l = 1, \ldots, \mathcal{L}$ (the layer index), where $x_{[l,i]}$ is the ith node output of the lth layer for sample t, $w_{[l-1,j][l,i]}$ is the weight, the adjustable parameter, connecting the jth node of the $(l-1)$th layer to the ith node of the lth layer, $b_{[l,i]}$ is the bias, also an adjustable parameter, of the ith node in the lth layer, and $\sigma_{[l,i]}(\cdot)$ is the discriminatory function of the i-th node in the l-th layer.

2. RMLP Networks

In RMLP networks, the cross talk adds memory to the network. Therefore, RMLP can be used for approximating dynamic systems. Each of the processing elements of the RMLP is governed by the following equations:

$$z_{[l,i]}(t) = \sum_{j=1}^{N_{[l]}} w_{[l,j][l,i]}\, x_{[l,j]}(t-1) + \sum_{j=1}^{N_{[l-1]}} w_{[l-1,j][l,i]}\, x_{[l-1,j]}(t) + b_{[l,i]}, \tag{3.8}$$

and

$$x_{[l,i]}(t) = \sigma_{[l,i]}\Big(z_{[l,i]}(t)\Big), \tag{3.9}$$

where $z_{[l,i]}(t)$ represents the internal state variable of the ith node at the lth layer for sample t; $x_{[l,i]}(t)$ is the ith node output of the lth layer for sample t, and $b_{[l,i]}$ is the bias of the node ; $w_{[l,j][l',i]}$ is the weight associated with the link between the jth node of the lth layer to the ith node of the l'th layer. Furthermore, t represents the discrete-time at which the node and network outputs are computed, with the node index $i = 1, \ldots, N_{[l]}$, and layer index $l = 1, \ldots, \mathcal{L}$, and with the $\sigma_{[l,i]}(\cdot)$ for the input and output layers ($l = 1$ and $l = \mathcal{L}$) being linear. The function $\sigma_{[l,i]}(\cdot)$ for the hidden-layer nodes is a squashing function and in this study $\tanh(\cdot)$ is used. The term $b_{[l,i]}$ provides the bias for each node.

D. Predictor Algorithms Using NNs

1. Single-Step-Ahead Prediction

Figure 5 shows the schematic of a SSP predictor. In SSP, the predictions for the output of the system at time *t+1* are obtained using all the inputs and outputs available up to time *t*. In case of a nonlinear system, the SSP of the system output,

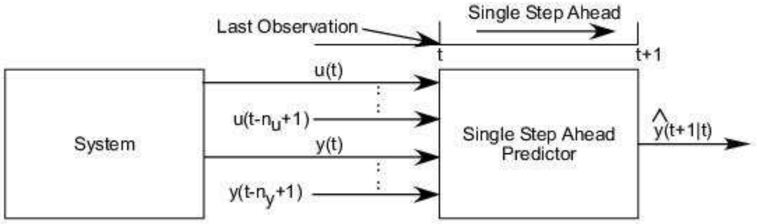

Fig. 5. Single-step-ahead predictor [Ref. 34, p. 44].

$\hat{y}(t+1|t)$, is expressed as

$$\hat{y}_i(t+1|t) = f_i(\mathcal{U}(t)), i = 1, \ldots, n, \tag{3.10}$$

where

$$\mathcal{U}(t) \equiv [y(t), \ldots, y(t-n_y+1), u(t), \ldots, u(t-n_u+1)], \tag{3.11}$$

The SSP and MSP representations for the FMLP network as shown in [35] is given in the following sub sections.

a. SSP with FMLP Networks

FMLP networks in SSP form represents a static set-up without recurrence. The single-step-ahead prediction scheme using FMLP networks can be represented as

$$\widehat{\mathbf{y}}_{NN}(t+1|t;\mathcal{W}) = \mathcal{F}\Big(\mathcal{U}(t);\mathcal{W}\Big), \tag{3.12}$$

where $\mathcal{W}$ is weight matrix which is to be determined by the learning algorithm, $\mathcal{F}$ represents the nonlinear transformation of the input, and the vector is defined as

$$\mathcal{U}(t) \equiv [\mathbf{y}(t), \ldots, \mathbf{y}(t-n_y+1), \mathbf{u}(t), \ldots, \mathbf{u}(t-n_u+1)] \tag{3.13}$$

where $\mathbf{y}(t)$ is the measured system outputs, $\widehat{\mathbf{y}}(\cdot \mid \cdot)$ is the system output predictions, $\mathbf{u}(t)$ is the inputs, n_y and n_u are the maximum number of lags in the output and input, respectively.

b. SSP with RMLP Networks

RMLP networks can be used for SSP. The single-step-ahead prediction scheme using RMLP networks can be represented as

$$\widehat{\mathbf{y}}_{NN}(t+1|t;\mathcal{W}) = \mathcal{F}(\mathcal{U}(t), \mathbf{z}_{[2]}(t), \ldots, \mathbf{z}_{[\mathcal{L}-1]}(t); \mathcal{W}), \tag{3.14}$$

where $\mathcal{W}$ is weight matrix to be determined by the learning algorithm, $\mathcal{F}$ represents the nonlinear transformation from the inputs and internal states to the outputs, $\mathbf{z}_{[l]}(t)$ is the internal state vector of the lth layer, and $\mathcal{L}$ is the total number of layers. RMLP in SSP form represents a static network with inherent recurrence.

2. Multi-Step-Ahead Prediction

Figure 6 shows the schematic of a MSP predictor. In MSP predictions of the output at time *t+1* is obtained using the inputs or their estimates up to time t and all the past predictions up to time t which are generated using the measurements up to time *t-p+1*, where p is greater than 1. The MSP of a nonlinear dynamic system output, $\hat{y}(t+1|t-p+1)$, is expressed as:

$$\hat{y}_i(t+1|t-p+1) = f_i(\hat{\mathcal{U}}(t)), i = 1, \ldots, n, \tag{3.15}$$

where

$$\hat{\mathcal{U}}(t) \equiv [\hat{y}(t|t-p+1), \ldots, \hat{y}(t-n_y+1|t-p+1), u(t), \ldots, u(t-n_u+1)], \tag{3.16}$$

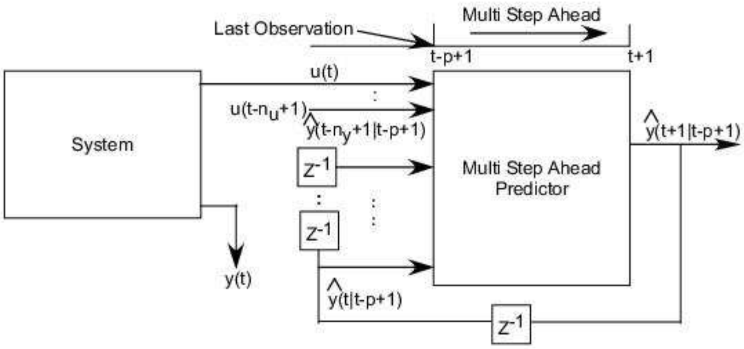

Fig. 6. Multi-step-ahead predictor [Ref. 34, p. 44].

where $y(t)$ is the system outputs, $u(t)$ is the inputs, n_y and n_u are the maximum number of lags in the output and input, respectively, and $f_i(\cdot)$ is a nonlinear function that best describes the model. If the inputs $u(t)$ for $t > t - p + 1$ are not available, best estimates of these quantities can be used instead. The MSP representations for FMLP and RMLP networks as shown in [35] is given in the following sub sections.

a. MSP with FMLP Networks

FMLP networks can be used for MSP. A MSP using FMLP can be represented by the following equation:

$$\widehat{\mathbf{y}}_{NN}(t+1|t-p+1;\mathcal{W}) = \mathcal{F}\Big(\widehat{\mathcal{U}}(t);\mathcal{W}\Big), \tag{3.17}$$

where $\mathcal{W}$ is weight matrix which is to be determined by the learning algorithm, $\mathcal{F}$ represents the nonlinear transformation of the input, and the vector is defined as

$$\hat{\mathcal{U}}(t) \equiv [\widehat{\mathbf{y}}(t|t-p+1), \ldots, \widehat{\mathbf{y}}(t-n_y+1|t-p+1), \mathbf{u}(t), \ldots, \mathbf{u}(t-n_u+1)], \tag{3.18}$$

where $\widehat{\mathbf{y}}(\cdot \mid \cdot)$ is the system output predictions, $\mathbf{u}(t)$ is the inputs, n_y and n_u are the maximum number of lags in the output and input, respectively. An FMLP network in MSP form represents a dynamic system without recurrence.

b. MSP with RMLP Networks

MSP using RMLP represents a dynamic network. The following equation describes a MSP with RMLP network

$$\hat{\mathbf{y}}_{NN}(t+1|t-p+1;\mathcal{W}) = \mathcal{F}(\hat{\mathcal{U}}(t), \mathbf{z}_{[2]}(t), \ldots, \mathbf{z}_{[\mathcal{L}-1]}(t); \mathcal{W}), \tag{3.19}$$

where $\mathcal{W}$ is weight matrix to be determined by the learning algorithm, $\mathcal{F}$ represents the nonlinear transformation from the inputs and internal states to the outputs, $\mathbf{z}_{[l]}(t)$ is the internal state vector of the lth layer, and $\mathcal{L}$ is the total number of layers. An RMLP network in MSP form represents a dynamic system with recurrence.

E. Learning Algorithms in Neuro-predictors

Learning refers to adjusting the weights and biases so that the resulting network best approximates the relationship between the input-output data. Back Propagation (BP) algorithm is most commonly used algorithm for training perceptrons like FMLP. The learning methods used for RMLPs is slightly different due to the cross-talk.

1. Learning Algorithm for FMLP Networks

Back Propagation (BP) algorithm is most commonly used algorithm for training perceptrons like FMLP Let us consider a three-layer network with $x_{[1,1]}, \ldots, x_{[1,N_{[1]}]}$ as inputs and $\widehat{y}_{[1]}, \ldots, \widehat{y}_{[N_{[\mathcal{L}]}]}$ as outputs to explain the algorithm of training FMLP networks for single-step-ahead prediction. $N_{[l]}$ is the number of nodes in layer l and the output layer is represented using the notation $\mathcal{L}$. Input signals, are propagated through the network from left to right. The symbol $w_{[i,j][m,n]}$ denotes the weight for the connection between neuron j of layer i and neuron n of layer m. Error is propagated from the output layer to the hidden layer. The error signal at the output of FMLP network at iteration k is defined by

$$E(k) = \sum_{j=1}^{N_{[\mathcal{L}]}} (x_{[\mathcal{L},n]}(k) - y_n(k))^2 \tag{3.20}$$

where, $x_{[\mathcal{L},n]}(k) {\equiv} \widehat{y_n}(k)$ and $y_n(k)$ denote the $n-th$ network output (prediction) and observed process output (measurement) respectively. Weight $w_{[\mathcal{L}-1,j][\mathcal{L},n]}$ is updated using a very straightforward procedure, as follows:

$$w_{[\mathcal{L}-1,j][\mathcal{L},n]}(k+1) = w_{[\mathcal{L}-1,j][\mathcal{L},n]}(k) + \Delta w_{[\mathcal{L}-1,j][\mathcal{L},n]}(k) \tag{3.21}$$

where $\Delta w_{[\mathcal{L}-1,j][\mathcal{L},n]}(k)$ is the weight correction. The weight correction in the multi-layer network is computed by:

$$\Delta w_{[\mathcal{L}-1,j][\mathcal{L},n]}(k) = \alpha {\times} \widehat{y}_n(k) {\times} \delta_{[\mathcal{L},n]}(k) \tag{3.22}$$

where $\delta_{[\mathcal{L},n]}(k)$ is the error gradient at neuron n in the output layer at iteration k. To calculate the weight correction for the hidden layer, the following equation is used:

$$\Delta w_{[1,i][\mathcal{L}-1,j]}(k) = \alpha {\times} x_{[1,i]}(k) \times \delta_{[\mathcal{L}-1,j]}(k) \tag{3.23}$$

where $\delta_{[\mathcal{L}-1,j]}(k)$ represents the error gradient at neuron j of layer $\mathcal{L}{-}1$ (hidden layer). For more information on the BP algorithm refer to [36] and [37]. In this thesis, this algorithm is referred as the FMLP algorithm.

2. Learning Algorithm for RMLP Networks

In this research we use two RMLP learning algorithms. The first learning algorithm of SSP using RMLP networks has been discussed in detail by Parlos, Chong, and Atiya [38]. A RMLP network can be used for both on-line as well as off-line training. During the off-line training, a set of K pairs of input-output data is presented to the neural network. After some fixed period of iterations the neural network reproduces the results of input-output data set within a certain degree of error tolerance. The network weights are updated using the *steepest descent approach*:

$$\Delta w_{[l-1,j][l,i]} = -\eta \sum_{k=1}^{K} \left(\frac{\partial E(k)}{\partial w_{[l-1,j][l,i]}} \right) \tag{3.24}$$

where η is the learning rate, and $E(k)$ is the squared error at time step k, given by

$$E(k) = \sum_{j=1}^{N_{[\mathcal{L}]}} (x_{[\mathcal{L},j]}(k) - y_j(k))^2 \tag{3.25}$$

where, $x_{[\mathcal{L},j]}(k) \equiv \hat{y_j}(k)$ and $y_j(k)$ denote the $j-th$ network output (prediction) and observed process output (measurement). For recurrent and cross-talk weights, as well as for the bias terms, a similar update rule is used [38].

When using NNs as MSPs, these networks need to be *trained* using data sets obtained under different operating conditions. Training requires a target value to be given to the network to update it parameters. If latest measurements are used as inputs to the NN, the approach is referred to as teacher forcing(TF), if the past predictions are used as inputs, it is called Global Feedback (GF). Hence TF is used for SSP and GF is used for MSP.

The learning algorithm for a RMLP structure with TF is based on gradient descent method and minimizes the Mean Squared Error (MSE) between the outputs predicted by the predictor and the measured values. In other words, the objective is to determine the change in the network parameters $w_{[l-1,i][l,j]}, w_{[l,i][l,j]}$ and $b_{[l,i]}$, for all i, j and l, such that the following function is minimized:

$$E \equiv \frac{1}{2}\sum_{t=0}^{NP-1} E(t+1) \equiv \frac{1}{2}\sum_{t=1}^{NP}\sum_{j=1}^{N_{[\mathcal{L}]}} \left[\widehat{y}_{\mathrm{NN},j}(t+1|t) - y_j(t+1)\right]^2, \qquad (3.26)$$

where NP is the total number of data pairs in the estimation data set. The learning algorithm for RMLP with GF is also a gradient descent method. It is slightly more complicated due to dependency of the gradient on the previous values. The objective function being minimized is as follows;

$$E \equiv \frac{1}{2}\sum_{t=0}^{NP-1} E(t+1) \equiv \frac{1}{2}\sum_{t=0}^{NP}\sum_{j=1}^{N_{[\mathcal{L}]}} \left[\widehat{y}_{\mathrm{NN},j}(t+1|t\!=\!0) - y_j(t+1)\right]^2, \qquad (3.27)$$

where, $\widehat{y}_{\mathrm{NN},j}(t+1|t\!=\!0)$ is the recursively predicted NN output, for $t = 0, \ldots, (NP-1)$, and $y_j(t+1)$ is the j^{th} target (sensed output) in the training set. Training refers to establishing the structure of the network along with the weights and biases, that result in a minimum error for the predicted outputs. The mathematical equations representing the learning algorithm for RMLP networks are too complex and numerous to be listed in this thesis. Parlos, Rais, and Atiya [32] provide a mathematically rigorous explanation of the learning algorithms for MSP using RMLP networks. In this thesis, this algorithm is referred as the RMLP algorithm.

The second RMLP learning algorithm used in this research is discussed in detail by Jaeger in [33]. This NNs trained using this algorithm are called Echo State Networks (ESNs). ESNs look at the RMLP networks from a new angle. ESN has an input layer, output layer and a bunch of neurons forming the hidden layer which is also called as the Dynamic Reservoir (DR). Large recurrent neural networks are interpreted as 'reservoirs' capable of storing complex, excitable dynamics. The output units can be considered as the outlets from these reservoirs. The key idea which differentiates ESNs from the conventional recurrent neural networks is that during training, only the weights from the network to the outputs are updated. Other network weights once initialized do not change during the course of the training.

Let us consider a network with K inputs, N internal nodes and L outputs. Activation of inputs units is $u(t) = (u_1(t), ..., u_K(t))$, of internal nodes is $x(t) = (x_1(t), ..., x_N(t))$ and of output nodes is $y(t) = (y_1(t), ..., y_L(t))$, where t represents the time step. The internal nodes are updated using:

$$x(t+1) = f(\mathcal{W}^{in}u(t+1) + \mathcal{W}x(t) + \mathcal{W}^{back}y(t)) \tag{3.28}$$

where $\mathcal{W}^{in}$ is an $N \times K$ input weight matrix, $\mathcal{W}$ is an $N \times N$ internal weight matrix, $\mathcal{W}^{back}$ is a $N \times L$ back weight matrix and f is the squashing function for the internal nodes. These three weight matrices are generated randomly and are not updated during the course of training. The prediction is done using the equation:

$$\widehat{y}_{NN}(t+1/t; \mathcal{W}^{out}) = f^{out}(\mathcal{W}^{out}(u(t+1), x(t+1), y(t))) \tag{3.29}$$

where $\mathcal{W}^{out}$ is a $L \times (K + N + L)$ output weight matrix and f^{out} is the squashing function for the output nodes. The training of ESN means the calculation of $\mathcal{W}^{out}$ matrix. It is done using the TF technique. The weights matrices are generated randomly and the network is allowed to run for the training data. The states of the network while training are stored in a $(T \times (K + N + L))$ size matrix (M), where T is the length of the training data set. The teacher forced output, i.e. $(f^{out})^{-1}y(t)$ is stored in a $T \times L$ size matrix $\mathcal{T}$. The mean squared error at the output of ESN is defined by

$$MSE = \frac{1}{t}\sum_{i=1}^{t}(d(i) - \mathcal{W}^{out}(x(i), u(i), y(i-1))^2 \tag{3.30}$$

where the signal d is the teacher forced output. The output weight matrix $\mathcal{W}^{out}$ is given by the equation

$$(W^{out})^T = (\mathcal{M})^{-1}\mathcal{T} \tag{3.31}$$

Further details on the ESN can be obtained from [33]. In this thesis, this algorithm is referred as the ESN algorithm.

F. Chapter Summary

The empirical modeling techniques used in this work are: ARX for linear modeling and FMLP, RMLP and ESN for non-linear modeling. In the past, researchers have used both linear and non-linear (mainly NNs) techniques to model MPEG-coded video source traffic. In this work, all the above mentioned empirical modeling techniques are used to develop multi-step-ahead predictors for the MPEG-coded video traces. The next chapter provides an insight on the video data traces used in this work. The statistical properties of these video traces is presented. The next chapter also highlights the temporal properties of these traces and presents the moving average calculations.

CHAPTER IV

ANALYSIS OF MPEG-4 VIDEO TRACES

Before getting into the modeling of MPEG-coded video source traffic, it is important to analyze the traffic data traces. In the past, researchers have used statistical parameters to describe the characteristics of video source traffic. In this chapter, a brief description of the data traces is provided. The video traces used in this research were obtained from [39]. This public archive has a very good collection of MPEG-4 and H.263 video traces. This internet site is maintained by the Telecommunication Networks Group of the Technical University of Berlin.

A. Generation of Video Data Traces

The procedure that was followed to generate the video trace data is explained in detail in [40]. A brief description of this procedure is presented in the following lines. The video was played from VHS tapes using an ordinary video cassette recorder. Uncompressed YUV information of each video was grabbed using the tool `bttvgrab` (version 0.15.10) [41] at a frame rate of 25 frames/sec in the QCIF format. The luminance resolution was 176×144 picture elements (pels) and 4:1:1 chrominance subsampling at a color depth of 8 bits. This information was stored on a disk. The stored YUV frame sequences were used as input for both the MPEG-4 encoder and the H.263 encoder. The encoding was not performed in real-time. All the traces used in this research were generated using this procedure.

B. MPEG-4 Video Data Traces

The repository in [39] has a large number of video traces ranging from very high quality to a very low quality. The traces also cover a large range of bit rates. Each video trace has been coded using three different quantization schemes. Depending on the level of quantization, encoded sequences have been categorized into high, medium and low quality. For this work, eight different video data traces were selected. These are the same traces as used in [30]. Following is a description of the data traces used in this research as presented in [34].

1. *Aladdin* (High Quality): *Aladdin* is an animated movie with lot of special effects. It has lot of scenes that change quite frequently. Most of the initial training of the neural networks described later in the current research have been performed using the trace of *Aladdin*.

2. *ARD Talk* Show (High Quality): A talk show does not involve rapid scene changes. A video trace of a talk show like *ARD Talk* is necessary to demonstrate the efficacy of generalized model of video source traffic. *ARD Talk* helps in simulating dynamics of video footage of a typical video conference. A very significant scope of the present research is in the field of video conferences.

3. *Die Hard III* (Medium Quality): *Die Hard III* trace has been encoded with medium quality. It belongs to the genre of action movies.

4. *Jurassic Park* I(High Quality): *Jurassic Park* I belongs to the genre of action and drama movies. It contains a lot of computer generated special effects. The model of the video traffic should be able to take care of the dynamics arising out of high content of special effects.

5. *Lecture Room* (Medium Quality): The trace of the *Lecture Room* encoded with medium quality will be used to demonstrate the effectiveness of the designed predictor to predict the traffic due to applications like video conference. It has no special effects and relatively less scene changes.

6. *Silence of the Lambs* (Low Quality): All the above sequences belong to either high quality or medium quality encoding. The devised model of video source traffic should be able to model the dynamics of any type of video footage irrespective of the encoding parameters. *Silence of the Lambs* provides the trace to validate the generated model in the case of different quantization parameter used during encoding.

7. *Skiing* (Low Quality): A sports clipping with fast movements and encoded with low quality effectively encompasses different kinds of data that will be used for validating the performance of the designed prediction scheme.

8. *StarWars* (High Quality): The action sequences of *StarWars* provide a challenge for modeling of VOP sizes accurately. Any model of the video source traffic should be able to predict the size of the VOPs of Star Wars.

C. Statistical Analysis of MPEG-4 Traces

The statistics of all the frames/VOPs of the traces selected for this work is given in Table I and Table II. It can be seen from the ratio of peak to mean bit rate that all the MPEG-4 traces have a variable bit rate. Additionally the bit rates of video traces used varies from 0.06 to 0.77 Mbps. This shows that the video traces used in this work cover a sufficiently large range of bit rates. Comparing the statistics of the trace of movies encoded with different qualities, it can be safely assumed that

Table I. VOP statistics of MPEG-4 traces

Trace	Compr. Ratio $(YUV : MP4)$	Mean $\overline{X}$ (KB)	Var $(KB)^2$	CoV($\mathcal{S}_x/\overline{X}$)
Aladdin	17.5	2.2	3.0	0.8
ARD Talk	13.9	2.7	3.0	0.6
Die Hard III	30.8	1.2	1.1	0.8
Jurassic Park I	9.9	3.8	5.1	0.6
Lecture Room	131.7	0.3	0.2	1.6
Silence of the Lambs	72.0	0.5	0.8	1.7
Skiing	40.6	0.9	1.1	1.1
StarWars	27.6	1.4	0.8	0.7

Table II. Peak/mean and mean bit rate (MBR) of MPEG traces

Trace	Peak/Mean($X_{max}/\overline{X}$)	MBR(Mbps)
Aladdin	7.1	0.44
ARD Talk	5.7	0.54
Die Hard III	6.6	0.25
Jurassic Park I	4.4	0.77
Lecture Room	11.9	0.06
Silence of the Lambs	21.4	0.11
Skiing	9.8	0.19
StarWars	6.8	0.28

the dynamics of all the traces broadly covers the entire range of activities that are captured using a video camera. It is this assumption that is used to generalize the results obtained in this work to all MPEG-coded video data traces.

D. Long Term Dependency of the VOPs

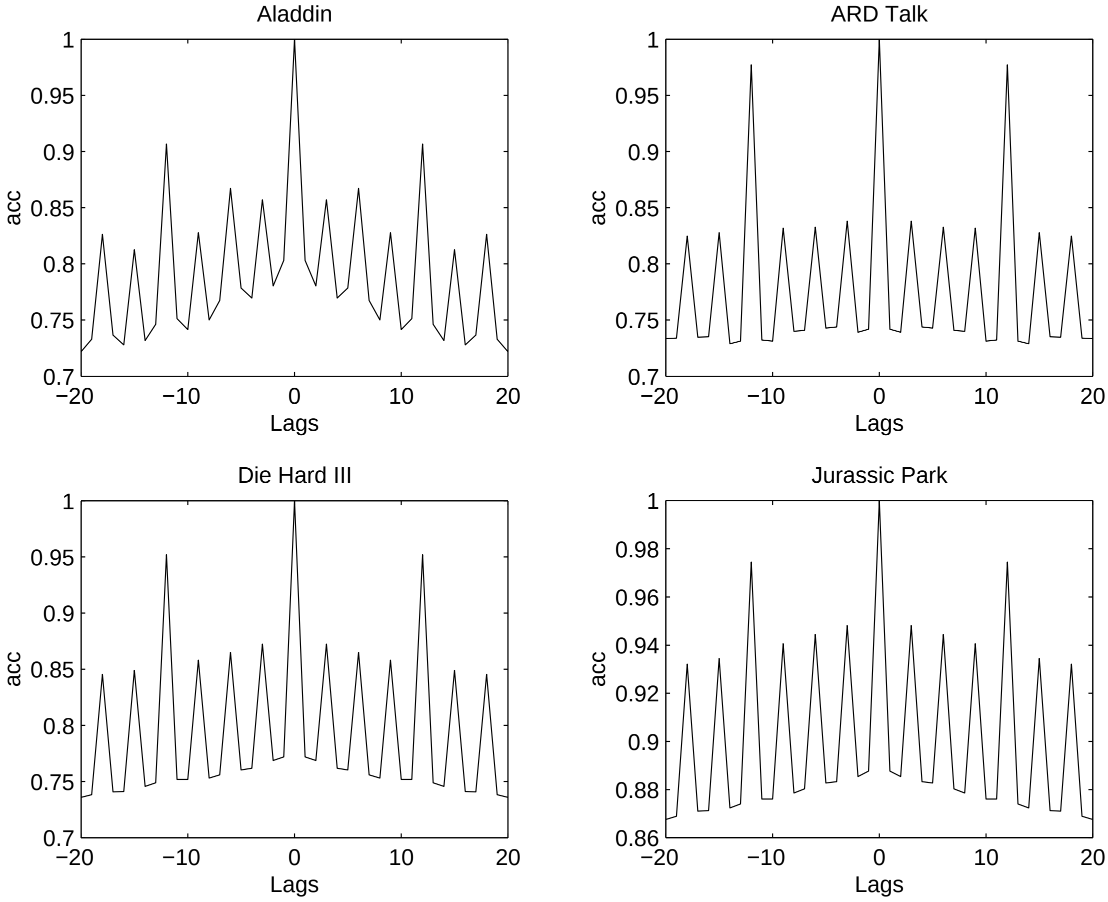

Fig. 7. Autocorrelation of VOPs of *Aladdin*, *ARD Talk*, *Die Hard III*, and *Jurassic Park*.

Figures 7 and 8 plot the autocorrelation functions of the VOPs for the video traces used in this research.

Study of the autocorrelation plots of the VOPs for different video traces provide useful insight into the nature of the data that is to be predicted. The rate at which the autocorrelation falls with increase in the lags is an indication of the long-range time dependency.

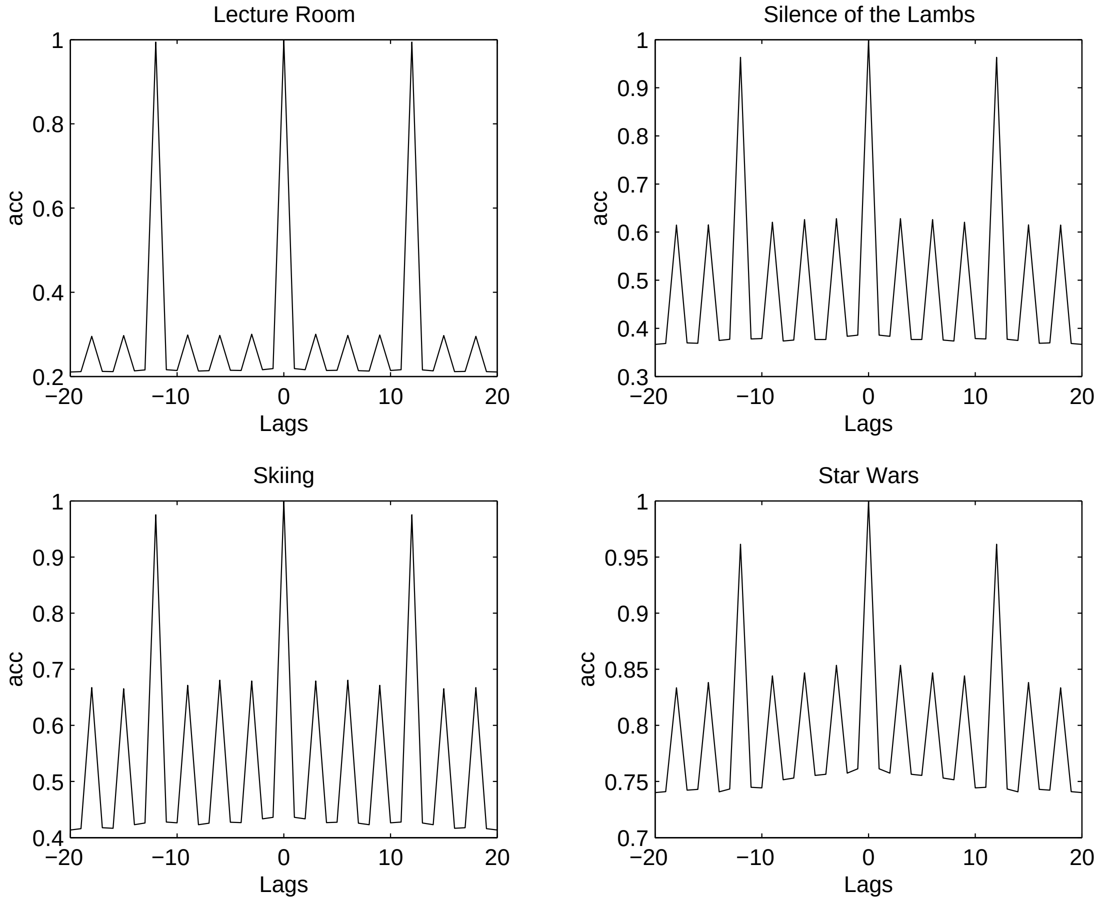

Fig. 8. Autocorrelation of VOPs of *Lecture Room*, *Silence of the Lambs*, *Skiing*, and *StarWars*.

It can be seen from the figures that the autocorrelation of these video traces is significantly high even after 20 lags. Because of this slowly decaying slope, it can be concluded that these video traces have a long-term temporal dependency. This is true in general for all the MPEG-coded video data traces.

The plots of the combined VOPs size autocorrelation functions in 7 and 8 also demonstrate a periodic spike pattern. This pattern is because of the repetitive GOV pattern of the data traces. The large positive spikes occur due to the $I-$VOPs. The $B-$VOPs are the reason behind the negative spikes. The $P-$VOPs also cause positive spikes but the magnitude of these spikes is much less than the ones caused due to the $I-$VOPs.

After looking at the definition of the $I-, B-$ and $P-$ VOPs, one is tempted to assume that the size of the $I-$ VOPs is always larger than the associated $P-$ and $B-$ VOPs. But this is not always the case. For videos with a high scene change rate, the size of $P-$ VOPs can exceed that of the neighboring $I-$ VOPs. This was found to be true in the case of the *Jurassic Park* video data trace used in this research.

E. Moving Average Time-series of VOP Sizes of Video Data Traces

It can be concluded from the available in the field of prediction of MPEG-coded video source traffic [30], that the multi-step-ahead predictions of individual VOP sizes has not been very successful. The errors in these predictions are significantly high making them unsuitable for further use. This is because the original time-series is extremely noisy. One way to solve this problem of excessive noise is to smooth the time-series by taking the moving average of original VOP sizes to generate a mean VOP size time-series.

The thrust behind this research work is the need to use the video coded time-series prediction as an input to a controller that controls the send rate of the MPEG-encoded video stream packets over the network in real-time. This controller would send periodic control signals and hence would work in discrete time.

The videos used in this work are encoded at a rate of 25 frames/sec. So if one is able to predict the size of every frame with reasonable accuracy, it would require a control signal every 40 ms. This would require a large bandwidth and would amount to an increase in the cost due to the control effort. In order to keep the cost down, one should have a low bandwidth and large control effort. Therefore, instead of using the prediction of each and every VOP size as an input to the controller, one can use the prediction of the moving average of the combined VOP size time-series. Moreover, since the prediction of individual VOP sizes is not very accurate, it would be better to work with an accurate multi-step-ahead prediction of the smoothed time-series.

An important thing to keep in mind is that the multi-step-ahead predictor is designed to facilitate the controller to achieve as efficient bandwidth allocation scheme. In case of multimedia networks, if the network receives the traffic from the output of the encoder, it is necessary to have the size of each VOP in the time-series. But when the controller is implemented between the encoder and the network, the bandwidth control scheme will depend on the controller output and not the encoder output. In this case, one no longer needs to predict each and every VOP size. In order to have a low cost control effort, the moving average time-series is used. This will work fine as long as the sampling frequency of the controller output is equal or higher than that of the input signal.

The bit rate of the time-series can be calculated using the equation:

$$X(k) = \frac{f}{w} \sum_{j=l}^{l+w-1} x(j) \tag{4.1}$$

where $l = 1, p, 2p, 3p, ..., np$, such that $np \leq M$, p is the period of shifting of window and $p \leq w$, M is the total number of VOPs in the series, f is frames per second, $X(k)$ is the k-th moving average of VOP source traffic rate in bytes/sec, and $x(j)$ is the j-th size of original VOP series. For this work, we predict the mean VOP size which is given by the equation

$$x_{MA}(k) = \frac{1}{w}\sum_{j=l}^{l+w-1} x(j) \tag{4.2}$$

where $x_{MA}(k)$ is the k-th moving average or the mean VOP size in bytes. For this work, w and p were selected to be 25 and 12 respectively.

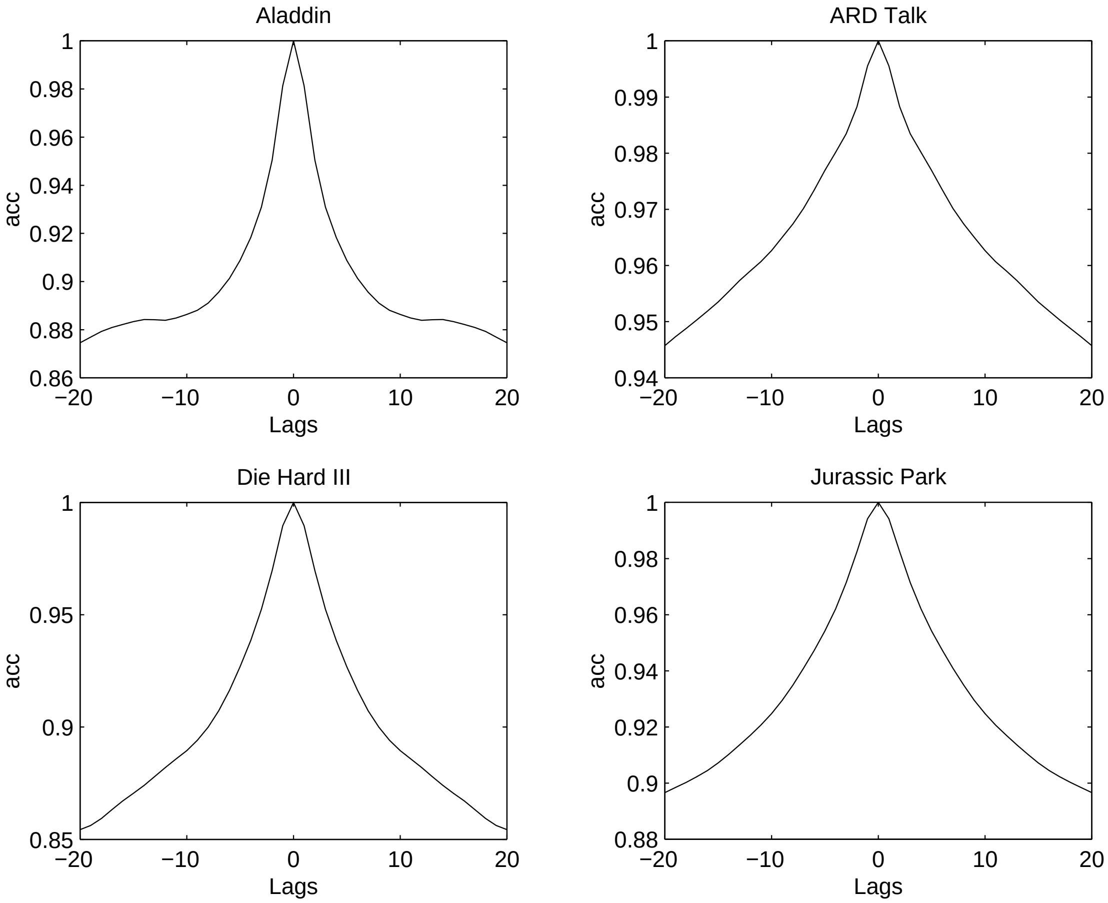

Fig. 9. Autocorrelation of the moving average or mean VOPs of *Aladdin*, *ARD Talk*, *Die Hard III*, and *Jurassic Park*.

Figures 9 and 10 present the autocorrelation functions of the moving average or mean VOP size time-series. The value of $w = 25$ corresponds to a time-frame of 1 sec.

It can be seen from the figures that the autocorrelation does not fall significantly after 20 lags, as in the case of the original VOP size time-series.

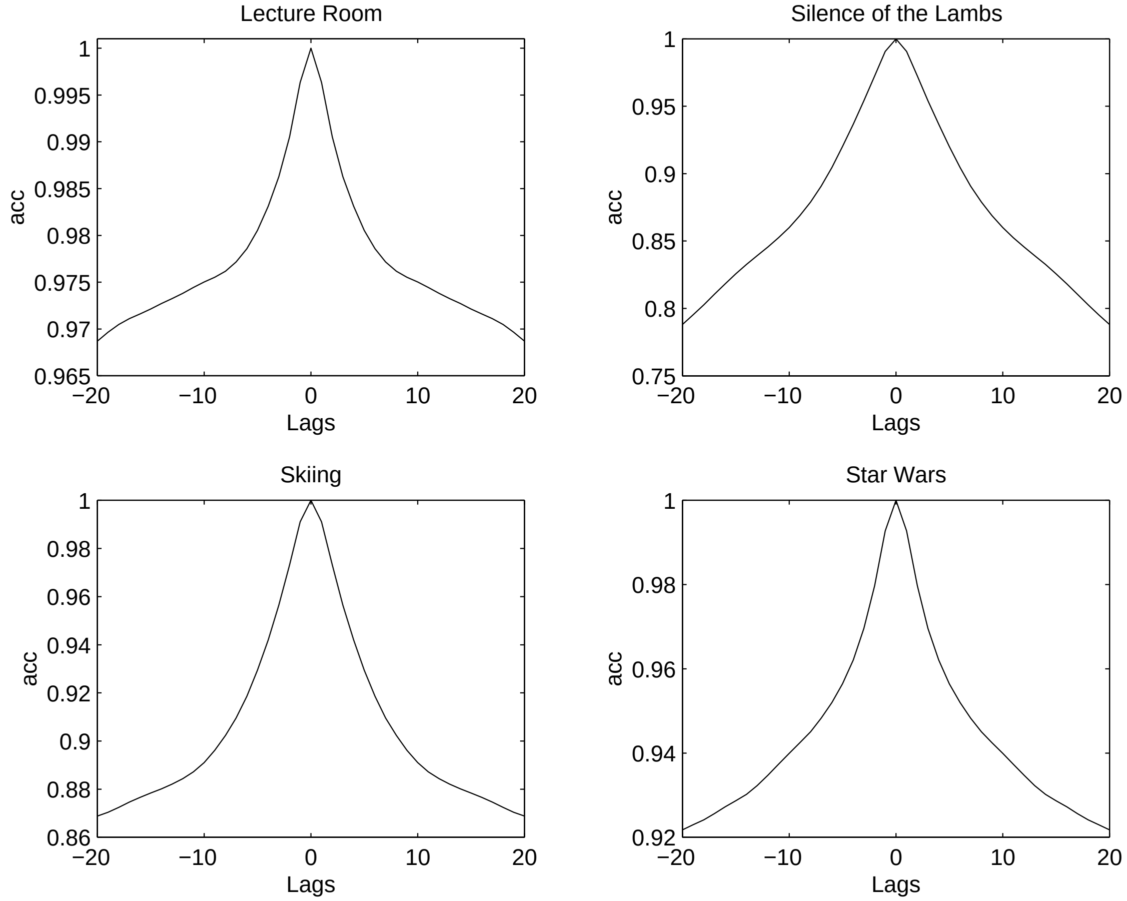

Fig. 10. Autocorrelation of the moving average or mean VOPs of *Lecture Room*, *Silence of the Lambs*, *Skiing*, and *StarWars*.

Figures 11 to 18 show the plots of the averaged time-series. The original time-series is shown in discrete points.

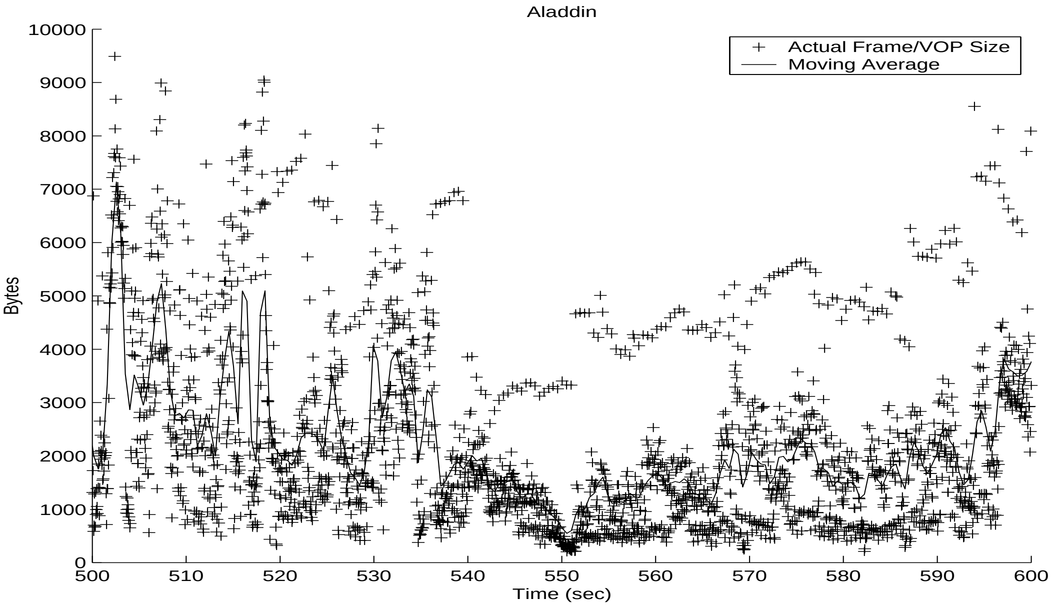

Fig. 11. Moving average time-series of *Aladdin.*

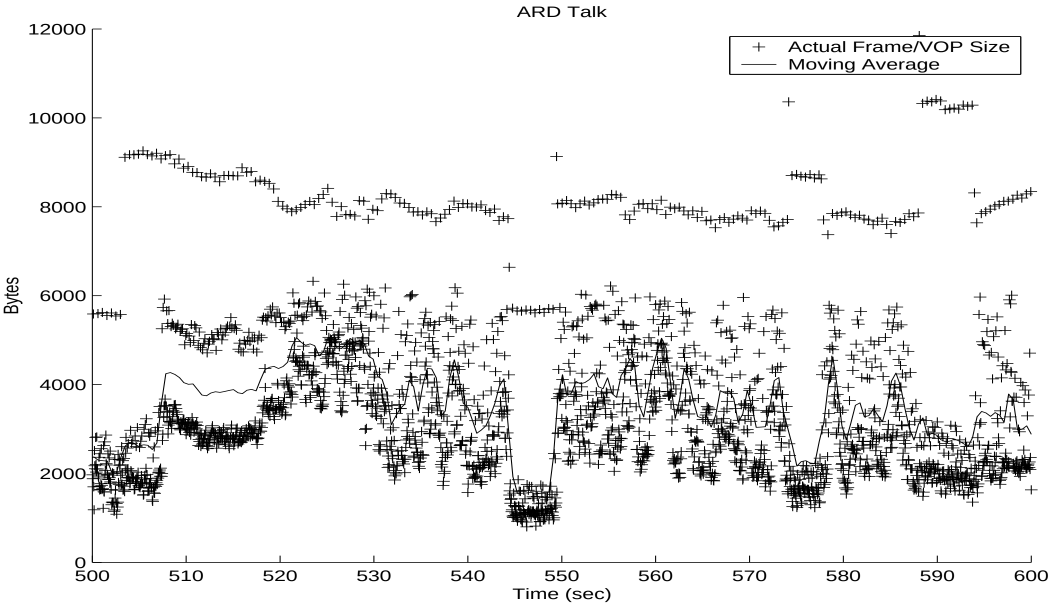

Fig. 12. Moving average time-series of *ARD Talk.*

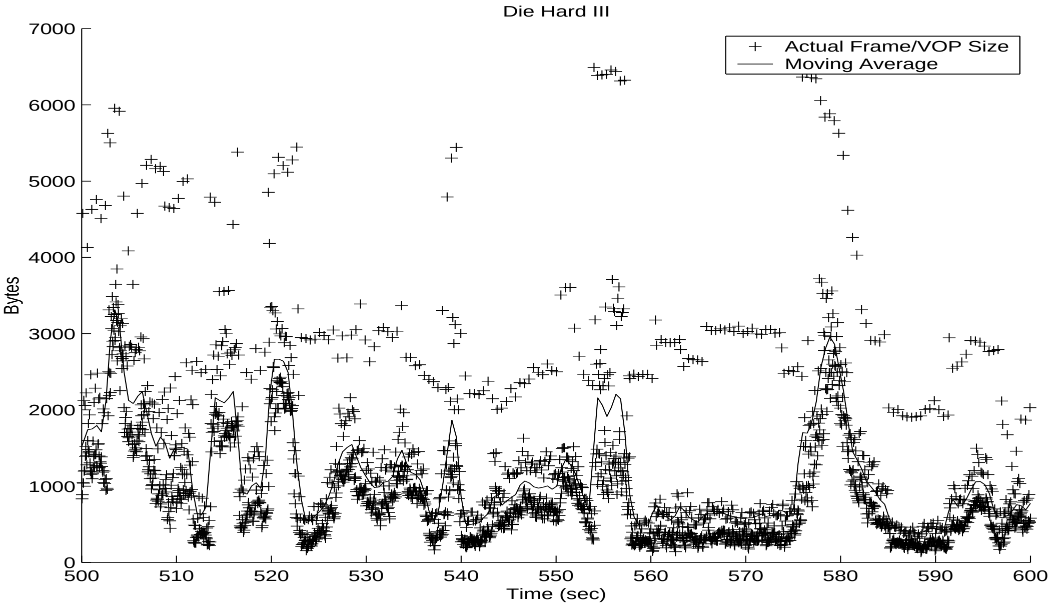

Fig. 13. Moving average time-series of *Die Hard III*.

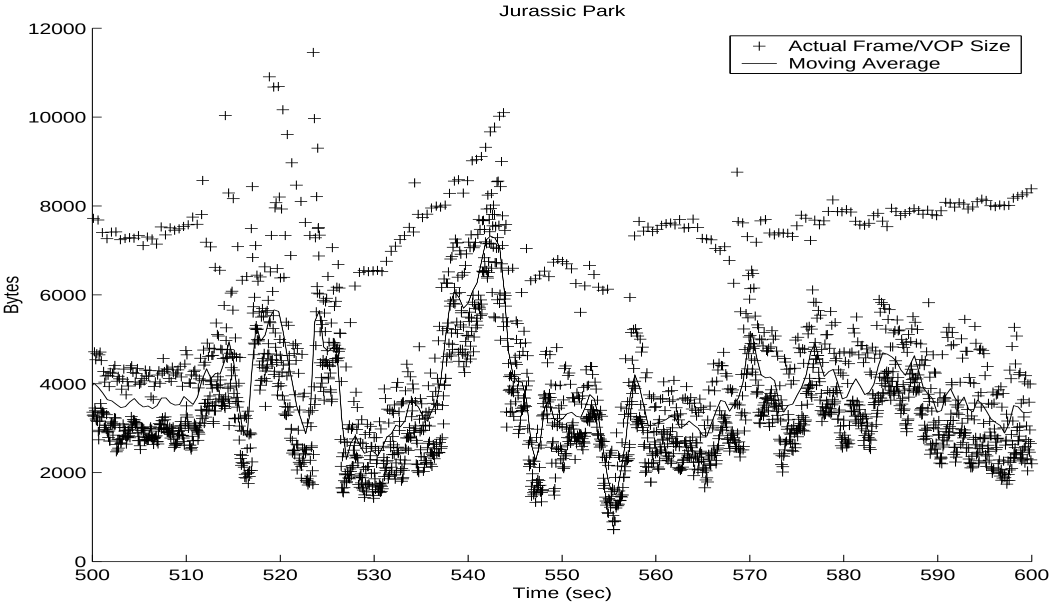

Fig. 14. Moving average time-series of *Jurassic Park*.

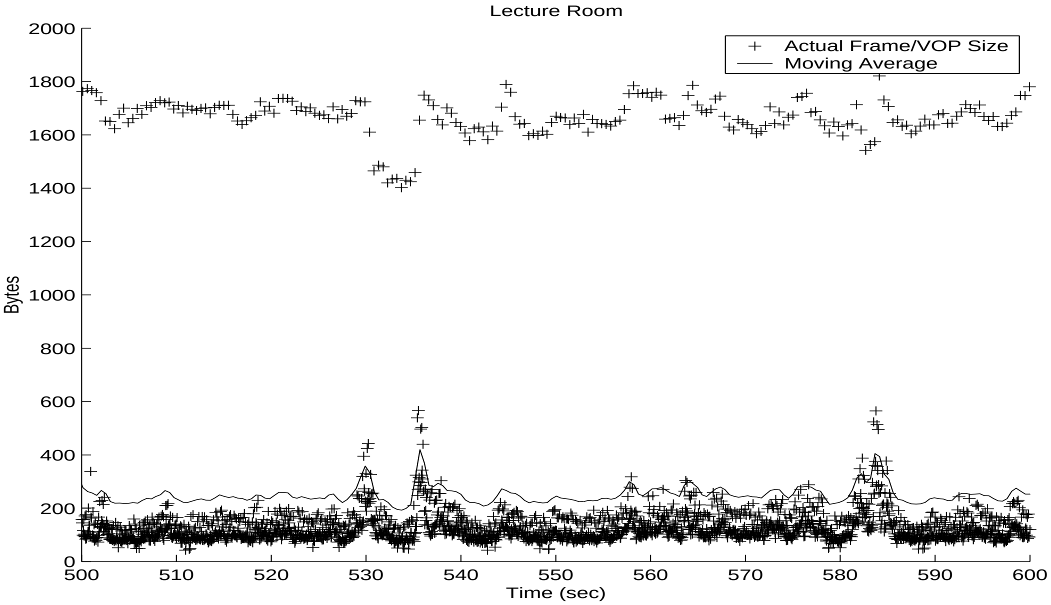

Fig. 15. Moving average time-series of *Lecture Room*.

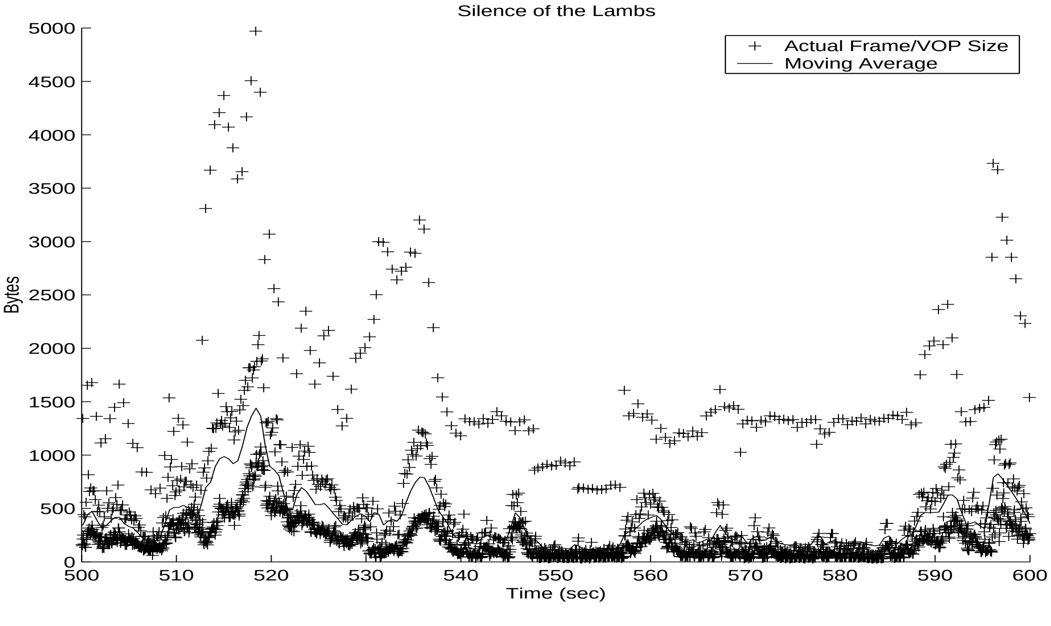

Fig. 16. Moving average time-series of *Silence of the Lambs*.

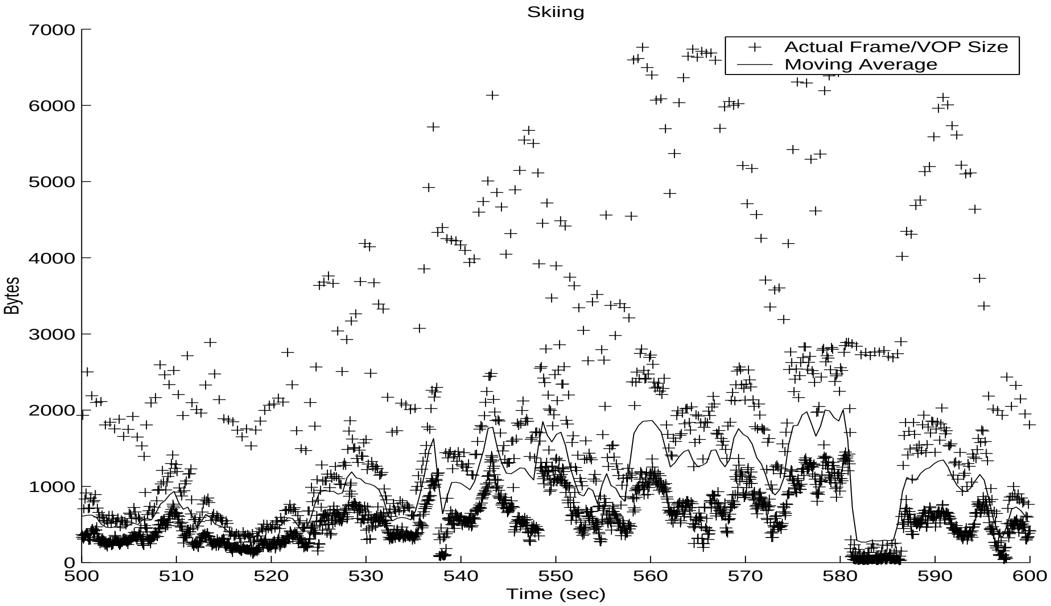

Fig. 17. Moving average time-series of *Skiing*.

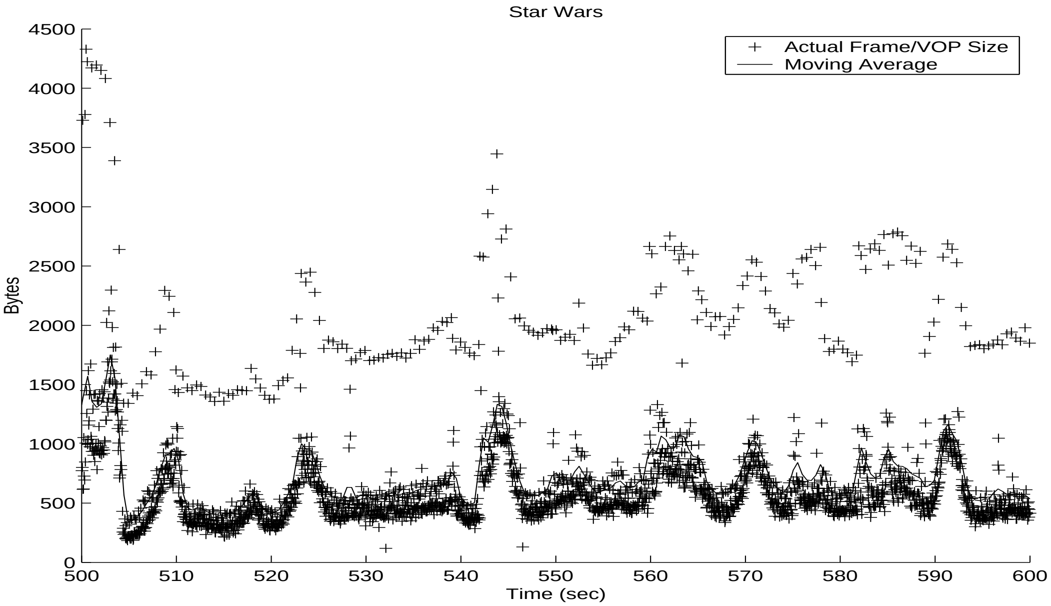

Fig. 18. Moving average time-series of *StarWars*.

F. Chapter Summary

This chapter provides a detailed analysis of the different video traces that are used in this work. Because these video traces cover a reasonably broad range of bit rates, the results of this analysis can be generalized to all MPEG-coded video data traces. The key properties of these video traces that concern this research are:

1. These traces show a significant long-range temporal dependency.

2. The time-series is extremely noisy.

It is because of these two properties that the prediction of this time-series is very difficult. One way to get across this problem it to predict the moving average of the VOP sizes instead of the original VOP sizes. In this work, the moving average is used because of two reasons:

1. The prediction of individual VOP sizes is not accurate.

2. Even in predicted accurately, the use of individual VOP size time-series as input to the source traffic controller will substantially increase the cost of the control effort.

In the following chapters, a detailed description of the predictors developed in this work is presented and their performance is analyzed.

CHAPTER V

PREDICTION OF MPEG-CODED VIDEO TRACES

In this chapter, the various predictor models developed in this research are presented. These models are developed using the empirical modeling techniques introduced in the earlier chapters. But first the definition and description of the terms, parameters and processes used in this research are presented.

A. Definitions and Descriptions

1. Performance Metrics

Here we define the performance metrics used to compare the performance of the different models developed in this research. Three types of errors were used as performance metric for the prediction schemes developed in this work. Let x_{MA} denote the moving average time-series which is to be predicted. The three performance metric are defined as:

1. Mean Square Error (MSE): MSE is the ratio between the sum of the square of the prediction error and the sum of the square of the input data. It is represented by the following equation:

$$MSE = \frac{\sum_{j=1}^{N}(x_{MA}(j) - \widehat{x}_{MA}(j))^2}{\sum_{j=1}^{N} x_{MA}(j)^2} \times 100 \tag{5.1}$$

   where N is the length of the moving average time-series, x_{MA} is the actual size of the j-th element of the moving average time-series and $\widehat{x}_{MA}$ is the prediction of the j-th element. MSE is a indicator of the overall quality of the prediction.

2. Maximum Absolute Error (MAE): MAE is the maximum error between the actual moving average of the VOP sizes and the predicted moving average of

the VOP sizes. It is given by the following equation:

$$MAE = \max_{1 \leq j \leq N} |x_{MA}(j) - \widehat{x}_{MA}(j)| \tag{5.2}$$

MAE is the maximum prediction error and provides the information about the worst case of failure of the prediction model.

3. Maximum Relative Error (MRE): MRE is the maximum of the ratio between the prediction error and the actual input data and is given by the equation:

$$MRE = \max_{1 \leq j \leq N} \frac{|x_{MA}(j) - \widehat{x}_{MA}(j)|}{|x_{MA}(j)|} \tag{5.3}$$

MRE is a measure of the relative comparison between the prediction error and the corresponding actual moving average value of the VOP size.

2. External Indicators

The present research deals with the prediction of a moving average time-series. But research has shown that the use of some external indicators along with the time-series, as inputs to the model improves the prediction process. For the problem at hand, the indicators identified by Bhattacharya et al. [30] have shown to provide significant improvement in the prediction of the moving average when the FMLP/RMLP algorithm is used. Since each video is coded differently, these indicators provide a deeper insight of the coding process. For example a large variation in the size of the $I-$VOPs is an indication of scene change. In the case of the moving average, a large variation of $I-$VOP size might not be noticed because of the averaging and hence the scene change might not be identified. Thus using the $I-$VOP size as an external indicator can improve the results. What follows next is the description of the indicators used in this research.

1. Size of the $I-$VOPs: For the video traces used in this work, each moving window ($w = 25$) has three $I-$VOPs. For the k-th moving average value, the three $I-$VOPs are denoted by $I_1(k), I_2(k)$ and $I_3(k)$ respectively.

$$\underbrace{\underbrace{I}_{I_3(k)} BBPBBPBBPBB \underbrace{I}_{I_2(k)} BBPBBPBBPBB \underbrace{I}_{I_1(k)}}_{x_{MA}(k)} \tag{5.4}$$

2. First order difference of the moving average δx_{MA}: This parameter is an indication of the gradient of the values represented by the moving average and is defined by:

$$\delta x_{MA}(k) = x_{MA}(k) - x_{MA}(k-1) \tag{5.5}$$

3. Second order difference of the moving average Δx_{MA}: This parameter provides knowledge about the direction of change in the gradient of the data points in the moving average time-series and is defined by:

$$\begin{aligned} \Delta x_{MA}(k) &= \delta x_{MA}(k) - \delta x_{MA}(k-1) \\ \Rightarrow \Delta x_{MA}(k) &= [x_{MA}(k) - x_{MA}(k-1)] - [x_{MA}(k-1) - x_{MA}(k-2)] \\ \Rightarrow \Delta x_{MA}(k) &= x_{MA}(k) - 2x_{MA}(k-1) + x_{MA}(k-2) \qquad (5.6) \end{aligned}$$

3. Scaling of the Data

For the purpose of non-linear modeling using NNs, the input data must be scaled to lie within certain bounds. The scaling of input data is a very important aspect of training the network. In order to prevent the saturation of nodes in NNs, the input data is forced to lie between -0.5 and 0.5. The activation function used in the development of NNs for this work is of the form:

$$\sigma(v) = \frac{(1.0 - e^{av})}{(1.0 + e^{av})} \tag{5.7}$$

where a is the slope parameter of the *hyperbolic tangent*(tanh) function and v is the internal activity level of a neuron. The values of $\sigma(v)$ range from -1 to 1. It is this sigmoid function which is responsible for the saturation of the nodes. The scaling of the data should be independent of the encoding of the video traces [34]. The scaling factor is determined using the following equation:

$$\widehat{b} = \alpha \frac{B_{max}}{f} \tag{5.8}$$

where $\widehat{b}$ is the scaling factor, B_{max} is the maximum of the Mean Bit Rate of the video traces used in this research, f is the frames rate in frames/sec and α is the multiplicative factor determined by the size of the maximum of the moving average of the VOP sizes in comparison with mean. The selection of α should be such that it takes care of the entire range of bit rates of the MPEG-coded videos for which this predictor is used.

4. Post Processing

For the non-linear models that were developed in this research, it was observed that the predicted data could capture the trends of the moving average time-series to some extent, but there was a significant offset from the original time-series. It was found to be true for all non-linear models irrespective of the algorithm used. The tweaking of modeling parameters also did not help. Therefore, the post processing scheme developed in [34] was used. In this scheme, the average of the errors between the predicted and the actual moving average time-series is added to all the predicted outputs. This is given by the following equation:

$$\widehat{x}_{MA}(k|k-p) = \widehat{x}^*_{MA}(k|k-p) + \frac{1}{(k-p)}\sum_{j=1}^{k-p}(x_{MA}(j) - \widehat{x}_{MA}(j)) \tag{5.9}$$

where p is the number of time step predicted ahead in the future, i.e. the prediction horizon, $\widehat{x}_{MA}(k|k-p)$ is the final predicted size of moving average of the VOPs at time step k after the post processing, $\widehat{x}^*_{MA}(k|k-p)$ is the output from the predictor, and $x_{MA}(j)$ is the actual size of moving average VOPs at time step j.

B. Single-step-ahead Prediction

1. Prediction of I-VOPs

The single-step-ahead prediction (SSP) of individual I-VOPs means predicting 0.48 seconds ahead. The training was done using the first 1500 points of the I-VOP size time-series of video trace *Aladdin*. The next 500 points were used for the validation of the model. The developed model was then used to generate SSP for the entire length for all the eight video traces.

a. SSP Using ARX Models

In the design of ARX models, external indicators were used as inputs to the predictor. The indicators used were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. Figures 19 to 22 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table III shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

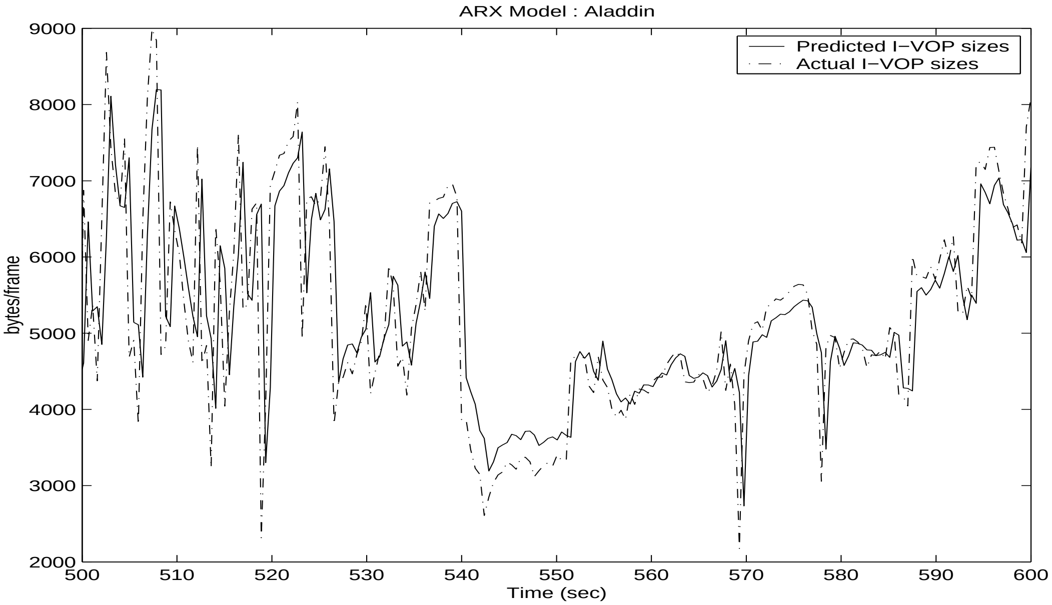

Fig. 19. SSP sizes of the I-VOPs of *Aladdin* using ARX models.

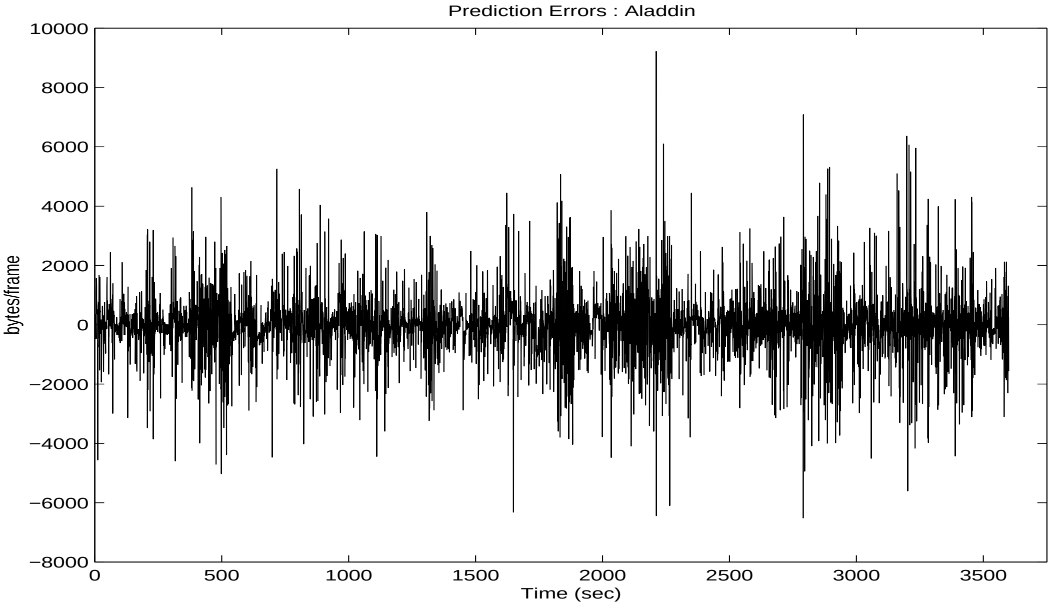

Fig. 20. SSP errors of I-VOP sizes of *Aladdin* using ARX models.

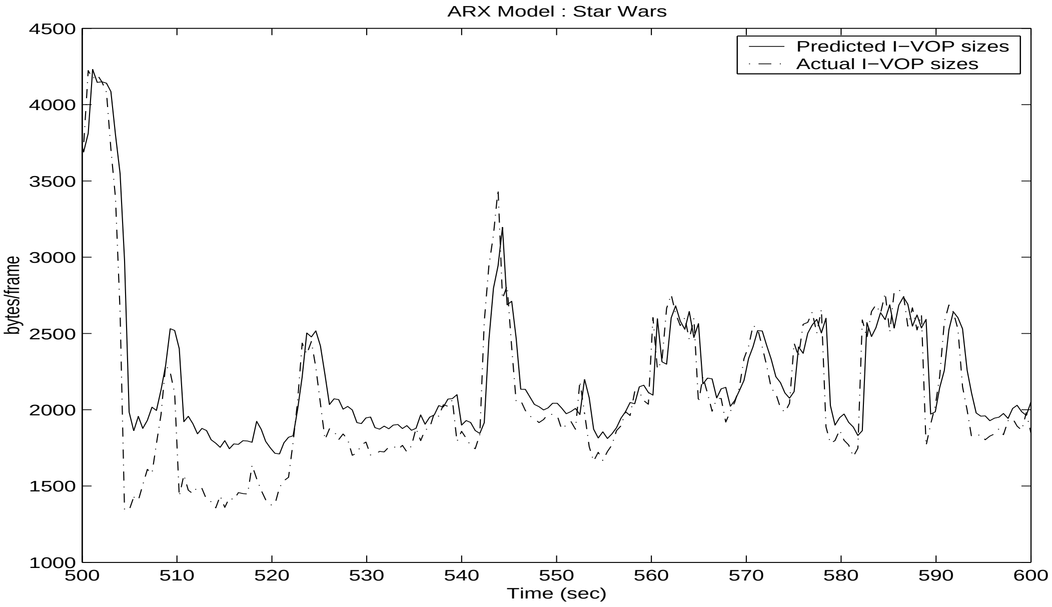

Fig. 21. SSP sizes of the I-VOPs of *StarWars* using ARX models.

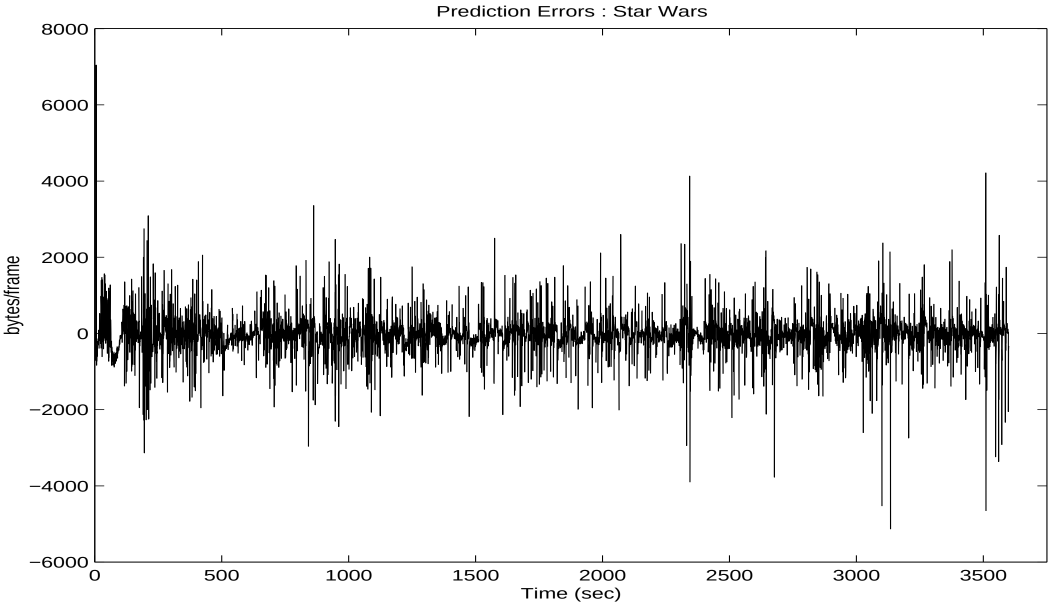

Fig. 22. SSP errors of I-VOP sizes of *StarWars* using ARX models.

Table III. Performance metrics of the I-VOP for SSP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	3.1	9216.9	16.1
ARD Talk	0.8	5605.5	2.1
Die Hard III	3.4	5040.7	11.4
Jurassic Park I	1.1	9570.2	8.2
Lecture Room	0.2	703.6	0.6
Silence of the Lambs	3.3	8070.5	11.2
Skiing	2.2	4716.2	2.5
StarWars	1.7	7042.8	7.8

b. SSP Using ESN Models

The network structure which gave the best results was $8 - 30 - 1$. The inputs were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. Figures 23 to 26 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table IV shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

2. Prediction of Moving Average of VOPs

The single-step-ahead prediction (SSP) represents a time horizon of 0.48 seconds.In this work, three separate prediction models are developed for SSP. The training was done using the first 1500 points of the moving average time-series of VOP sizes of video trace *Aladdin.* The next 500 points were used for the validation. The developed

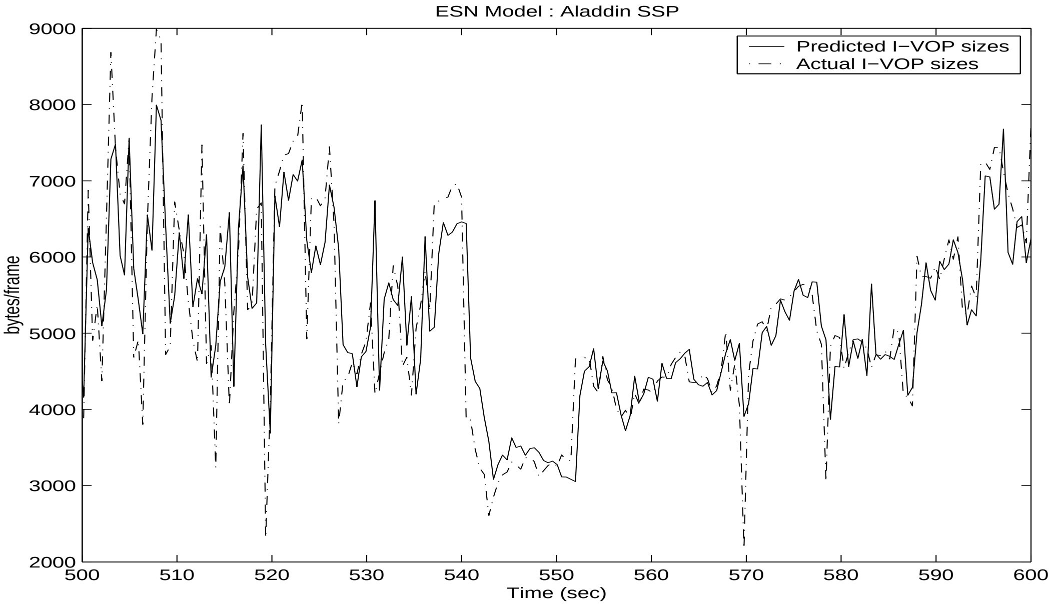

Fig. 23. SSP sizes of the I-VOPs of *Aladdin* using ESN models.

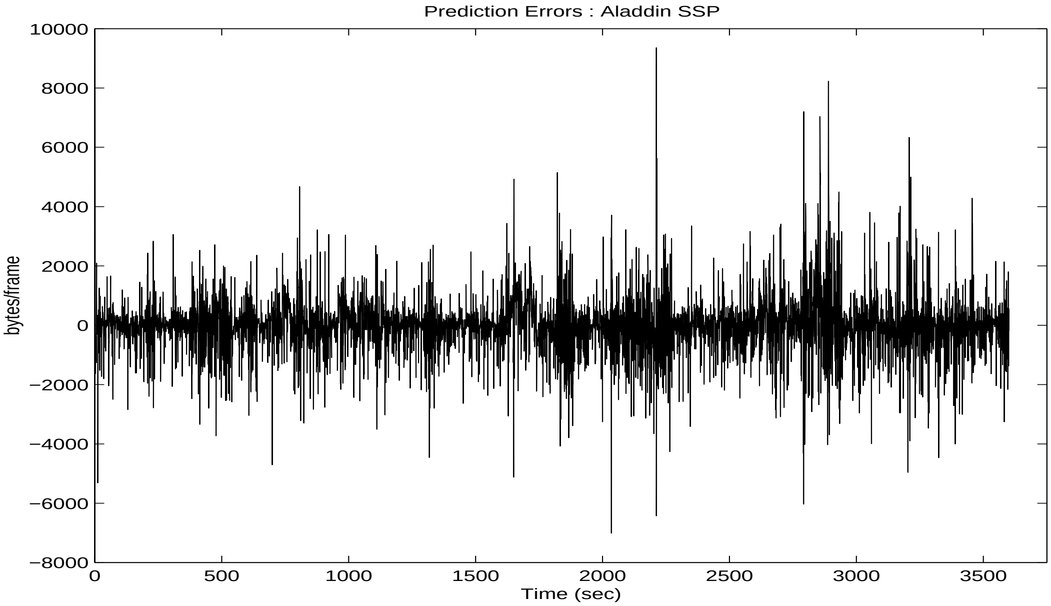

Fig. 24. SSP errors of I-VOP sizes of *Aladdin* using ESN models.

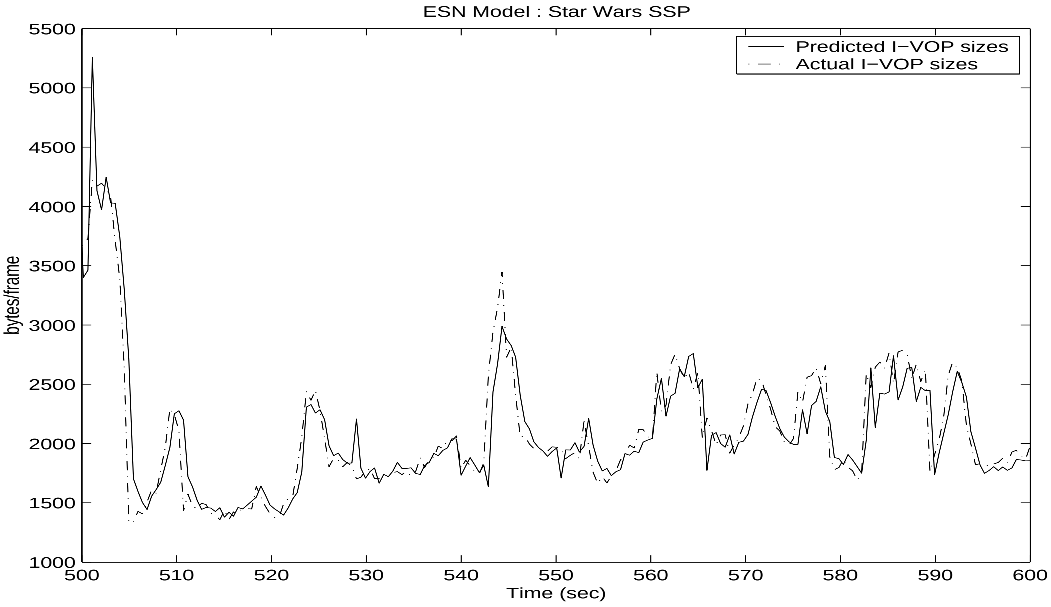

Fig. 25. SSP sizes of the I-VOPs of *StarWars* using ESN models.

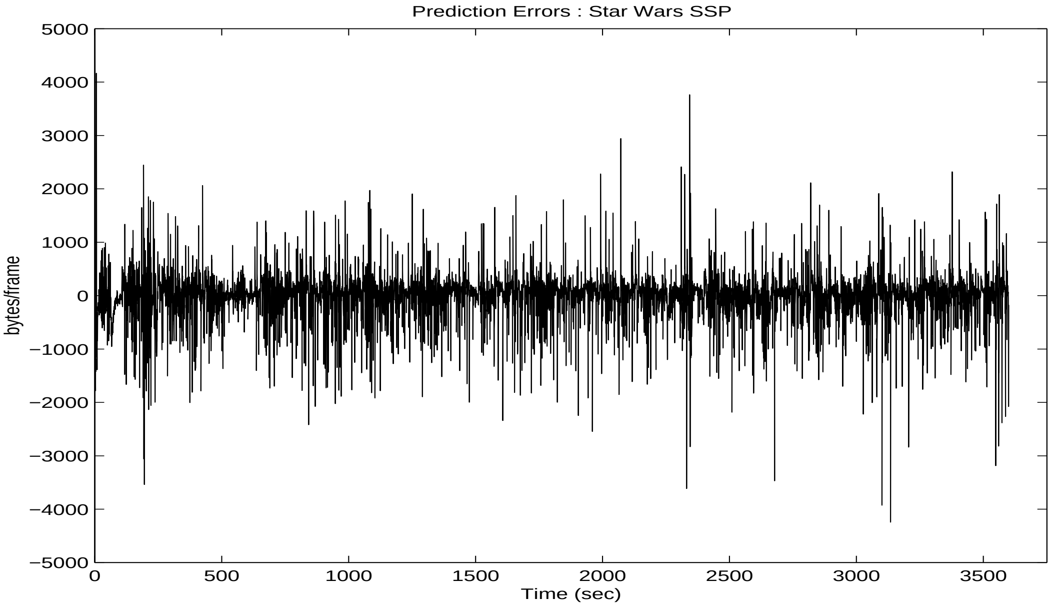

Fig. 26. SSP errors of I-VOP sizes of *StarWars* using ESN models.

Table IV. Performance metrics of the I-VOP for SSP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	3.0	9358.8	13.3
ARD Talk	1.6	8714.4	1.9
Die Hard III	3.0	4699.8	11.7
Jurassic Park I	3.8	10763	7.3
Lecture Room	0.1	1162.3	0.8
Silence of the Lambs	3.8	6765.9	10.6
Skiing	1.9	5183.5	2.6
StarWars	1.5	4403.6	6.5

model was then used to generate SSP for the entire length for all the eight video traces.

a. SSP Using AR Models

The AR model that gives the best results for SSP is the one with 17 lags. Figures 27 to 30 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table V shows the performance of the AR predictor for the video traces used in this research in terms of the performance metrics defined earlier in the chapter.

b. SSP Using ARX Models

In the design of ARX models, external indicators were used as inputs to the predictor. The inputs that were used are x_{MA}, δx_{MA}, Δx_{MA} and $I-$VOPs respectively. The ARX model which gave the best results for SSP had the following structure: $n_y = 2$,

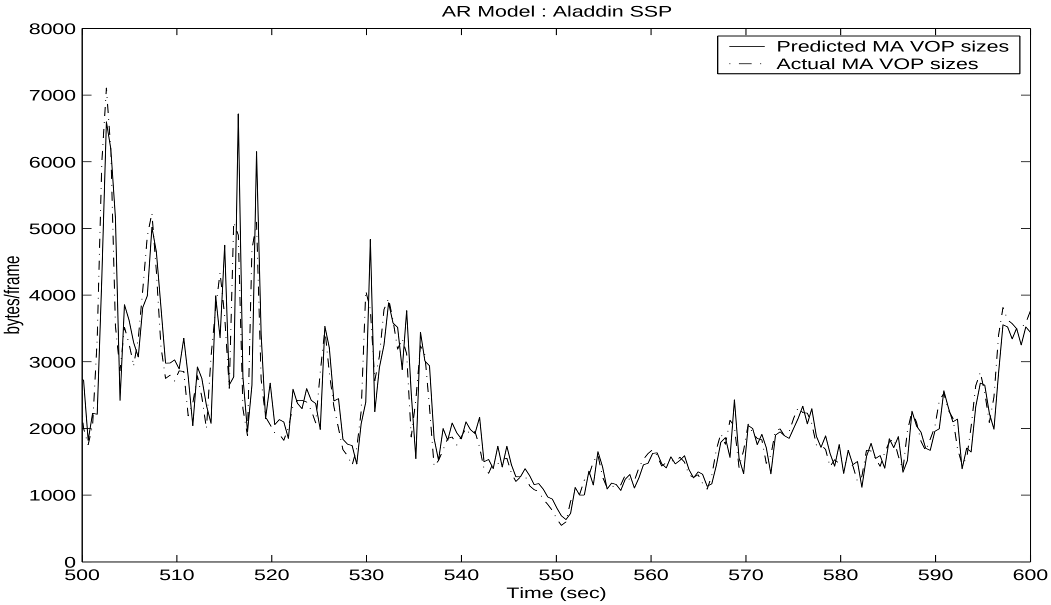

Fig. 27. SSP sizes of the moving average VOPs of *Aladdin* using AR models.

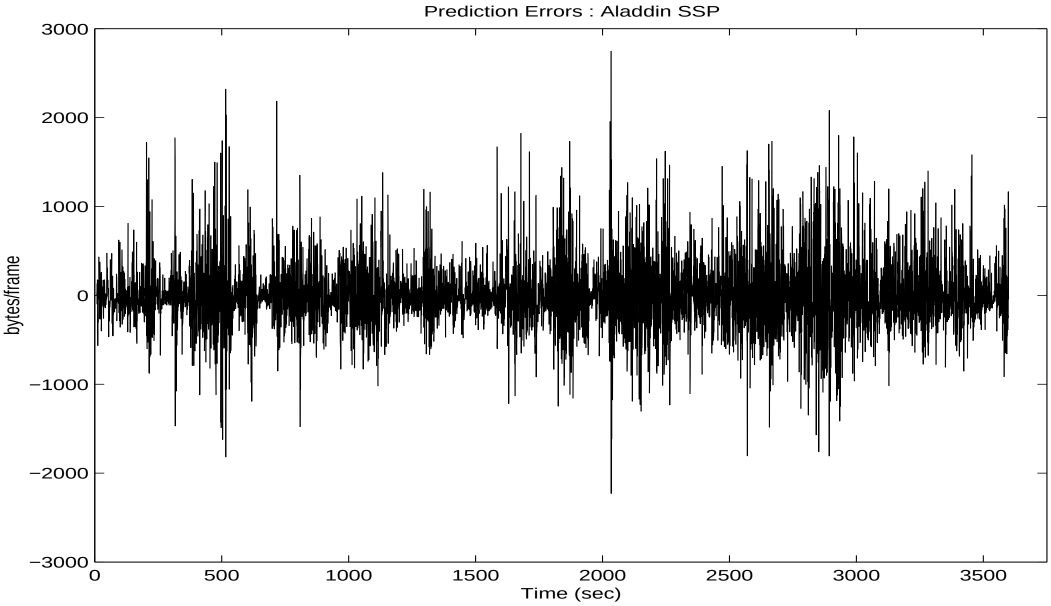

Fig. 28. SSP errors of moving average VOP sizes of *Aladdin* using AR models.

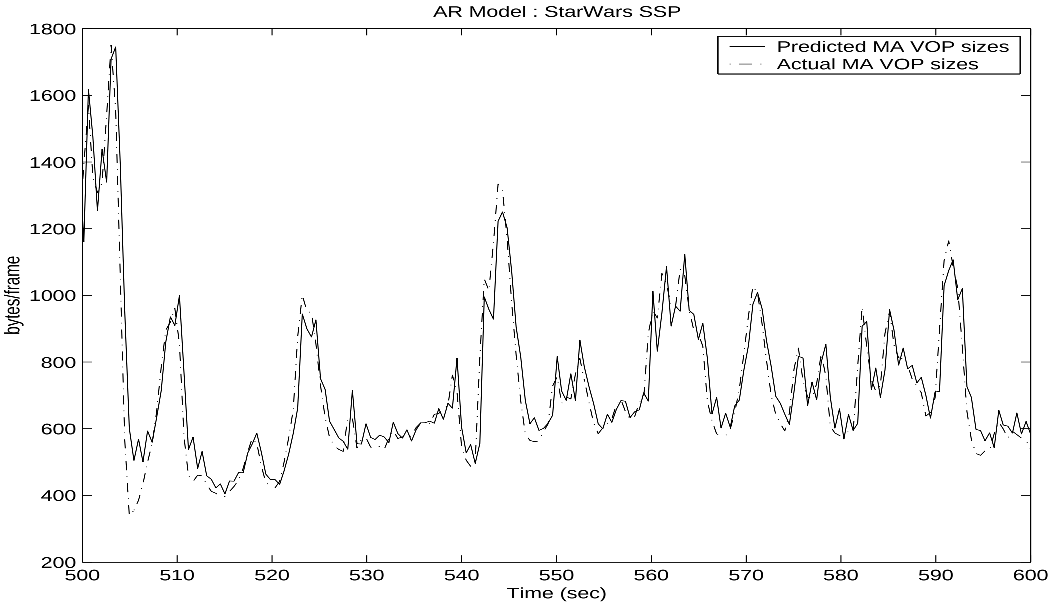

Fig. 29. SSP sizes of the moving average VOPs of *StarWars* using AR models.

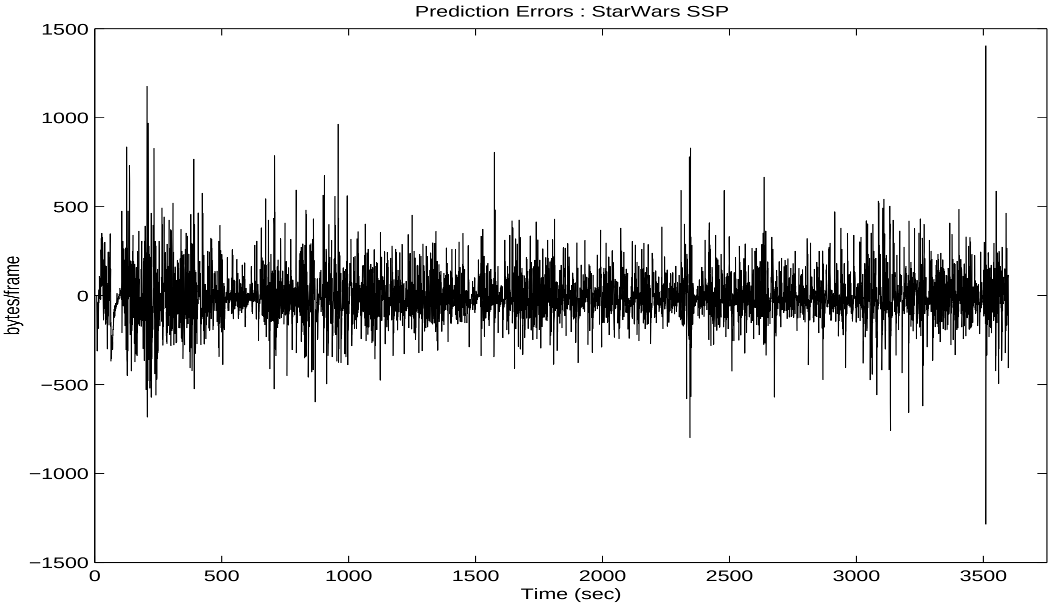

Fig. 30. SSP errors of moving average VOP sizes of *StarWars* using AR models.

Table V. Performance metrics of the SSP for AR models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	3.2	3476.9	4.3
ARD Talk	0.8	2416.5	1.5
Die Hard III	2.3	2508.7	1.5
Jurassic Park I	1.2	4643.6	1.2
Lecture Room	0.8	272.9	0.4
Silence of the Lambs	3.2	1665.6	2.9
Skiing	2.1	777.2	1.4
StarWars	1.5	1404.0	1.6

$n_u = [1111]$ and $n_k = [1111]$. Figures 31 to 34 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table VI shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

c. SSP Using ESN Models

For the ESN model, the moving average time-series x_{MA} along with the external indicators δx_{MA}, Δx_{MA} and $I-$VOPs were given as inputs. The network structure which gave the best results was $4 - 40 - 1$. In order to predict the moving average of the VOP sizes at time step k the four inputs were $x_{MA}(k-1)$, $\delta x_{MA}(k-1)$, $\Delta x_{MA}(k-1)$ and $I_1(k-1)$. Figures 35 to 38 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

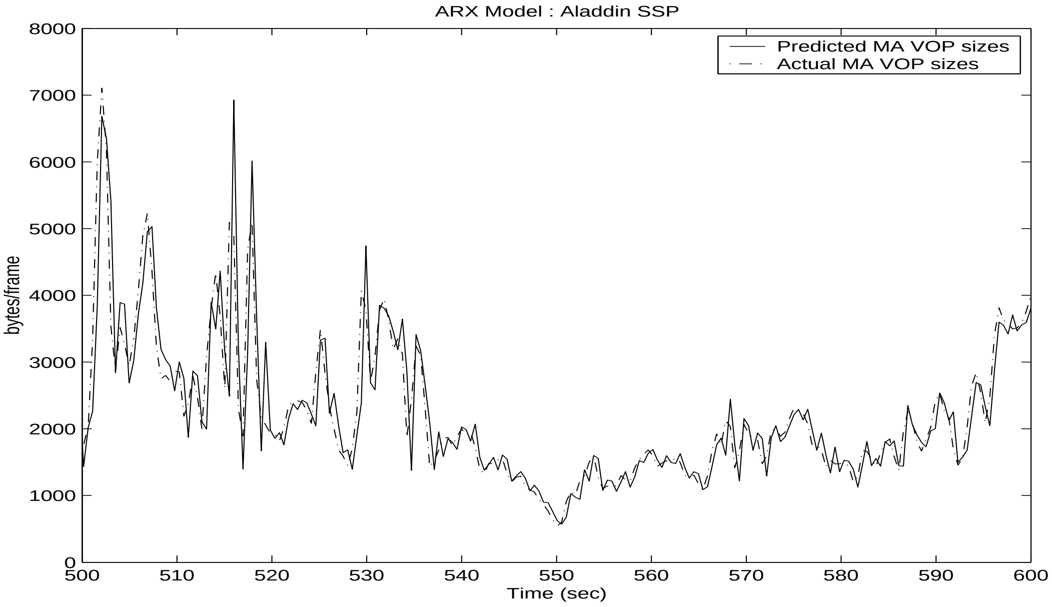

Fig. 31. SSP sizes of the moving average VOPs of *Aladdin* using ARX models.

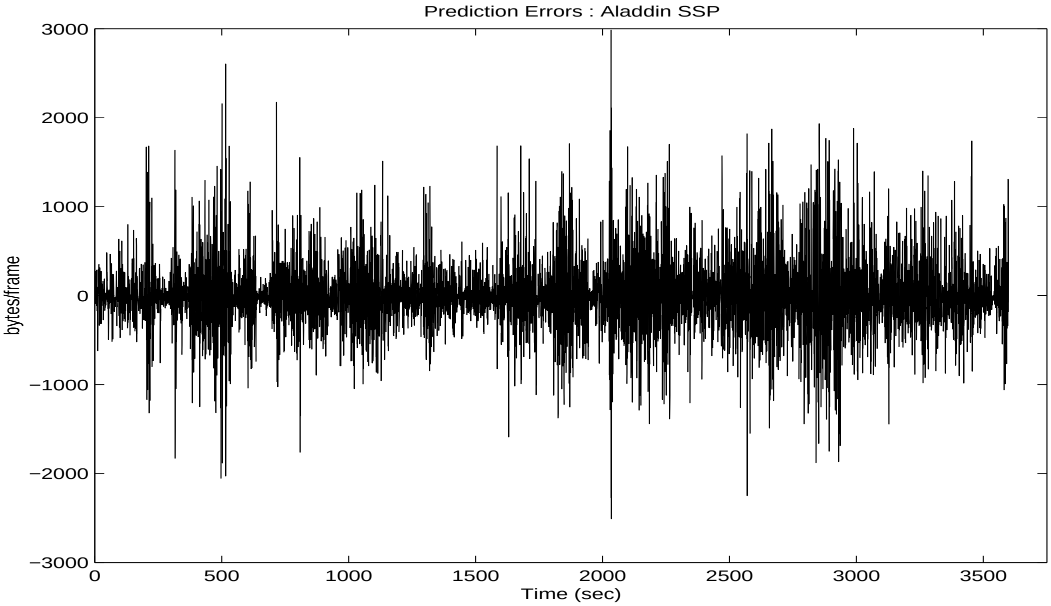

Fig. 32. SSP errors of moving average VOP sizes of *Aladdin* using ARX models.

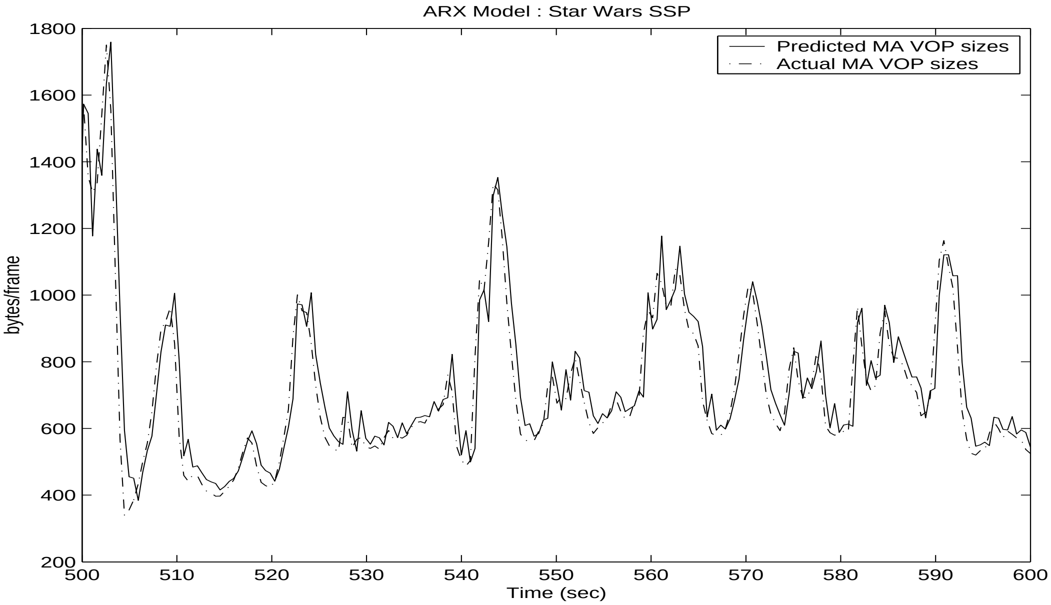

Fig. 33. SSP sizes of the moving average VOPs of *StarWars* using ARX models.

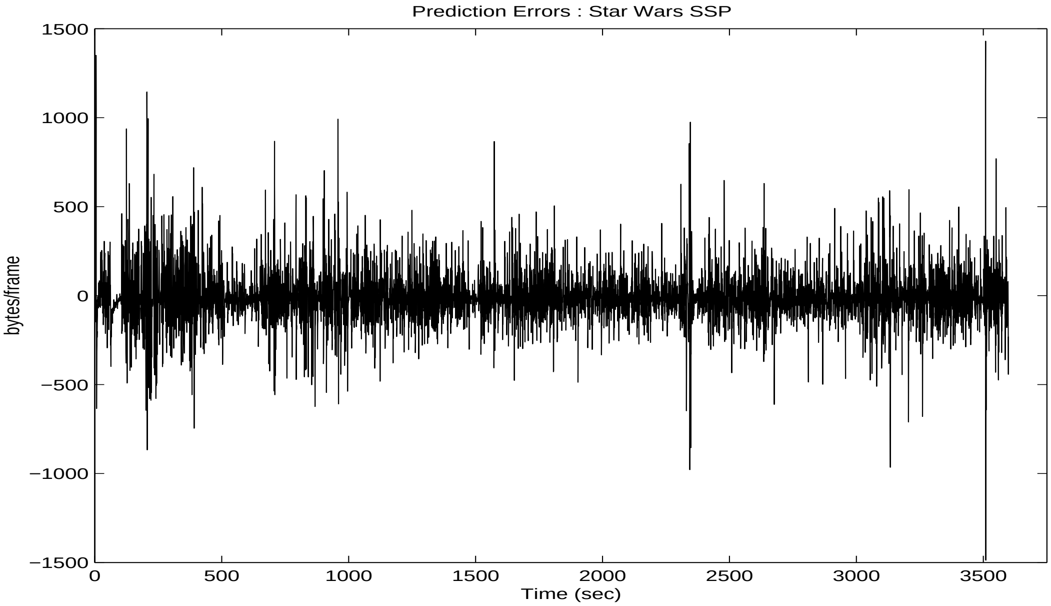

Fig. 34. SSP errors of moving average VOP sizes of *StarWars* using ARX models.

Table VI. Performance metrics of the SSP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	3.5	3984.1	4.8
ARD Talk	1.1	2514.3	1.5
Die Hard III	2.4	2827.5	1.8
Jurassic Park I	1.8	4633.1	1.0
Lecture Room	0.9	262.7	0.5
Silence of the Lambs	3.3	1903.6	2.7
Skiing	2.2	880.2	2.0
StarWars	2.0	1486.7	1.8

Table VII shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

C. Two-step-ahead Prediction

1. Prediction of I-VOPs

The two-step-ahead prediction (Two-SP) of individual I-VOPs means predicting 0.96 seconds ahead. The training was done using the first 1500 points of the I-VOP size time-series of video trace *Aladdin*. The next 500 points were used for the validation of the model.

a. Two-step-ahead Prediction Using ARX Models

The indicators used for the ARX model were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. The model structure which gives the best results is $n_y = 3$, $n_u = 1$ and $n_k = 1$.

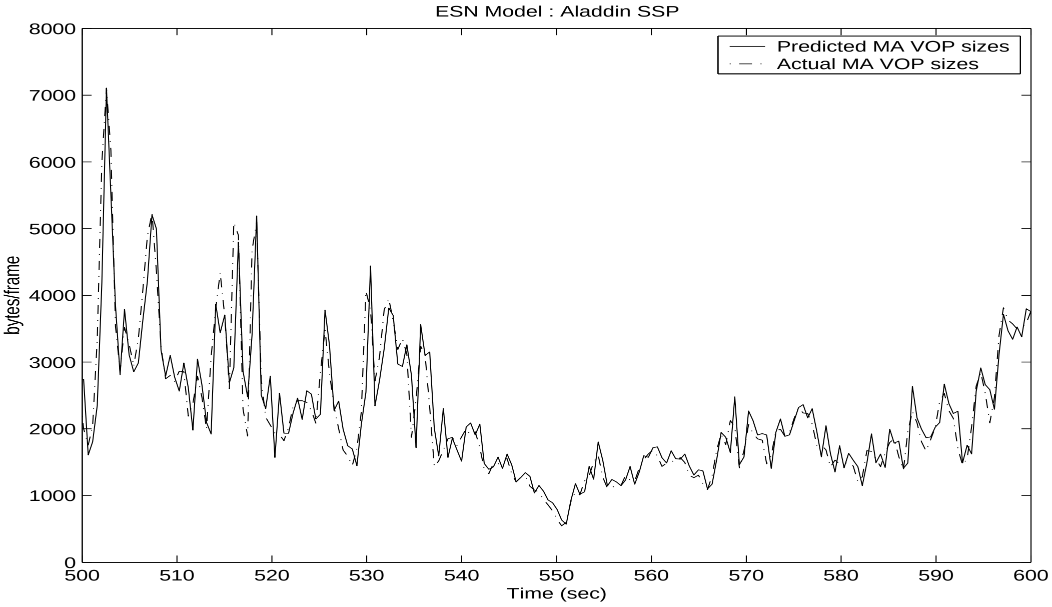

Fig. 35. SSP sizes of the moving average VOPs of *Aladdin* using ESN models.

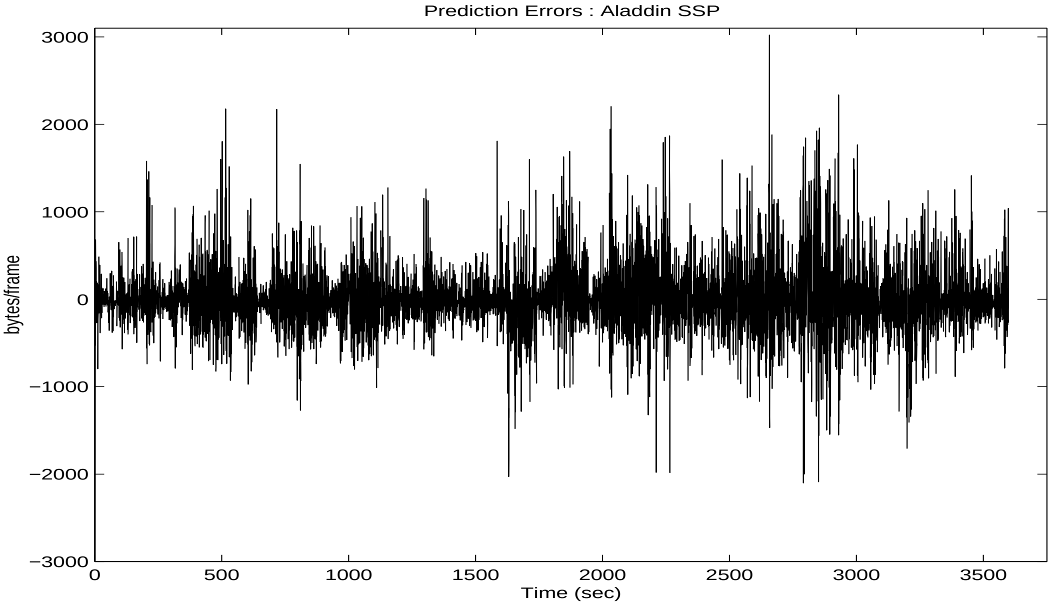

Fig. 36. SSP errors of moving average VOP sizes of *Aladdin* using ESN models.

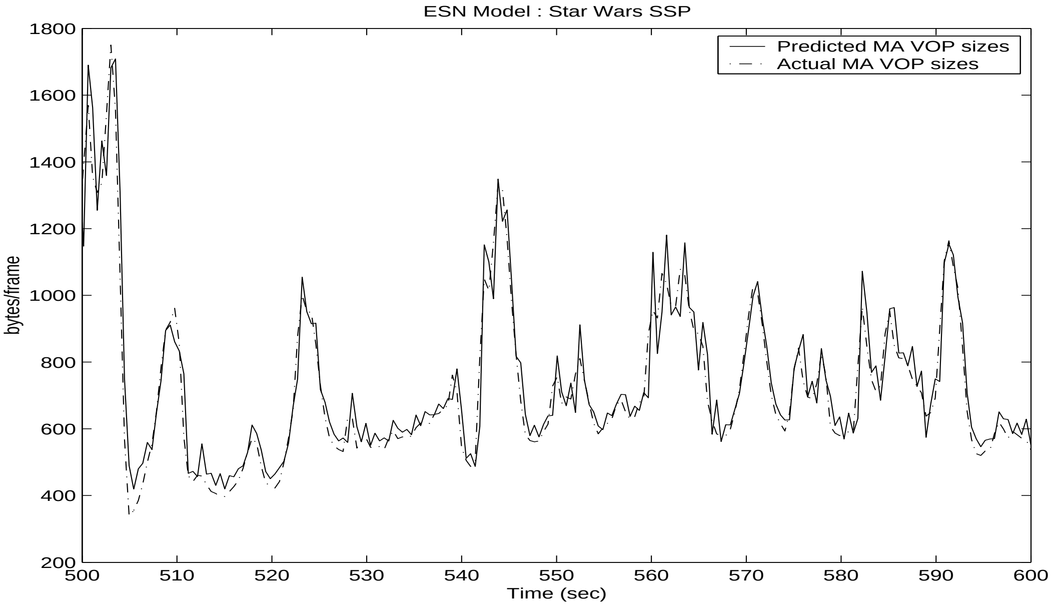

Fig. 37. SSP sizes of the moving average VOPs of *StarWars* using ESN models.

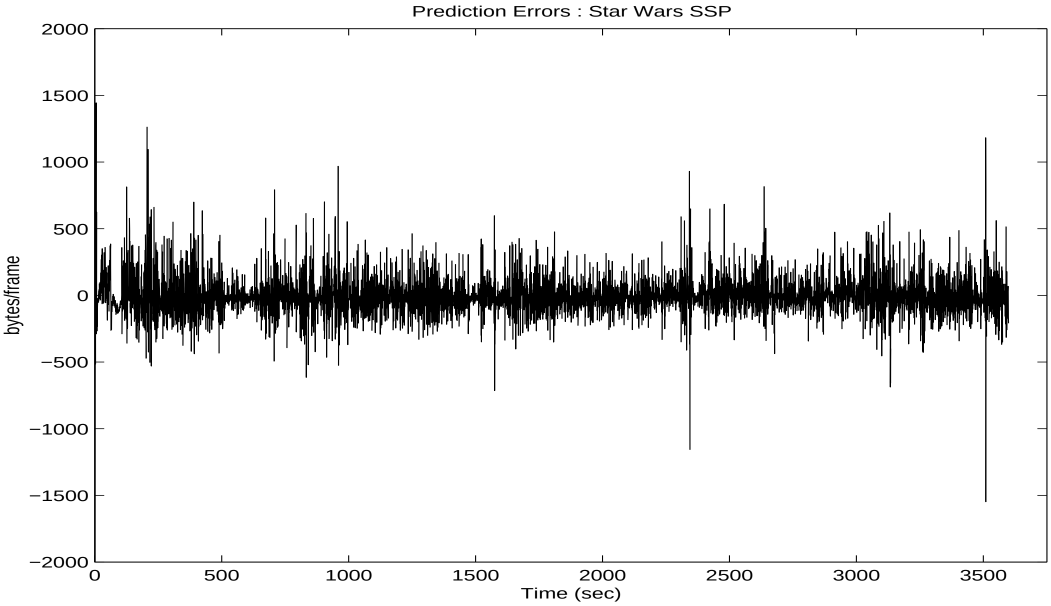

Fig. 38. SSP errors of moving average VOP sizes of *StarWars* using ESN models.

Table VII. Performance metrics of the SSP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	2.3	3022.0	9.9
ARD Talk	1.4	5245.8	0.8
Die Hard III	1.4	1354.4	2.4
Jurassic Park I	2.1	11956	2.2
Lecture Room	1.3	280.1	0.6
Silence of the Lambs	2.7	1738.2	2.8
Skiing	1.5	1049.6	2.5
StarWars	0.8	1547.6	2.2

Figures 39 to 42 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table VIII shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

b. Two-step-ahead Prediction Using ESN Models

The network structure which gave the best results was $8 - 25 - 1$. The inputs and indicators used were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. Figures 43 to 46 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table IX shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

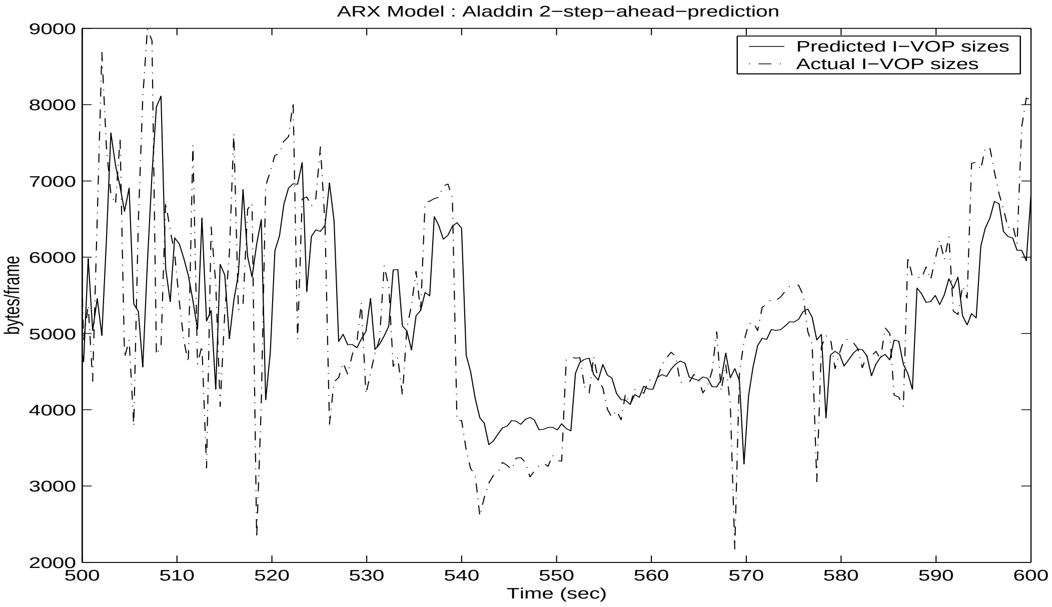

Fig. 39. Two-SP sizes of the I-VOPs of *Aladdin* using ARX models.

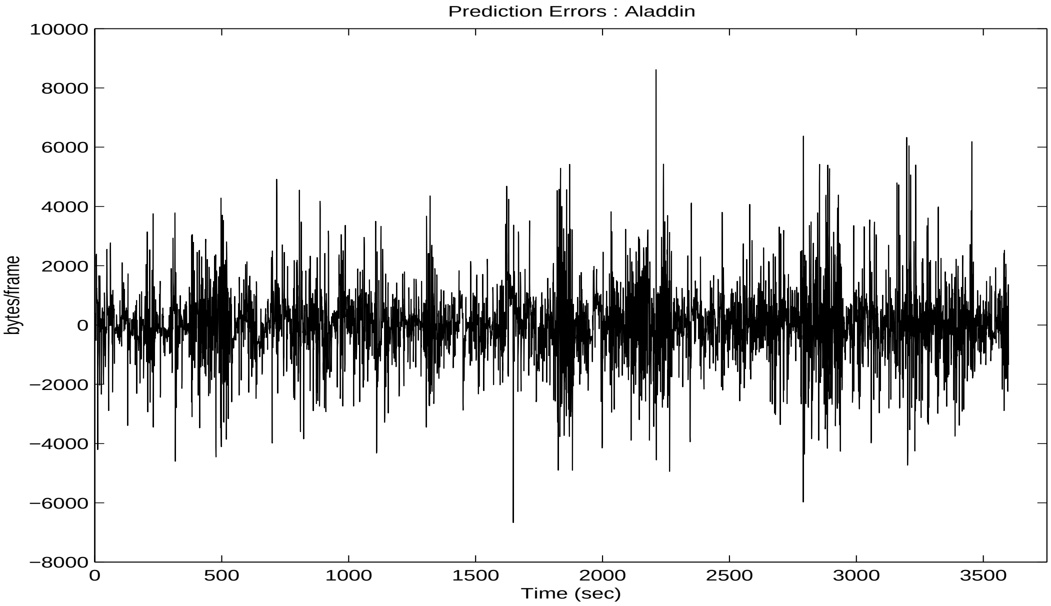

Fig. 40. Two-SP errors of I-VOP sizes of *Aladdin* using ARX models.

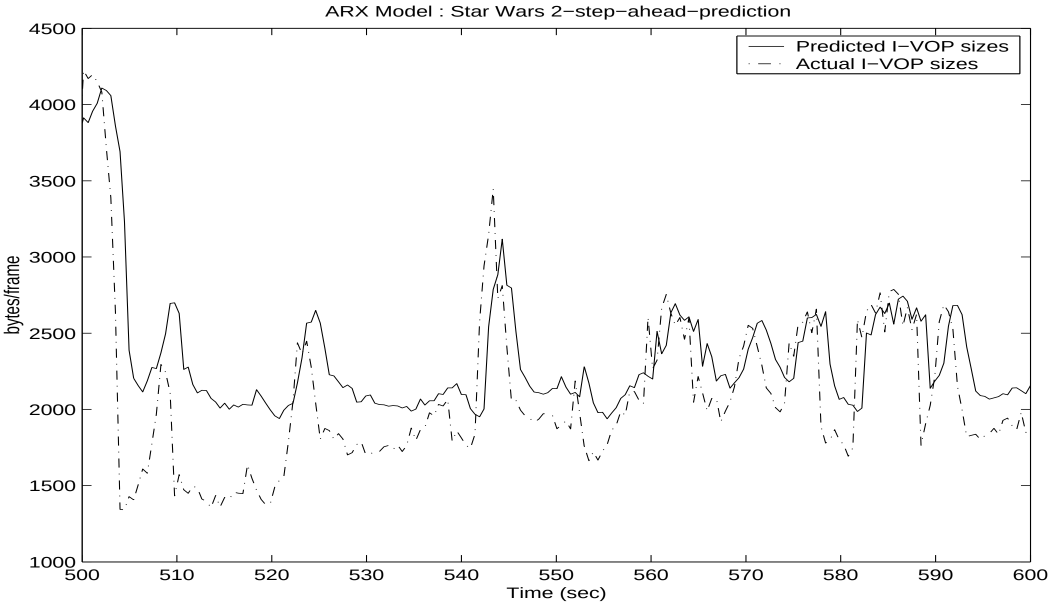

Fig. 41. Two-SP sizes of the I-VOPs of *StarWars* using ARX models.

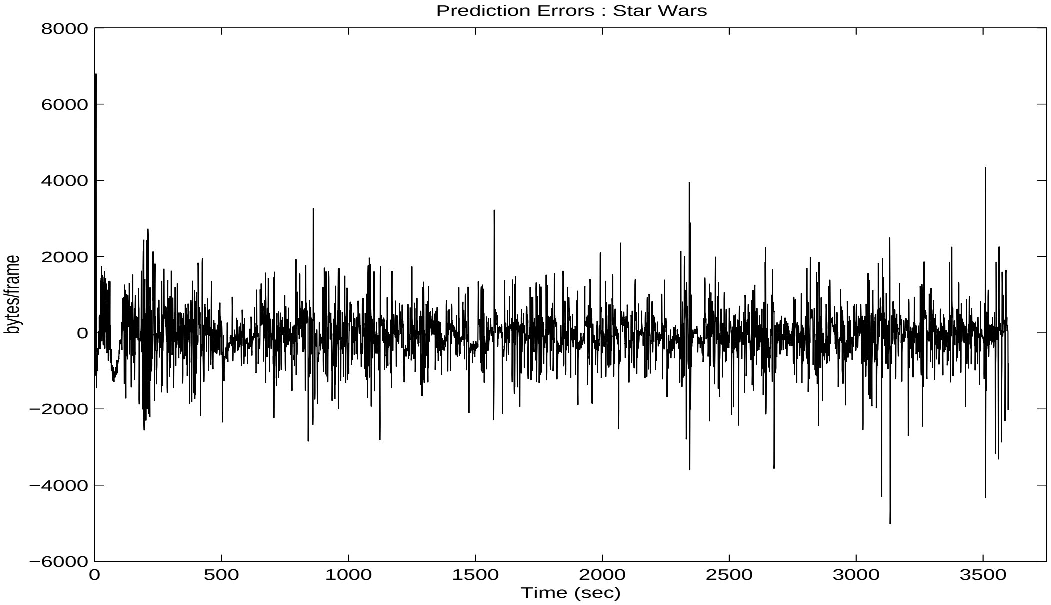

Fig. 42. Two-SP errors of I-VOP sizes of *StarWars* using ARX models.

Table VIII. Performance metrics of the I-VOP for the two-SP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	4.8	8616.1	16.9
ARD Talk	1.7	5877.4	2.1
Die Hard III	5.6	4898.1	11.1
Jurassic Park I	2.0	9625.2	7.9
Lecture Room	0.7	1192.0	1.1
Silence of the Lambs	6.9	7892.0	9.1
Skiing	4.3	4713.1	2.2
StarWars	3.0	6797.5	7.7

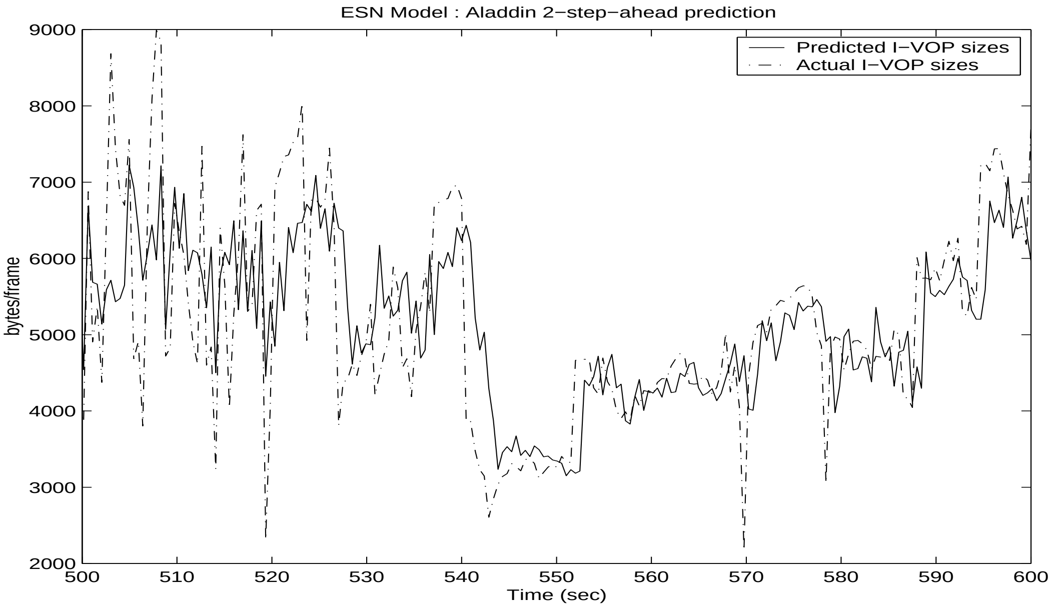

Fig. 43. Two-SP sizes of the I-VOPs of *Aladdin* using ESN models.

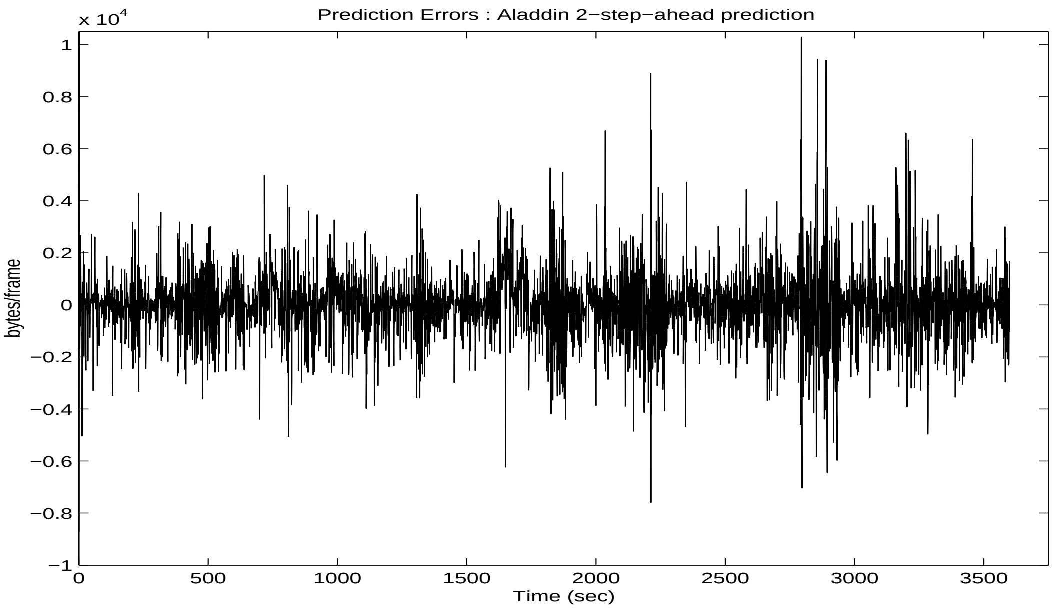

Fig. 44. Two-SP errors of I-VOP sizes of *Aladdin* using ESN models.

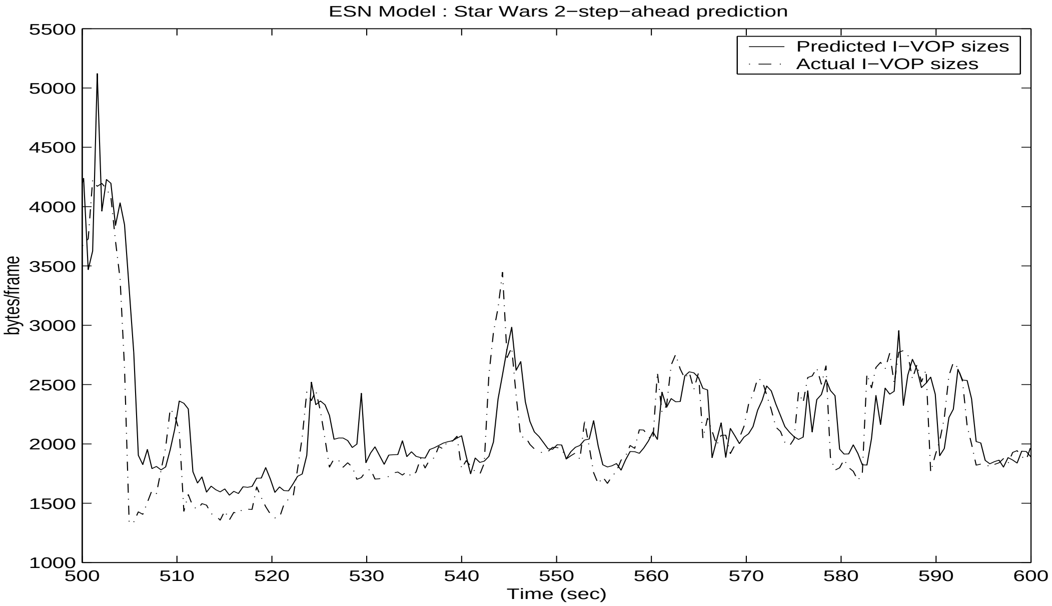

Fig. 45. Two-SP sizes of the I-VOPs of *StarWars* using ESN models.

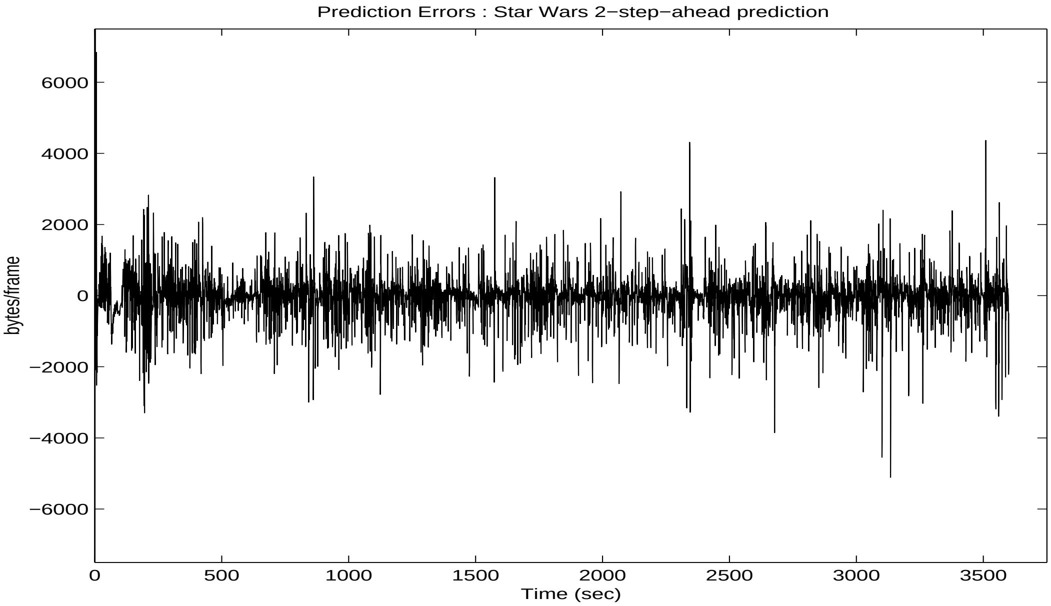

Fig. 46. Two-SP errors of I-VOP sizes of *StarWars* using ESN models.

Table IX. Performance metrics of the I-VOP for the two-SP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	5.4	10294	15.8
ARD Talk	2.8	9349.4	1.6
Die Hard III	5.4	5021.1	11.4
Jurassic Park I	6.3	12426	7.4
Lecture Room	0.2	1110.8	0.8
Silence of the Lambs	7.0	9103.6	9.7
Skiing	3.7	5080.3	2.5
StarWars	2.7	7429.5	7.8

2. Prediction of Moving Average of VOPs

The two-step-ahead prediction (Two-SP) represents a time horizon of 0.96 seconds. For the two -step-ahead prediction, training was done using the first 1500 points of the moving average time-series of VOP sizes of video trace Aladdin. The next 500 points were used for the validation of the model. The developed model was then used to generate two-step-ahead predictions for all eight video traces.

a. Two-step-ahead Prediction Using AR Models

The AR model used for the two-step-ahead prediction was the same model that was developed for SSP. For this model, $n_y = 17$. Figures 47 to 50 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

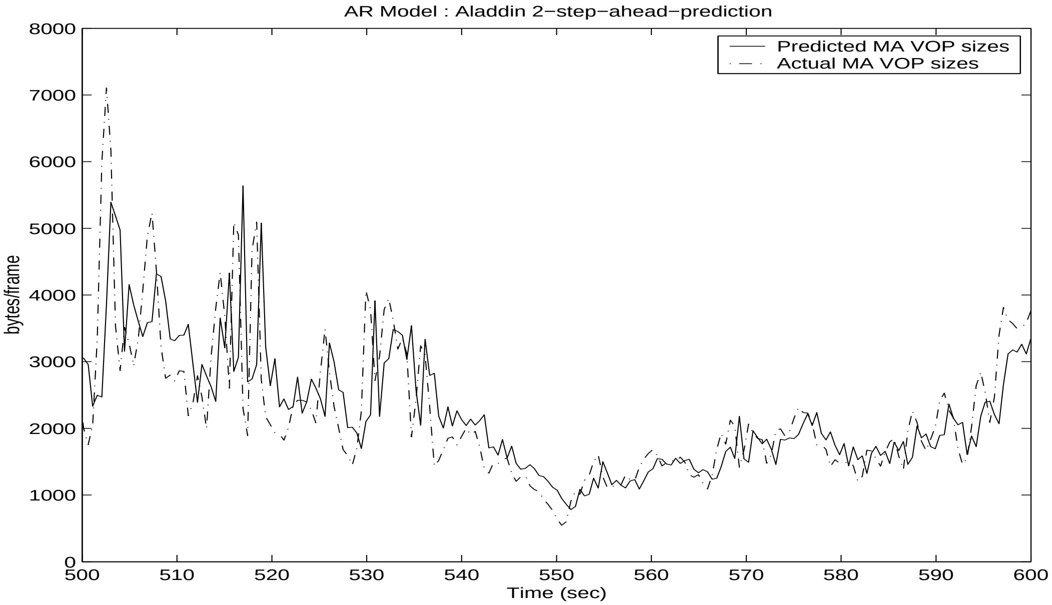

Fig. 47. Two-SP sizes of the moving average VOPs of *Aladdin* using AR models.

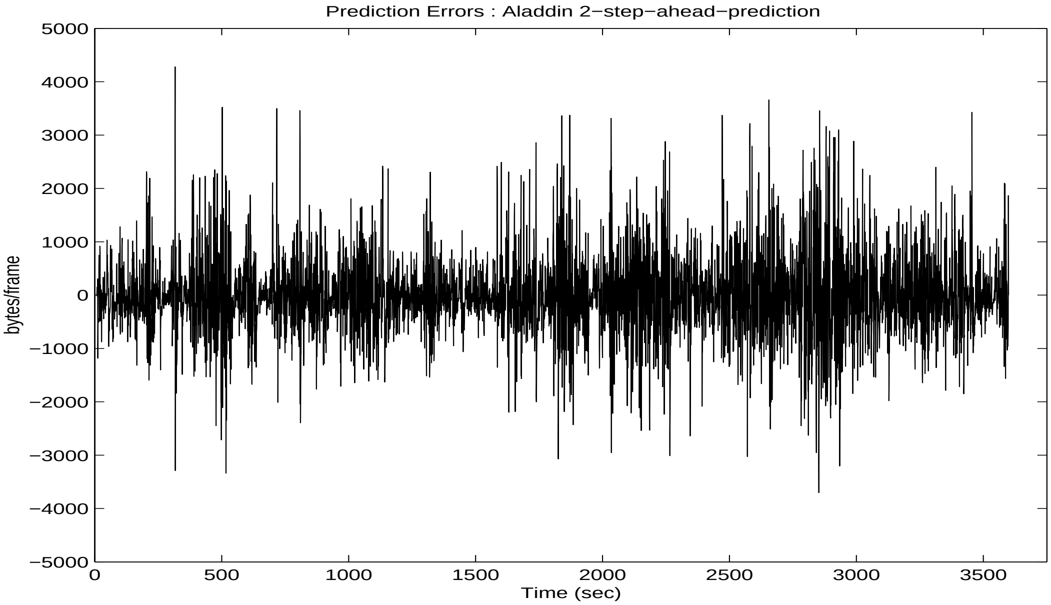

Fig. 48. Two-SP errors of moving average VOP sizes of *Aladdin* using AR models.

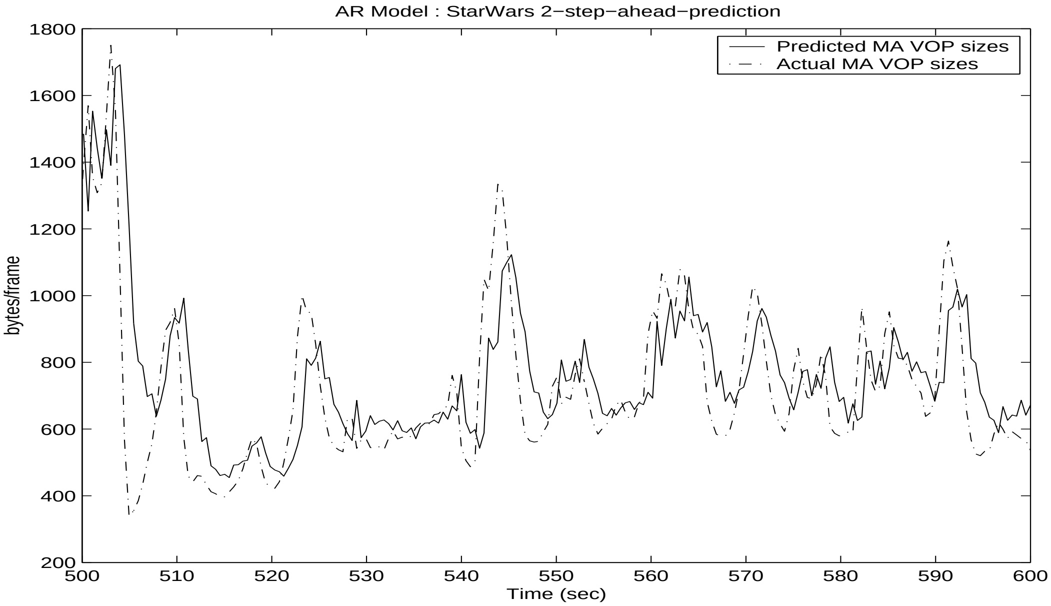

Fig. 49. Two-SP sizes of the moving average VOPs of *StarWars* using AR models.

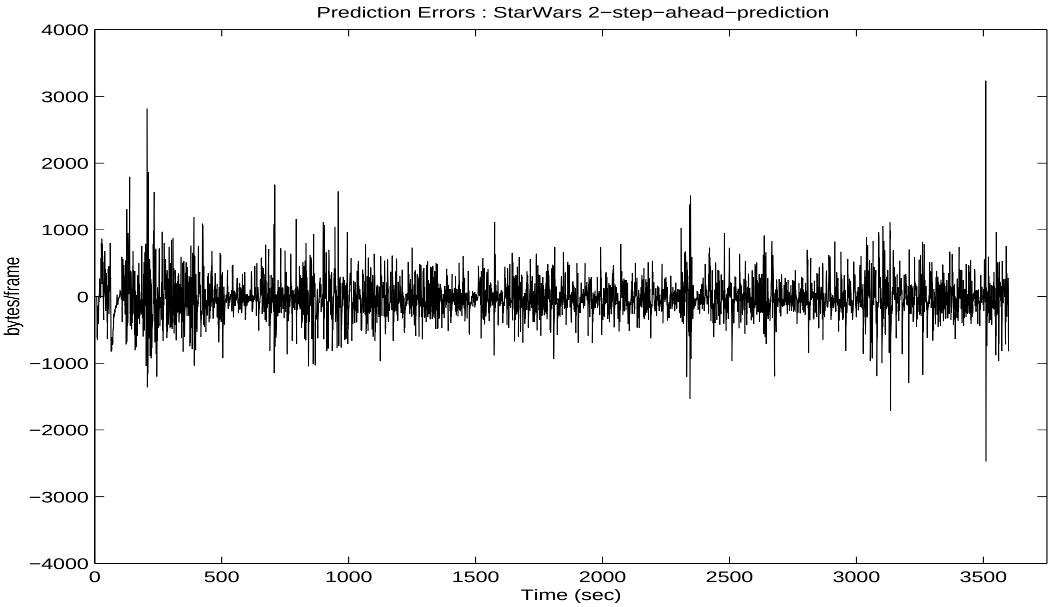

Fig. 50. Two-SP errors of moving average VOP sizes of *StarWars* using AR models.

Table X shows the performance of the AR predictor for the video traces used in this research in terms of the performance metrics defined earlier in the chapter.

b. Two-step-ahead Prediction Using ARX Models

In the design of ARX models for two-step-ahead prediction, other than the time-series x_{MA}, external indicators δx_{MA}, Δx_{MA} and $I-$VOPs were also used as additional inputs. The model which gave the best results had the structure: $n_y = 3$, $n_u = [1111]$ and $n_k = [1111]$. Figures 51 to 54 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XI shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

Table X. Performance metrics of the two-SP for AR models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	9.2	9147.9	21.2
ARD Talk	3.1	7573.4	3.8
Die Hard III	7.4	9132.0	20.4
Jurassic Park I	4.7	9275.9	13.8
Lecture Room	13.8	8809.0	11.4
Silence of the Lambs	8.8	7380.7	5.3
Skiing	6.3	1504.8	5.6
StarWars	4.5	3232.8	7.2

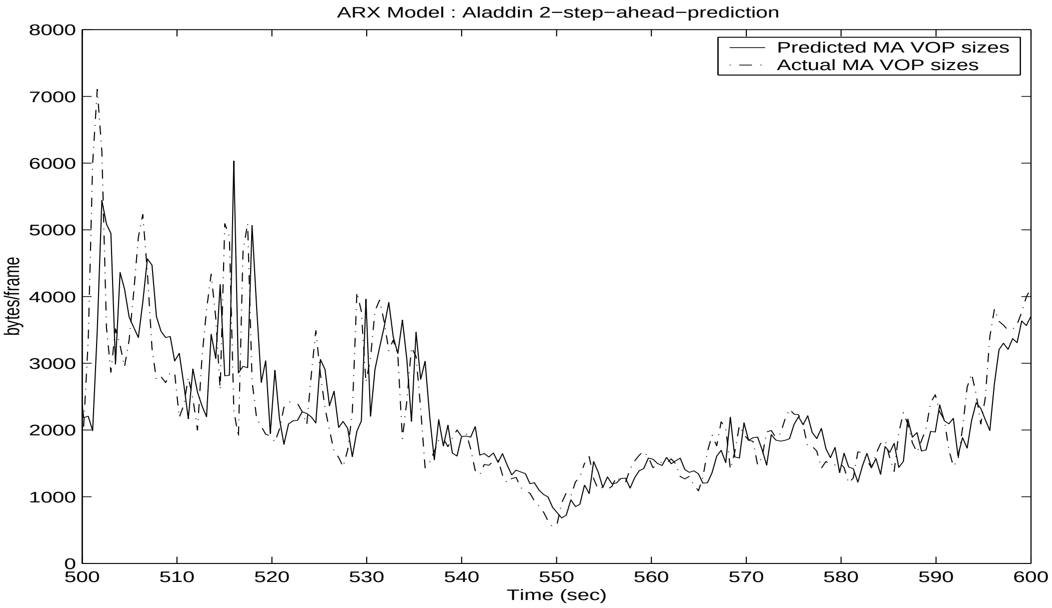

Fig. 51. Two-SP sizes of the moving average VOPs of *Aladdin* using ARX models.

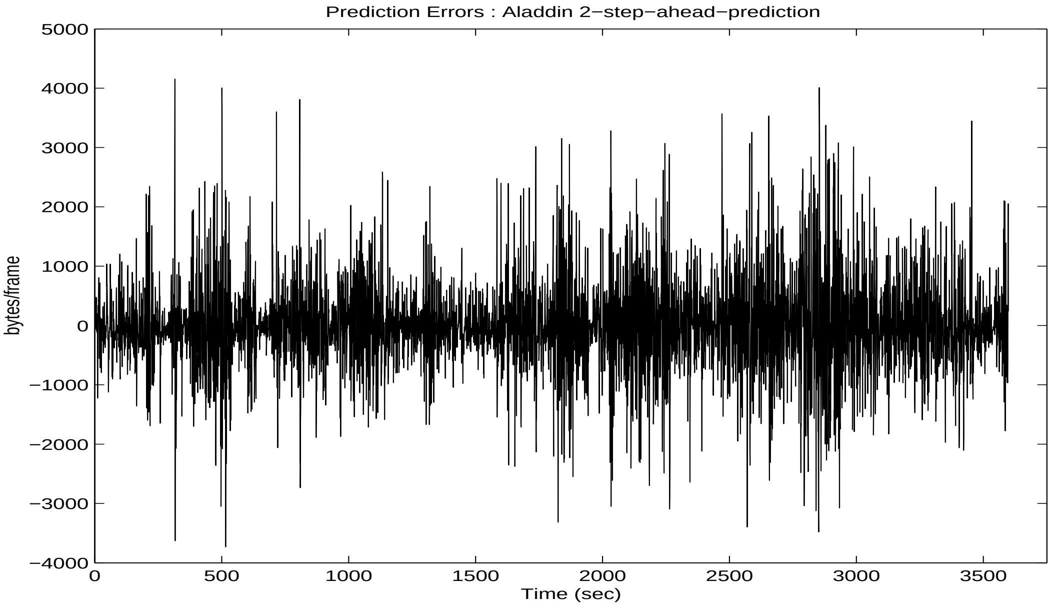

Fig. 52. Two-SP errors of moving average VOP sizes of *Aladdin* using ARX models.

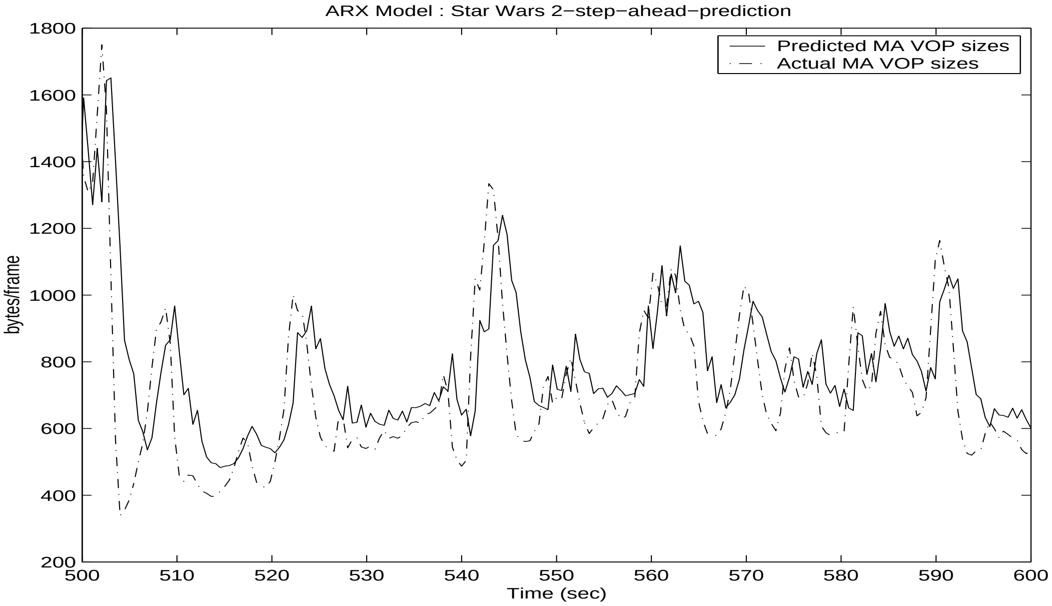

Fig. 53. Two-SP sizes of the moving average VOPs of *StarWars* using ARX models.

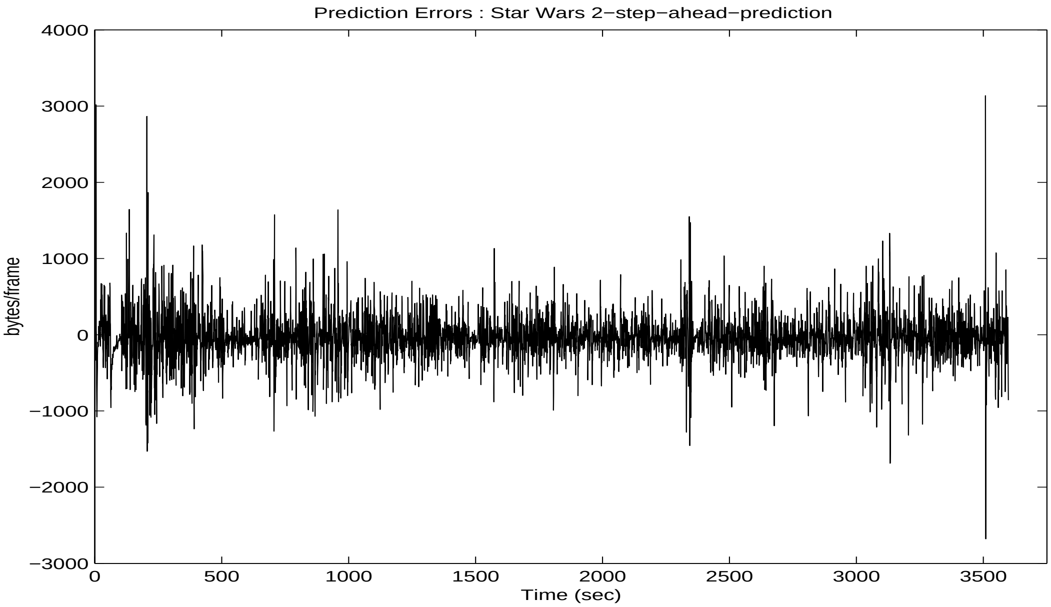

Fig. 54. Two-SP errors of moving average VOP sizes of *StarWars* using ARX models.

Table XI. Performance metrics of the two-SP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	9.6	9157.4	14.9
ARD Talk	3.0	5318.6	2.0
Die Hard III	7.9	5588.7	6.2
Jurassic Park I	4.5	9147.1	6.7
Lecture Room	6.8	431.6	1.5
Silence of the Lambs	8.5	2816.3	4.9
Skiing	6.5	1606.1	6.6
StarWars	4.7	3137.1	7.3

c. Two-step-ahead Prediction Using FMLP Models

The two-step-ahead FMLP model was designed using the external indicators. The FMLP structure which gave the best results was $12 - 28 - 1$. For the prediction of $x_{MA}(k)$, the 12 inputs were $x_{MA}(k-2)$, $x_{MA}(k-3)$, $x_{MA}(k-4)$, $\delta x_{MA}(k-2)$, $\delta x_{MA}(k-3)$, $\delta x_{MA}(k-4)$, $\Delta x_{MA}(k-2)$, $\Delta x_{MA}(k-3)$, $\Delta x_{MA}(k-4)$, $I_1(k-2)$, $I_2(k-2)$, and $I_3(k-2)$. Figures 55 to 58 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

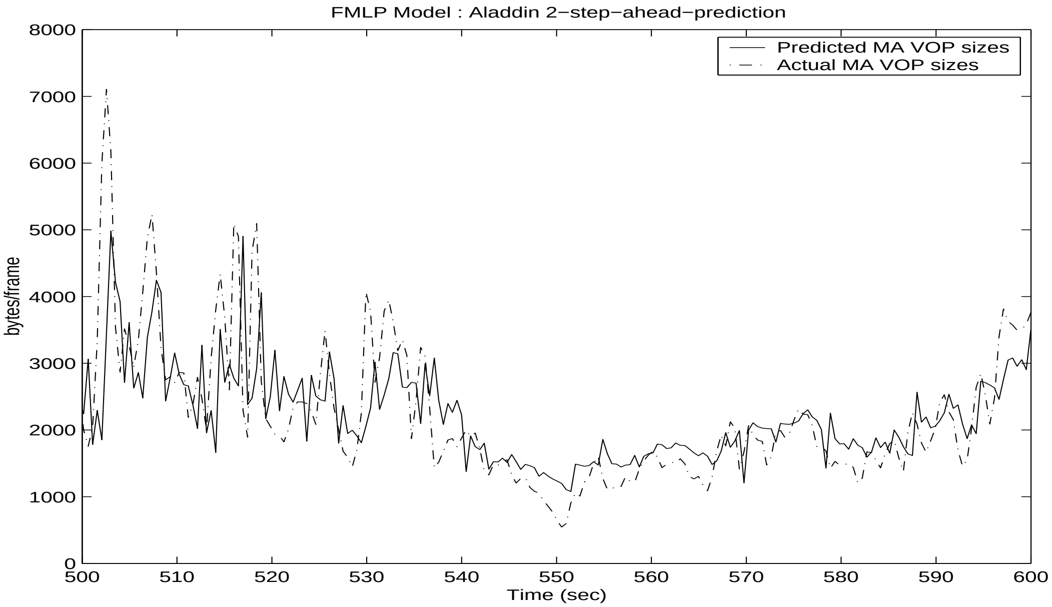

Fig. 55. Two-SP sizes of the moving average VOPs of *Aladdin* using FMLP models.

Table XII shows the performance of the FMLP two-step-ahead predictor for the video traces used in this research in terms of the performance metrics.

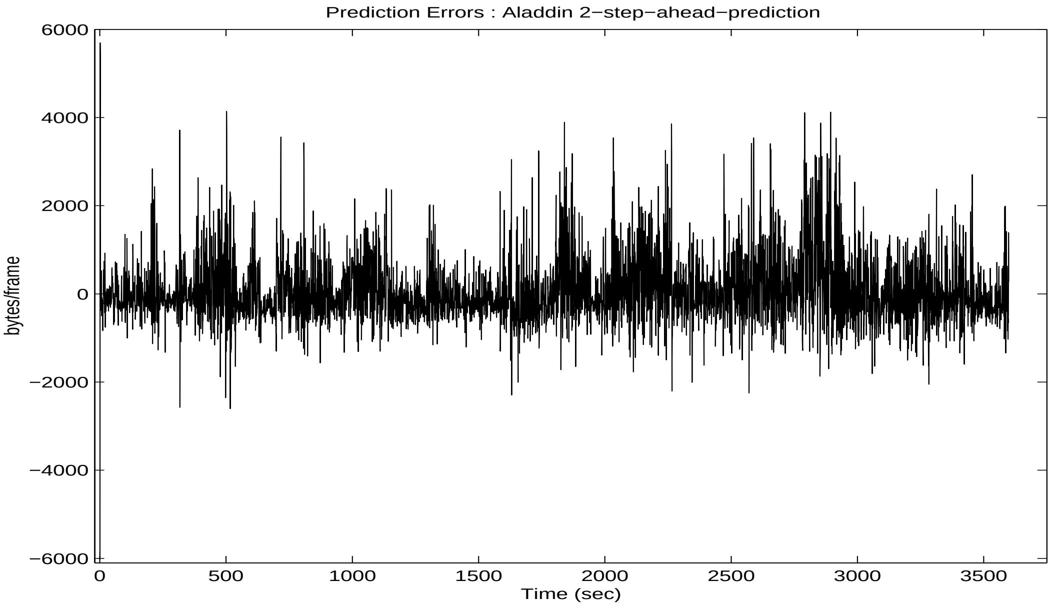

Fig. 56. Two-SP errors of moving average VOP sizes of *Aladdin* using FMLP models.

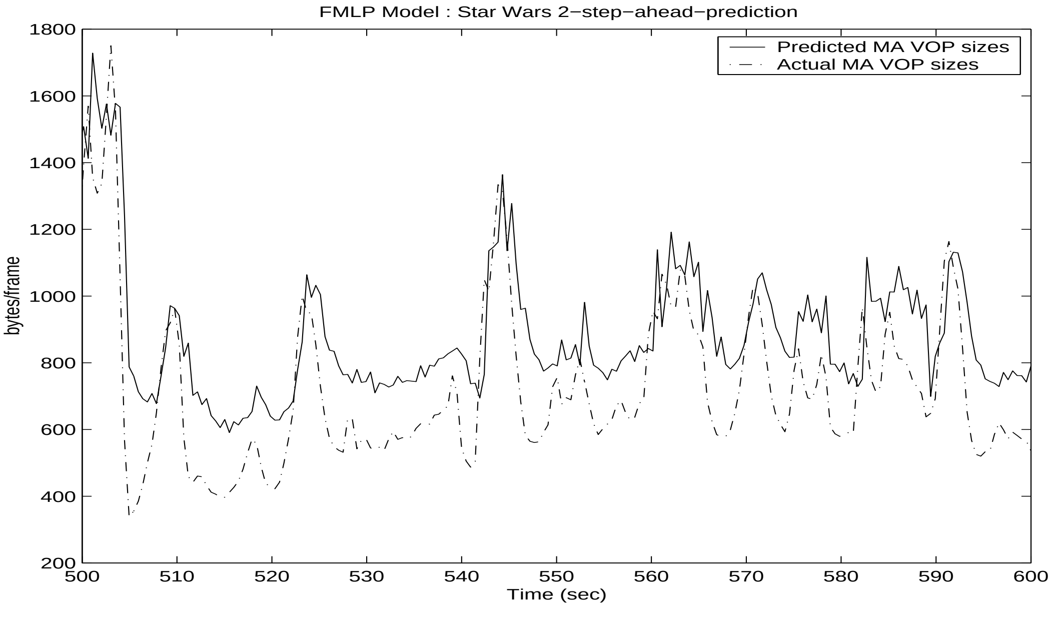

Fig. 57. Two-SP sizes of the moving average VOPs of *StarWars* using FMLP models.

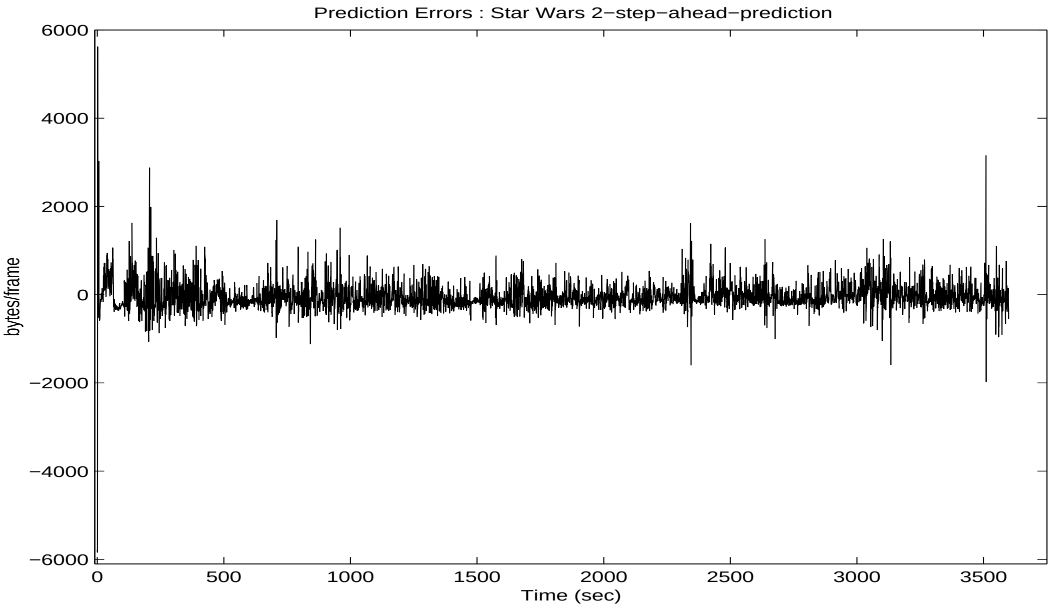

Fig. 58. Two-SP errors of moving average VOP sizes of *StarWars* using FMLP models.

Table XII. Performance metrics of the two-SP for FMLP models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	7.8	6017.9	23.9
ARD Talk	2.9	5741.6	2.7
Die Hard III	7.2	5823.2	13.0
Jurassic Park I	4.8	8960.4	8.9
Lecture Room	49.3	5672.8	9.5
Silence of the Lambs	14.2	4523.4	4.4
Skiing	10.2	6011.6	23.3
StarWars	4.1	5841.7	13.6

d. Two-step-ahead Prediction Using ESN Models

The network structure which gave the best results for two-step-ahead prediction of the moving average of VOP size time-series was $4 - 30 - 1$. In order to predict the moving average of the VOP sizes at time step k the four inputs were $x_{MA}(k-2)$, $\delta x_{MA}(k-2)$, $\Delta x_{MA}(k-2)$ and $I_1(k-2)$. Figures 59 to 62 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XIII shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

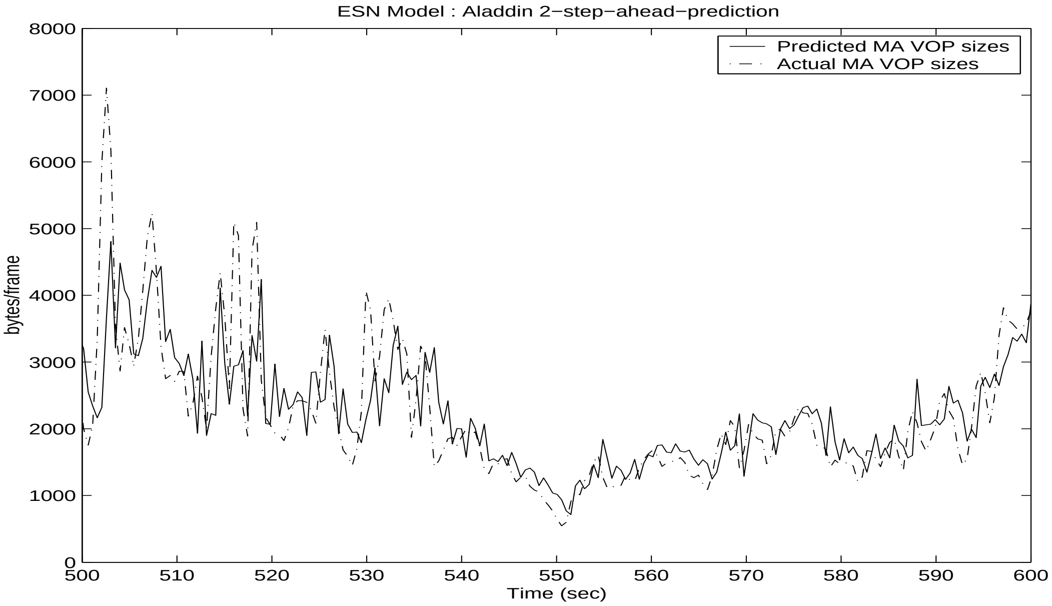

Fig. 59. Two-SP sizes of the moving average VOPs of *Aladdin* using ESN models.

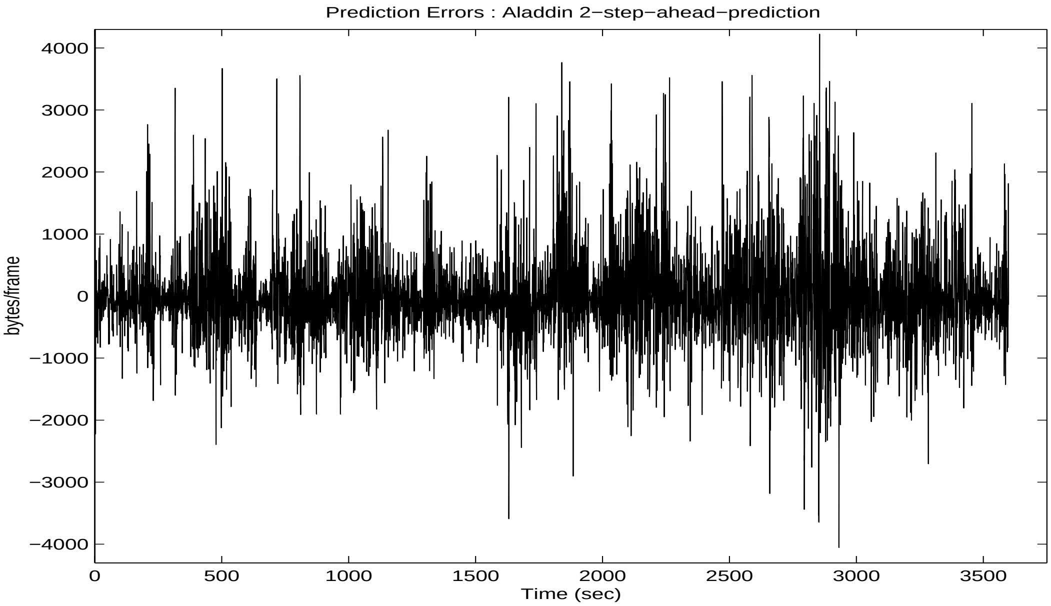

Fig. 60. Two-SP errors of moving average VOP sizes of *Aladdin* using ESN models.

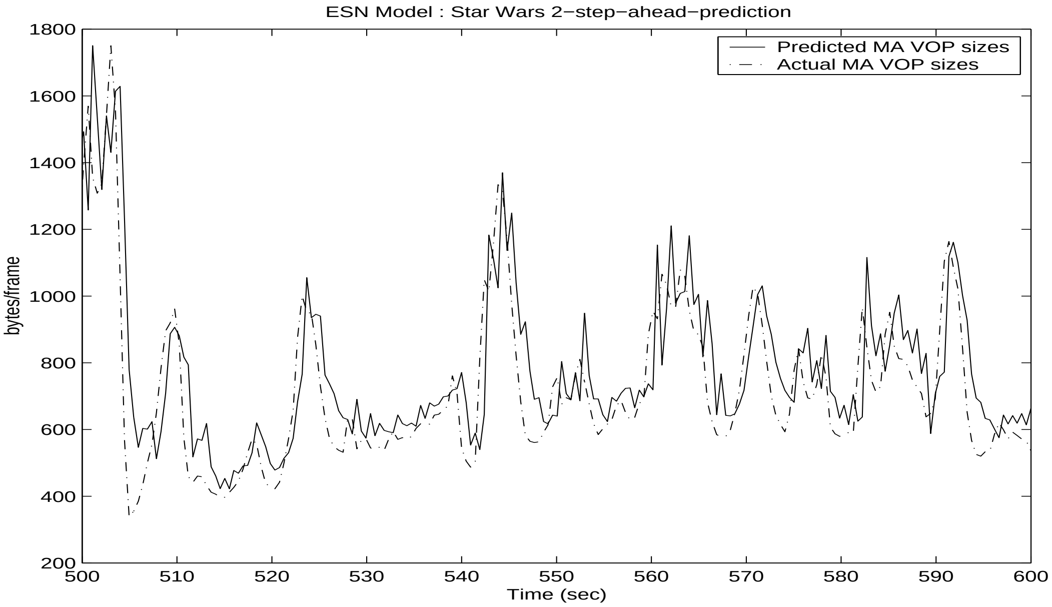

Fig. 61. Two-SP sizes of the moving average VOPs of *StarWars* using ESN models.

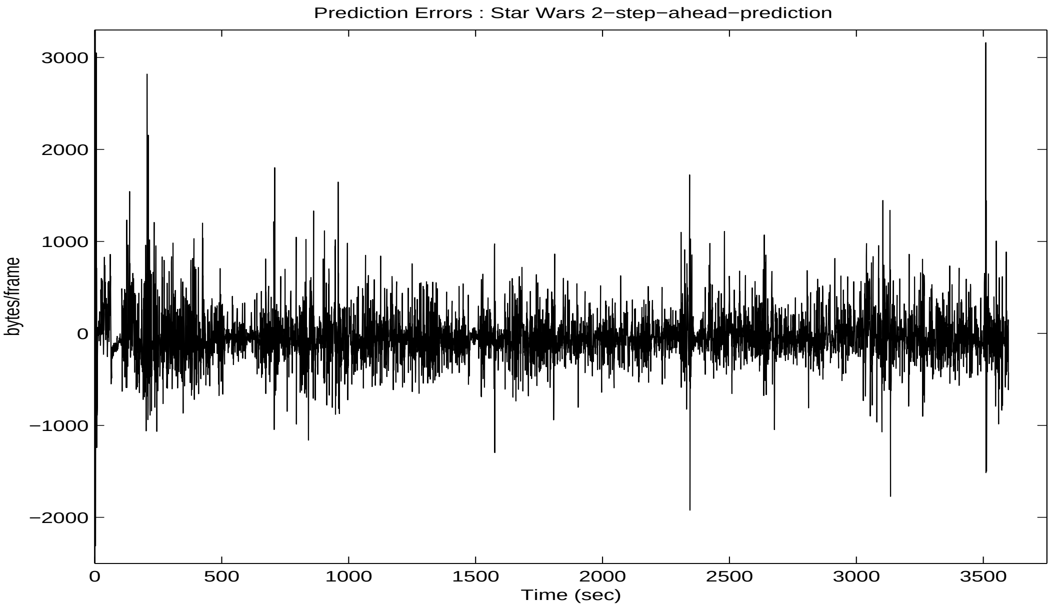

Fig. 62. Two-SP errors of moving average VOP sizes of *StarWars* using ESN models.

Table XIII. Performance metrics of the two-SP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	7.8	4222.5	4.6
ARD Talk	2.7	5318.4	1.0
Die Hard III	5.4	2791.9	3.4
Jurassic Park I	6.6	10303	4.2
Lecture Room	1.5	378.4	1.0
Silence of the Lambs	8.0	2338.8	3.4
Skiing	5.6	1757.6	5.2
StarWars	3.1	3185.4	4.2

D. Four-step-ahead Prediction

1. Prediction of I-VOPs

The four-step-ahead prediction (Four-SP) of individual I-VOPs means predicting 1.92 seconds ahead. The training was done using the first 1500 points of the I-VOP size time-series of video trace *Aladdin*. The next 500 points were used for the validation of the model.

a. Four-step-ahead Prediction Using ARX Models

The indicators used for the ARX model were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. The model structure which gives the best results is $n_y = 7$, $n_u = 1$ and $n_k = 3$. Figures 63 to 66 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XIV shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

b. Four-step-ahead Prediction Using ESN Models

The network structure which gave the best results was $8 - 23 - 1$. The inputs and indicators used were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. Figures 67 to 70 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XV shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

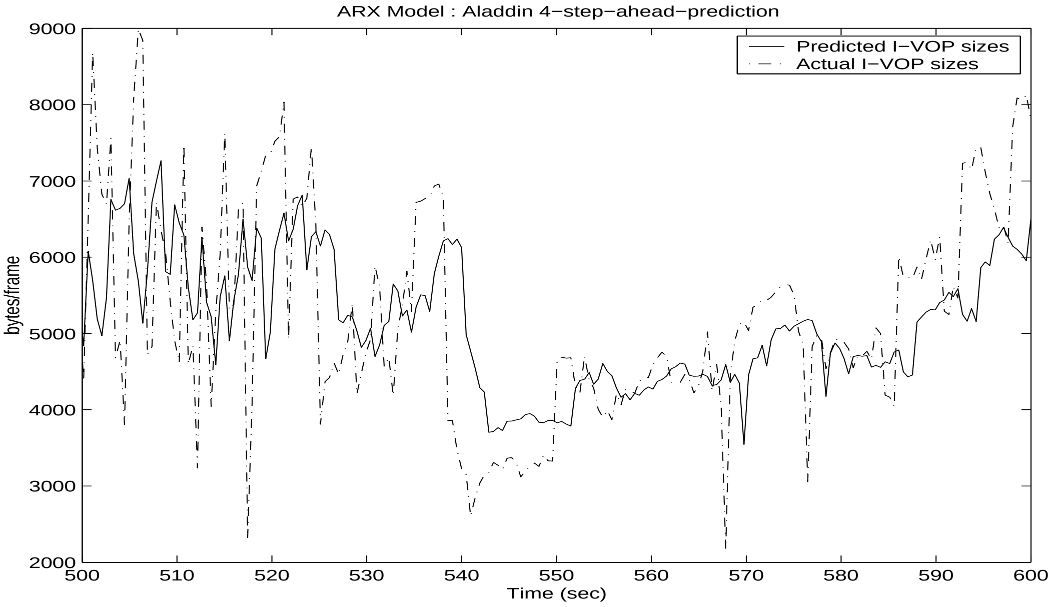

Fig. 63. Four-SP sizes of the I-VOPs of *Aladdin* using ARX models.

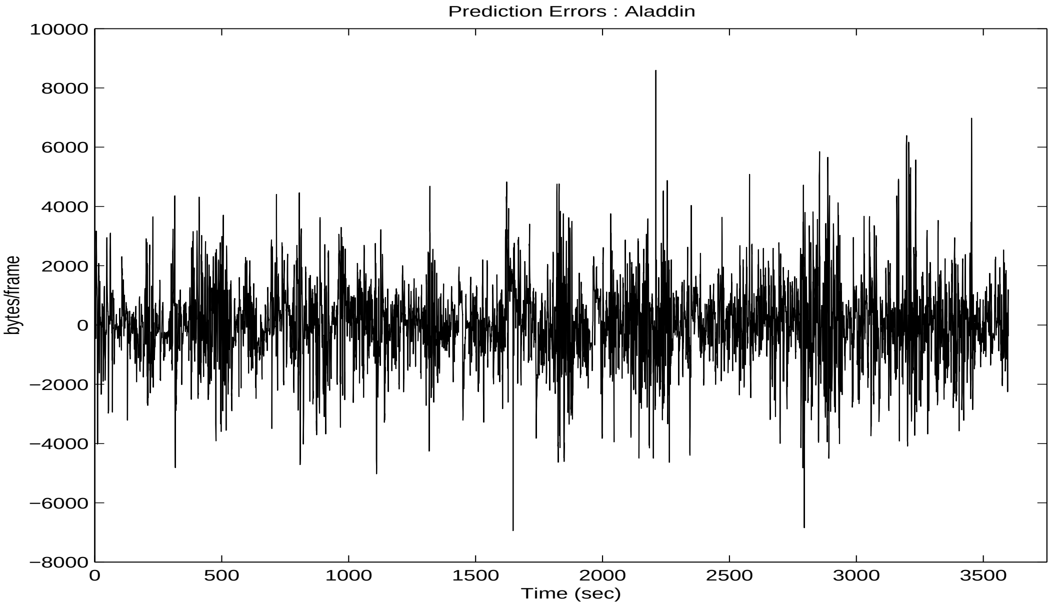

Fig. 64. Four-SP errors of I-VOP sizes of *Aladdin* using ARX models.

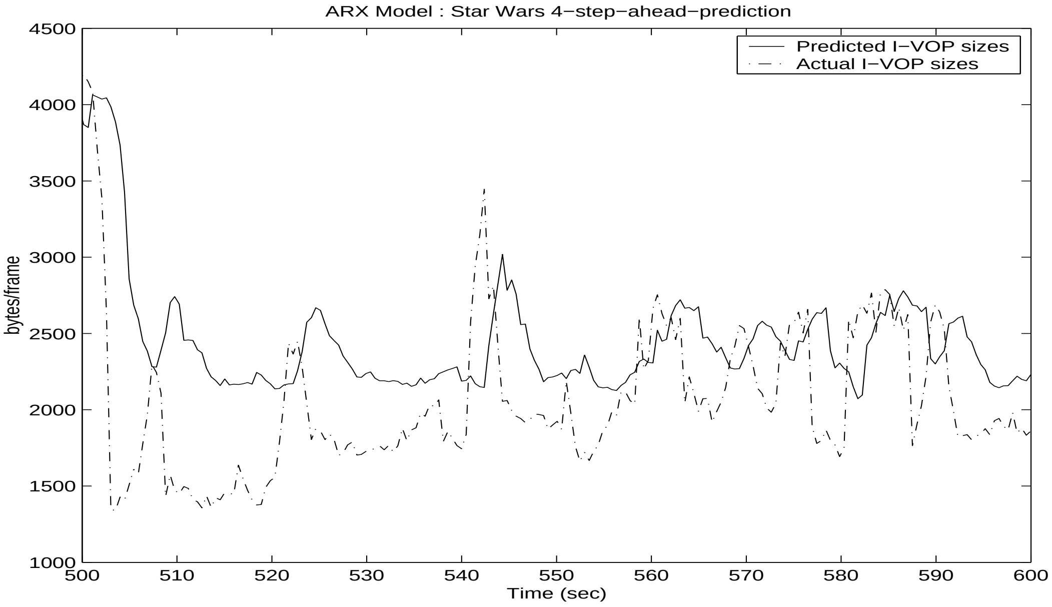

Fig. 65. Four-SP sizes of the I-VOPs of *StarWars* using ARX models.

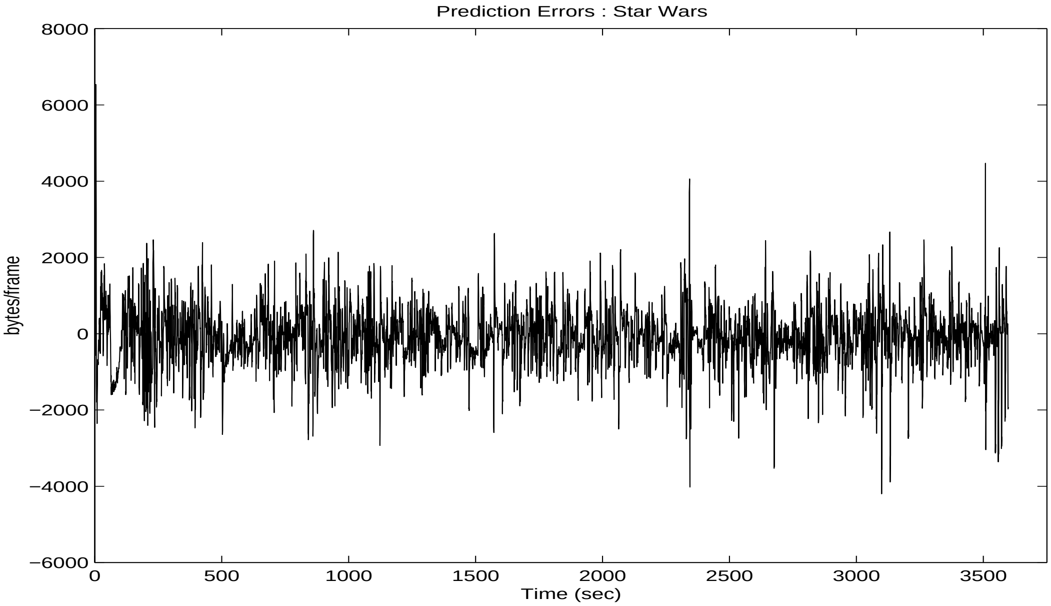

Fig. 66. Four-SP errors of I-VOP sizes of *StarWars* using ARX models.

Table XIV. Performance metrics of the I-VOP for the four-SP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	6.6	8597.2	17.6
ARD Talk	3.0	5680.6	2.0
Die Hard III	8.5	4719.6	10.5
Jurassic Park I	3.4	9172.3	8.6
Lecture Room	1.6	1667.4	1.6
Silence of the Lambs	10.8	7761.2	6.3
Skiing	7.0	5338.4	2.4
StarWars	4.5	6536.4	7.5

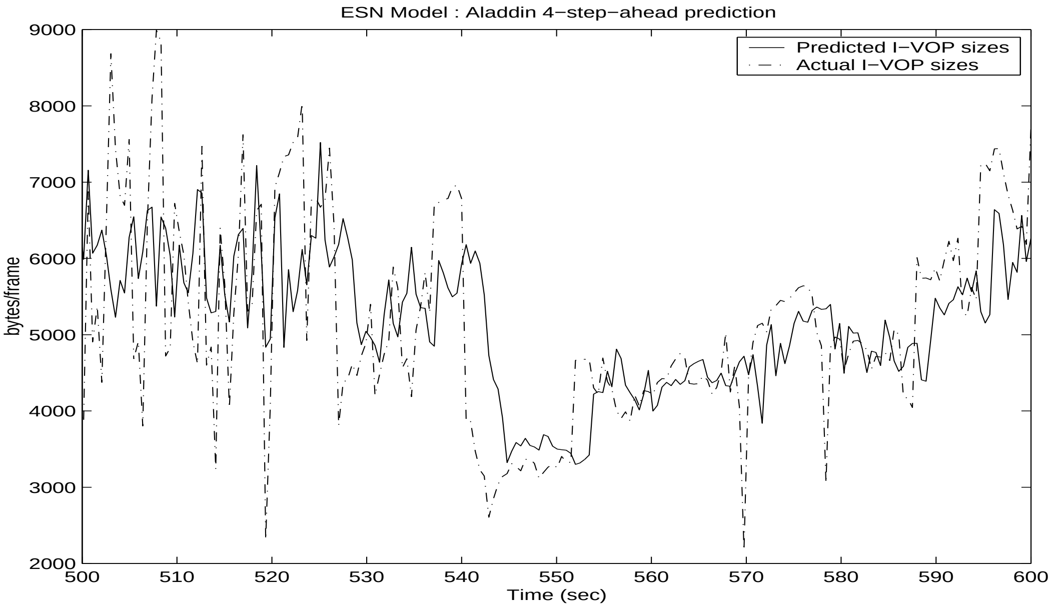

Fig. 67. Four-SP sizes of the I-VOPs of *Aladdin* using ESN models.

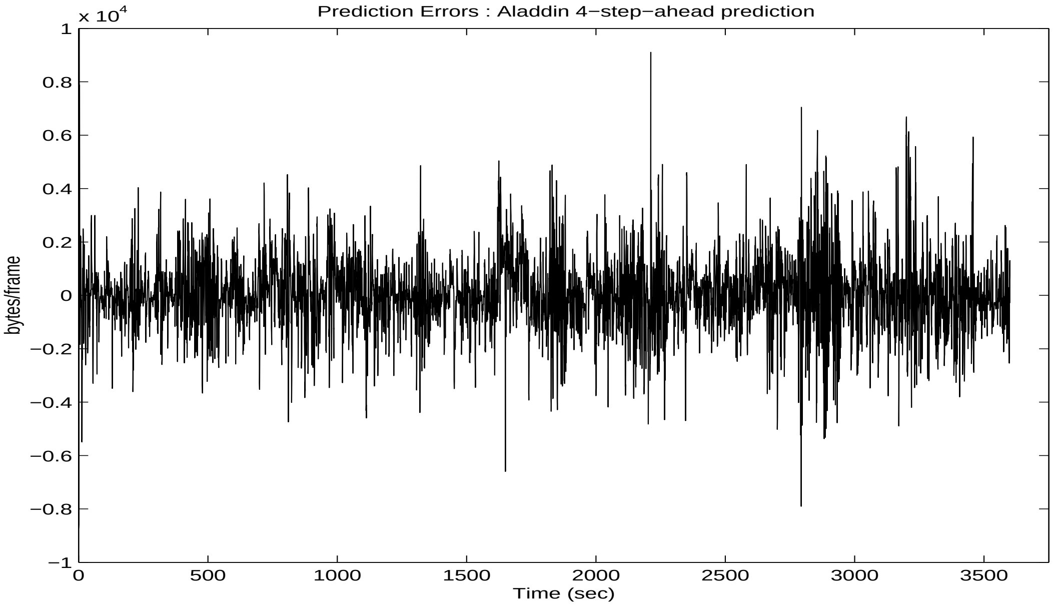

Fig. 68. Four-SP errors of I-VOP sizes of *Aladdin* using ESN models.

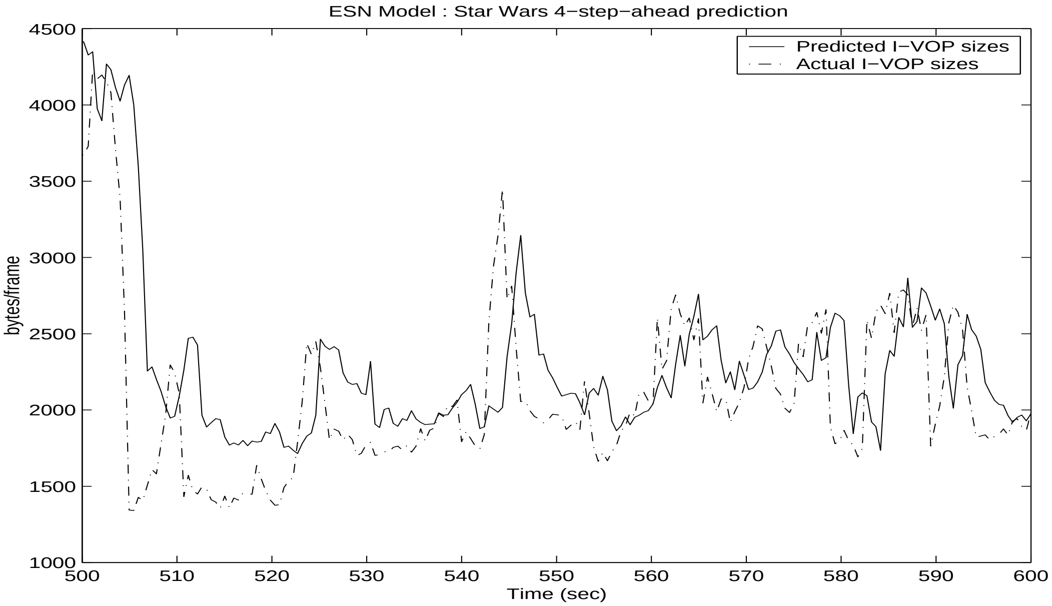

Fig. 69. Four-SP sizes of the I-VOPs of *StarWars* using ESN models.

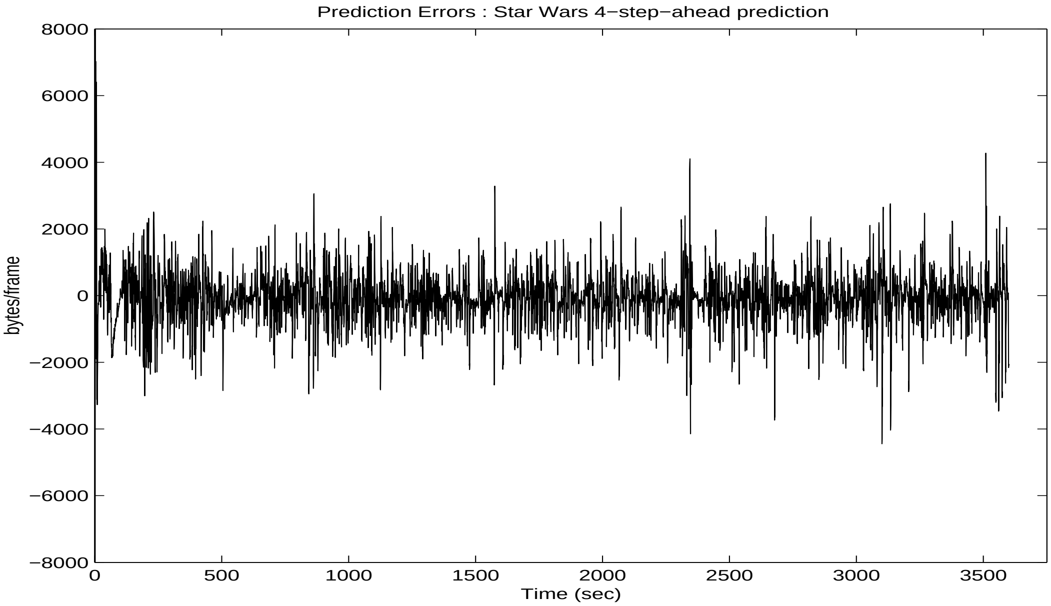

Fig. 70. Four-SP errors of I-VOP sizes of *StarWars* using ESN models.

Table XV. Performance metrics of the I-VOP for the four-SP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	6.8	9108.6	16.8
ARD Talk	3.9	12063	1.6
Die Hard III	8.3	4904.7	11.0
Jurassic Park I	6.6	14928	8.9
Lecture Room	0.4	1466.3	1.4
Silence of the Lambs	11.0	8227.5	6.3
Skiing	6.7	5455.6	2.5
StarWars	4.4	7148.3	7.4

2. Prediction of Moving Average of VOPs

The four-step-ahead prediction (Four-SP) represents a time horizon of 1.92 seconds. For four-step-ahead prediction, training was done using the first 1500 points of the moving average time-series of VOP sizes of video trace *Aladdin*. The next 500 points were used for the validation of the model. The developed model was then used to generate four-step-ahead predictions for the entire length for all the eight video traces.

a. Four-step-ahead Prediction Using AR Models

The AR model used for four-step-ahead prediction was the same model that was developed for SSP. For this model, $n_y = 17$. Figures 71 to 74 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

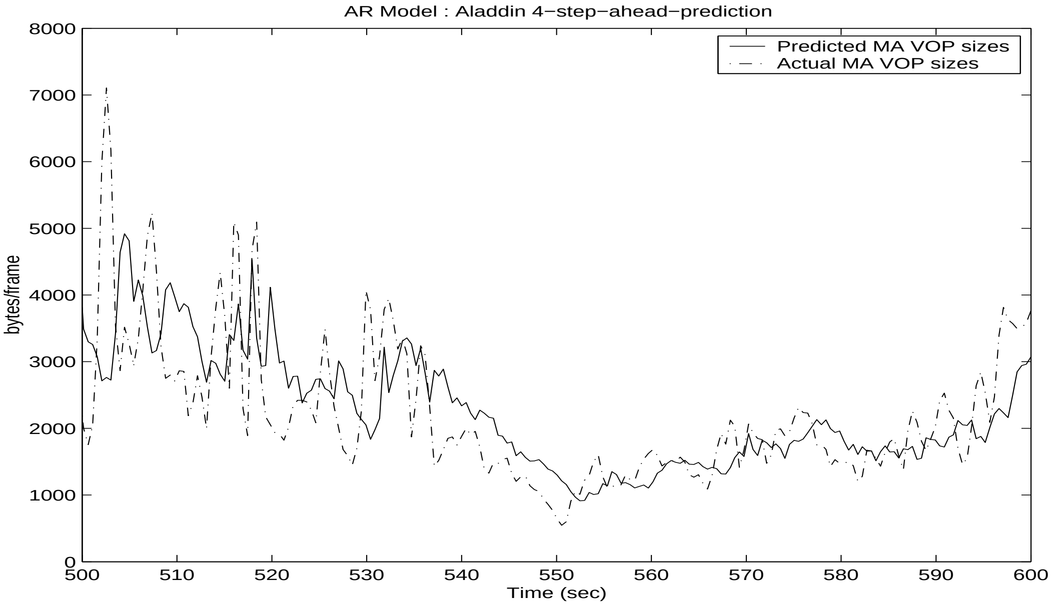

Fig. 71. Four-SP sizes of the moving average VOPs of *Aladdin* using AR models.

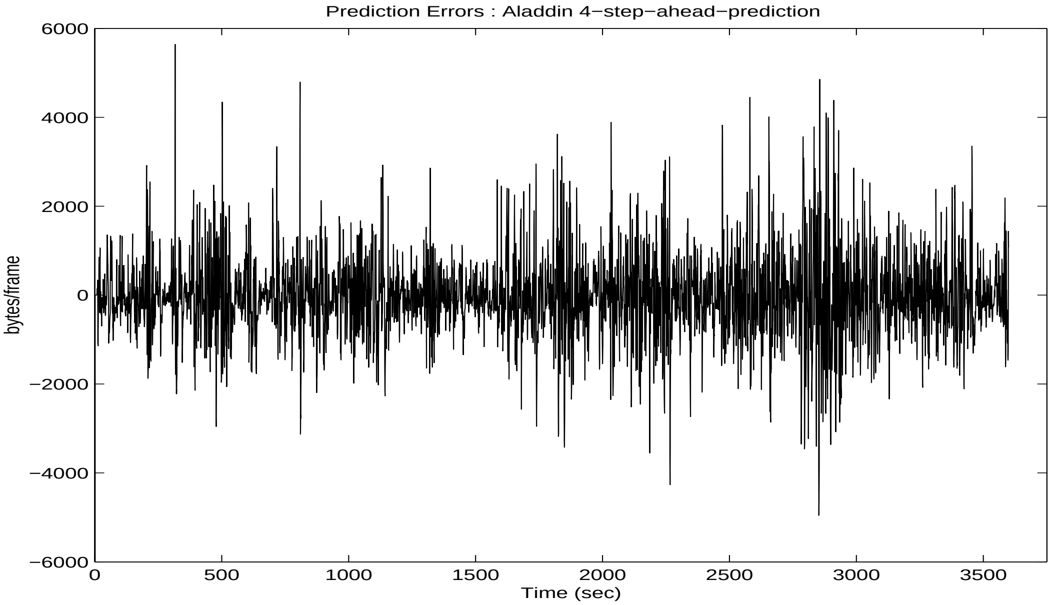

Fig. 72. Four-SP errors of moving average VOP sizes of *Aladdin* using AR models.

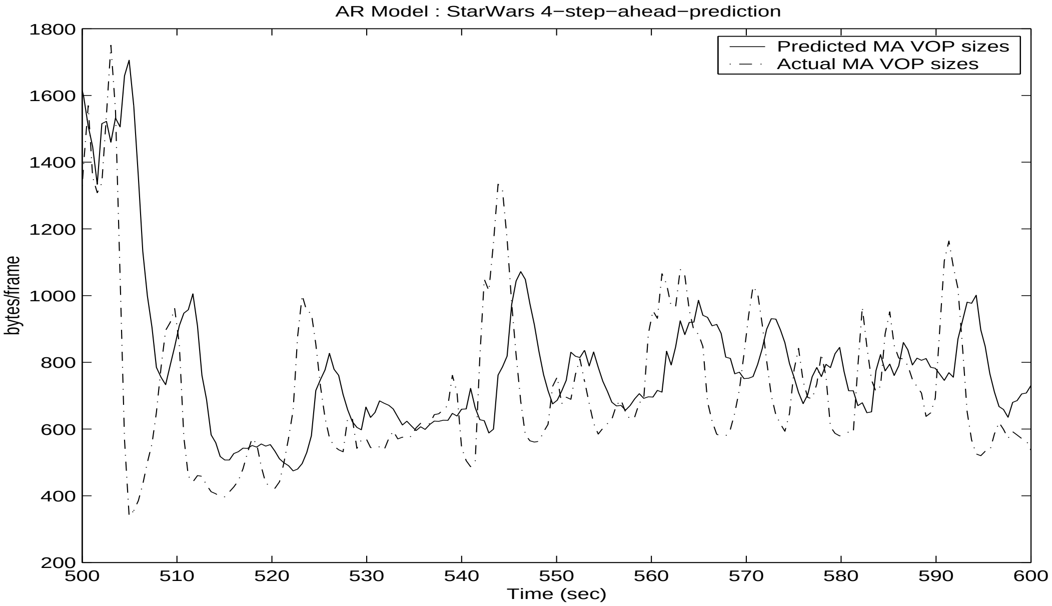

Fig. 73. Four-SP sizes of the moving average VOPs of *StarWars* using AR models.

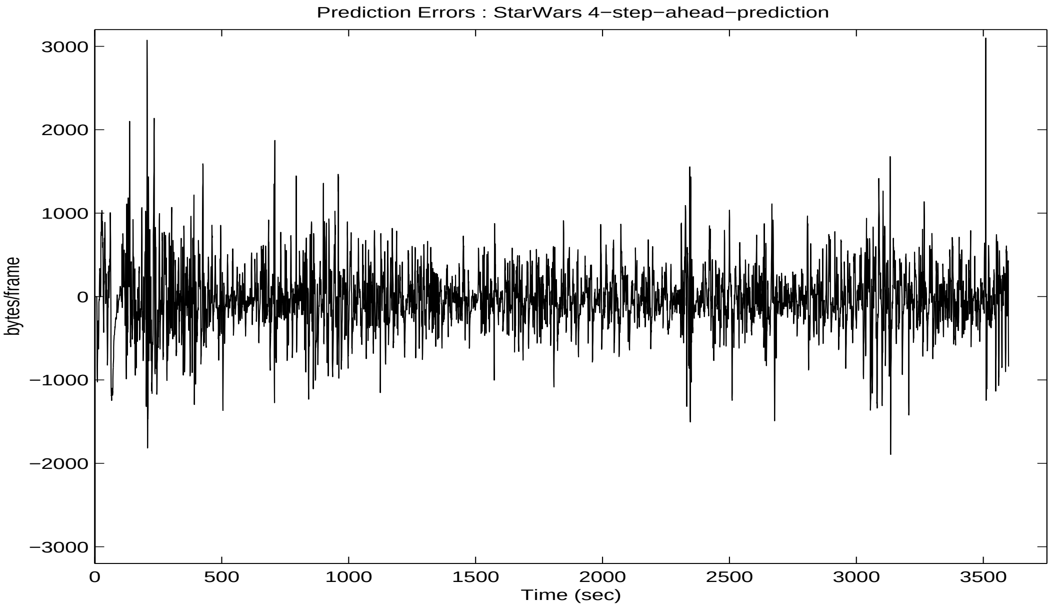

Fig. 74. Four-SP errors of moving average VOP sizes of *StarWars* using AR models.

Table XVI shows the performance of the AR predictor for the video traces used in this research in terms of the performance metrics defined earlier in the chapter.

b. Four-step-ahead Prediction Using ARX Models

In the design of ARX models for four-step-ahead prediction, other than the time-series x_{MA}, external indicators δx_{MA}, Δx_{MA} and $I-$VOPs were also used as additional inputs. The model which gave the best results had the structure: $n_y = 6$, $n_u = [1111]$ and $n_k = [2222]$. Figures 75 to 78 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XVII shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

Table XVI. Performance metrics of the four-SP for AR models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	15.2	9326.8	37.0
ARD Talk	4.9	7868.6	4.6
Die Hard III	15.3	9132.0	20.4
Jurassic Park I	9.9	9563.7	14.2
Lecture Room	40.5	9007.1	15.8
Silence of the Lambs	20.9	7832.2	10.4
Skiing	13.0	1677.9	7.4
StarWars	8.5	3099.9	9.5

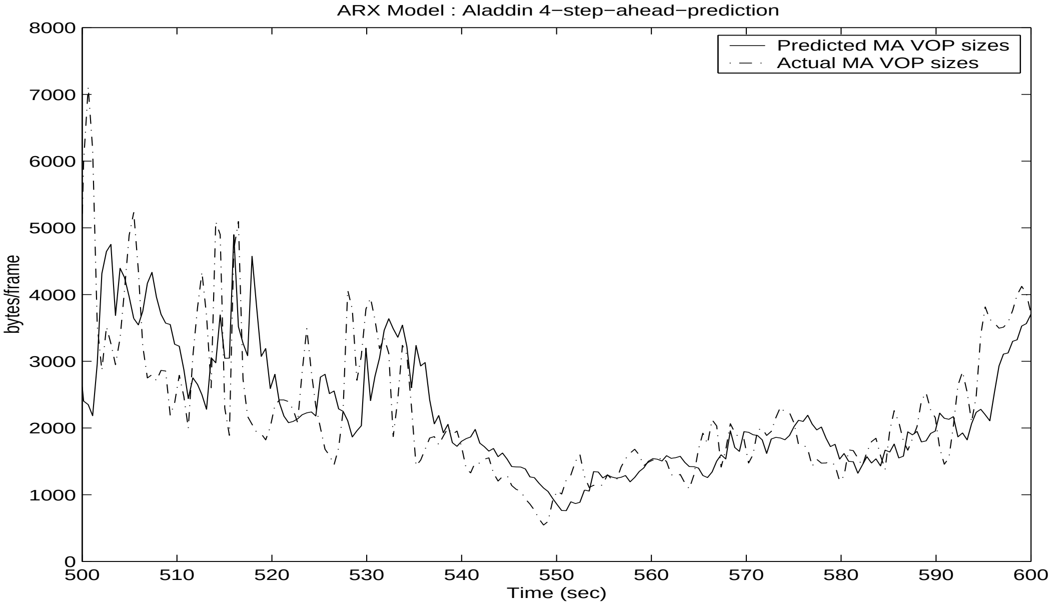

Fig. 75. Four-SP sizes of the moving average VOPs of *Aladdin* using ARX models.

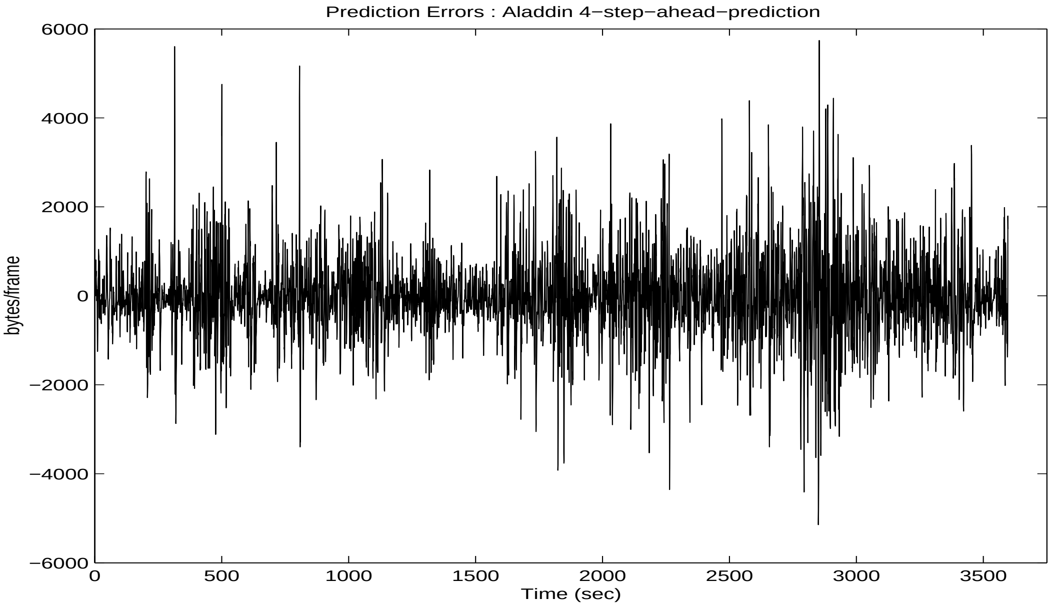

Fig. 76. Four-SP errors of moving average VOP sizes of *Aladdin* using ARX models.

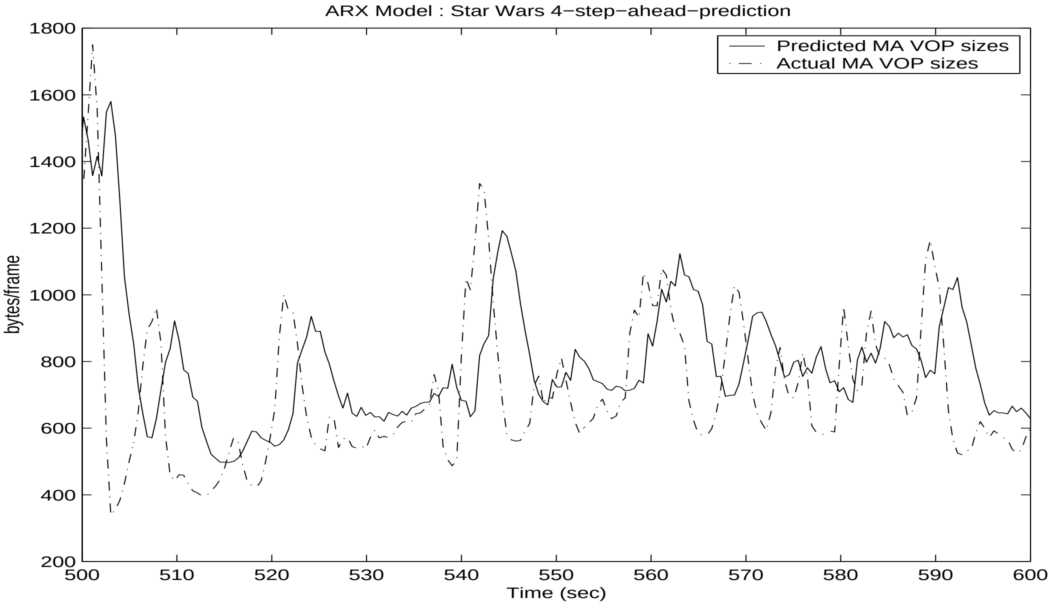

Fig. 77. Four-SP sizes of the moving average VOPs of *StarWars* using ARX models.

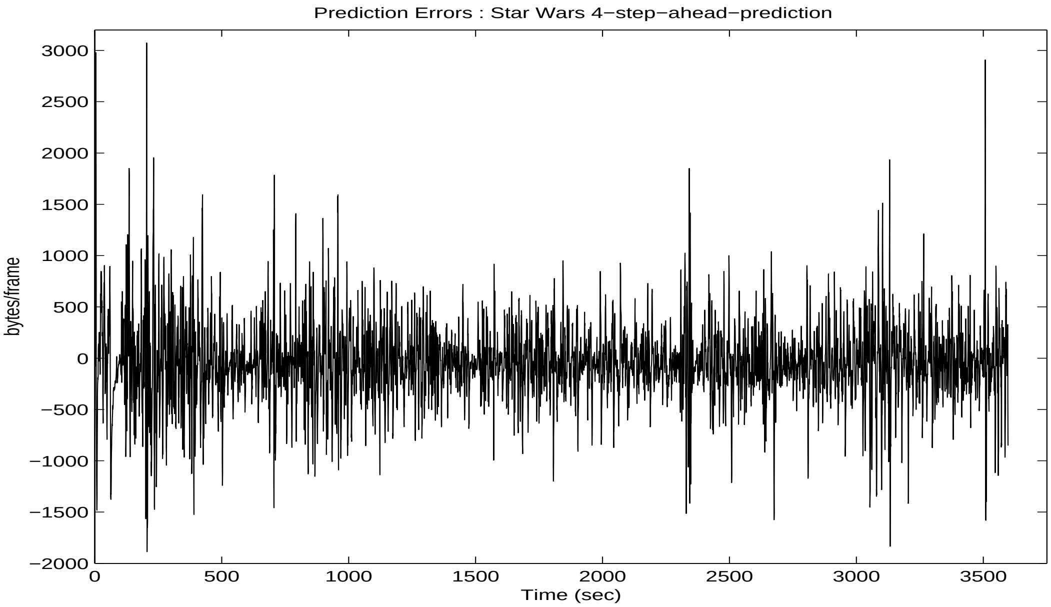

Fig. 78. Four-SP errors of moving average VOP sizes of *StarWars* using ARX models.

Table XVII. Performance metrics of the four-SP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	14.9	5739.2	17.6
ARD Talk	4.4	5522.0	2.3
Die Hard III	14.0	4972.0	7.0
Jurassic Park I	9.8	9324.9	7.0
Lecture Room	7.4	601.8	3.3
Silence of the Lambs	17.0	4760.6	9.5
Skiing	13.6	1965.7	8.7
StarWars	8.8	3074.4	9.4

c. Four-step-ahead Prediction Using FMLP Models

The four-step-ahead FMLP model was designed using the external indicators. The FMLP structure which gave the best results was $12 - 30 - 1$. For the prediction of $x_{MA}(k)$, the 12 inputs were $x_{MA}(k-4)$, $x_{MA}(k-5)$, $x_{MA}(k-6)$, $\delta x_{MA}(k-4)$, $\delta x_{MA}(k-5)$, $\delta x_{MA}(k-6)$, $\Delta x_{MA}(k-4)$, $\Delta x_{MA}(k-5)$, $\Delta x_{MA}(k-6)$, $I_1(k-4)$, $I_2(k-4)$, and $I_3(k-4)$. Figures 79 to 82 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

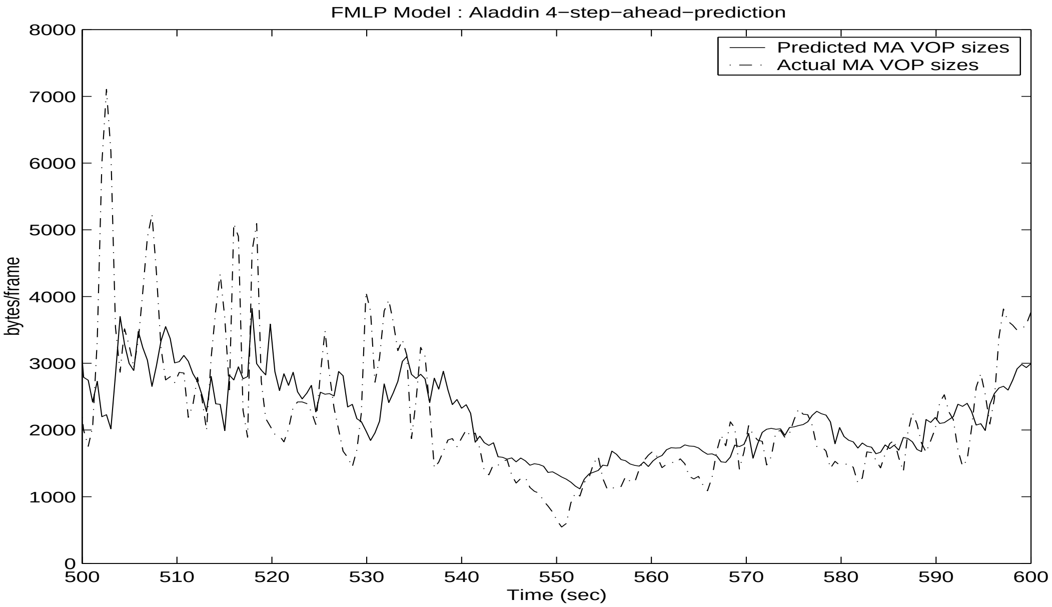

Fig. 79. Four-SP sizes of the moving average VOPs of *Aladdin* using FMLP models.

Table XVIII shows the performance of the FMLP four-step-ahead predictor for the video traces used in this research in terms of the performance metrics.

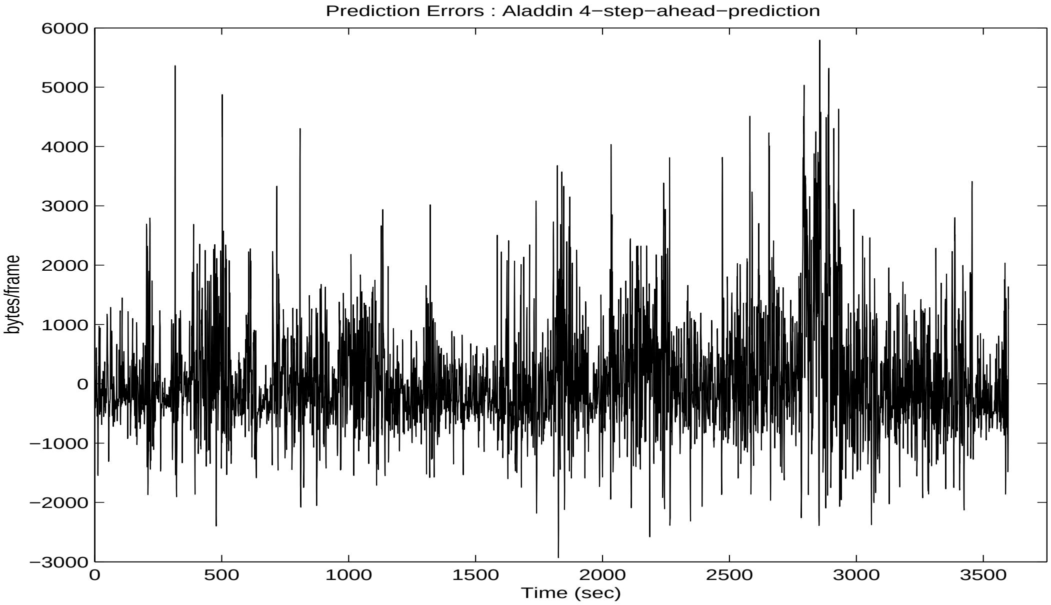

Fig. 80. Four-SP errors of moving average VOP sizes of *Aladdin* using FMLP models.

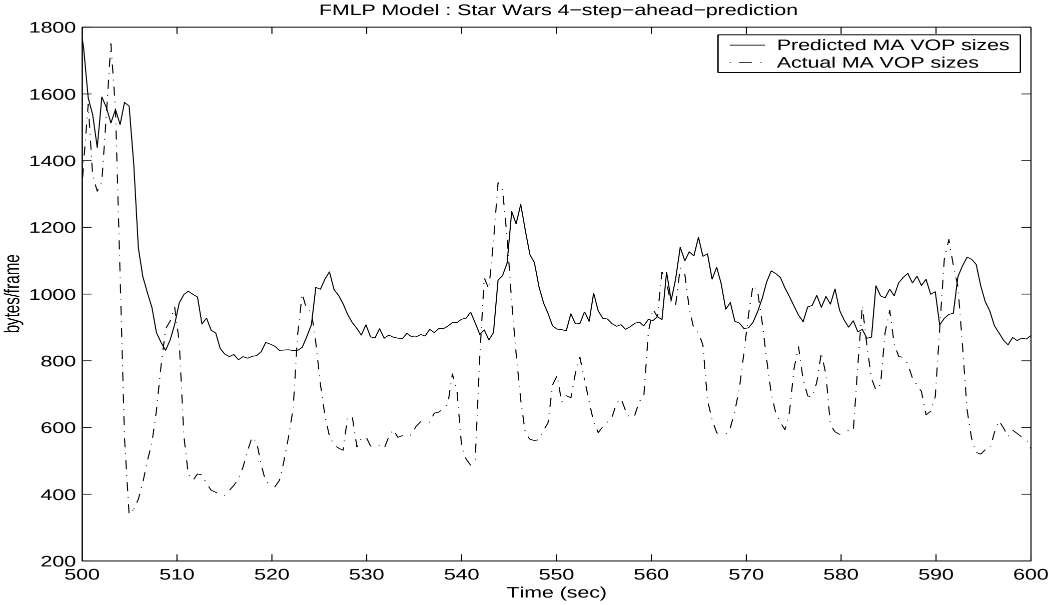

Fig. 81. Four-SP sizes of the moving average VOPs of *StarWars* using FMLP models.

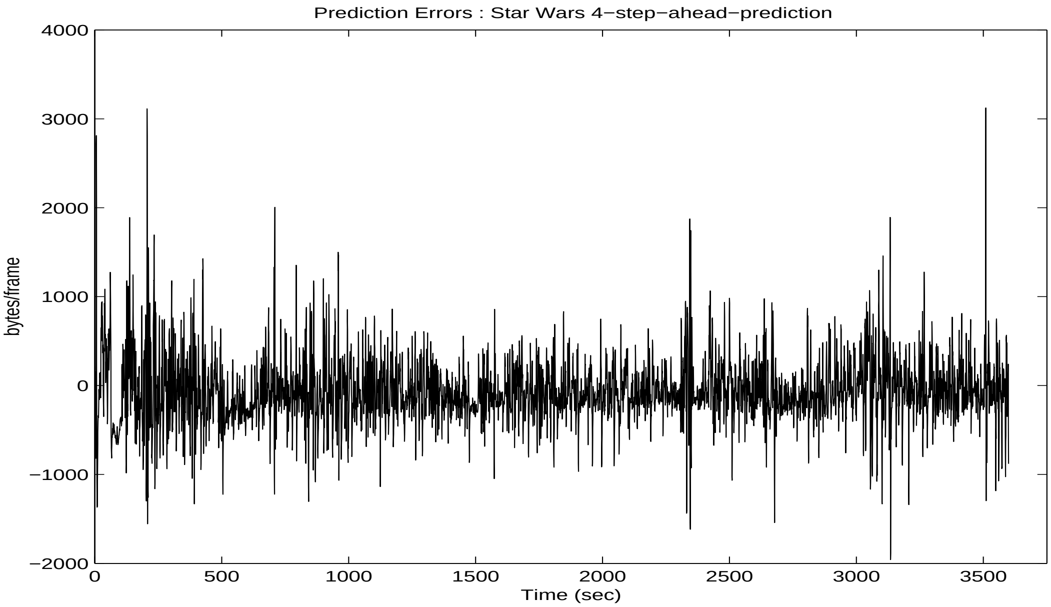

Fig. 82. Four-SP errors of moving average VOP sizes of *StarWars* using FMLP models.

Table XVIII. Performance metrics of the four-SP for FMLP models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	13.2	5794.5	20.7
ARD Talk	5.0	5957.3	1.4
Die Hard III	11.6	3574.1	7.0
Jurassic Park I	23.1	15254	4.2
Lecture Room	81.9	1017.6	2.4
Silence of the Lambs	25.9	3030.3	9.6
Skiing	13.8	1886.8	9.2
StarWars	6.3	3123.6	9.9

d. Four-step-ahead Prediction Using ESN Models

The network structure which gave the best results for four-step-ahead prediction of the moving average of VOP size time-series was $4-30-1$. In order to predict the moving average of the VOP sizes at time step k the four inputs were $x_{MA}(k-4)$, $\delta x_{MA}(k-4)$, $\Delta x_{MA}(k-4)$ and $I_1(k-4)$. Figures 67 to 86 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XIX shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

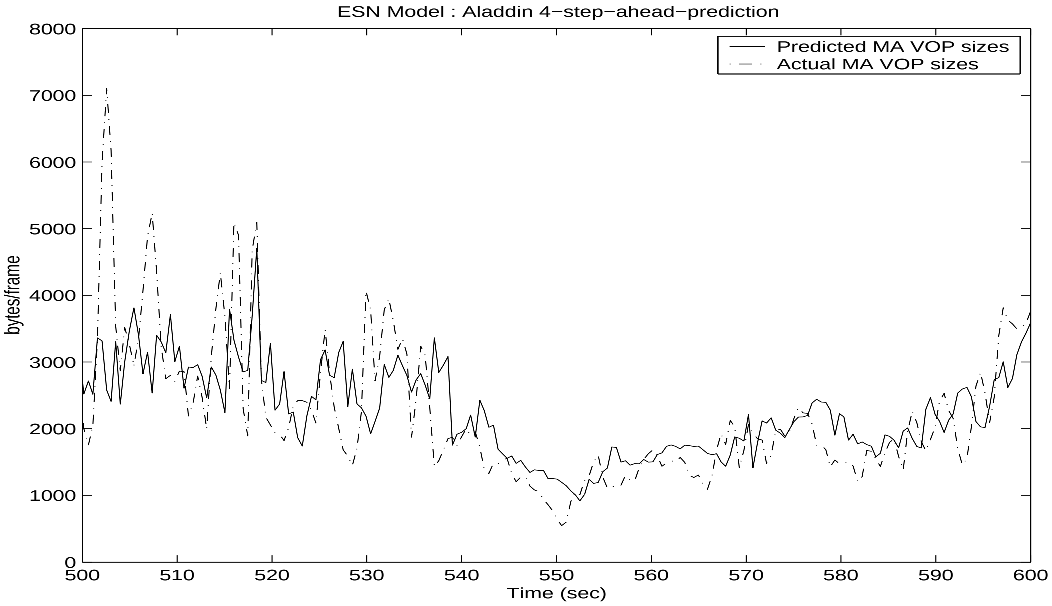

Fig. 83. Four-SP sizes of the moving average VOPs of *Aladdin* using ESN models.

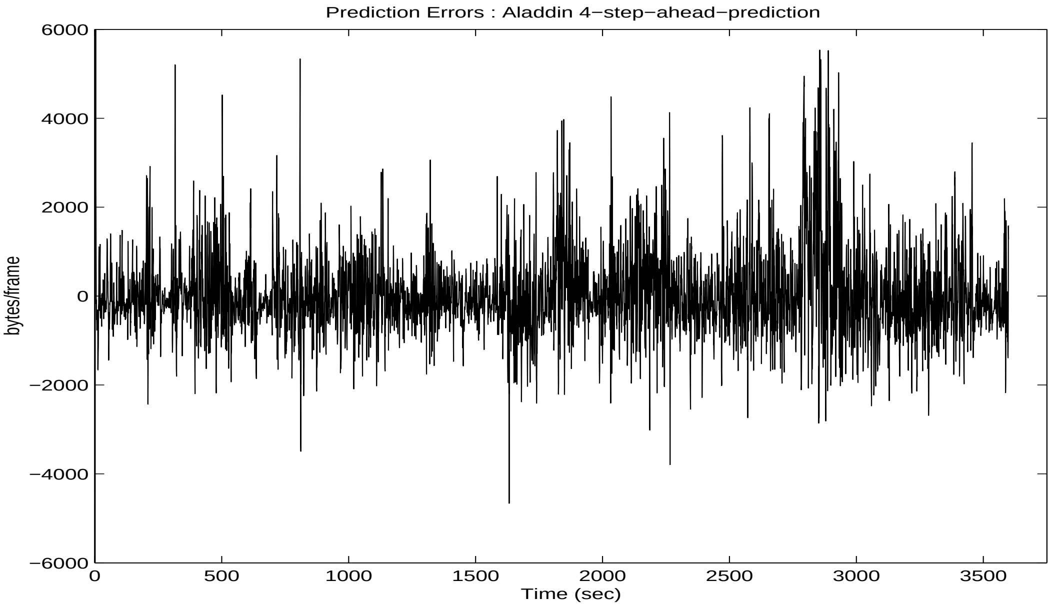

Fig. 84. Four-SP errors of moving average VOP sizes of *Aladdin* using ESN models.

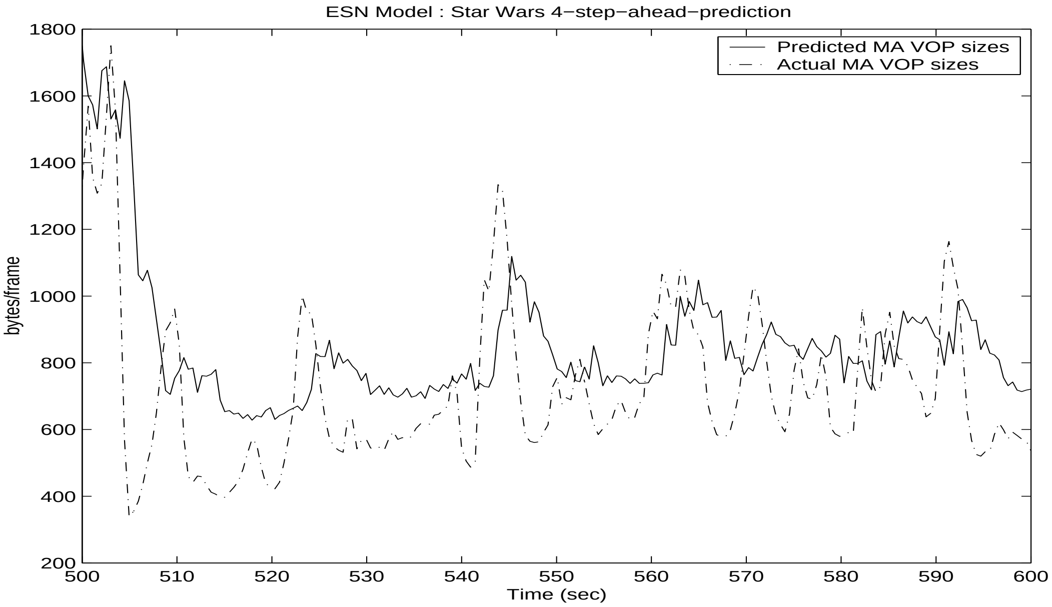

Fig. 85. Four-SP sizes of the moving average VOPs of *StarWars* using ESN models.

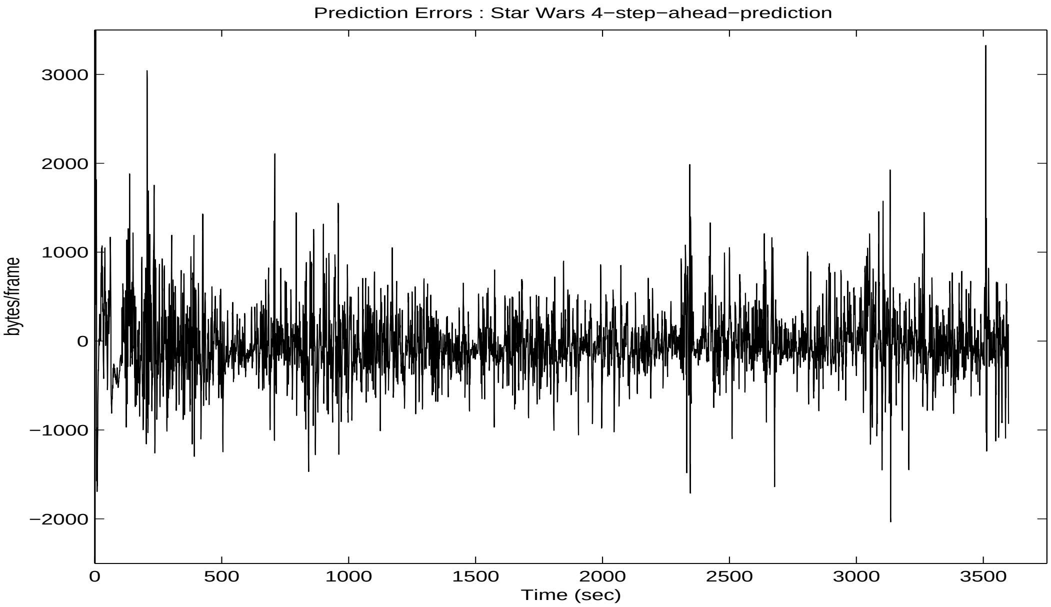

Fig. 86. Four-SP errors of moving average VOP sizes of *StarWars* using ESN models.

Table XIX. Performance metrics of the four-SP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	13.7	5543.3	10.6
ARD Talk	4.6	5775.2	1.4
Die Hard III	12.3	3958.3	5.9
Jurassic Park I	15.2	16257	5.9
Lecture Room	2.8	654.29	1.7
Silence of the Lambs	16.6	2563.4	9.2
Skiing	11.8	1965.4	9.0
StarWars	8.4	3350.0	9.1

E. Six-step-ahead Prediction

1. Prediction of I-VOPs

The six-step-ahead prediction (Six-SP) of individual I-VOPs means predicting 2.88 seconds ahead. The training was done using the first 1500 points of the I-VOP size time-series of video trace *Aladdin*. The next 500 points were used for the validation of the model.

a. Six-step-ahead Prediction Using ARX Models

The indicators used for the ARX model were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. The model structure which gives the best results is $n_y = 7$, $n_u = 1$ and $n_k = 1$. Figures 87 to 90 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XX shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

b. Six-step-ahead Prediction Using ESN Models

The network structure which gave the best results was $8 - 25 - 1$. The inputs and indicators used were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. Figures 91 to 94 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XXI shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

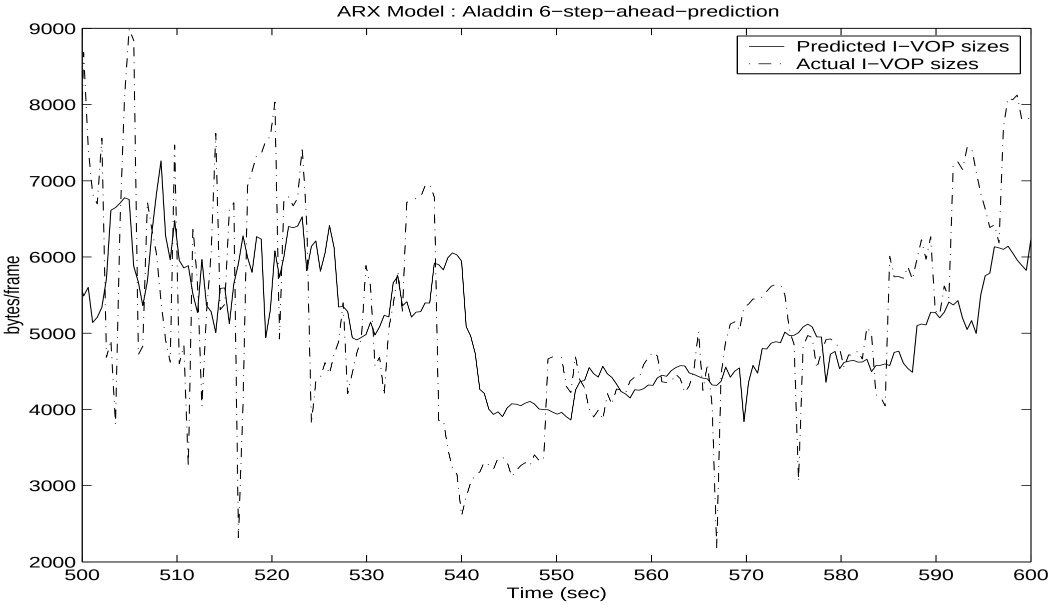

Fig. 87. Six-SP sizes of the I-VOPs of *Aladdin* using ARX models.

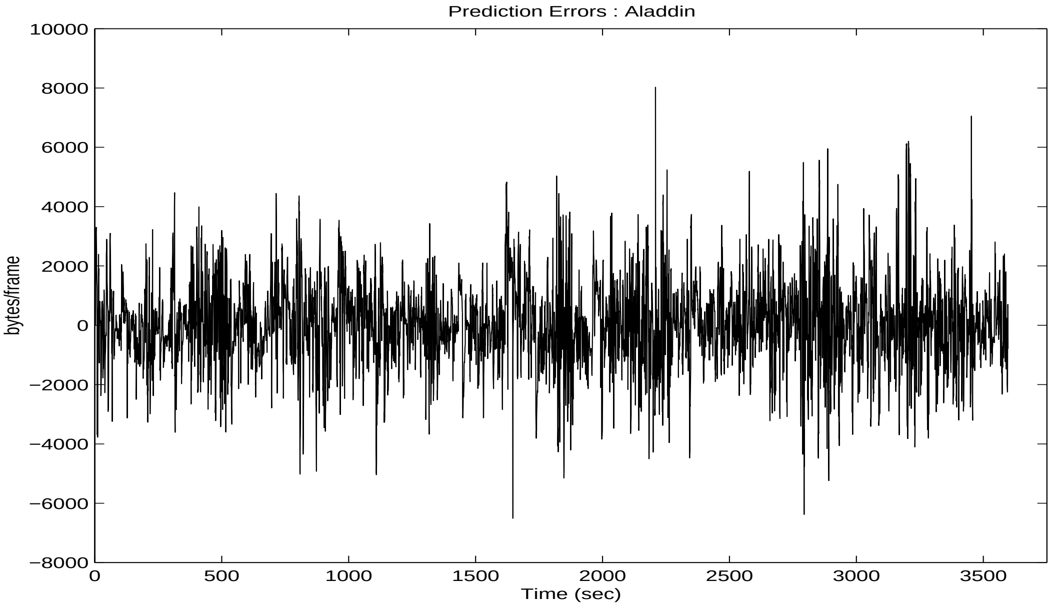

Fig. 88. Six-SP errors of I-VOP sizes of *Aladdin* using ARX models.

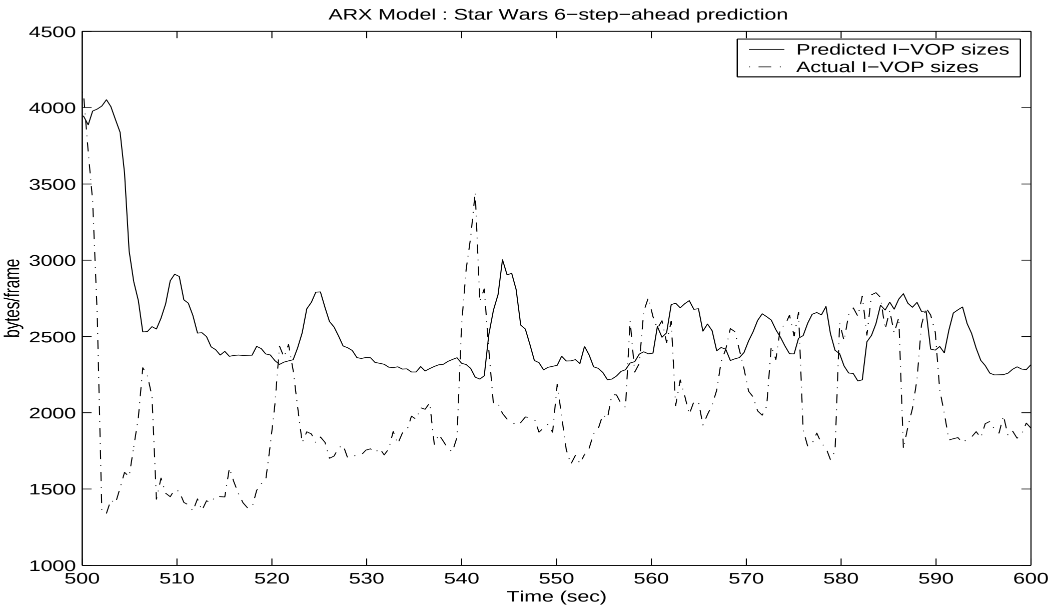

Fig. 89. Six-SP sizes of the I-VOPs of *StarWars* using ARX models.

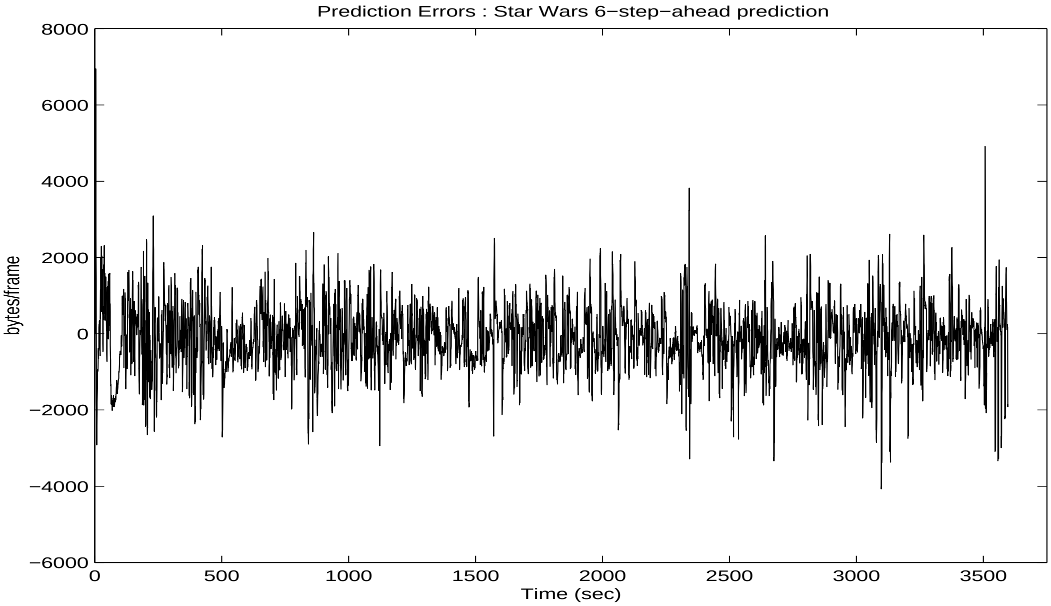

Fig. 90. Six-SP errors of I-VOP sizes of *StarWars* using ARX models.

Table XX. Performance metrics of the I-VOP for the six-SP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	7.6	8029.5	16.5
ARD Talk	4.0	5665.1	2.1
Die Hard III	10.8	5105.9	10.2
Jurassic Park I	4.3	8926.8	8.1
Lecture Room	2.7	1841.7	1.9
Silence of the Lambs	13.2	7546.6	8.0
Skiing	8.7	5210.3	2.3
StarWars	5.6	6952.2	7.5

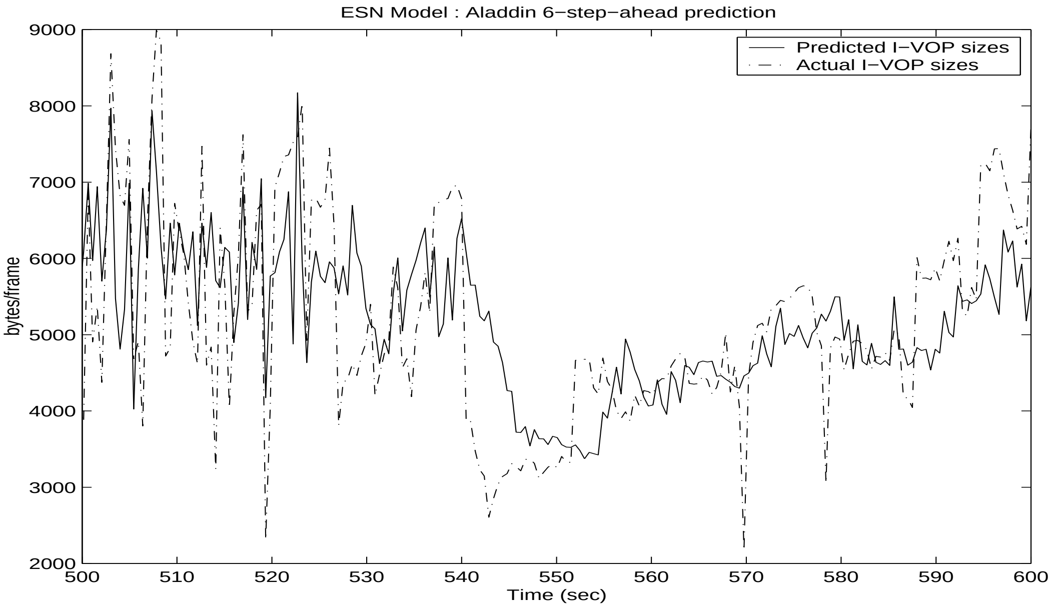

Fig. 91. Six-SP sizes of the I-VOPs of *Aladdin* using ESN models.

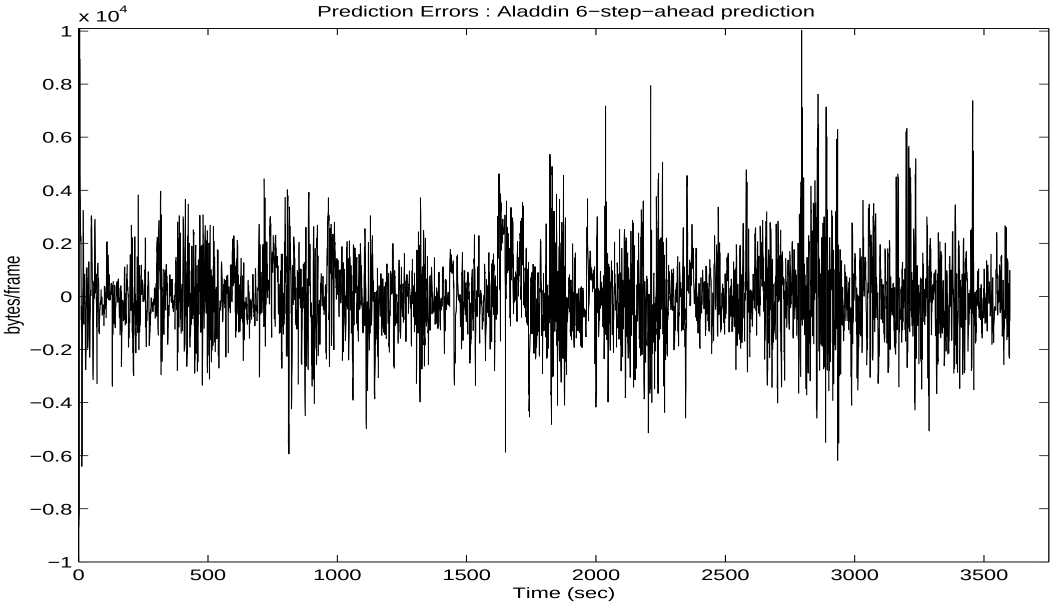

Fig. 92. Six-SP errors of I-VOP sizes of *Aladdin* using ESN models.

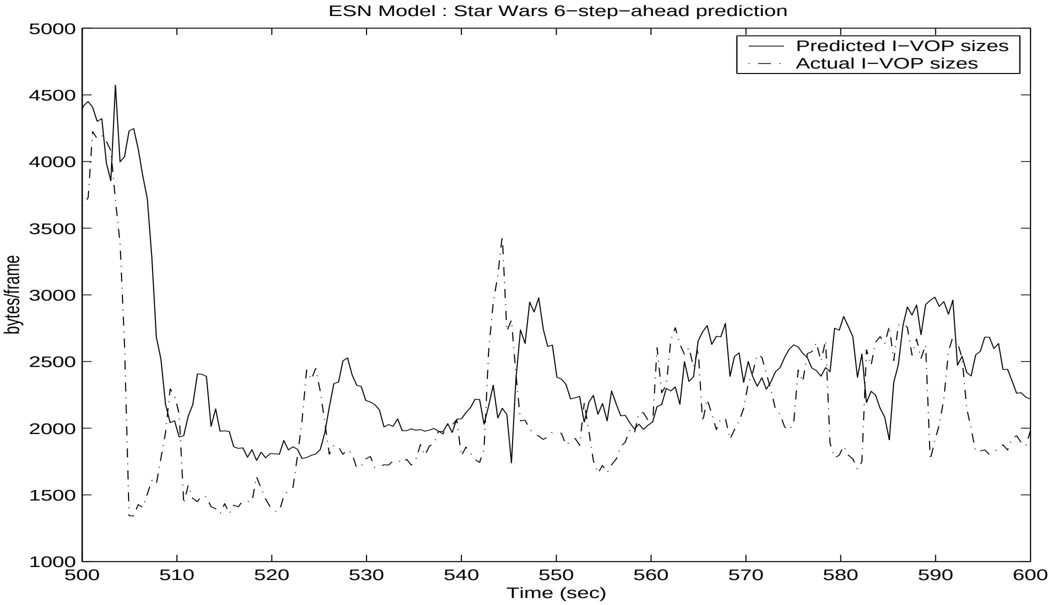

Fig. 93. Six-SP sizes of the I-VOPs of *StarWars* using ESN models.

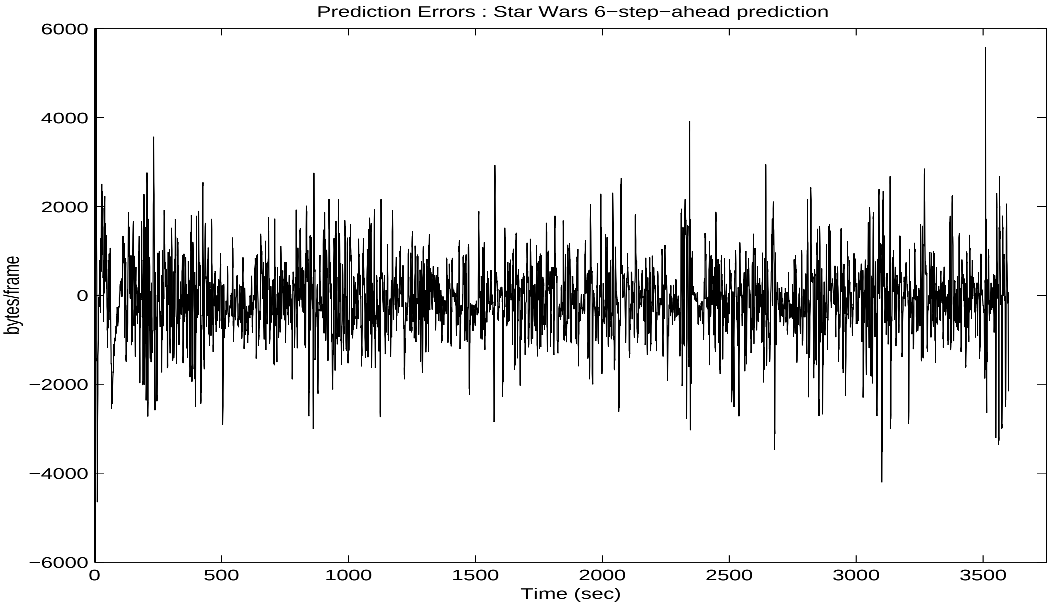

Fig. 94. Six-SP errors of I-VOP sizes of *StarWars* using ESN models.

Table XXI. Performance metrics of the I-VOP for the six-SP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	8.2	10031	15.8
ARD Talk	5.4	10759	2.0
Die Hard III	10.2	5322.5	11.1
Jurassic Park I	7.8	14490	8.8
Lecture Room	2.4	1648.8	1.6
Silence of the Lambs	14.8	8319.0	8.5
Skiing	8.4	5372.5	2.5
StarWars	5.4	5814.1	7.0

2. Prediction of Moving Average of VOPs

The six-step-ahead prediction (Six-SP) represents a time horizon of 2.88 seconds. For six-step-ahead prediction, training was done using the first 1500 points of the moving average time-series of VOP sizes of video trace *Aladdin*. The next 500 points were used for the validation of the model. The developed model was then used to generate six-step-ahead predictions for the entire length for all the eight video traces.

a. Six-step-ahead Prediction Using AR Models

The AR model used for six-step-ahead prediction was the same model that was developed for SSP. For this model, $n_y = 17$. Figures 95 to 98 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

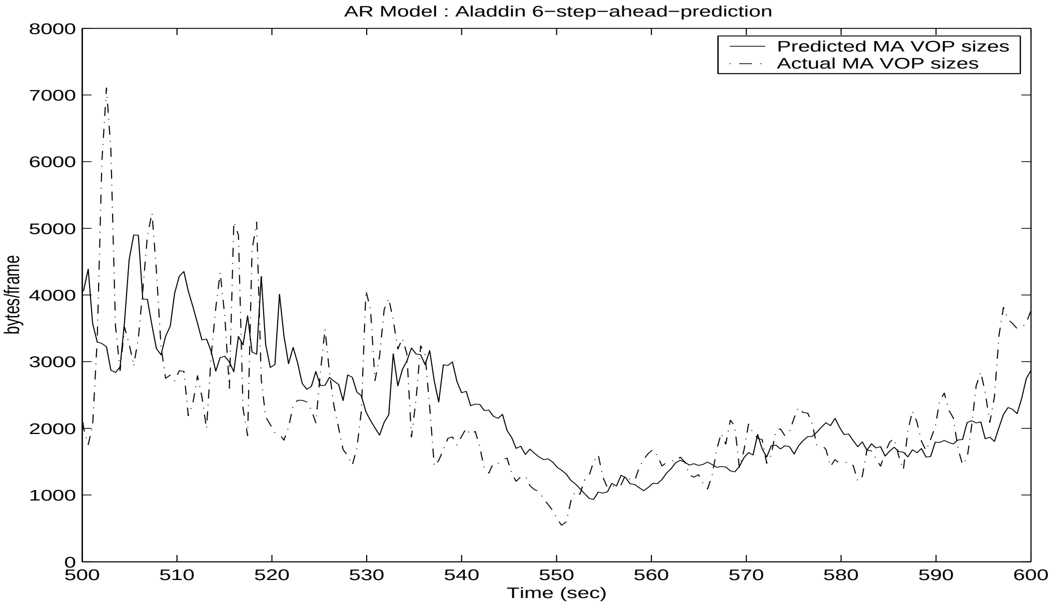

Fig. 95. Six-SP sizes of the moving average VOPs of *Aladdin* using AR models.

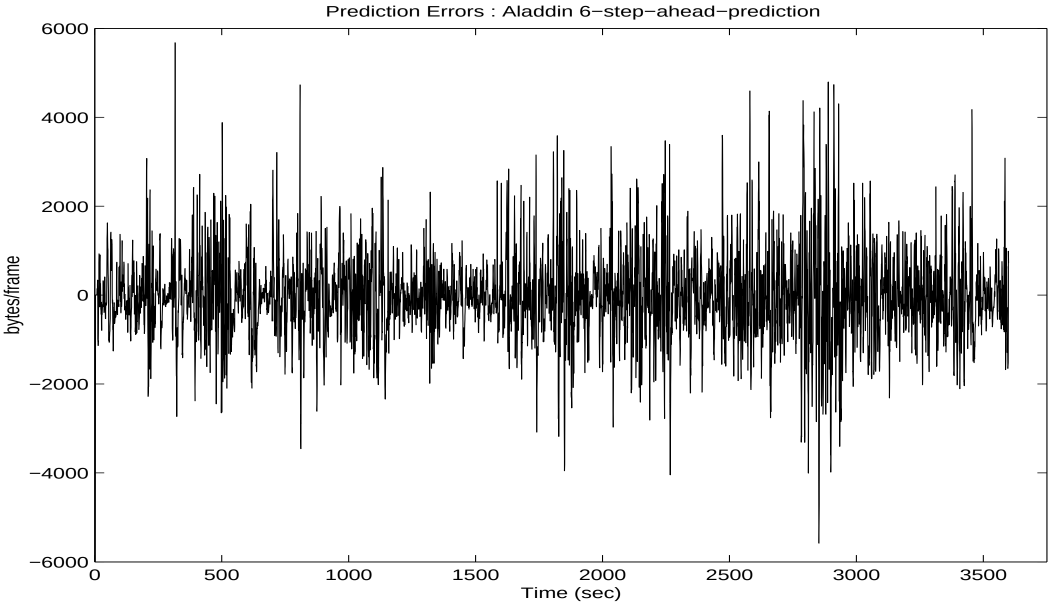

Fig. 96. Six-SP errors of moving average VOP sizes of *Aladdin* using AR models.

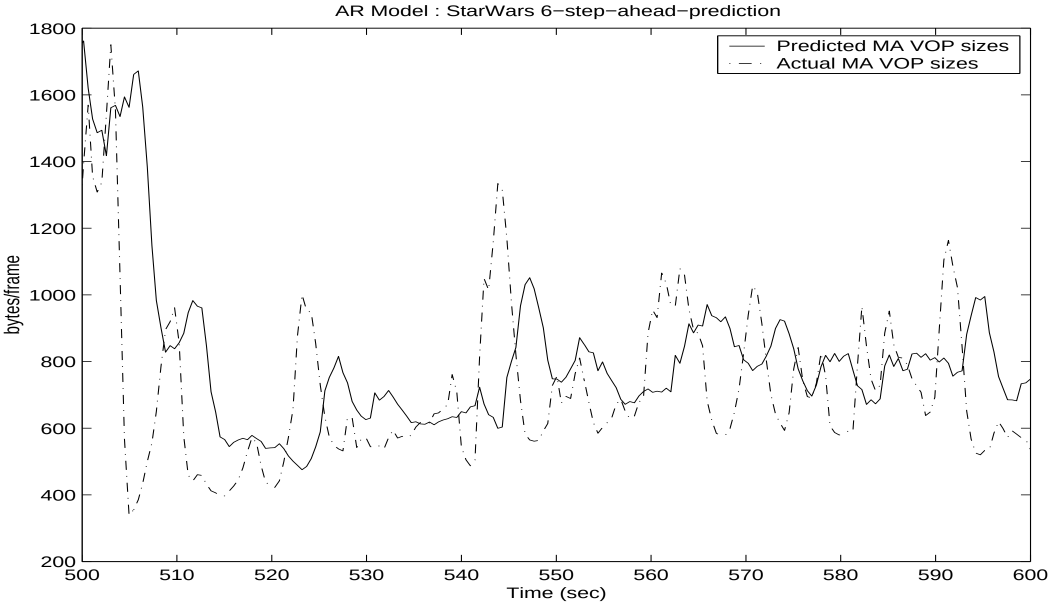

Fig. 97. Six-SP sizes of the moving average VOPs of *StarWars* using AR models.

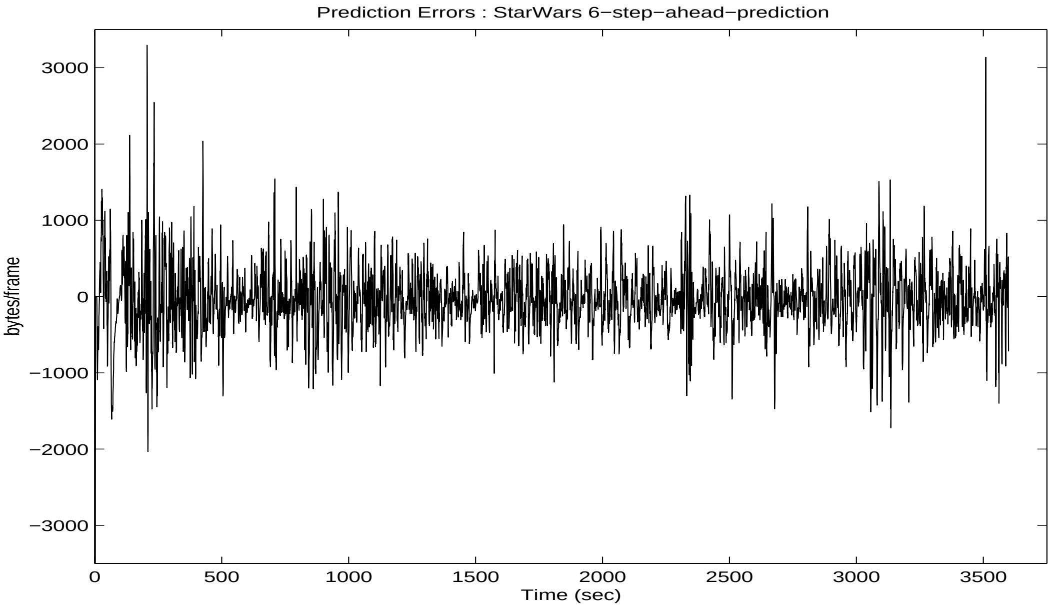

Fig. 98. Six-SP errors of moving average VOP sizes of *StarWars* using AR models.

Table XXII shows the performance of the AR predictor for the video traces used in this research in terms of the performance metrics defined earlier in the chapter.

b. Six-step-ahead Prediction Using ARX Models

In the design of ARX models for six-step-ahead prediction, other than the time-series x_{MA}, external indicators δx_{MA}, Δx_{MA} and $I-$VOPs were also used as additional inputs. The model which gave the best results had the structure: $n_y = 7$, $n_u = [1111]$ and $n_k = [1111]$. Figures 99 to 102 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XXIII shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

Table XXII. Performance metrics of the six-SP for AR models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	16.5	9326.8	37.0
ARD Talk	6.1	7880.3	4.6
Die Hard III	19.0	9132.0	20.4
Jurassic Park I	12.4	9554.1	14.2
Lecture Room	66.7	9013.5	15.9
Silence of the Lambs	28.4	7940.1	11.4
Skiing	16.1	1817.5	7.1
StarWars	9.9	3294.8	9.5

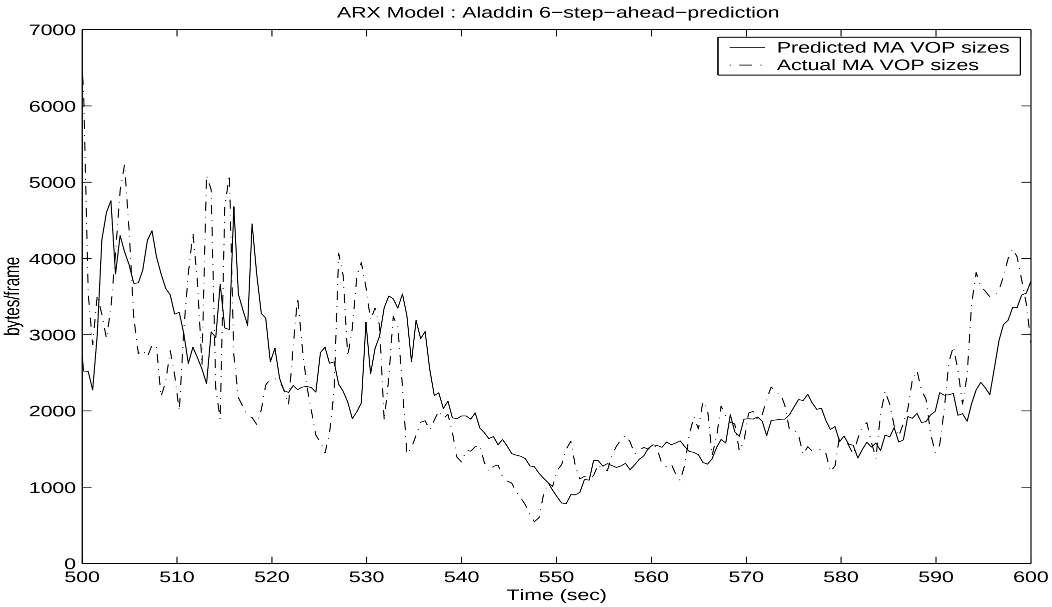

Fig. 99. Six-SP sizes of the moving average VOPs of *Aladdin* using ARX models.

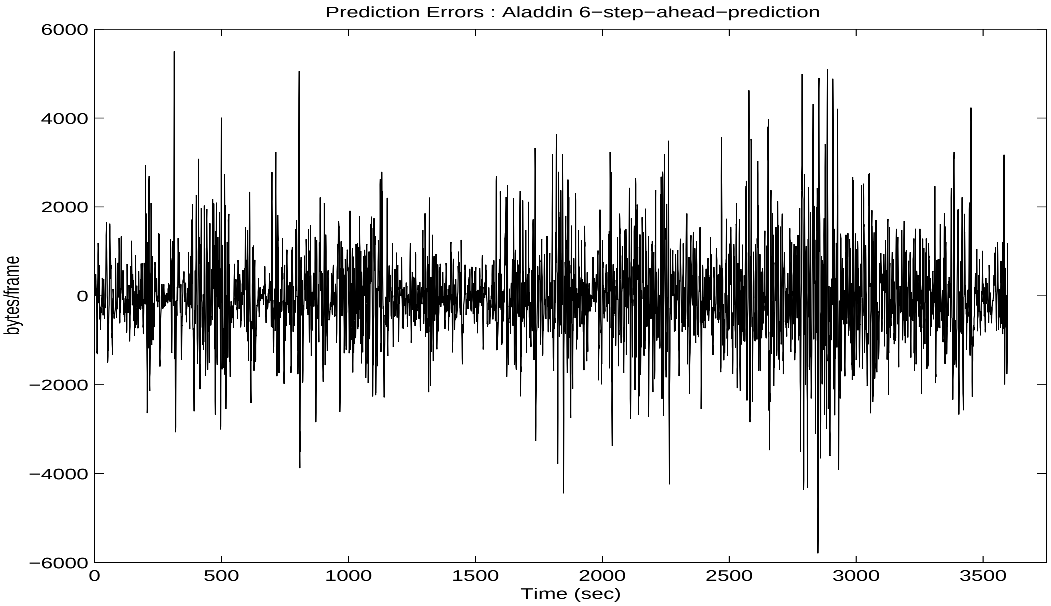

Fig. 100. Six-SP errors of moving average VOP sizes of *Aladdin* using ARX models.

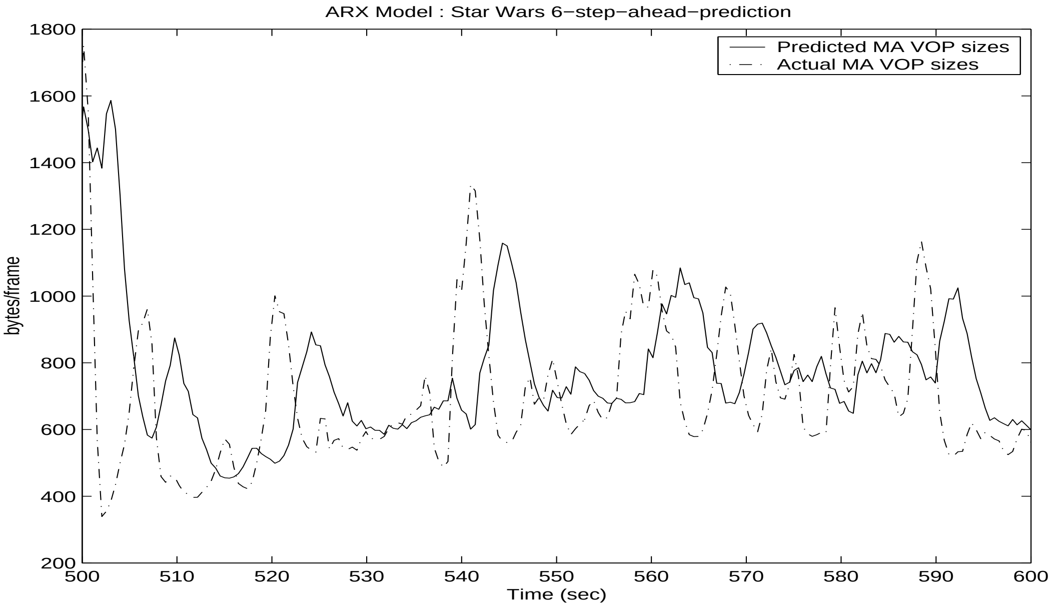

Fig. 101. Six-SP sizes of the moving average VOPs of *StarWars* using ARX models.

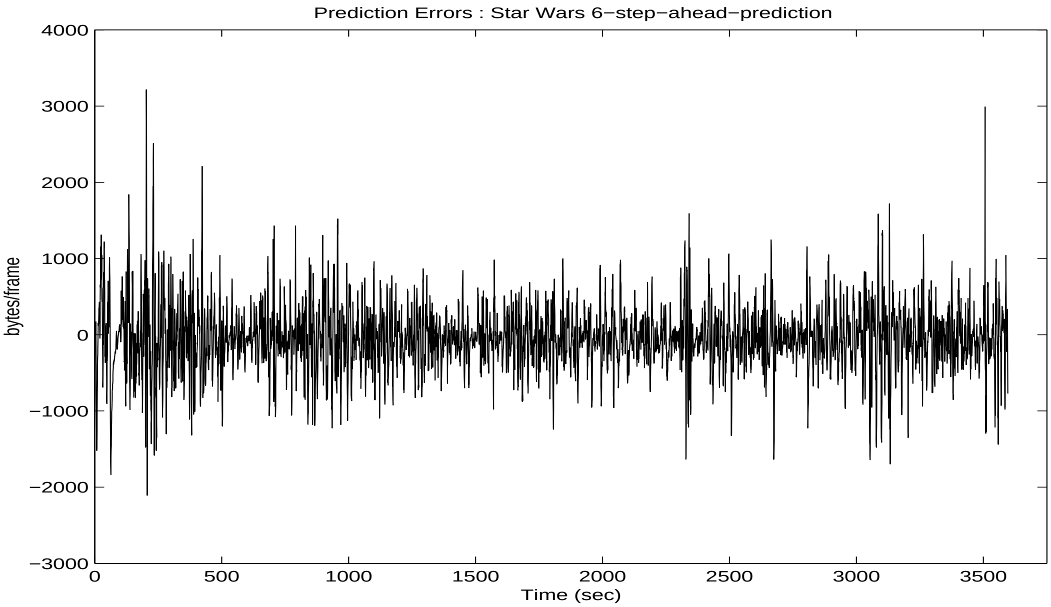

Fig. 102. Six-SP errors of moving average VOP sizes of *StarWars* using ARX models.

Table XXIII. Performance metrics of the six-SP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	18.1	5791.8	18.5
ARD Talk	6.4	5131.1	2.4
Die Hard III	17.6	4611.3	7.6
Jurassic Park I	12.3	9246.3	7.1
Lecture Room	10.7	630.1	3.7
Silence of the Lambs	21.5	4356.5	10.5
Skiing	17.5	2039.8	8.8
StarWars	10.4	3215.2	9.2

c. Six-step-ahead Prediction Using FMLP Models

The six-step-ahead FMLP model was designed using the external indicators. The FMLP structure which gave the best results was $12 - 30 - 1$. For the prediction of $x_{MA}(k)$, the 12 inputs were $x_{MA}(k-6)$, $x_{MA}(k-7)$, $x_{MA}(k-8)$, $\delta x_{MA}(k-6)$, $\delta x_{MA}(k-7)$, $\delta x_{MA}(k-8)$, $\Delta x_{MA}(k-6)$, $\Delta x_{MA}(k-7)$, $\Delta x_{MA}(k-8)$, $I_1(k-6)$, $I_2(k-6)$, and $I_3(k-6)$. Figures 103 to 106 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

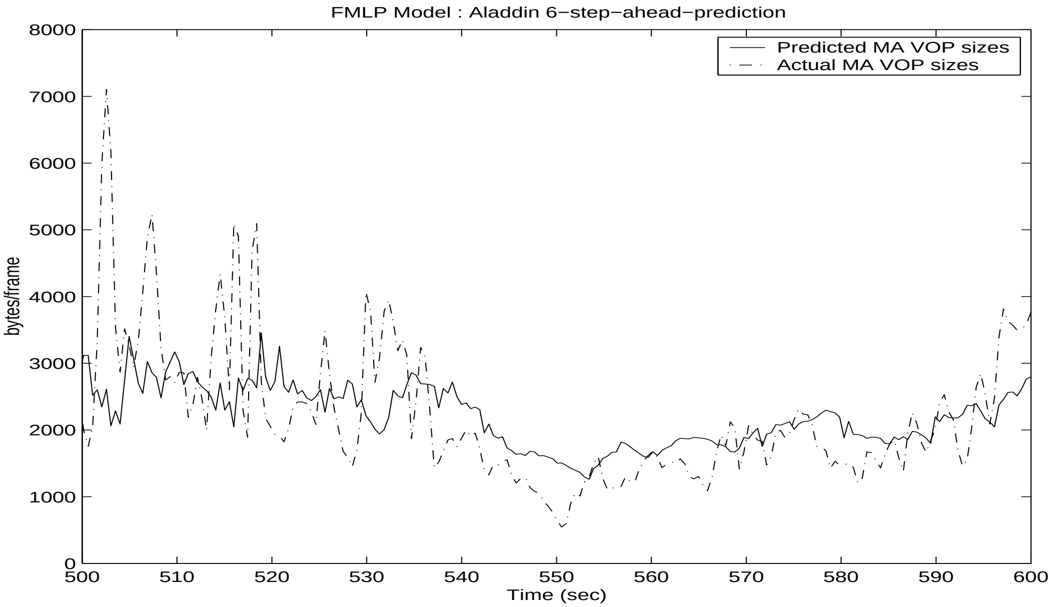

Fig. 103. Six-SP sizes of the moving average VOPs of *Aladdin* using FMLP models.

Table XXIV shows the performance of the FMLP six-step-ahead predictor for the video traces used in this research in terms of the performance metrics.

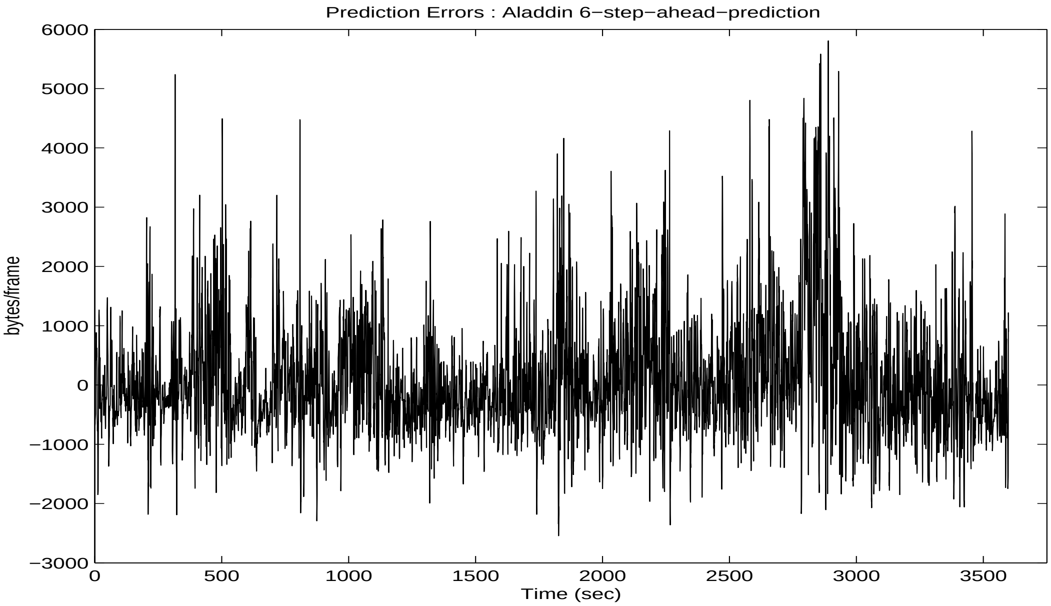

Fig. 104. Six-SP errors of moving average VOP sizes of *Aladdin* using FMLP models.

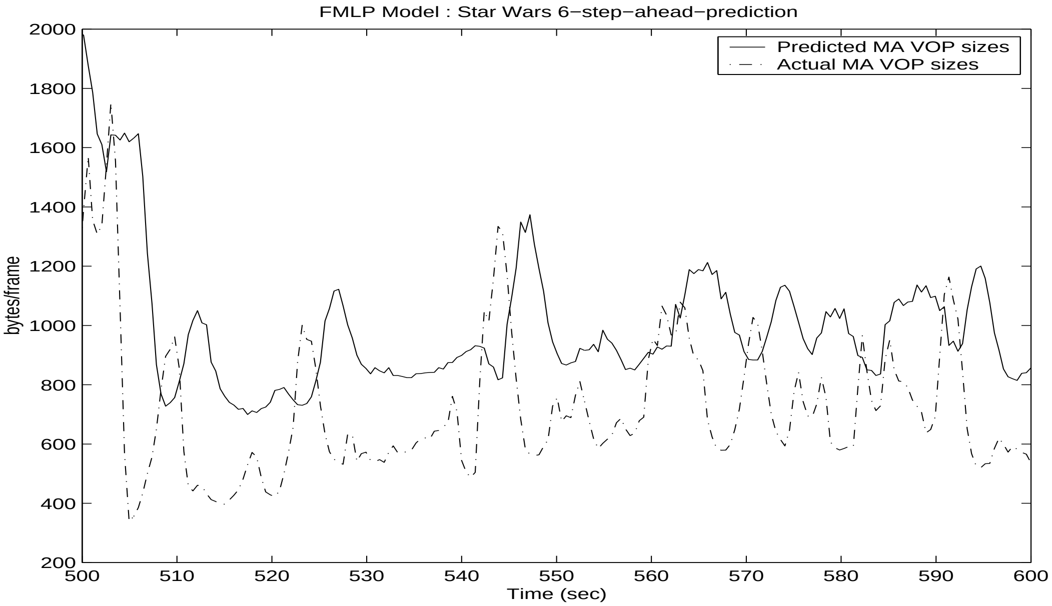

Fig. 105. Six-SP sizes of the moving average VOPs of *StarWars* using FMLP models.

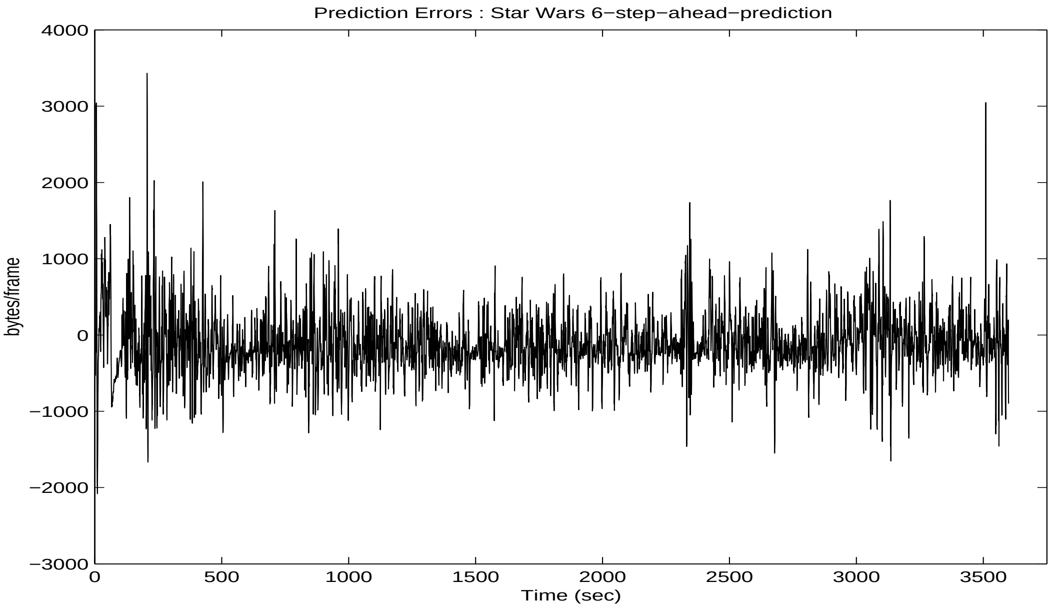

Fig. 106. Six-SP errors of moving average VOP sizes of *StarWars* using FMLP models.

Table XXIV. Performance metrics of the six-SP for FMLP models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	15.3	5807.7	25.3
ARD Talk	6.1	6359.1	1.3
Die Hard III	15.3	3986.8	8.9
Jurassic Park I	21.6	13105	3.8
Lecture Room	73.3	677.1	2.7
Silence of the Lambs	33.3	2945.3	12.3
Skiing	20.0	1825.9	9.5
StarWars	8.2	3432.8	10.6

d. Six-step-ahead Prediction Using RMLP Models

As in [34], the six-step-ahead RMLP model was designed using the external indicators. The RMLP structure which gave the best results was $16 - 17 - 1$. The training was done using global feedback. For the prediction of $x_{MA}(k)$, 11 of the 16 inputs were $x_{MA}(k-6)$, $x_{MA}(k-7)$, $x_{MA}(k-8)$, $\delta x_{MA}(k-6)$, $\delta x_{MA}(k-7)$, $\delta x_{MA}(k-8)$, $\Delta x_{MA}(k-6)$, $\Delta x_{MA}(k-7)$, $\Delta x_{MA}(k-8)$, $I_1(k-6)$ and $I_2(k-6)$. The remaining 5 inputs were the past predictions of the model, that is $\widehat{x}^*_{MA}(k-1)$, $\widehat{x}^*_{MA}(k-2)$, $\widehat{x}^*_{MA}(k-3)$, $\widehat{x}^*_{MA}(k-4)$ and $\widehat{x}^*_{MA}(k-5)$. Figures 107 to 110 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

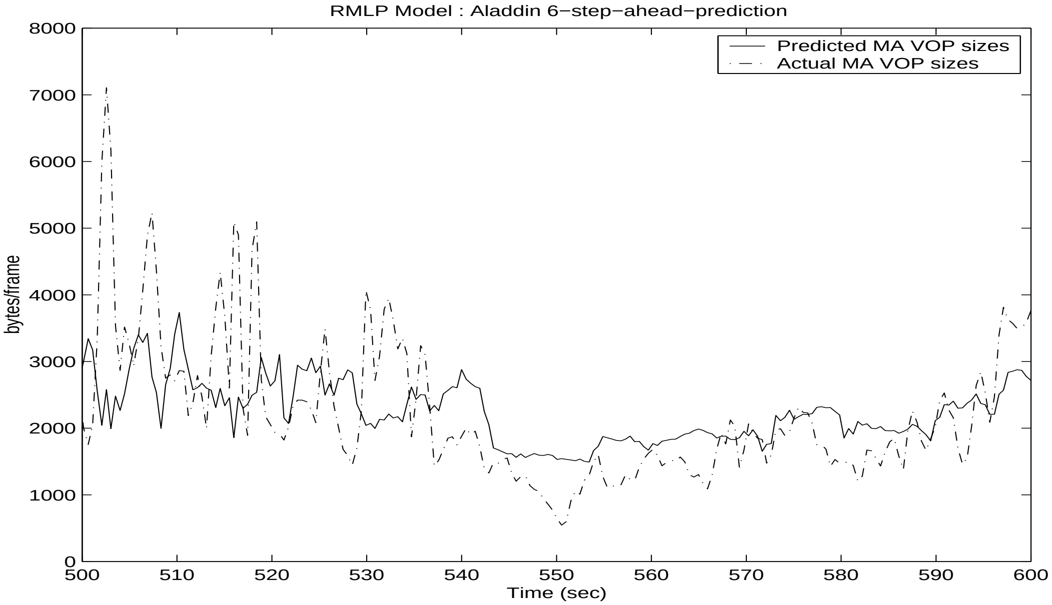

Fig. 107. Six-SP sizes of the moving average VOPs of *Aladdin* using RMLP models.

Table XXV shows the performance of the RMLP six-step-ahead predictor for the video traces used in this research in terms of the performance metrics.

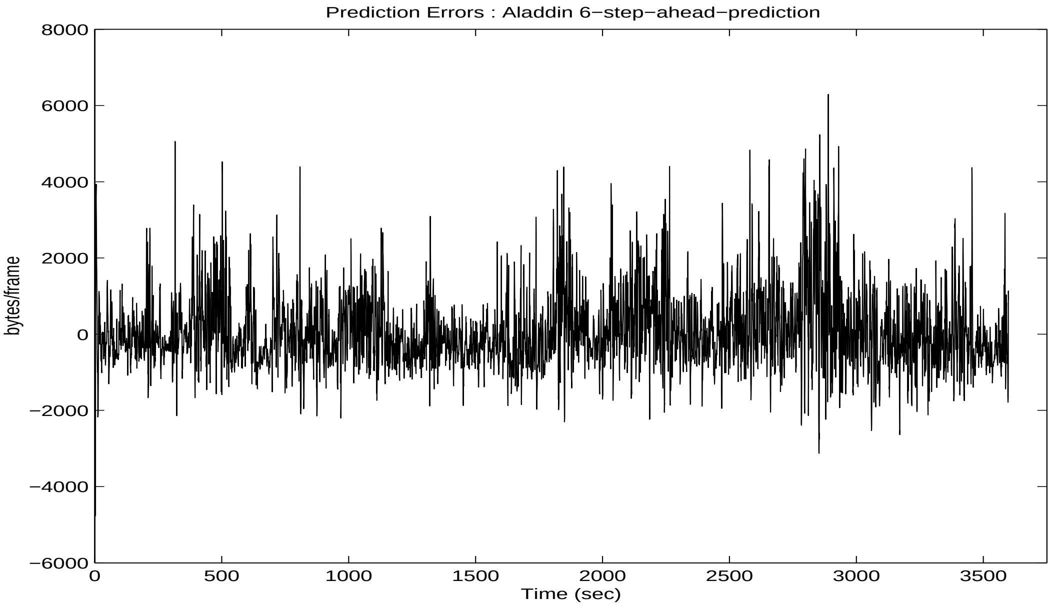

Fig. 108. Six-SP errors of moving average VOP sizes of *Aladdin* using RMLP models.

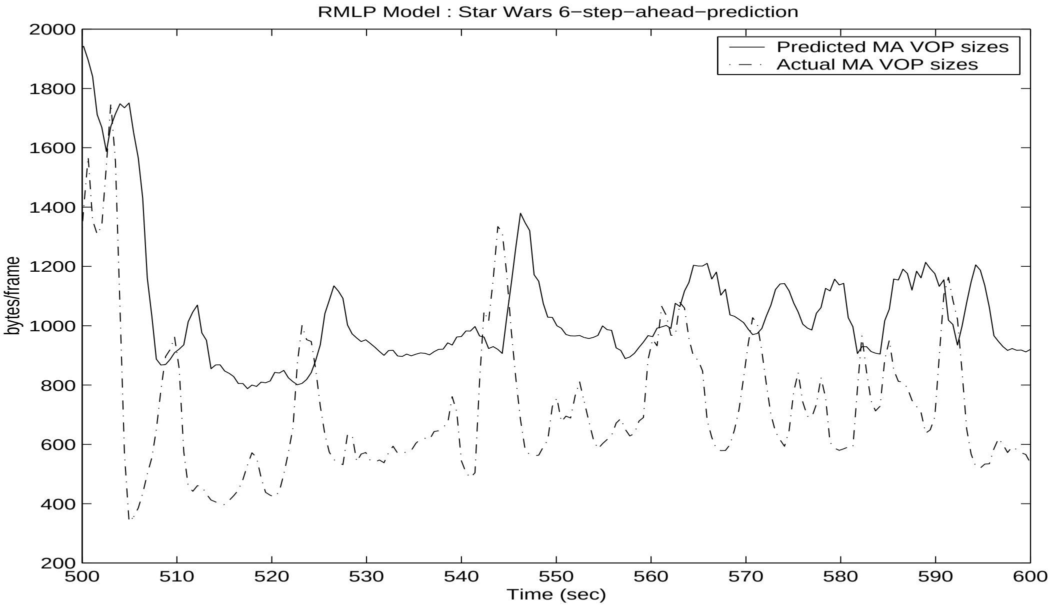

Fig. 109. Six-SP sizes of the moving average VOPs of *StarWars* using RMLP models.

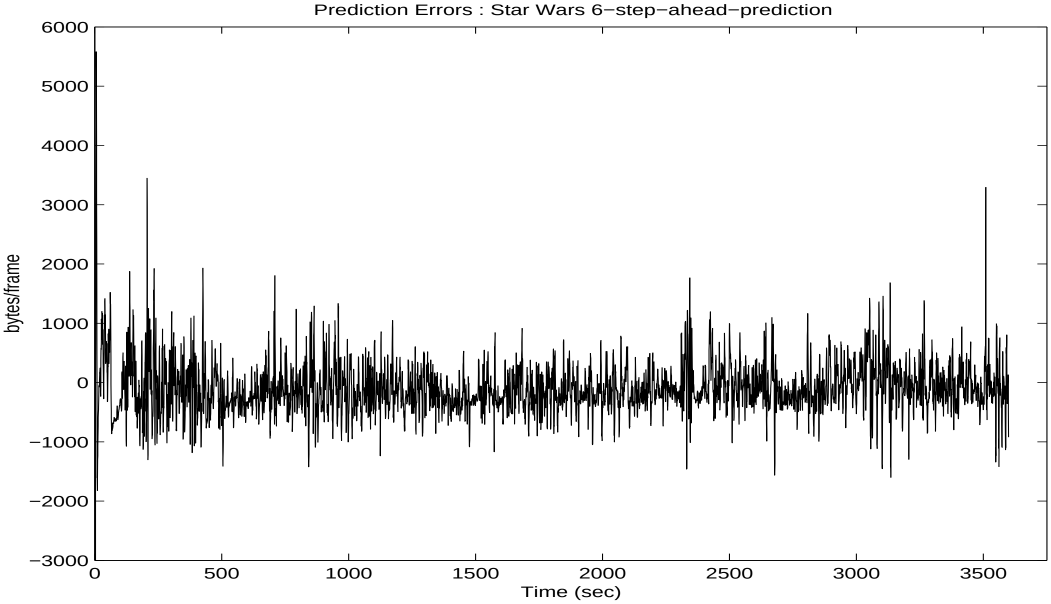

Fig. 110. Six-SP errors of moving average VOP sizes of *StarWars* using RMLP models.

Table XXV. Performance metrics of the six-SP for RMLP models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	15.6	6296.4	29.8
ARD Talk	5.9	5676.8	2.1
Die Hard III	18.5	4611.3	10.8
Jurassic Park I	11.3	9125.0	15.2
Lecture Room	140.0	4699.9	9.5
Silence of the Lambs	45.3	3473.9	13.3
Skiing	29.2	4934.4	19.0
StarWars	9.8	5581.4	36.2

e. Six-step-ahead Prediction Using ESN Models

The network structure which gave the best results for six-step-ahead prediction of the moving average of VOP size time-series was $4 - 30 - 1$. In order to predict the moving average of the VOP sizes at time step k the four inputs were $x_{MA}(k-6)$, $\delta x_{MA}(k-6)$, $\Delta x_{MA}(k-6)$ and $I_1(k-6)$. Figures 111 to 114 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XXVI shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

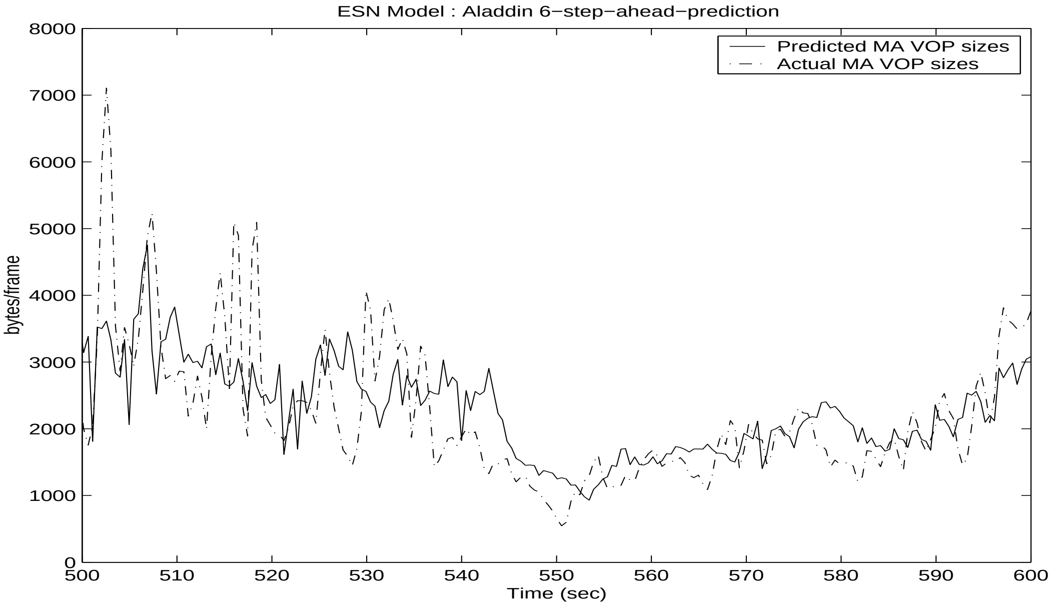

Fig. 111. Six-SP sizes of the moving average VOPs of *Aladdin* using ESN models.

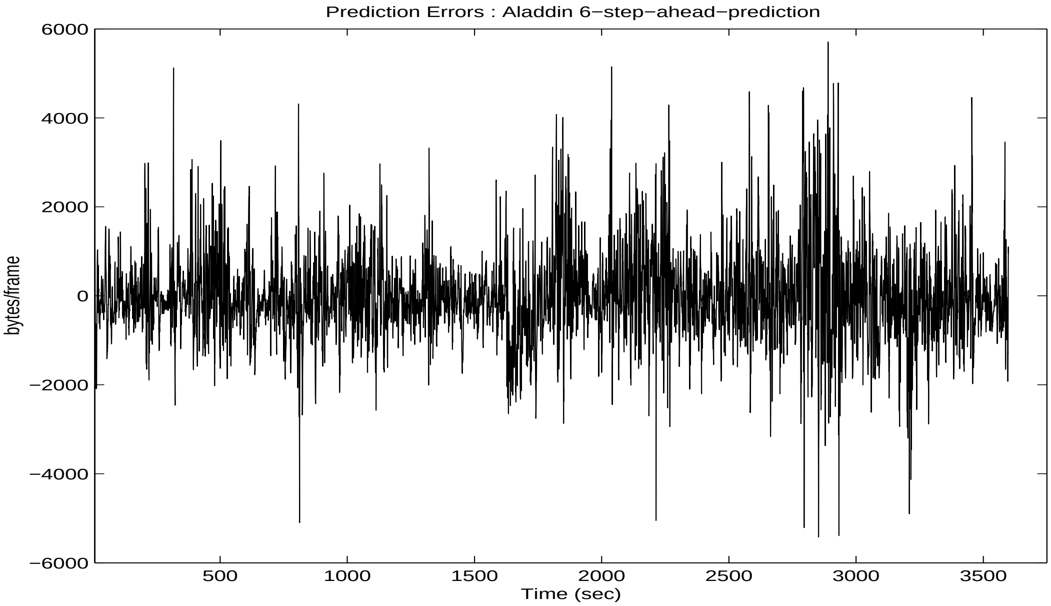

Fig. 112. Six-SP errors of moving average VOP sizes of *Aladdin* using ESN models.

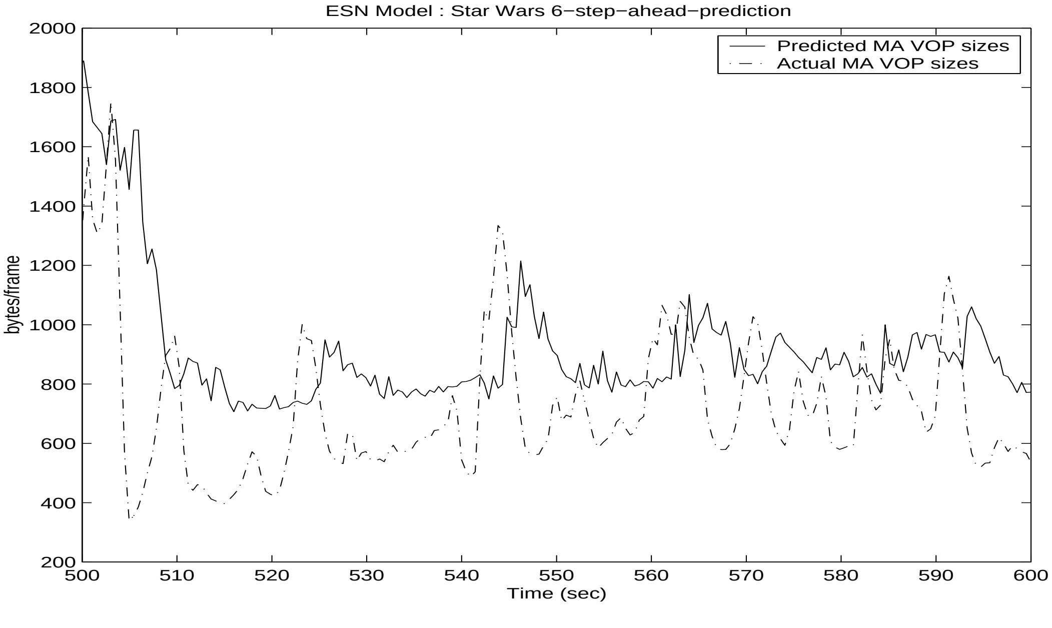

Fig. 113. Six-SP sizes of the moving average VOPs of *StarWars* using ESN models.

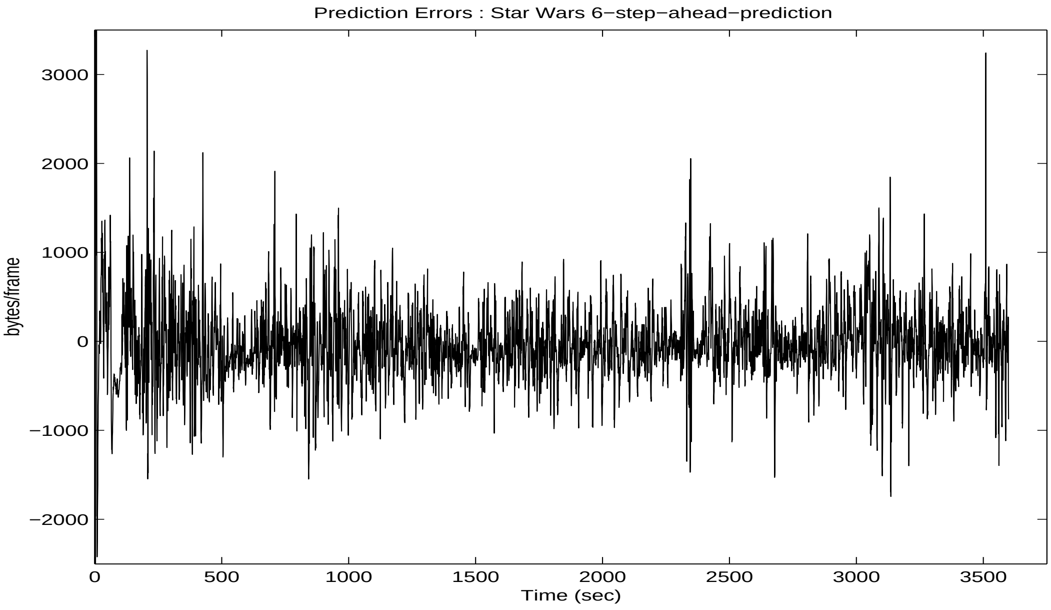

Fig. 114. Six-SP errors of moving average VOP sizes of *StarWars* using ESN models.

Table XXVI. Performance metrics of the six-SP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	14.6	5727.6	11.9
ARD Talk	9.1	6140.6	2.1
Die Hard III	15.3	4324.9	5.9
Jurassic Park I	11.5	11938	6.6
Lecture Room	3.8	700.6	2.1
Silence of the Lambs	32.6	2945.3	12.3
Skiing	22.6	3508.3	10.4
StarWars	8.1	3269.7	9.7

F. Ten-step-ahead Prediction

1. Prediction of I-VOPs

The ten-step-ahead prediction (ten-SP) of individual I-VOPs means predicting 4.8 seconds ahead. The training was done using the first 1500 points of the I-VOP size time-series of video trace *Aladdin*. The next 500 points were used for the validation of the model.

a. Ten-step-ahead Prediction Using ARX Models

The indicators used for the ARX model were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. The model structure which gives the best results is $n_y = 11$, $n_u = 1$ and $n_k = 1$. Figures 115 to 118 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XXVII shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

b. Ten-step-ahead Prediction Using ESN Models

The network structure which gave the best results was $8 - 20 - 1$. The inputs and indicators used were $I-$VOPs, δI, ΔI and the $P-$VOPs [34]. Figures 119 to 122 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XXVIII shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

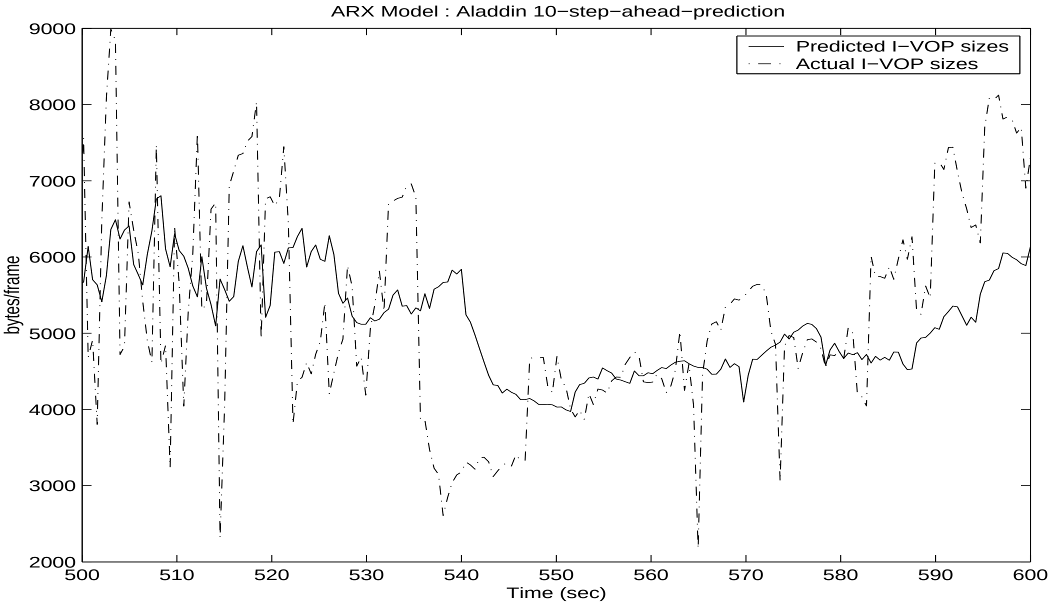

Fig. 115. Ten-SP sizes of the I-VOPs of *Aladdin* using ARX models.

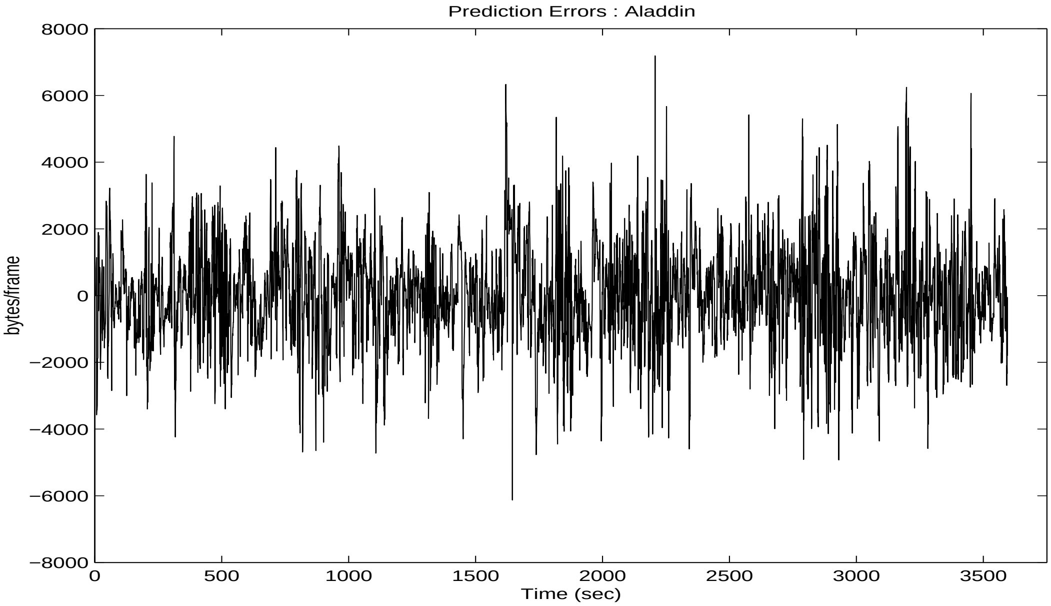

Fig. 116. Ten-SP errors of I-VOP sizes of *Aladdin* using ARX models.

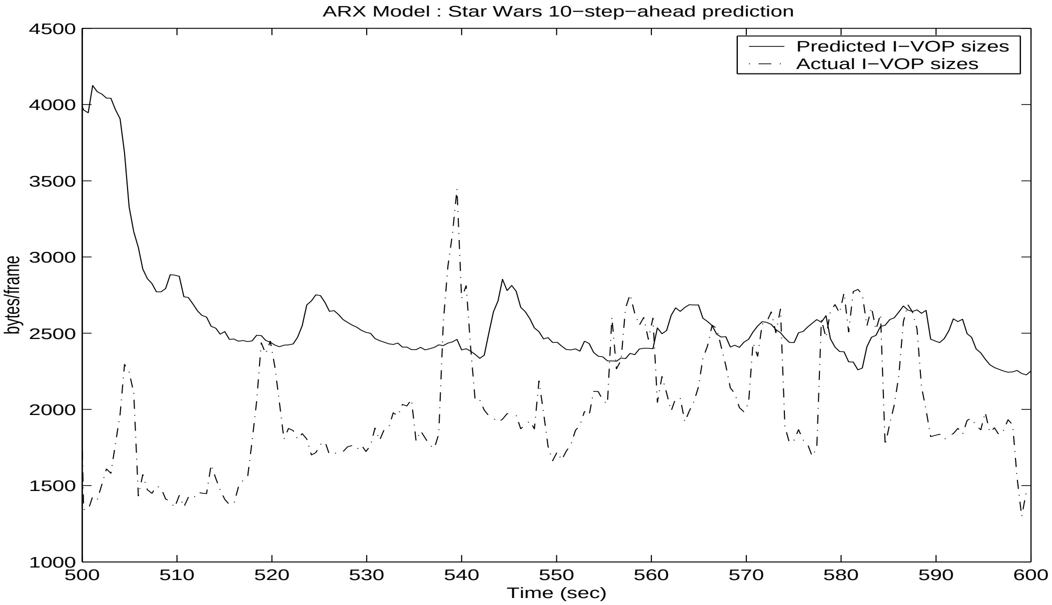

Fig. 117. Ten-SP sizes of the I-VOPs of *StarWars* using ARX models.

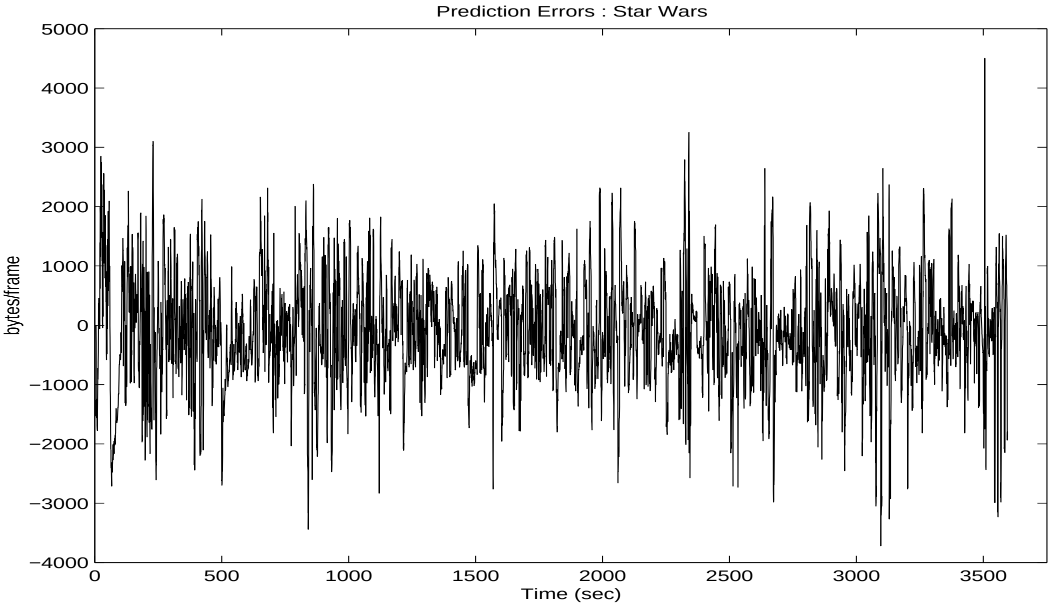

Fig. 118. Ten-SP errors of I-VOP sizes of *StarWars* using ARX models.

Table XXVII. Performance metrics of the I-VOP for the ten-SP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	8.1	7187.7	15.6
ARD Talk	5.0	6071.5	2.7
Die Hard III	11.5	4858.1	10.3
Jurassic Park I	5.3	8471.0	7.0
Lecture Room	3.3	1920.5	2.0
Silence of the Lambs	15.7	6780.9	6.3
Skiing	10.2	5022.6	2.6
StarWars	6.2	4500.5	7.2

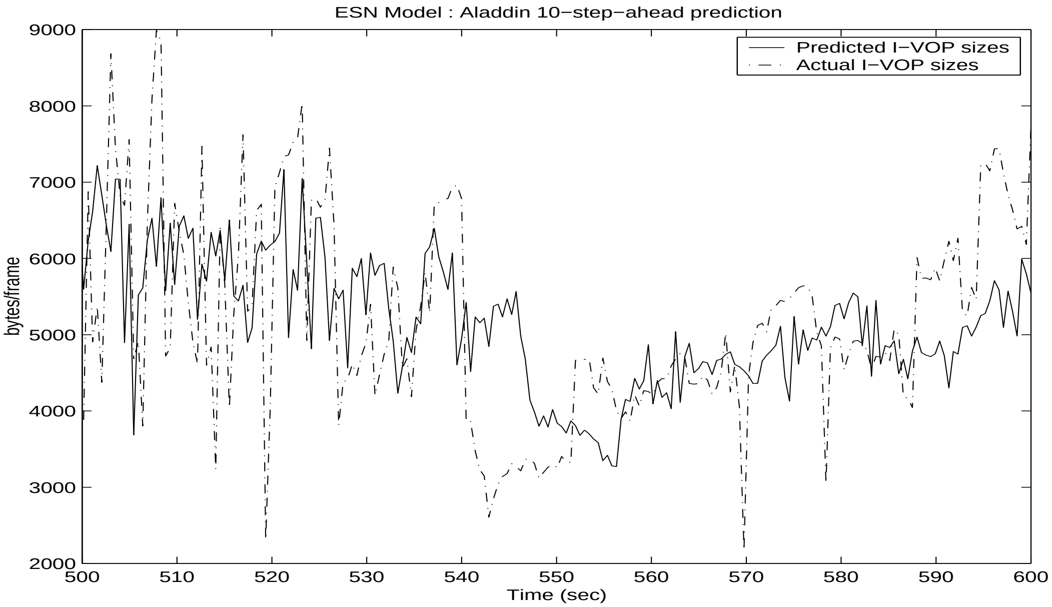

Fig. 119. Ten-SP sizes of the I-VOPs of *Aladdin* using ESN models.

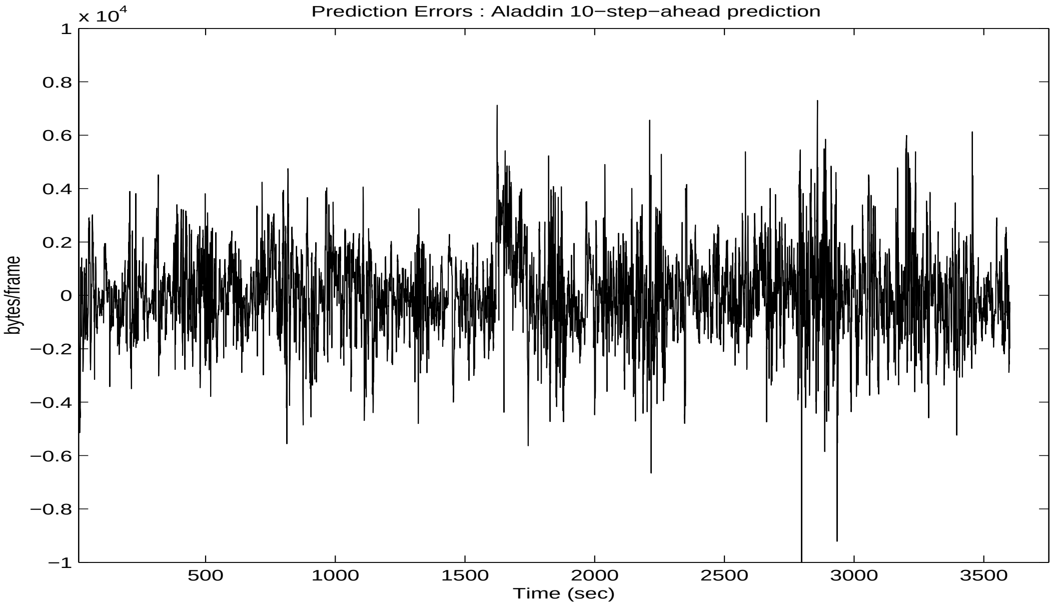

Fig. 120. Ten-SP errors of I-VOP sizes of *Aladdin* using ESN models.

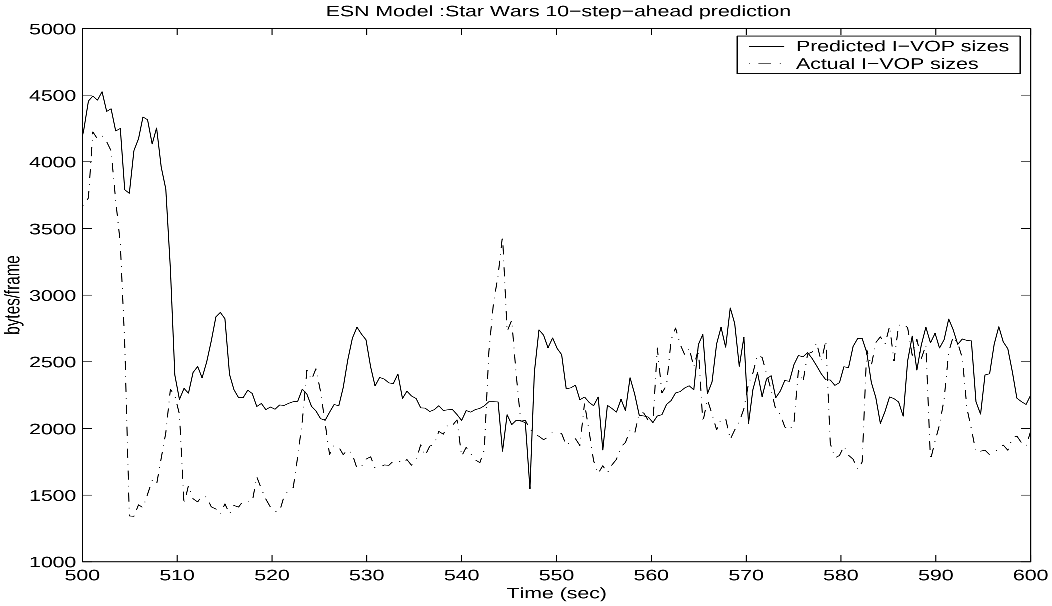

Fig. 121. Ten-SP sizes of the I-VOPs of *StarWars* using ESN models.

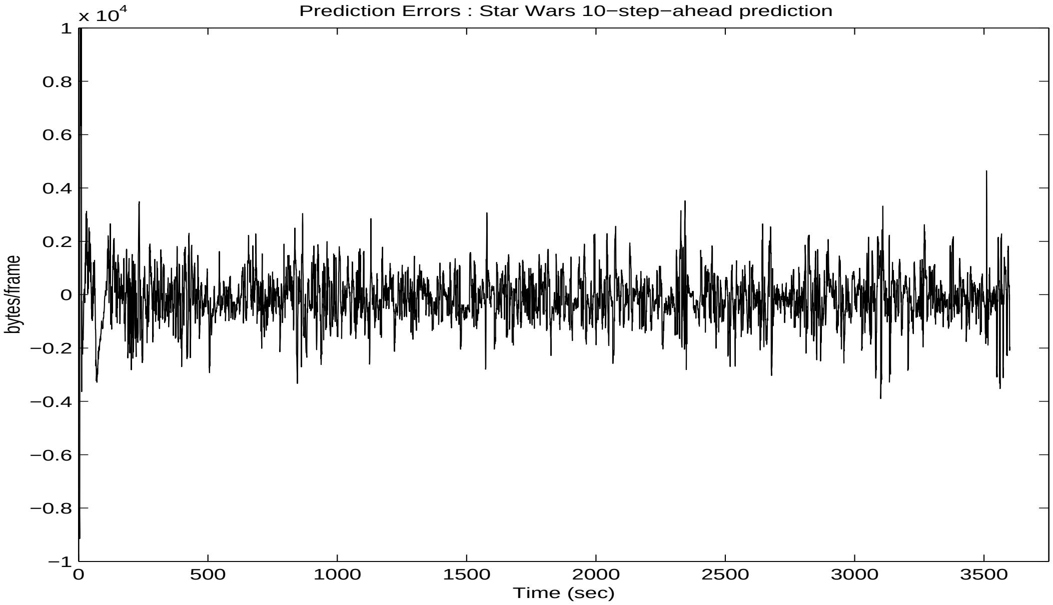

Fig. 122. Ten-SP errors of I-VOP sizes of *StarWars* using ESN models.

Table XXVIII. Performance metrics of the I-VOP for the ten-SP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	8.5	9167.4	14.4
ARD Talk	6.4	6746.3	1.7
Die Hard III	11.3	7328.1	9.9
Jurassic Park I	8.8	14991	9.8
Lecture Room	2.9	1723.1	3.1
Silence of the Lambs	17.0	8404.5	7.6
Skiing	10.1	5687.0	2.7
StarWars	6.9	9544.7	18.7

2. Prediction of Moving Average of VOPs

The ten-step-ahead prediction (Ten-SP) represents a time horizon of 4.80 seconds. For ten-step-ahead prediction, training was done using the first 1500 points of the moving average time-series of VOP sizes of video trace *Aladdin*. The next 500 points were used for the validation of the model. The developed model was then used to generate ten-step-ahead predictions for the entire length for all the eight video traces.

a. Ten-step-ahead Prediction Using AR Models

The AR model used for ten-step-ahead prediction was the same model that was developed for SSP. For this model, $n_y = 17$. Figures 123 to 126 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

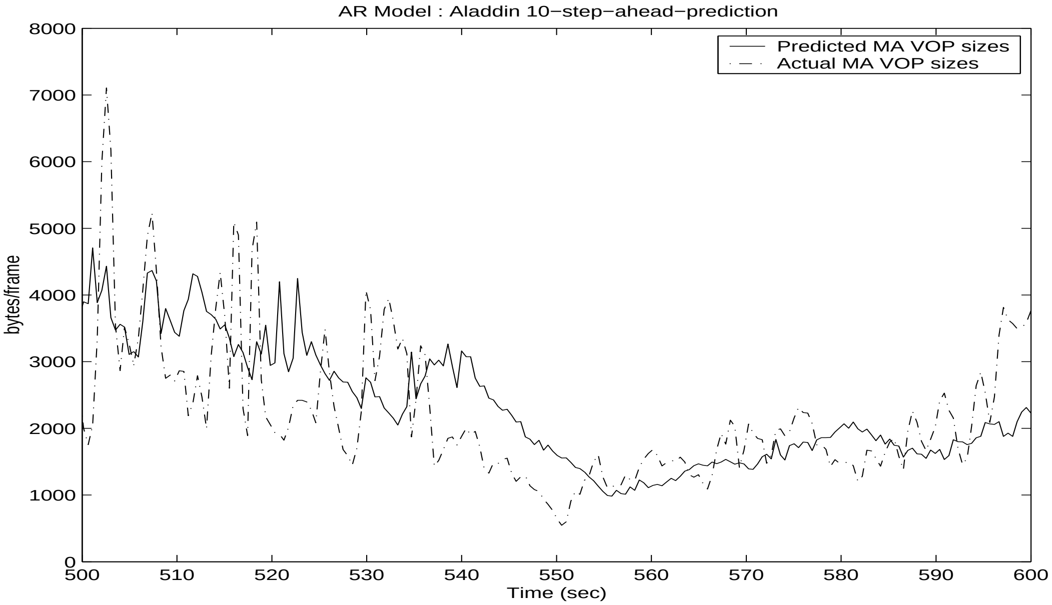

Fig. 123. Ten-SP sizes of the moving average VOPs of *Aladdin* using AR models.

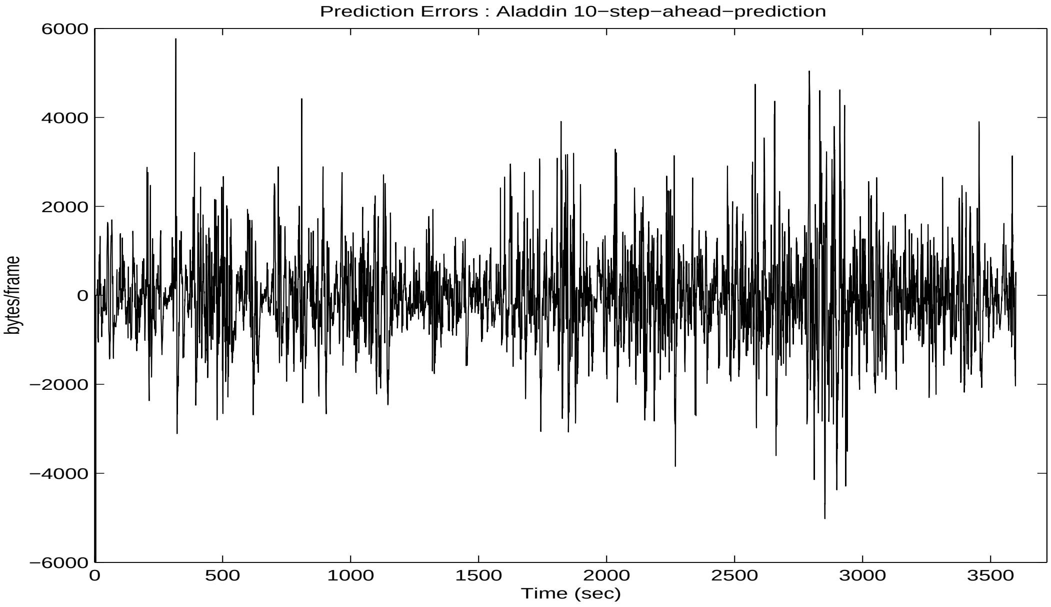

Fig. 124. Ten-SP errors of moving average VOP sizes of *Aladdin* using AR models.

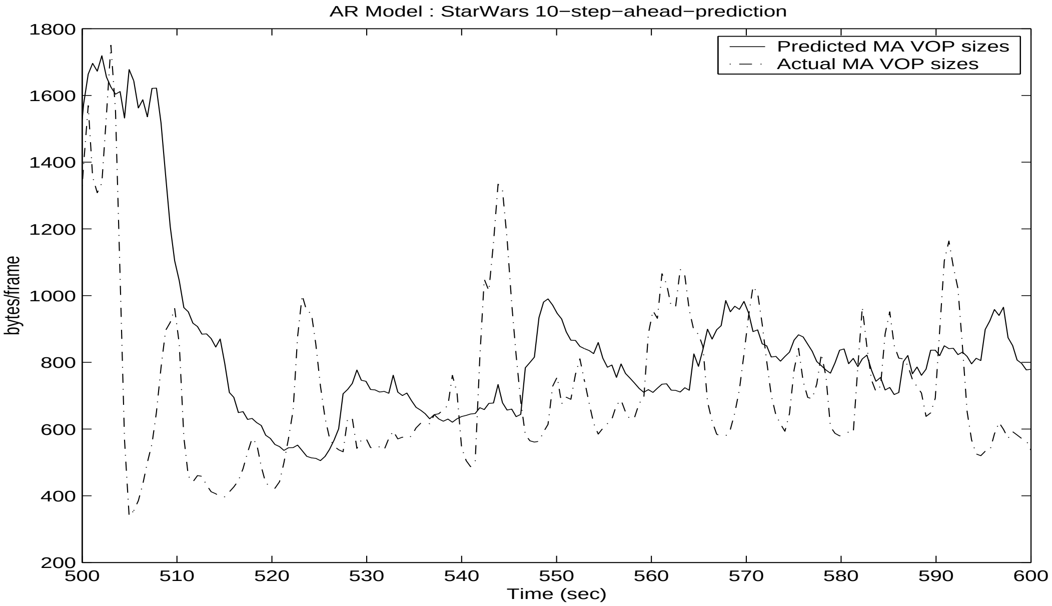

Fig. 125. Ten-SP sizes of the moving average VOPs of *StarWars* using AR models.

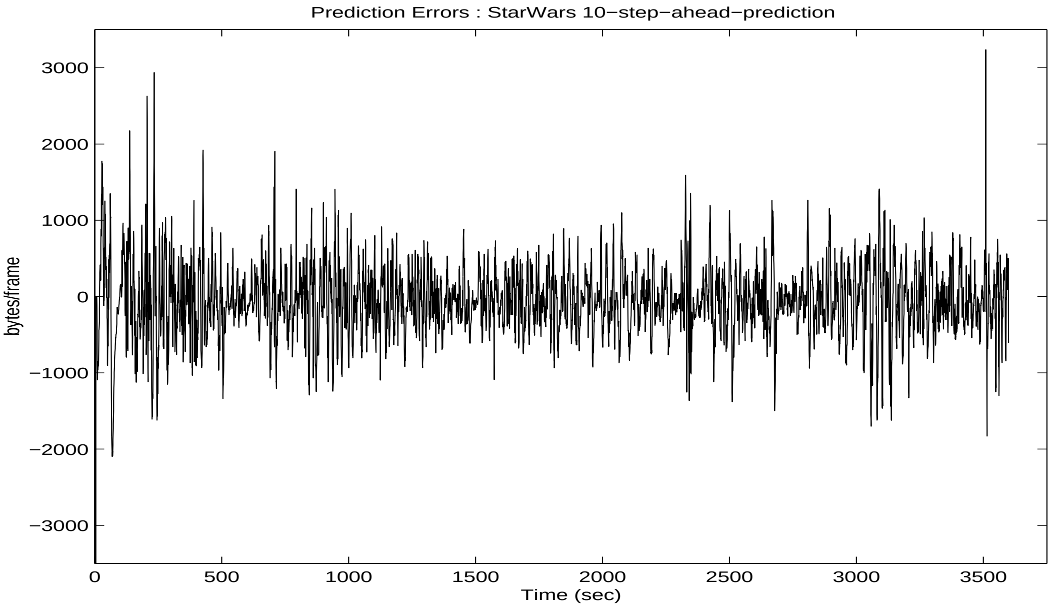

Fig. 126. Ten-SP errors of moving average VOP sizes of *StarWars* using AR models.

Table XXIX shows the performance of the AR predictor for the video traces used in this research in terms of the performance metrics defined earlier in the chapter.

b. Ten-step-ahead Prediction Using ARX Models

In the design of ARX models for ten-step-ahead prediction, other than the time-series x_{MA}, external indicators δx_{MA}, Δx_{MA} and $I-$VOPs were also used as additional inputs. The model which gave the best results had the structure: $n_y = 11$, $n_u = [1111]$ and $n_k = [1111]$. Figures 127 to 130 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XXX shows the performance of the ARX predictor for the video traces used in this research in terms of the three performance metrics.

Table XXIX. Performance metrics of the ten-SP for AR models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	17.9	9326.8	37.0
ARD Talk	7.7	7930.1	4.8
Die Hard III	24.9	9132.0	20.4
Jurassic Park I	16.1	9571.2	28.0
Lecture Room	120.4	9170.5	22.5
Silence of the Lambs	41.5	7940.1	11.1
Skiing	19.7	1770.0	6.8
StarWars	11.8	3237.7	10.3

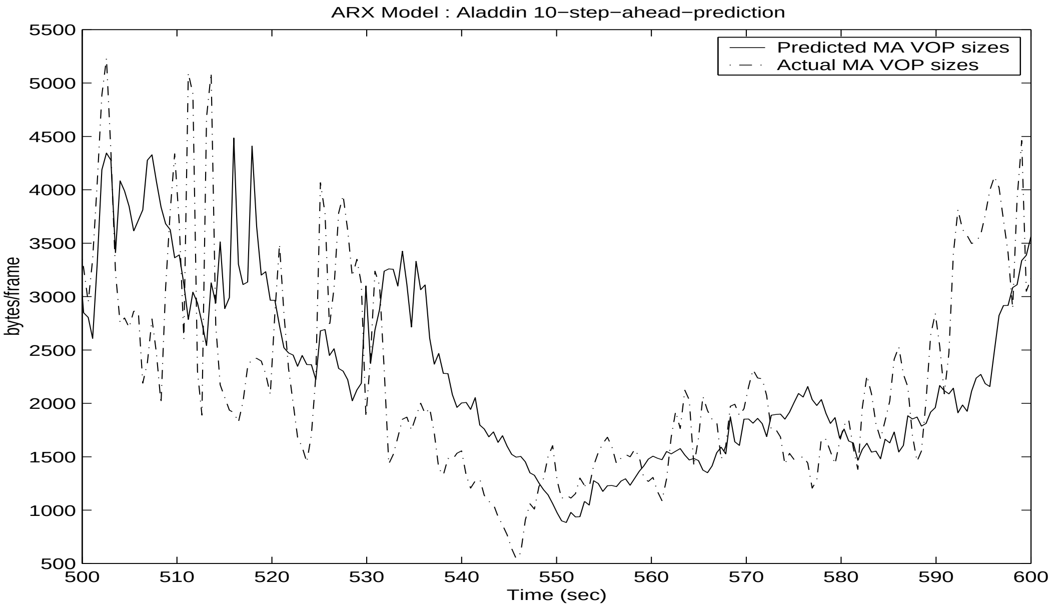

Fig. 127. Ten-SP sizes of the moving average VOPs of *Aladdin* using ARX models.

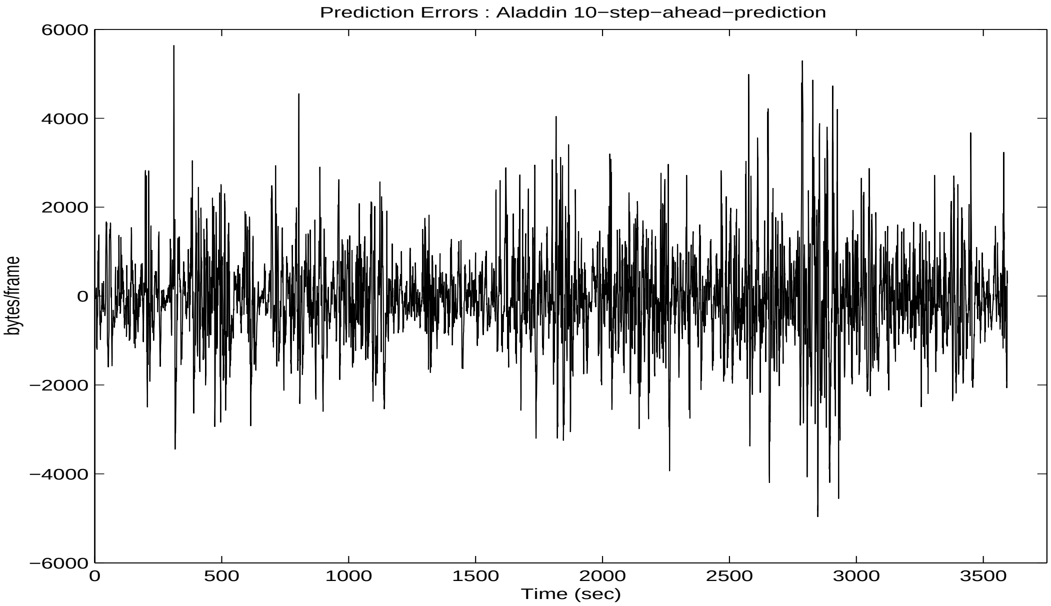

Fig. 128. Ten-SP errors of moving average VOP sizes of *Aladdin* using ARX models.

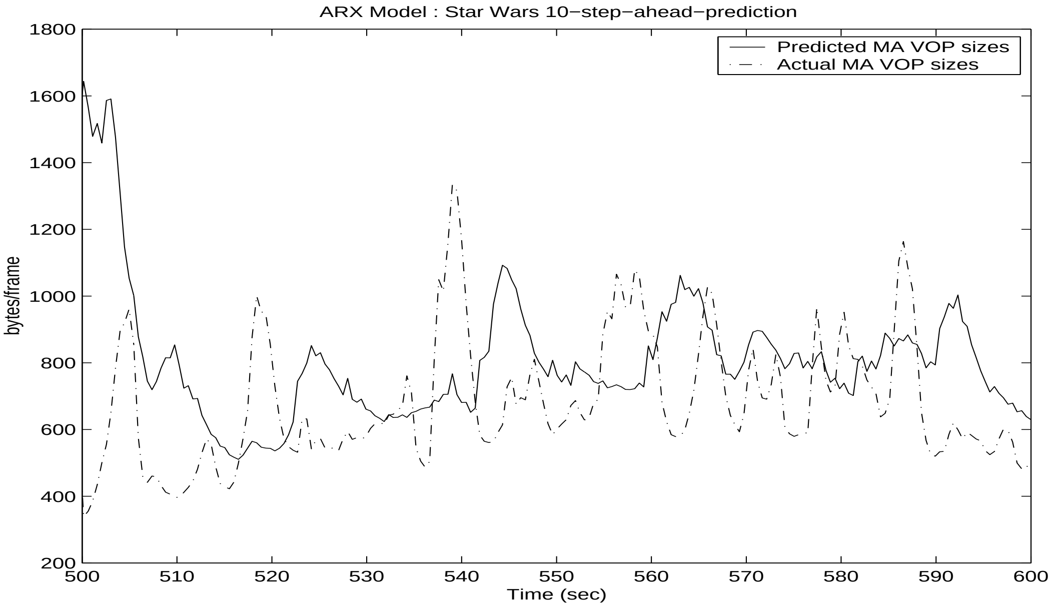

Fig. 129. Ten-SP sizes of the moving average VOPs of *StarWars* using ARX models.

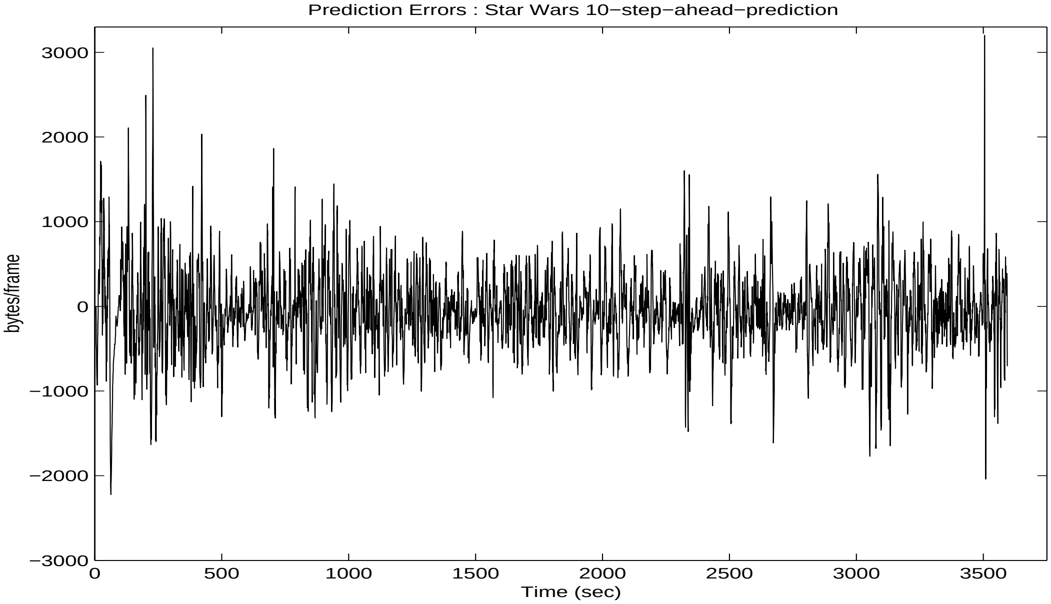

Fig. 130. Ten-SP errors of moving average VOP sizes of *StarWars* using ARX models.

Table XXX. Performance metrics of the ten-SP for ARX models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	18.9	6636.5	17.9
ARD Talk	7.9	5897.9	2.4
Die Hard III	21.2	3456.6	9.1
Jurassic Park I	15.6	9416.8	8.4
Lecture Room	11.5	716.2	2.5
Silence of the Lambs	28.6	3938.3	10.7
Skiing	20.3	1887.6	7.1
StarWars	12.0	3201.1	10.2

c. Ten-step-ahead Prediction Using FMLP Models

The ten-step-ahead FMLP model was designed using the external indicators. The FMLP structure which gave the best results was $12-27-1$. For the prediction of $x_{MA}(k)$, the 12 inputs were $x_{MA}(k-10)$, $x_{MA}(k-11)$, $x_{MA}(k-12)$, $\delta x_{MA}(k-10)$, $\delta x_{MA}(k-11)$, $\delta x_{MA}(k-12)$, $\Delta x_{MA}(k-10)$, $\Delta x_{MA}(k-11)$, $\Delta x_{MA}(k-12)$, $I_1(k-10)$, $I_2(k-10)$, and $I_3(k-10)$. Figures 131 to 134 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

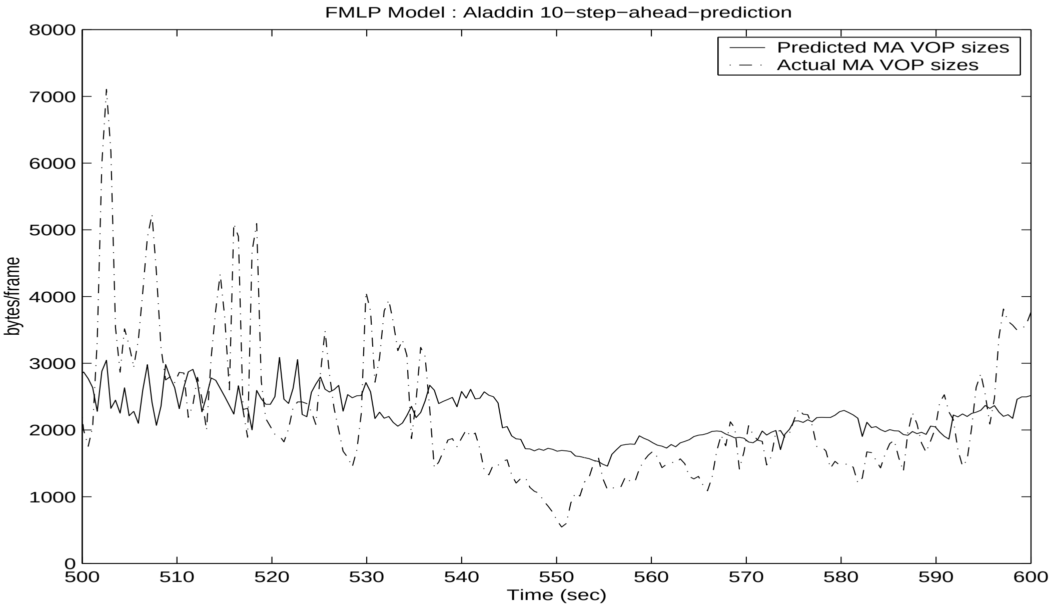

Fig. 131. Ten-SP sizes of the moving average VOPs of *Aladdin* using FMLP models.

Table XXXI shows the performance of the FMLP ten-step-ahead predictor for the video traces used in this research in terms of the performance metrics.

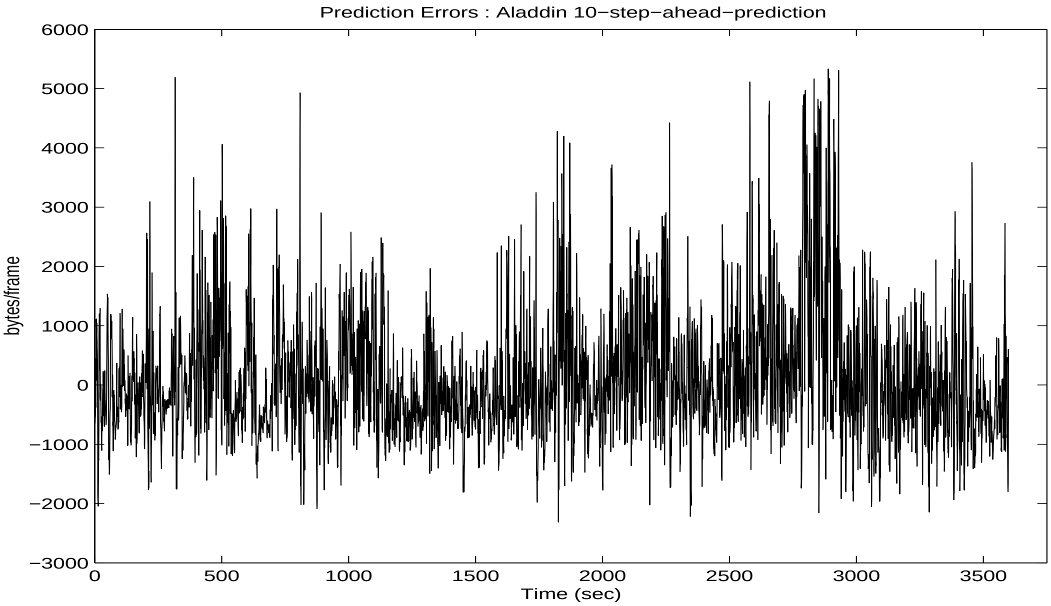

Fig. 132. Ten-SP errors of moving average VOP sizes of *Aladdin* using FMLP models.

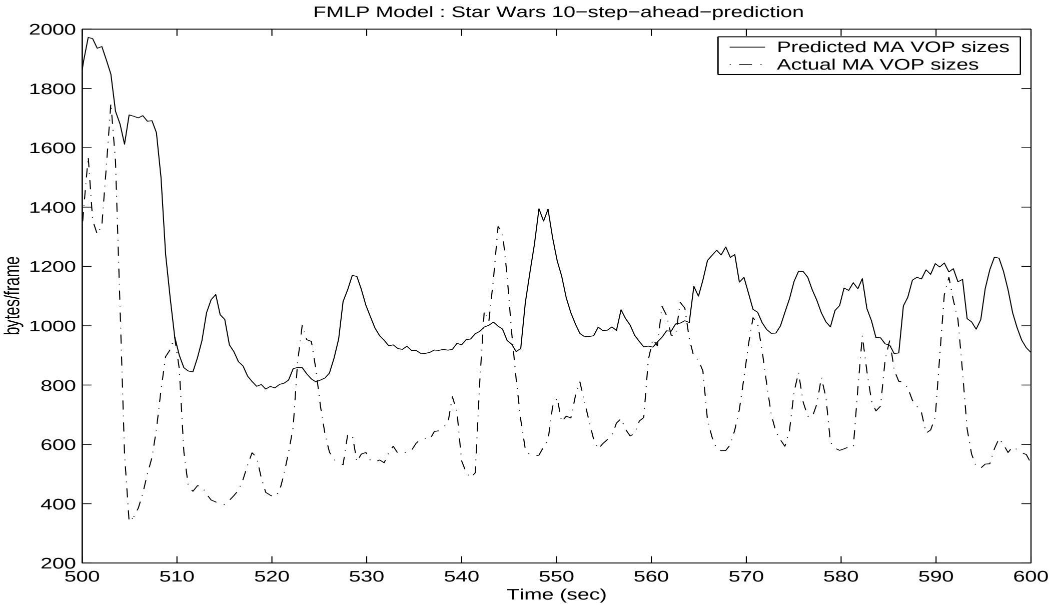

Fig. 133. Ten-SP sizes of the moving average VOPs of *StarWars* using FMLP models.

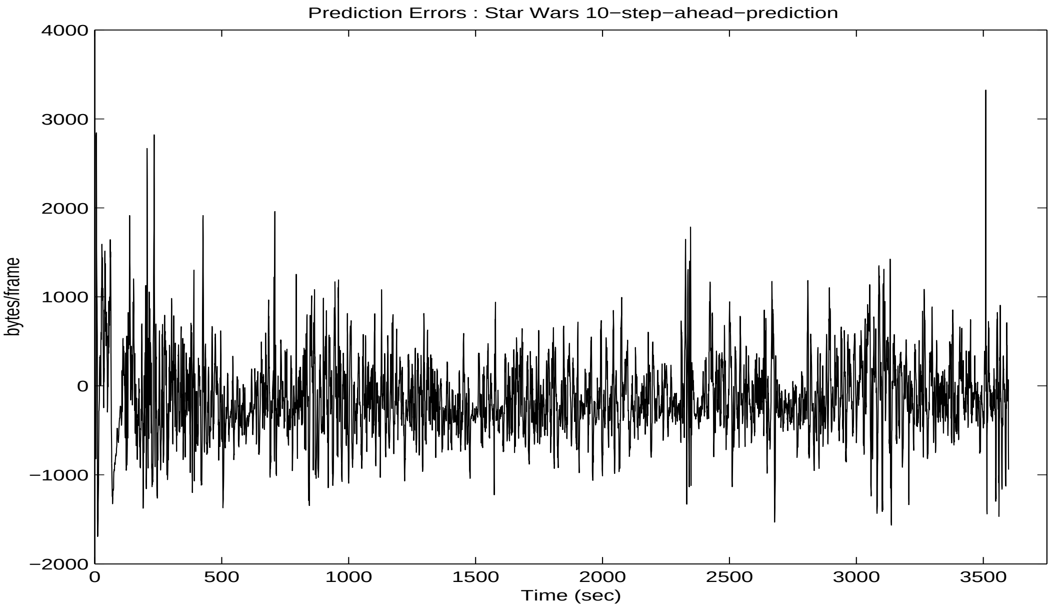

Fig. 134. Ten-SP errors of moving average VOP sizes of *StarWars* using FMLP models.

Table XXXI. Performance metrics of the ten-SP for FMLP models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	17.1	5331.9	28.8
ARD Talk	7.3	6430.9	1.4
Die Hard III	19.7	4127.5	11.9
Jurassic Park I	22.0	12262	6.3
Lecture Room	116.8	1043.3	3.2
Silence of the Lambs	49.5	3359.2	14.1
Skiing	27.2	1818.7	9.7
StarWars	10.2	3324.1	11.6

d. Ten-step-ahead Prediction Using RMLP Models

As in [34], the ten-step-ahead RMLP model was designed using the external indicators. The RMLP structure which gave the best results was $21-10-1$. The training was done using global feedback. For the prediction of $x_{MA}(k)$, 12 of the 21 inputs were $x_{MA}(k-10)$, $x_{MA}(k-11)$, $x_{MA}(k-12)$, $\delta x_{MA}(k-10)$, $\delta x_{MA}(k-11)$, $\delta x_{MA}(k-12)$, $\Delta x_{MA}(k-10)$, $\Delta x_{MA}(k-11)$, $\Delta x_{MA}(k-12)$, $I_1(k-10)$, $I_2(k-10)$ and $I_3(k-10)$. The remaining 9 inputs were the past predictions of the model, that is $\widehat{x}^*_{MA}(k-1)$, $\widehat{x}^*_{MA}(k-2)$, $\widehat{x}^*_{MA}(k-3)$, $\widehat{x}^*_{MA}(k-4)$, $\widehat{x}^*_{MA}(k-5)$, $\widehat{x}^*_{MA}(k-6)$, $\widehat{x}^*_{MA}(k-7)$, $\widehat{x}^*_{MA}(k-8)$ and $\widehat{x}^*_{MA}(k-9)$. Figures 135 to 138 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

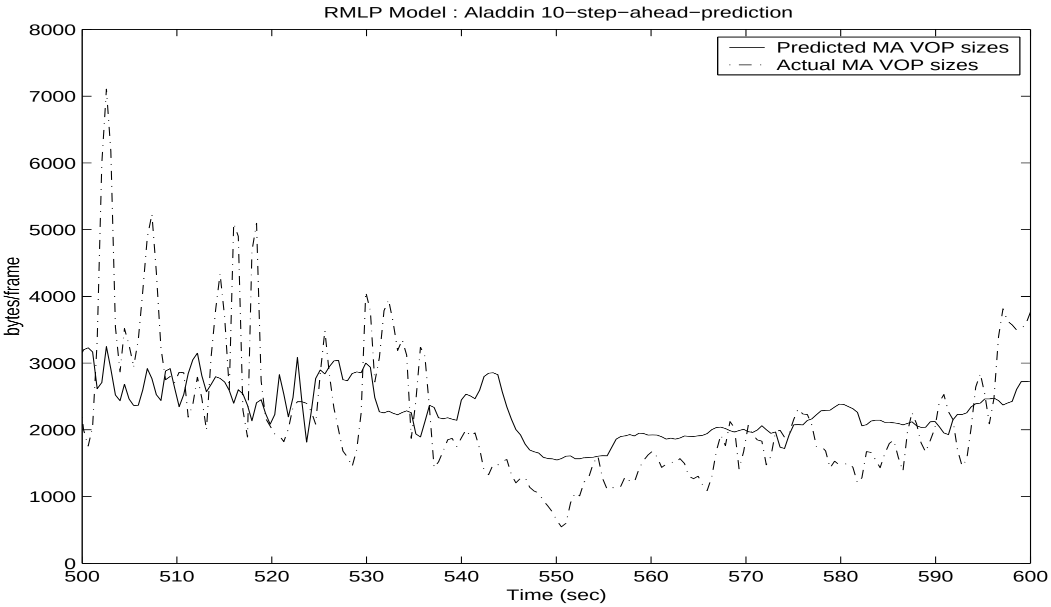

Fig. 135. Ten-SP sizes of the moving average VOPs of *Aladdin* using RMLP models.

Table XXXII shows the performance of the RMLP ten-step-ahead predictor for

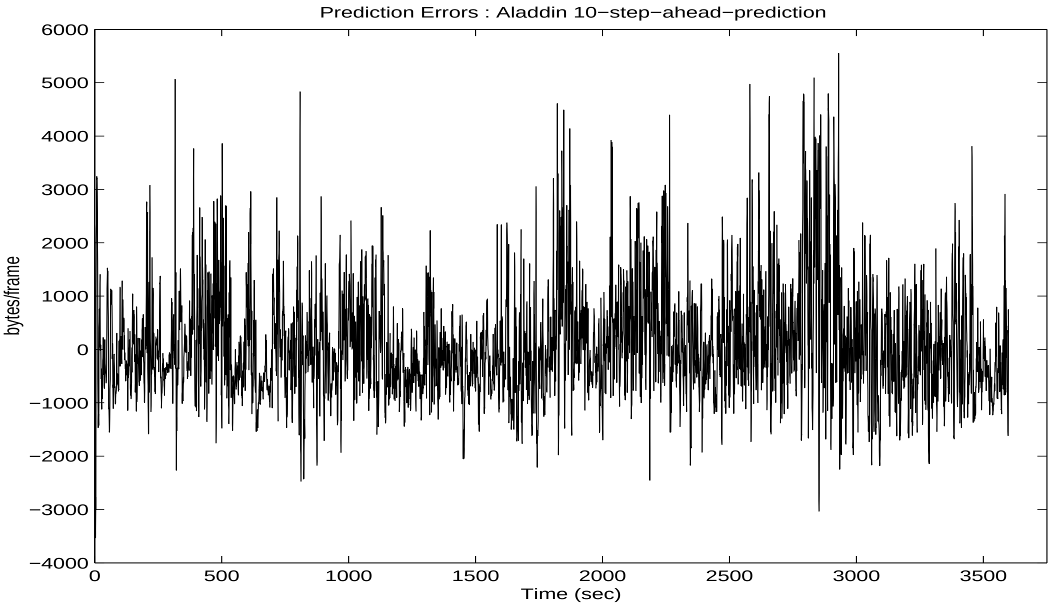

Fig. 136. Ten-SP errors of moving average VOP sizes of *Aladdin* using RMLP models.

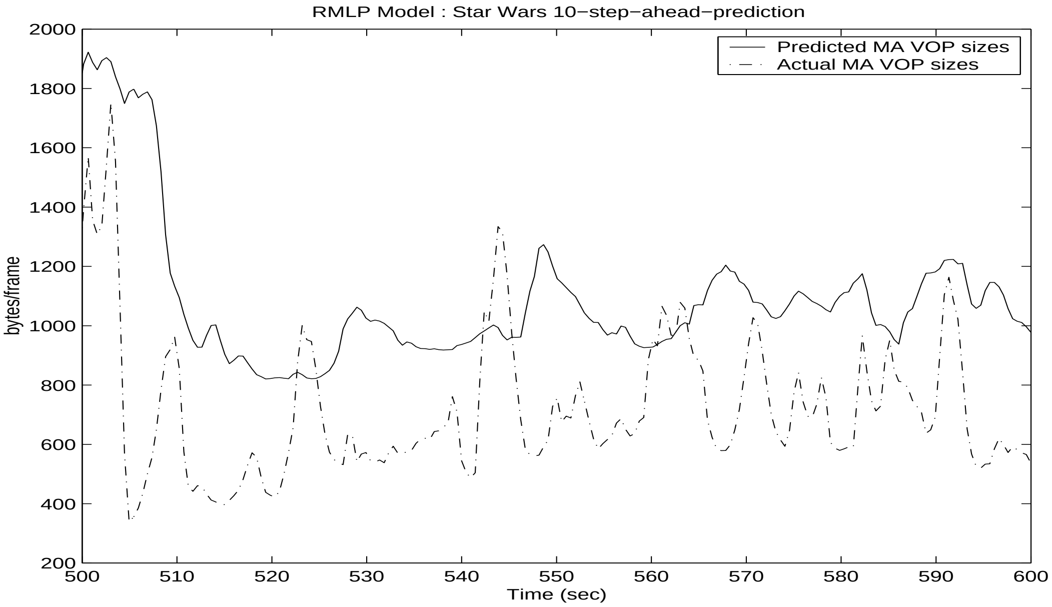

Fig. 137. Ten-SP sizes of the moving average VOPs of *StarWars* using RMLP models.

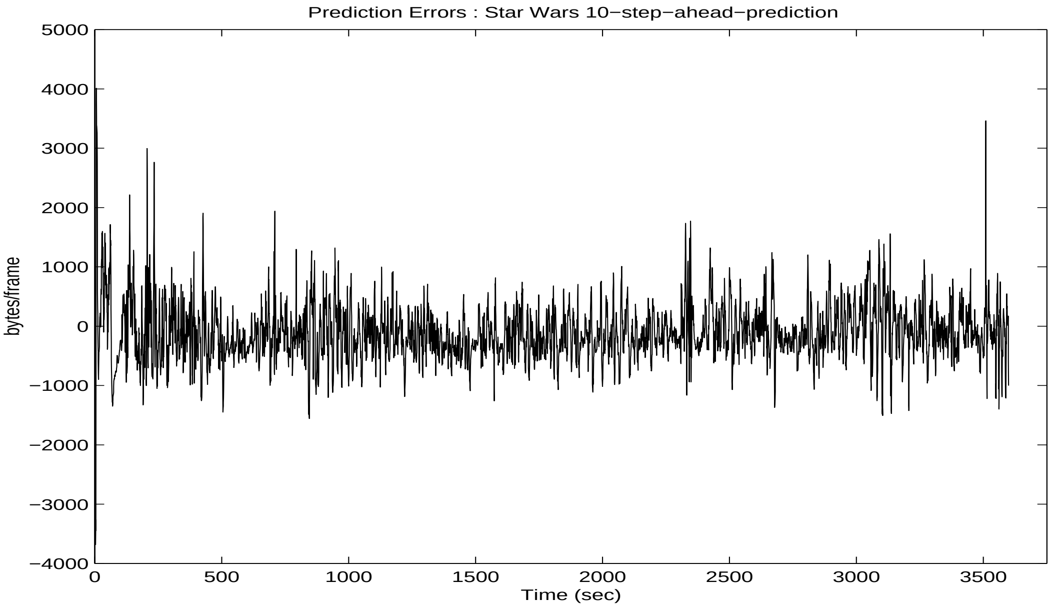

Fig. 138. Ten-SP errors of moving average VOP sizes of *StarWars*using RMLP models.

the video traces used in this research in terms of the performance metrics.

e. Ten-step-ahead Prediction Using ESN Models

The network structure which gave the best results for ten-step-ahead prediction of the moving average of VOP size time-series was $4 - 25 - 1$. In order to predict the moving average of the VOP sizes at time step k the four inputs were $x_{MA}(k-10)$, $\delta x_{MA}(k-10)$, $\Delta x_{MA}(k-10)$ and $I_1(k-10)$. Figures 119 to 142 show the performance of the designed predictor for a time window of 100 seconds and errors for the entire lengths of the video traces *Aladdin* and *StarWars* respectively.

Table XXXIII shows the performance of the ESN predictor for all the video traces in terms of the three performance metrics.

Table XXXII. Performance metrics of the ten-SP for RMLP models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	16.9	5550.6	20.7
ARD Talk	7.3	6002.5	1.9
Die Hard III	22.7	4217.8	14.0
Jurassic Park I	14.6	9017.0	14.0
Lecture Room	165.3	3624.7	8.6
Silence of the Lambs	59.2	3115.5	16.5
Skiing	37.8	3786.3	14.5
StarWars	11.7	4012.9	37.1

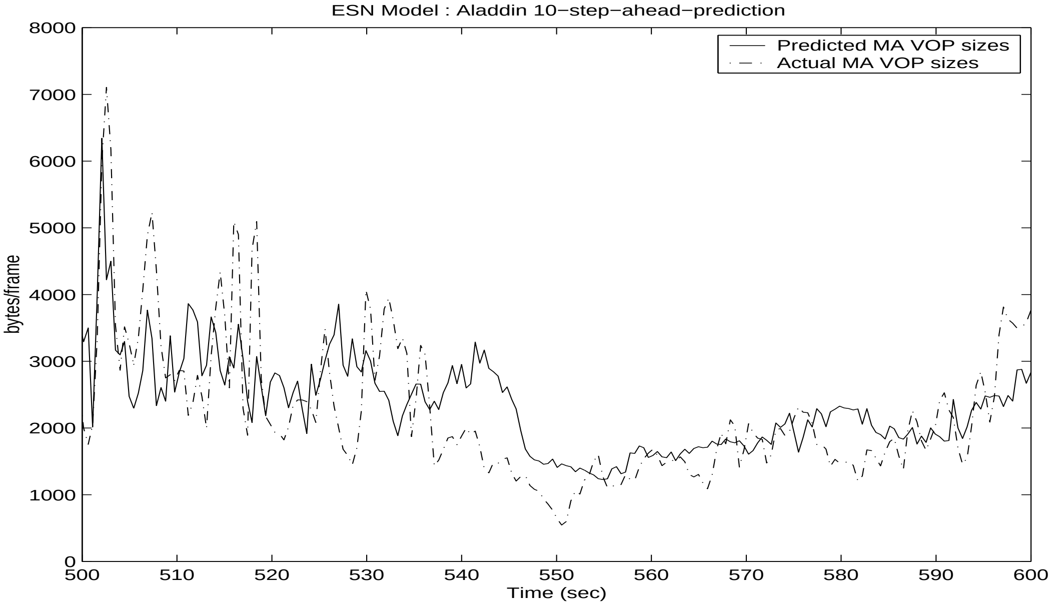

Fig. 139. Ten-SP sizes of the moving average VOPs of *Aladdin* using ESN models.

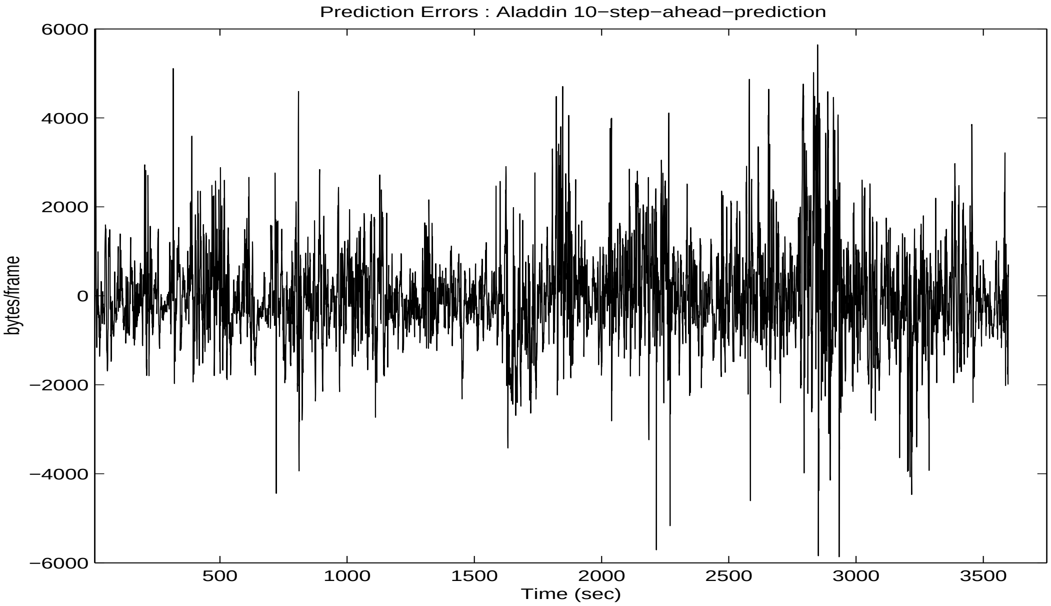

Fig. 140. Ten-SP errors of moving average VOP sizes of *Aladdin* using ESN models.

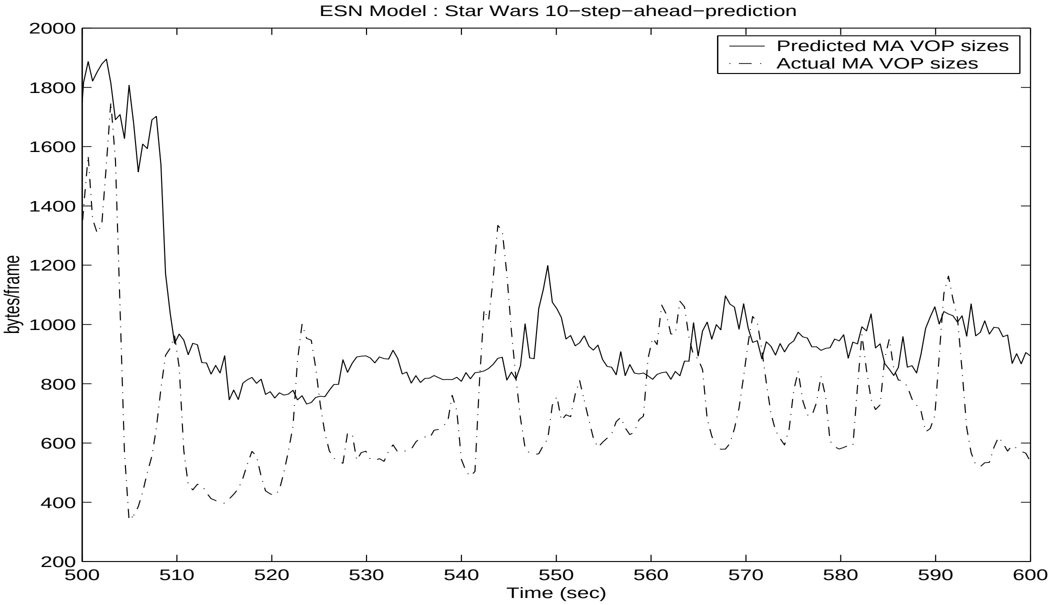

Fig. 141. Ten-SP sizes of the moving average VOPs of *StarWars* using ESN models.

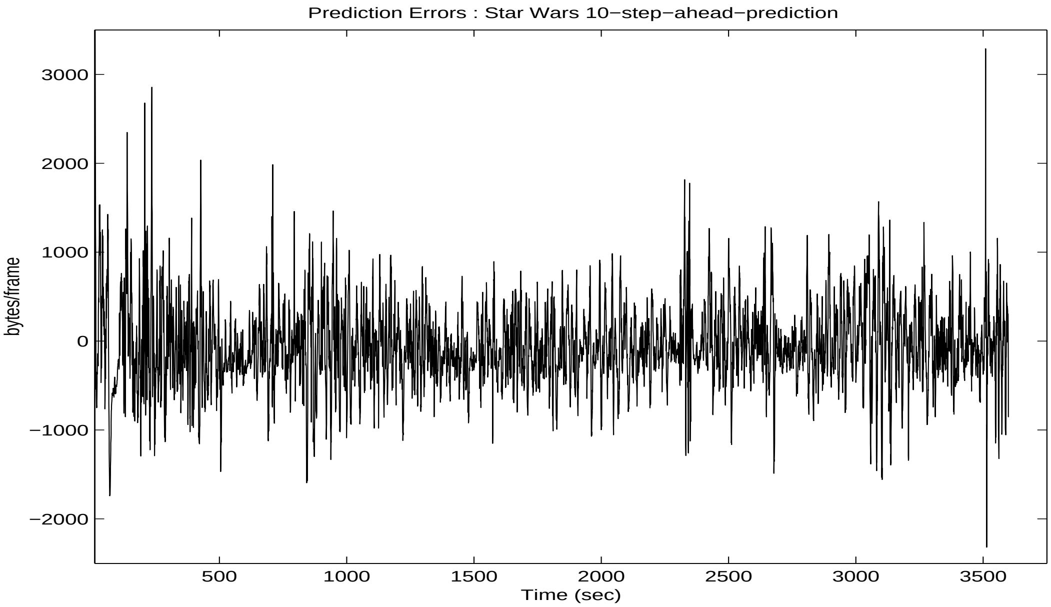

Fig. 142. Ten-SP errors of moving average VOP sizes of *StarWars* using ESN models.

Table XXXIII. Performance metrics of the ten-SP for ESN models

Video Trace	MSE (%)	MAE (in bytes)	MRE
Aladdin	16.9	5844.6	11.4
ARD Talk	12.9	5914.2	2.4
Die Hard III	20.1	3923.5	8.3
Jurassic Park I	29.2	19997	10.8
Lecture Room	4.7	915.81	1.7
Silence of the Lambs	44.9	5824.1	12.7
Skiing	29.5	4001.0	13.8
StarWars	10.2	3335.2	10.7

G. Chapter Summary

In this chapter, the single and multi-step-ahead predictors for both the I-VOPs and moving average of the VOP sizes are presented. Their performance on all the traces is tabulated in the terms of the standard performance metrics. In the next chapter, these results are compared against the published literature. A comparison is also made among these prediction schemes developed in this research. Based on this analysis some important observations and conclusions are also presented in the next chapter.

CHAPTER VI

SUMMARY AND CONCLUSIONS

This chapter summarizes the work presented in this thesis. First, a comparison of the different models developed in this research is presented. This is followed by the conclusions and recommendations for future work.

A. Comparison of the Models

In this section, the different empirical models are compared in terms of the performance metric MSE. Some of the results used here have been derived from [30]. The conclusions made later in this chapter are based on this comparison. From the published literature it can be accessed that the prediction of individual VOPs has not been very successful. In this work, we develop one linear (ARX) and one non-linear (ESN) predictor for I-VOPs. This is done to substantiate the claim that the prediction of the moving average is more accurate than the prediction of individual VOPs.

1. Single-step-ahead Prediction Models

a. Prediction Models for I-VOPs

Table XXXIV presents the MSE in the single-step-ahead prediction of individual I-VOPs for the models developed in this work along with the the results published in the literature. The results for the RMLP model are obtained from [30]. The table also shows the MBR of the video traces.

Figures 143 and 144 show the performance of the prediction schemes developed in this research for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively.

Table XXXIV. MSE of the single-step-ahead prediction of I-VOPs

Video Trace	MBR (Mbps)	ARX Model	RMLP Model [30]	ESN Model
Aladdin	0.44	3.1	2.6	3.0
ARD Talk	0.54	0.8	0.9	1.6
Die Hard III	0.25	3.4	2.9	3.0
Jurassic Park I	0.77	1.1	0.8	3.8
Lecture Room	0.06	0.2	0.2	0.1
Silence of the Lambs	0.11	3.3	3.6	3.8
Skiing	0.19	2.2	2.0	1.9
StarWars	0.28	1.7	1.5	1.5

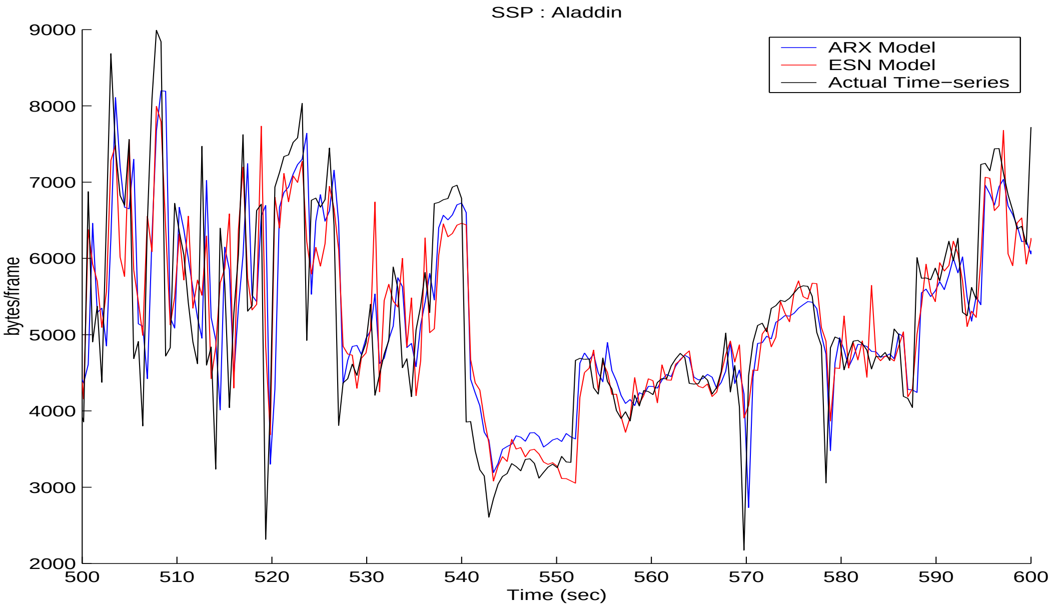

Fig. 143. SSP sizes of the I-VOPs of *Aladdin*.

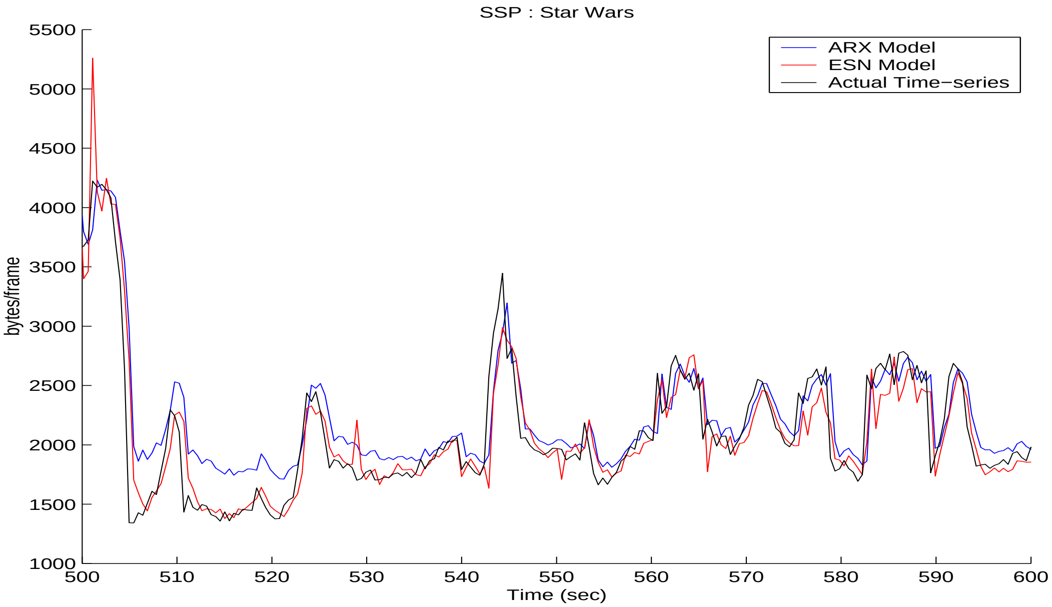

Fig. 144. SSP sizes of the I-VOPs of *StarWars*.

From the table it can be observed that:

1. The performance of non-linear models is better than the linear models.

2. For high bit rates (MBR $\geq 0.4 \sim 0.5$ Mbps), the RMLP predictor is better and for low bit rates the ESN model is better than other models.

b. Prediction Models for Moving Average

Table XXXV presents the MSE in the single-step-ahead prediction of the moving average VOP size time-series for different methods used in this research and the published literature. The results for the FMLP model are obtained from the research paper by Bhattacharya et al. [30]. The table also shows the MBR of the video traces.

Figures 145 and 146 show the performance of these prediction schemes for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively.

Table XXXV. MSE of single-step-ahead prediction for all models

Video Trace	MBR (Mbps)	AR Model	ARX Model	FMLP Model [30]	ESN Model
Aladdin	0.44	3.2	3.5	2.5	2.3
ARD Talk	0.54	0.8	1.1	0.8	1.4
Die Hard III	0.25	2.3	2.4	2.4	1.4
Jurassic Park I	0.77	1.2	1.8	1.0	2.1
Lecture Room	0.06	0.8	0.9	14.8	1.3
Silence of the Lambs	0.11	3.3	3.3	3.4	2.7
Skiing	0.19	2.1	2.2	2.7	1.5
StarWars	0.28	1.5	2.0	1.7	0.8

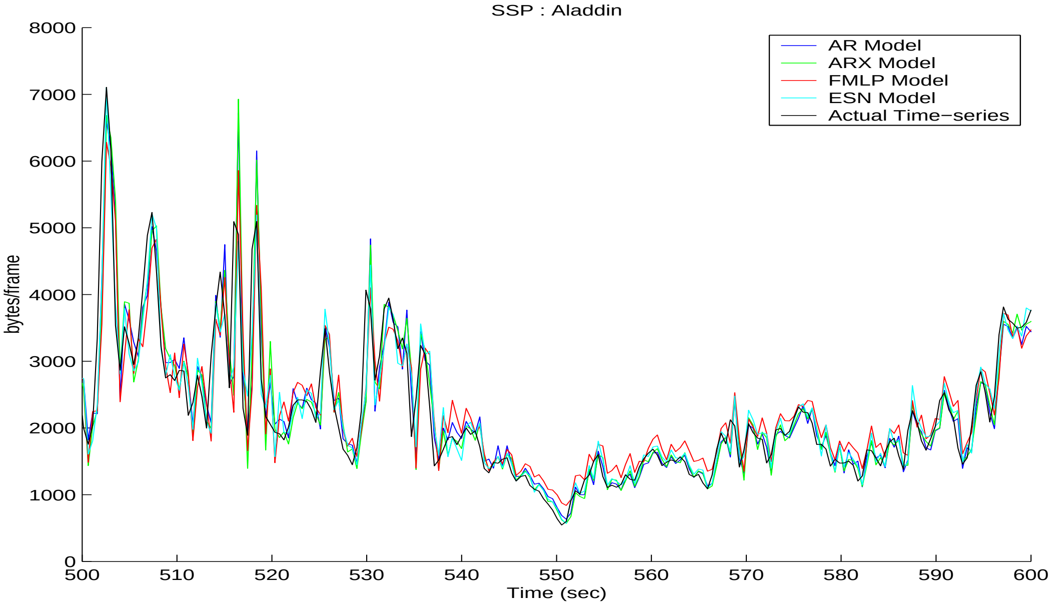

Fig. 145. SSP sizes of the moving average VOPs of *Aladdin* for all models.

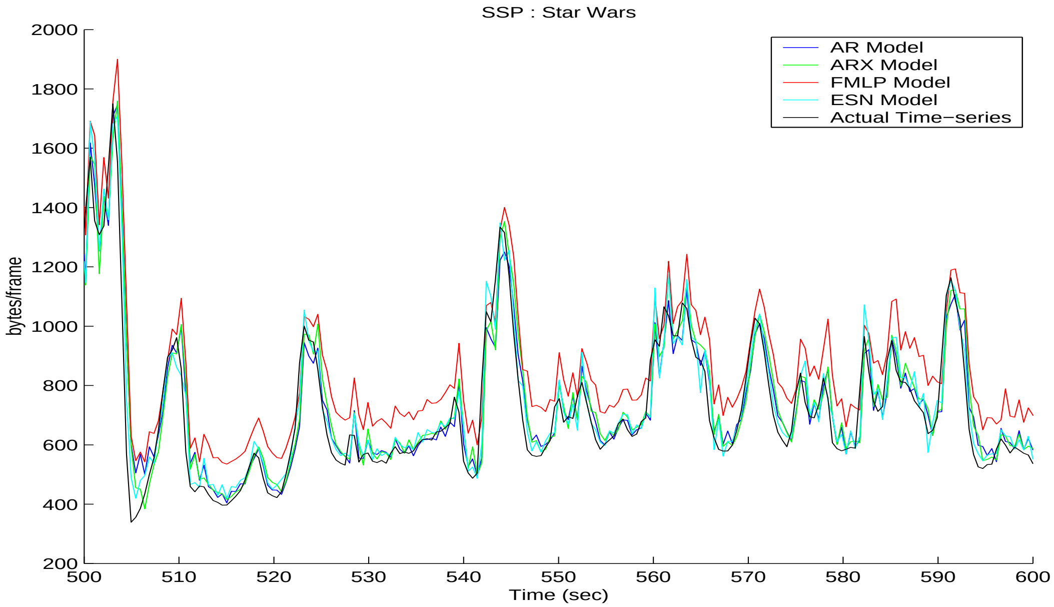

Fig. 146. SSP sizes of the moving average VOPs of *StarWars* for all models.

From Table XXXV the following observations can be made:

1. It can be seen that the non-linear models perform much better than the linear models for almost all the video traces (except for the video trace *Lecture Room*). It is safe to assume in general that the performance of the non-linear models is better than the linear ones for single-step-ahead prediction and the extra cost (in terms of the modeling time and complexity) of the non-linear models is justified.

2. Of the various non-linear models developed in this research and those published in the literature [30], there is no one model which performs well for all the video traces. If we relate the performance of the models with the MBR of the video traces, it can be seen from Table XXXV that the FMLP model performs better for high bit rate video traces (MBR $\geq 0.4 \sim 0.5$ Mbps) such as *ARD Talk*

and *Jurassic Park* than the ESN model. On the other hand, the ESN model supersedes the FMLP model in performance for low bit rate videos.

Thus it can be generalized that for single-step-ahead prediction, the FMLP model performs better for high bits rates whereas for low bit rates the performance of ESN model is better.

2. Two-step-ahead Prediction Models

a. Prediction Models for I-VOPs

Table XXXVI presents the MSE in the two-step-ahead prediction of individual I-VOPs for the models developed in this work. The table also shows the MBR of the video traces.

Figures 147 and 148 show the performance of the prediction schemes developed in this research for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively.

It can be observed from Table XXXVI that the performance of the ESN model is better than the ARX model for low bit rate video traces.

b. Prediction Models for Moving Average

Table XXXVII presents the MSE in the two-step-ahead prediction of the moving average VOP size time-series for different methods used in this research and the published literature. The results for the RMLP model are obtained from the research paper by Bhattacharya et al. [30]. The MBR of the video traces is also tabulated in Table XXXVII.

Figures 149 and 150 show the performance of these prediction schemes for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively.

Table XXXVI. MSE of the two-step-ahead prediction of I-VOPs

Video Trace	MBR (Mbps)	ARX Model	ESN Model
Aladdin	0.44	4.8	5.4
ARD Talk	0.54	1.7	2.8
Die Hard III	0.25	5.6	5.4
Jurassic Park I	0.77	2.0	6.3
Lecture Room	0.06	0.7	0.2
Silence of the Lambs	0.11	6.9	7.0
Skiing	0.19	4.3	3.7
StarWars	0.28	3.0	2.7

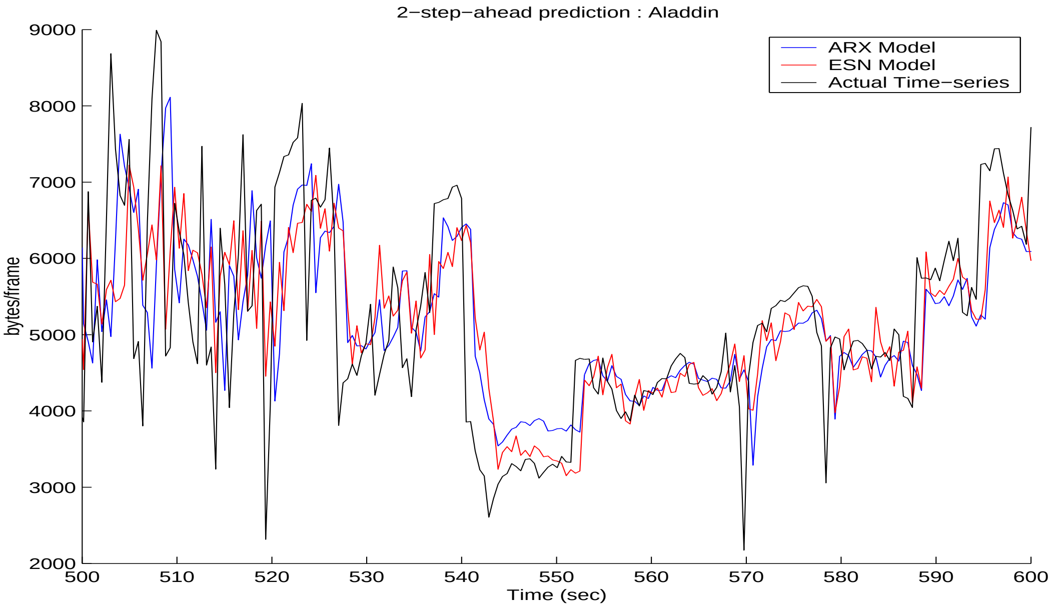

Fig. 147. Two-SP sizes of the I-VOPs of *Aladdin*.

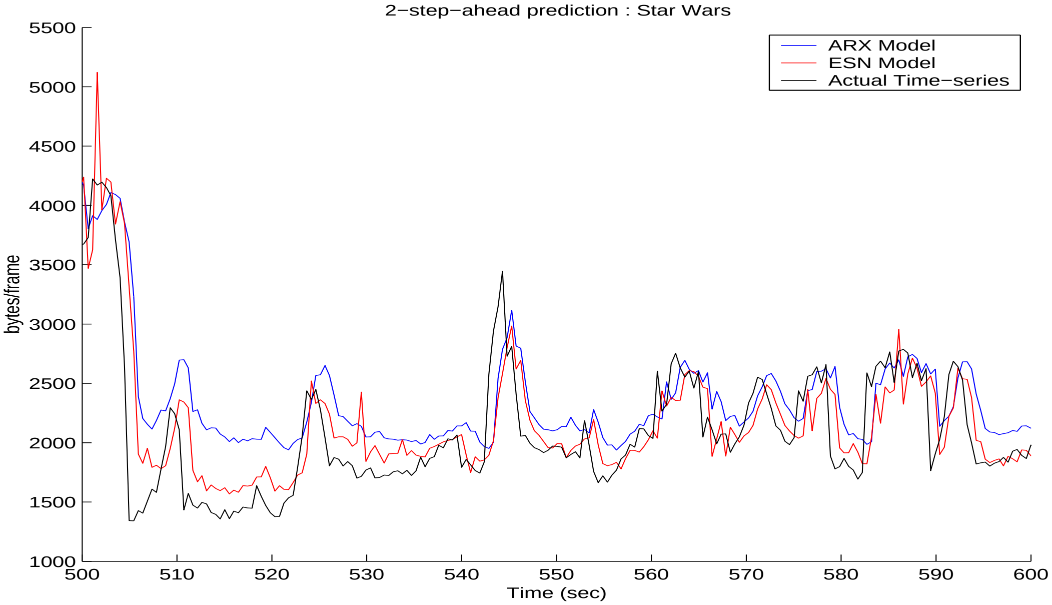

Fig. 148. Two-SP sizes of the I-VOPs of *StarWars*.

Table XXXVII. MSE of the two-step-ahead prediction for all models

Video Trace	MBR (Mbps)	AR Model	ARX Model	FMLP Model	RMLP Model [30]	ESN Model
Aladdin	0.44	9.2	9.6	7.9	8.2	7.8
ARD Talk	0.54	3.1	3.0	2.9	2.4	2.7
Die Hard III	0.25	7.4	7.9	7.2	7.8	5.4
Jurassic Park I	0.77	4.7	4.5	4.8	3.8	6.6
Lecture Room	0.06	13.8	6.8	49.3	22.6	1.5
Silence of the Lambs	0.11	8.8	8.5	14.2	9.3	8.0
Skiing	0.19	6.3	6.5	10.2	7.9	5.6
StarWars	0.28	4.5	4.7	4.1	4.9	3.1

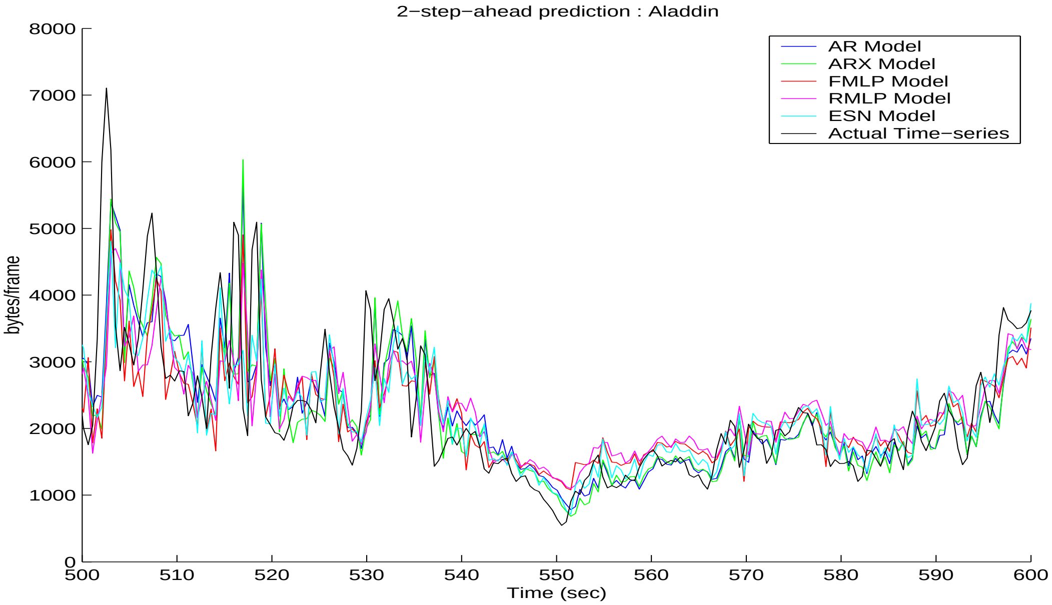

Fig. 149. Two-SP sizes of the moving average VOPs of *Aladdin* for all models.

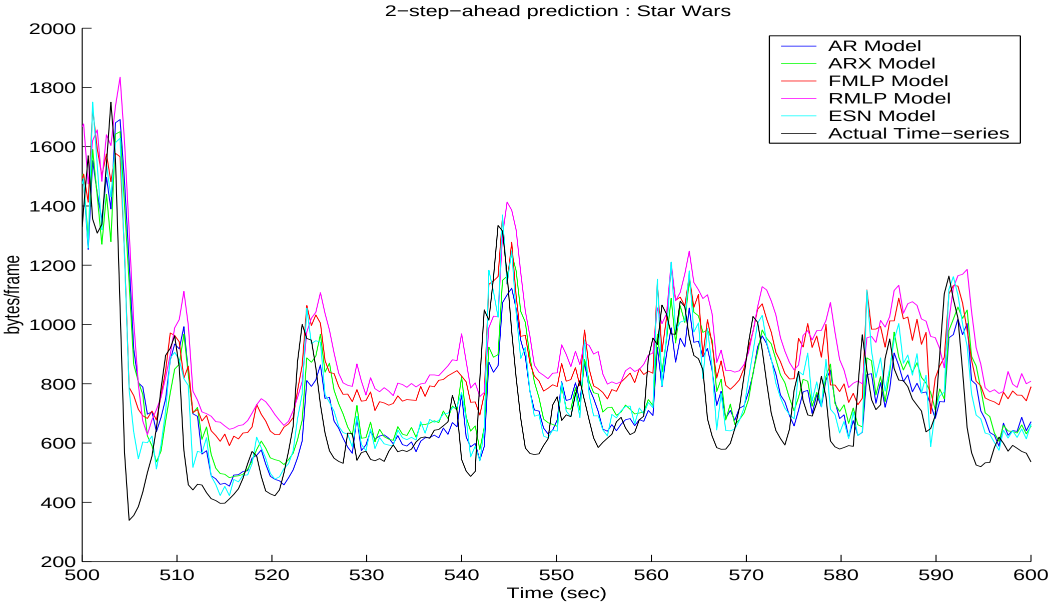

Fig. 150. Two-SP sizes of the moving average VOPs of *StarWars* for all models.

Following observations can be made from Table XXXVII:

1. The non-linear empirical modeling techniques perform better than the linear techniques for two-step-ahead prediction in terms of the performance metric MSE.

2. The recurrent neural network models perform better than the feed-forward neural networks for two-step-ahead prediction. For high bit rate video traces (MBR $\geq 0.4 \sim 0.5$ Mbps), the RMLP model performs better whereas for the low bit rate traces the performance of the ESN model is better.

Thus it can be generalized from the above discussion that the recurrent neural networks give best results for two-step-ahead predictions with the RMLP predictor for high bit rate video traces and ESN predictor for low bit rate video traces.

3. Four-step-ahead Prediction Models

a. Prediction Models for I-VOPs

Table XXXVIII presents the MSE in the four-step-ahead prediction of individual I-VOPs for the models developed in this work. The table also shows the MBR of the video traces.

Figures 151 and 152 show the performance of the prediction schemes developed in this research for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively.

It can be observed from Table XXXVIII that the ESN model still performs better than the ARX model for low bit rate video traces.

Table XXXVIII. MSE of the four-step-ahead prediction of I-VOPs

Video Trace	MBR (Mbps)	ARX Model	ESN Model
Aladdin	0.44	6.6	6.8
ARD Talk	0.54	3.0	3.9
Die Hard III	0.25	8.5	8.3
Jurassic Park I	0.77	3.4	6.6
Lecture Room	0.06	1.6	0.4
Silence of the Lambs	0.11	10.8	11.0
Skiing	0.19	7.0	6.7
StarWars	0.28	4.5	4.4

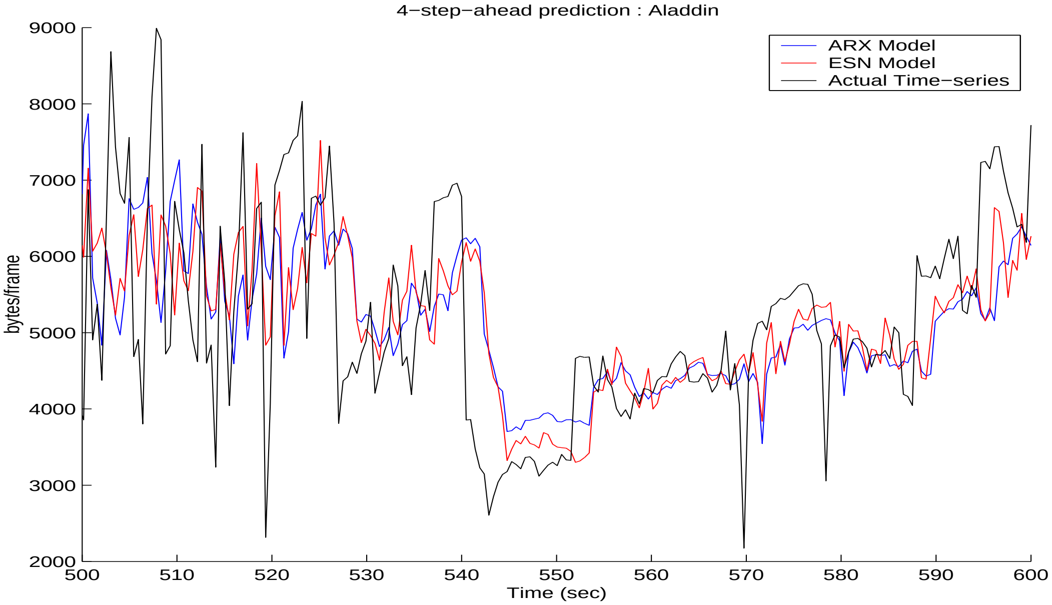

Fig. 151. Four-SP sizes of the I-VOPs of *Aladdin*.

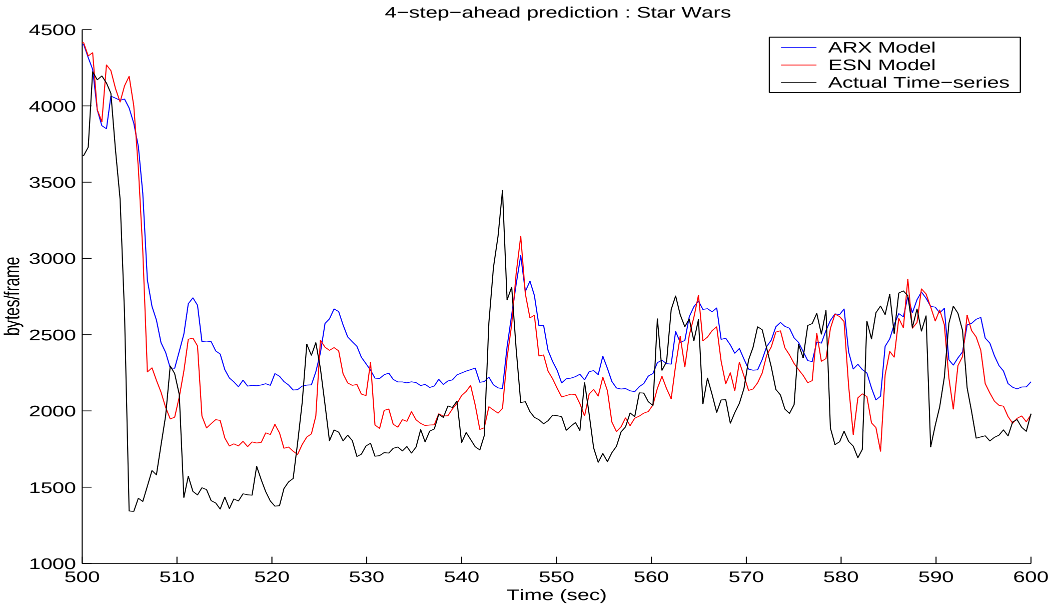

Fig. 152. Four-SP sizes of the I-VOPs of *StarWars*.

b. Prediction Models for Moving Average

Table XXXIX presents the MSE in the four-step-ahead prediction of the moving average VOP size time-series for different methods used in this research and the published literature. The results for the RMLP model are obtained from the research paper by Bhattacharya et al. [30]. The MBR of the video traces is also tabulated in Table XXXIX.

Figures 153 and 154 show the performance of these prediction schemes for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively.

From Table XXXIX the following can be observed:

1. The non-linear empirical modeling techniques perform better than the linear techniques for the four-step-ahead predictions.

Table XXXIX. MSE of the four-step-ahead prediction for all models

Video Trace	MBR (Mbps)	AR Model	ARX Model	FMLP Model	RMLP Model [30]	ESN Model
Aladdin	0.44	15.2	14.9	13.2	13.0	13.7
ARD Talk	0.54	4.9	4.4	5.0	4.2	4.6
Die Hard III	0.25	15.3	14.0	11.6	14.9	12.3
Jurassic Park I	0.77	9.9	9.8	23.1	8.0	15.2
Lecture Room	0.06	40.5	7.4	81.9	33.3	2.8
Silence of the Lambs	0.11	20.9	17.0	25.9	19.0	16.6
Skiing	0.19	13.0	13.6	13.8	15.1	11.9
StarWars	0.28	8.5	8.8	6.3	8.6	8.4

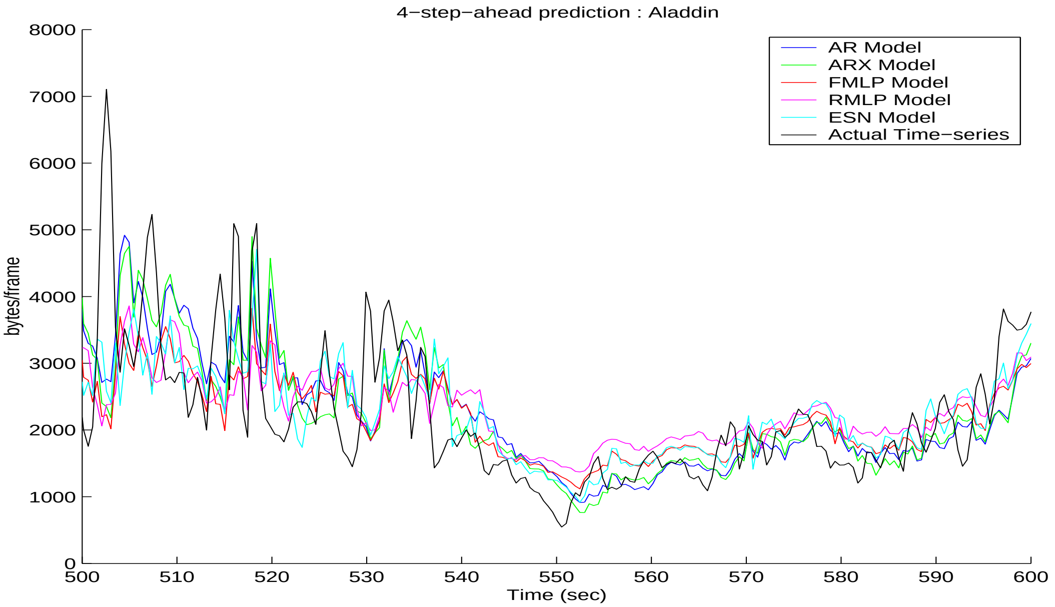

Fig. 153. Four-SP sizes of the moving average VOPs of *Aladdin* for all models.

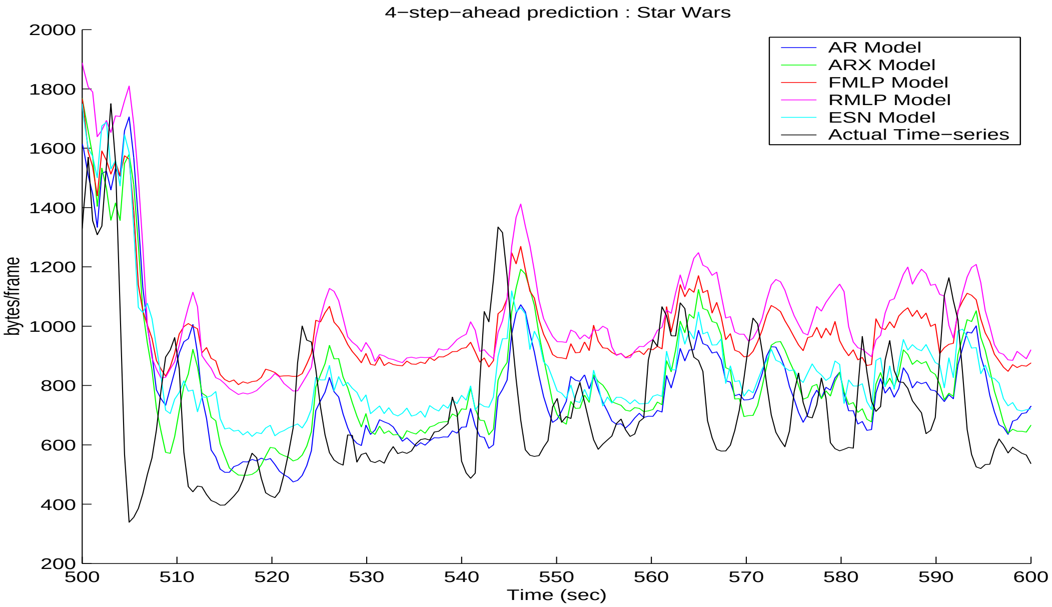

Fig. 154. Four-SP sizes of the moving average VOPs of *StarWars* for all models.

2. For high bit rate video traces the RMLP model performs better than the other models whereas for the low bit rate traces we get mixed results. The ESN model performs better than other non-linear models for very low bit rate video traces (MBR $\geq$ 0.2 Mbps) such as *Lecture Room*, *Silence of the Lambs*, and *Skiing*. For the range where $0.2 \leq \text{MBR} \leq 0.4 \sim 0.5$ Mbps, the FMLP model gives better results.

Thus it can be concluded from the above observations that the ESN model continues to perform better than others for very low bit rate videos. But for medium bit rate ranges, its performance is inferior to the FMLP model. The RMLP model continues to perform better than all other models for high bit rate video traces.

4. Six-step-ahead Prediction Models

a. Prediction Models for I-VOPs

Table XL presents the MSE in the six-step-ahead prediction of individual I-VOPs for the models developed in this work. The table also shows the MBR of the video traces.

Figures 155 and 156 show the performance of the prediction schemes developed in this research for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively.

Table XL. MSE of the six-step-ahead prediction of I-VOPs

Video Trace	MBR (Mbps)	ARX Model	ESN Model
Aladdin	0.44	7.6	8.2
ARD Talk	0.54	4.0	5.4
Die Hard III	0.25	10.8	10.2
Jurassic Park I	0.77	4.3	7.8
Lecture Room	0.06	2.7	2.4
Silence of the Lambs	0.11	13.2	14.9
Skiing	0.19	8.7	8.4
StarWars	0.28	5.6	5.4

It can be observed from Table XL that though the ESN model is still better than the ARX model at low bit rates, the errors in both models have increased considerably.

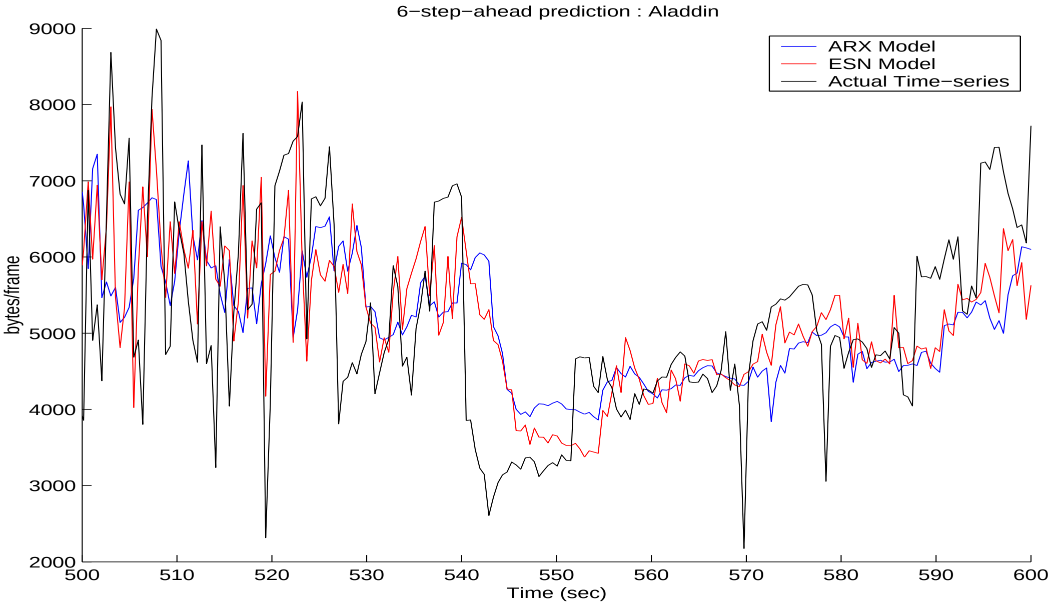

Fig. 155. Six-SP sizes of the I-VOPs of *Aladdin*.

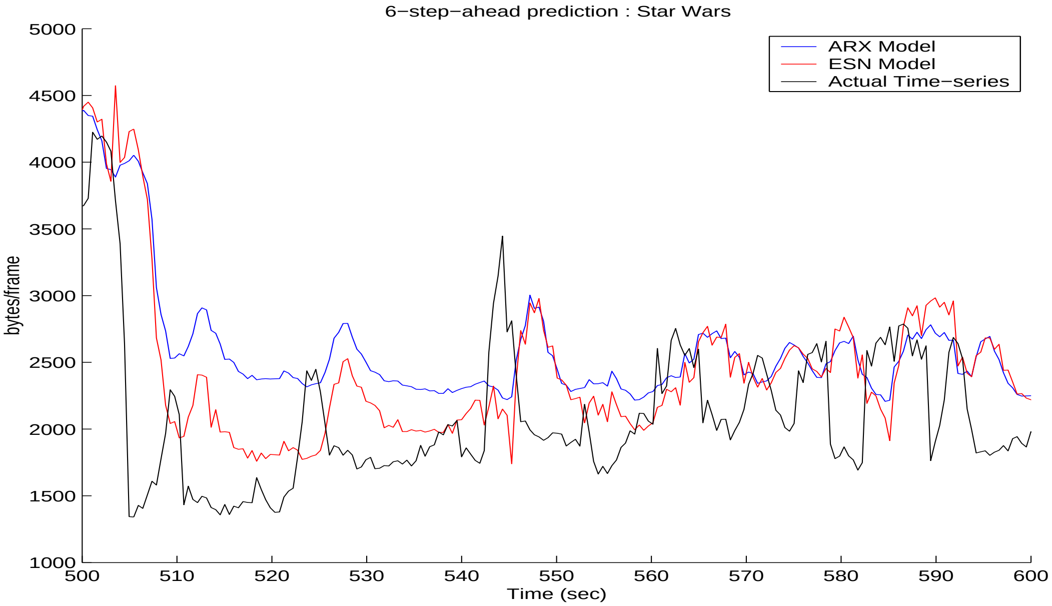

Fig. 156. Six-SP sizes of the I-VOPs of *StarWars*.

b. Prediction Models for Moving Average

Table XLI presents the MSE in the Six-step-ahead prediction of the moving average VOP size time-series for different methods used in this research. The MBR of the video traces is also tabulated in Table XXXIX.

Figures 157 and 158 show the performance of these prediction schemes for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively.

Table XLI. MSE of the six-step-ahead prediction for all models

Video Trace	MBR (Mbps)	AR Model	ARX Model	FMLP Model	RMLP Model	ESN Model
Aladdin	0.44	16.5	18.1	15.3	15.6	14.6
ARD Talk	0.54	6.1	6.4	6.1	5.9	9.1
Die Hard III	0.25	19.0	17.6	15.3	18.5	15.3
Jurassic Park I	0.77	12.4	12.3	21.6	11.3	11.5
Lecture Room	0.06	66.7	10.7	73.3	140.0	3.8
Silence of the Lambs	0.11	28.4	21.5	33.3	45.3	32.6
Skiing	0.19	16.1	17.5	20.0	29.2	22.6
StarWars	0.28	9.9	10.4	8.2	9.8	8.1

It can be observed from Table XLI that:

1. Though the RMLP predictor is better than other predictors for high bit rate video traces, the prediction errors are large.
2. There is no single model which performs well for all low bit rate traces. The prediction is equally bad for both linear and non-linear models.

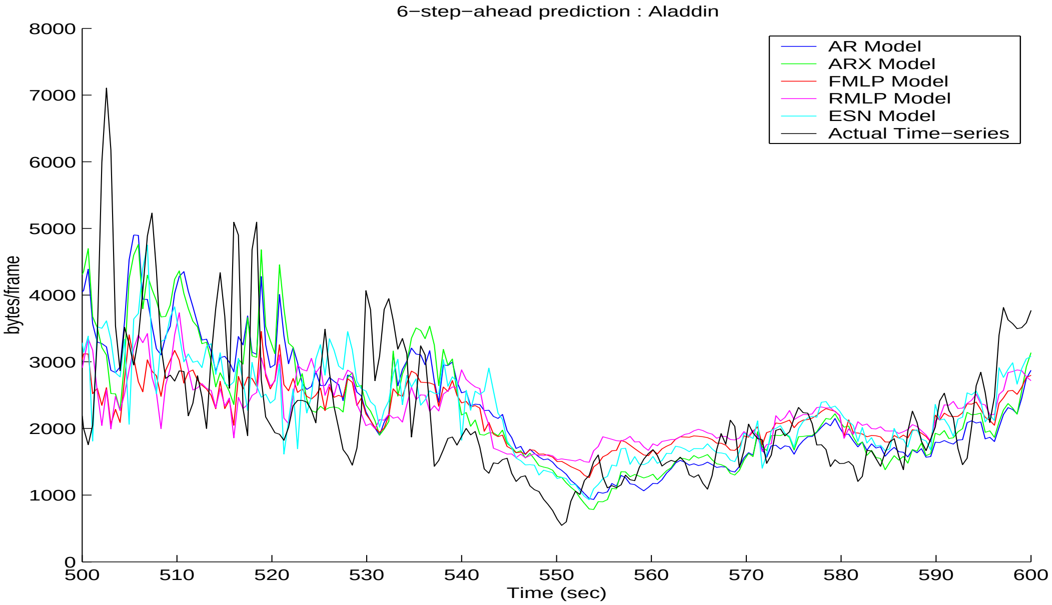

Fig. 157. Six-SP sizes of the moving average VOPs of *Aladdin* for all models.

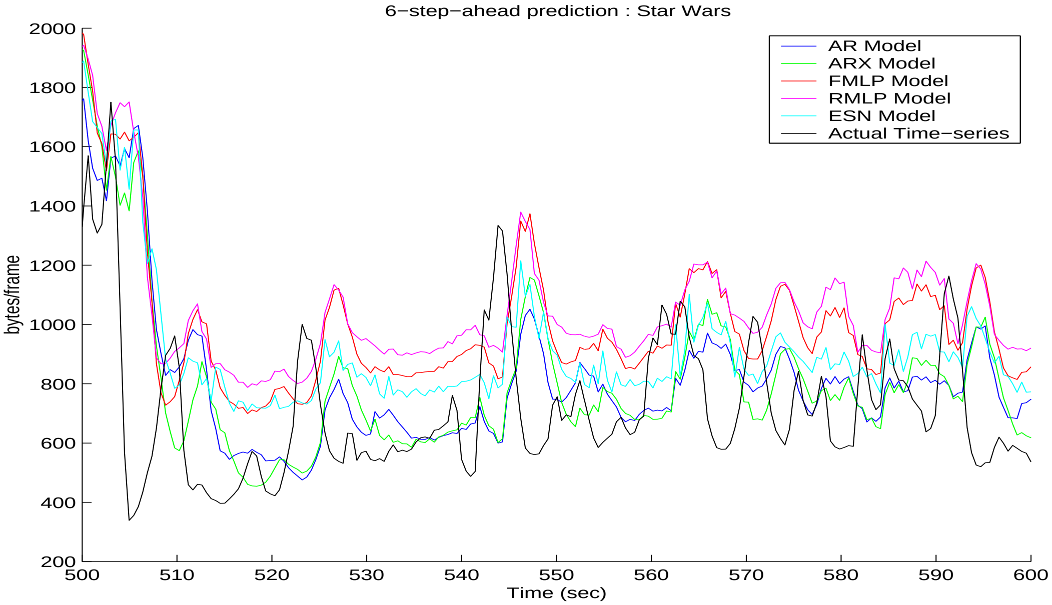

Fig. 158. Six-SP sizes of the moving average VOPs of *StarWars* for all models.

Thus it can be concluded that the all prediction models begin to falter as the prediction horizon is increased to six-steps-ahead. Next the results for ten-step-ahead predictions are presented.

5. Ten-step-ahead Prediction Models

a. Prediction Models for I-VOPs

Table XLII presents the MSE in the ten-step-ahead prediction of individual I-VOPs for the models developed in this work. The table also shows the MBR of the video traces.

Figures 159 and 160 show the performance of the prediction schemes developed in this research for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively.

Table XLII. MSE of the ten-step-ahead prediction of I-VOPs

Video Trace	MBR (Mbps)	ARX Model	ESN Model
Aladdin	0.44	8.1	8.4
ARD Talk	0.54	5.0	6.4
Die Hard III	0.25	11.5	11.3
Jurassic Park I	0.77	5.3	8.8
Lecture Room	0.06	3.3	2.9
Silence of the Lambs	0.11	15.7	17.0
Skiing	0.19	10.2	10.1
StarWars	0.28	6.2	6.9

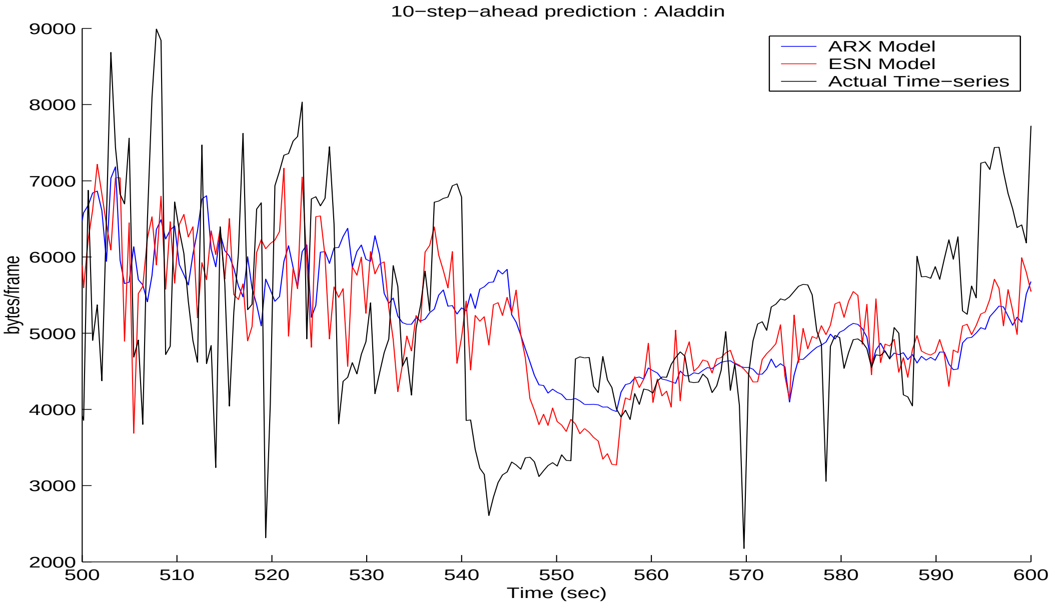

Fig. 159. Ten-SP sizes of the I-VOPs of *Aladdin*.

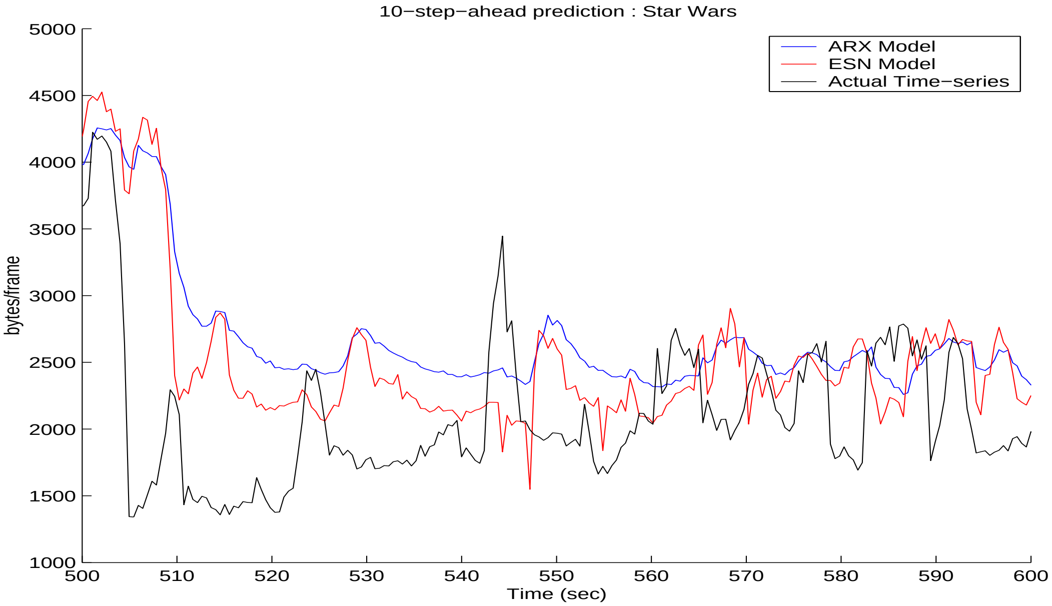

Fig. 160. Ten-SP sizes of the I-VOPs of *StarWars*.

It can be observed from Table XLII that the errors in both the models are significantly high and the performance is comparable.

b. Prediction Models for Moving Average

Table XLIII presents the MSE in the Ten-step-ahead prediction of the moving average VOP size time-series for different methods used in this research. The MBR of the video traces is also tabulated in Table XLIII.

Table XLIII. MSE of the ten-step-ahead prediction for all models

Video Trace	MBR (Mbps)	AR Model	ARX Model	FMLP Model	RMLP Model	ESN Model
Aladdin	0.44	17.9	18.9	17.1	16.9	16.9
ARD Talk	0.54	7.7	7.9	7.3	7.3	12.9
Die Hard III	0.25	24.9	21.2	19.7	22.7	20.1
Jurassic Park I	0.77	16.1	15.6	22.0	14.6	29.2
Lecture Room	0.06	120.4	11.5	116.8	165.3	4.7
Silence of the Lambs	0.11	41.5	28.6	49.5	59.2	44.9
Skiing	0.19	19.7	20.3	27.2	37.8	29.5
StarWars	0.28	11.8	12.0	10.2	11.7	10.2

Figures 161 and 162 show the performance of these prediction schemes for a 100 second window for the movie traces *Aladdin* and *StarWars* respectively. It can be observed from the table that for both the linear and non-linear models, the prediction errors are high. Though for the high bit rate videos the RMLP predictor is still better than the others, the errors in predictions are significantly high. Thus it can

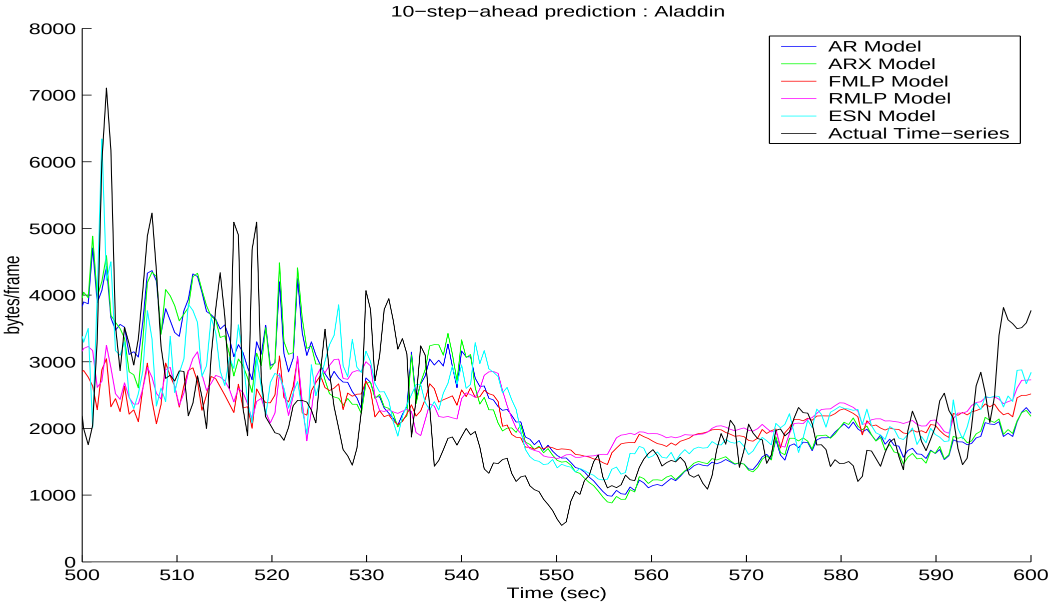

Fig. 161. Ten-SP sizes of the moving average VOPs of *Aladdin* for all models.

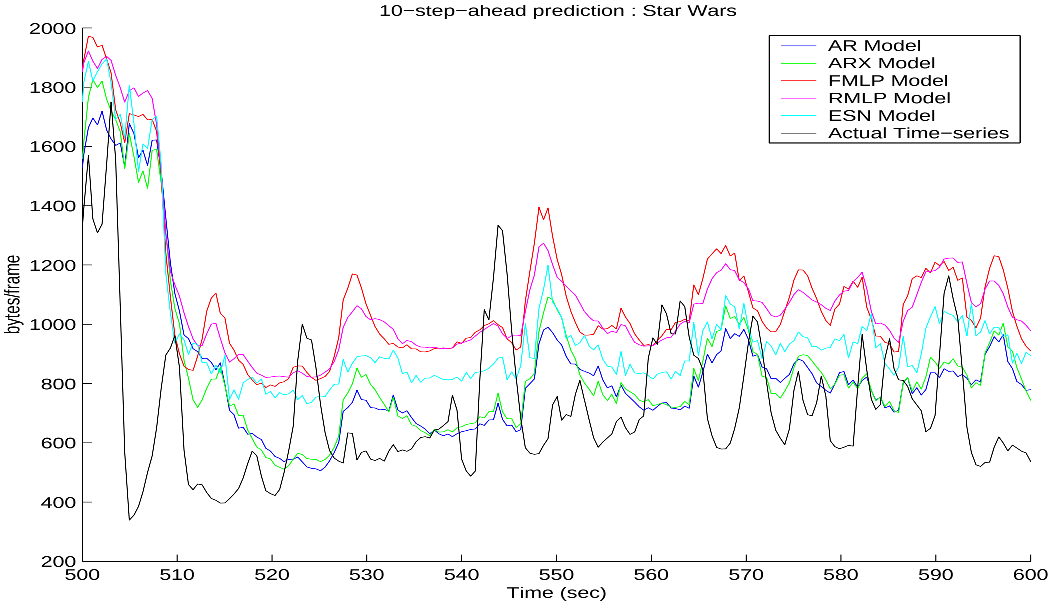

Fig. 162. Ten-SP sizes of the moving average VOPs of *StarWars* for all models.

be concluded that all the models fail to predict the MPEG-coded video source traffic ten-step-ahead.

B. Conclusions

The objective of the current research is to develop multi-step-ahead prediction schemes for the prediction of MPEG-coded video source traffic. This work mainly focuses on the development of prediction models for the moving average time-series of the VOP sizes. Few models for the prediction of I-VOPs are also developed. A glance at the I-VOPs prediction results shows that the prediction errors for I-VOPs are comparable with the prediction errors of the moving average time-series. Literature shows that the I-VOPs are easier to predict than the P- or B-VOPs. In order to have a good estimate of the GOV size and hence the bit rate, one needs to have accurate predictions of at least I- and P-VOPs. But since the errors in the prediction of P-VOPs are much higher ([21], [22] and [30]), the combined result of I- and P-VOP predictions is bound to be worse than the moving average predictions. This justifies the use of moving average of the VOP size time-series over the individual VOP siz time-series.

This work can be concluded with the following statements.

1. Recurrent neural network models give good results for the prediction of the video traffic than linear or non-recurrent neural networks.

2. The RMLP models perform better for high bit rate video traces whereas the ESN models perform well for low bit rates.

3. The prediction deteriorates significantly with increase in the horizon beyond four-steps-ahead.

4. This work suggests the ESN prediction model which performs better than the

existing models for multi-step-ahead predictions for low bit rate videos.

C. Recommendations for Future Work

This work shows that there is no single model that performs well for all bit rates. Some recommendations for future work are:

1. Use of more than one model for multi-step-ahead prediction of the source video traffic. This requires the design of a scheme which switches between the prediction models depending on the bit rate of the video traffic.

2. Design of non-linear prediction models which can be adapted online. Till now researchers have used linear, non-linear and adaptive linear models for the prediction of MPEG-coded video source traffic. The domain non-linear modeling techniques which can be adapted online for the prediction of MPEG-coded video source traffic has not been explored.

3. Design of a control scheme for efficient delivery of multimedia traffic using the output of the empirical models described in this research work.

REFERENCES

[1] H. Kalva, *Delivering MPEG-4 Based Audio-Visual Services*. Boston: Kluwer Academic Services, 2001.

[2] P. Symes, *Video Compression Demystified*, New York: McGraw Hill, 2001.

[3] R. M. Losee, “A discipline independent definition of information,” *Journal of American Society of Information Science*, vol. 48, No. 3, pp. 254–269, March 1997.

[4] J.J. Bae and T. Suda, “Survey of traffic control schemes and protocols in ATM networks,” in *Proceedings of the IEEE*, vol. 79, pp. 170–189, February 1991.

[5] V.S. Frost and B. Melamed, “Traffic modeling for telecommunications networks,” in *IEEE Commun. Mag.*, pp. 70–81, March 1994.

[6] M. Krunz and H. Hughes, “A traffic model for MPEG-coded VBR streams,” in *Proceedings of Joint International Conference on Measurement and Modeling of Computer Systems*, Ottawa, Canada, May 1995, pp. 47–55.

[7] D.P. Heyman and T.V. Lakshman, “Source models for VBR broadcast-video traffic,” in *IEEE/ACM Transactions on Networking*, vol. 4, pp. 40–48, February 1996.

[8] X. Wang, S. Jung, and J.S. Meditch, “VBR broadcast video traffic modeling-a wavelet decomposition approach,” in *Global Telecommunications Conference, GLOBECOM, IEEE*, Phoenix, November 1997, pp. 1052–1056.

[9] N. Rožić and M. Vojnović, "Source modeling of MPEG video," in *Global Telecommunications Conference, GLOBECOM, IEEE*, Phoenix, November 1997, pp. 1429–1433.

[10] M. Krunz and S.K. Tripathi, "On the characterization of VBR MPEG streams," in *Proceedings of ACM SIGMETRICS*, Seattle, June 1997, pp. 192–202.

[11] D. Liu, E.I. Sára, and W. Sun, "Nested auto-regressive processes for MPEG-encoded video traffic modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 169–183, February 2001.

[12] M. Krunz and A.M. Makowski, "Modeling video traffic using $M/G/\infty$ input processes: a compromise between Markovian and LRD models," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 733–747, June 1998.

[13] J. Feng and K.T. Lo, "A simple hierarchical traffic model of VBR MPEG video," in *International Performance, Computing and Communications Conference, IEEE*, Tempe/Phoenix, February 1998, pp. 147–153.

[14] G. Chiruvolu, T.K. Das, R. Sankar, and N. Ranganathan, "A scene-based generalized Markov chain model for VBR video traffic," in *IEEE International Conference on Communications, ICC, IEEE*, Atlanta, June 1998, pp. 554–558.

[15] H. Liu, N. Ansari, and Y.Q. Shi, "Modeling VBR video traffic by Markov-modulated self-similar processes," presented at the *IEEE 3rd Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, September 1999.

[16] P. Manzoni, P. Cremonesi, and G. Serazzi, "Workload models of VBR video traffic and their use in resource allocation policies," in *IEEE/ACM Transactions on Networking*, vol. 7, pp. 387–397, June 1999.

[17] N.D. Doulamis, A.D. Doulamis, G.E. Konstantoulakis, and G.I. Stassinopoulos, "Efficient modeling of VBR MPEG-1 coded video sources," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 93–112, February 2000.

[18] K. Chandra and A.R. Reibman, "Modeling of one- and two-layer variable bit rate video," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 398–413, June 1999.

[19] J.A. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.

[20] E. He, F. Du, X. Dong, L.M. Ni, and H.D. Hughes, "Video traffic modeling over wireless networks," in *IEEE International Conference on Communications. ICC, IEEE*, New Orleans, June 2000, pp. 536–542.

[21] A. M. Adas, "Using adaptive linear prediction to support real-time VBR video under RCBR network service model," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 635–644, October 1998.

[22] S.J. Yoo, "Efficient traffic prediction scheme for real-time VBR MPEG video transmission over high-speed networks," *IEEE Transactions on Broadcasting*, vol. 48, pp. 10–18, March 2002.

[23] A. Chodorek and R.R. Chodorek, "An MPEG-2 video traffic prediction based on phase space analysis and its application to on-line dynamic bandwidth allocation," in *2nd European Conference on Universal Multiservice Networks. EDUMN*, Vienna, April 8–10 2002, pp. 44–55.

[24] A. Chodorek, "A fast and efficient model of an MPEG-4 video traffic based on phase space linearised decomposition," in *14th European Simulation Symposium*

and Exhibition, Dresden, Germany, October 2002, pp. 249–254.

[25] A.A. Tarraf, I.W. Habib, T.N. Saadawi, and S.A. Ahmed, "ATM multimedia traffic prediction using neural networks," in *Proceedings of Global Data Networking*, IEEE, Cairo, December 1993, pp. 77–84.

[26] W.M. Moh, M.J. Chen, N.M. Chu, and C.D. Liao, "Traffic prediction and dynamic bandwidth allocation over ATM: A neural network approach," *Journal of Computer Communications*, vol. 18, pp. 563–571, August 1995.

[27] P.R. Chang and J.T. Hu, "Optimal nonlinear adaptive prediction and modeling of MPEG video in ATM networks using pipelined recurrent neural networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 1087–1100, August 1997.

[28] A.D. Doulamis, N.D. Doulamis, and S.D. Kollias, "Non-linear traffic modeling of VBR MPEG-2 video sources," in *IEEE International Conference on Multimedia and Expo. ICME, IEEE*, New York, July-August 2000, pp. 1318–1321.

[29] A.D. Doulamis, N.D. Doulamis, and S.D. Kollias, "Recursive non linear models for on line traffic prediction of VBR MPEG coded video sources," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN, IEEE*, Como, Italy, July 2000, pp. 114–119.

[30] A. Bhattacharya, A. G. Parlos, A. F. Atiya, "Prediction of MPEG-coded video source traffic using recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2177–2190, August 2003.

[31] O. Rose, "Index of /MPEG," University of Würzburg, Würzburg, Germany. [Online]. Available: http://nero.informatik.uni-wuerzburg.de/MPEG.

[32] A.G. Parlos, O.T. Rais, and A.F. Atiya, “Multi-step-ahead prediction using dynamic recurrent neural networks,” *Neural Networks*, vol. 13, pp. 765–786, September 2000.

[33] H. Jaeger, “The echo state approach to analyzing and training recurrent neural networks,” in *GMD Report 148, GMD - German National Research Institute for Computer Science*, 2001. http://www.gmd.de/People/Herbert.Jaeger/Publications.html

[34] A. Bhattacharya, “Prediction of MPEG-coded video source traffic using recurrent neural networks,” M.S. thesis, Texas A&M University, College Station, Texas, 2002.

[35] K. Kim, “Sensorless fault diagnosis of induction motors,” Ph.D. dissertation, Texas A&M University, College Station, Texas, 2001.

[36] S. Haykin, *Neural Networks - A Comprehensive Foundation*. Upper Saddle River, New Jersey: Prentice Hall, 1994.

[37] M. Negnevitsky, *Artificial Intelligence - A Guide to Intelligent Systems*. Essex, England: Addison Wesley, 2002.

[38] A.G. Parlos, K.T. Chong, and A.F. Atiya, “Application of the recurrent multilayer perceptron in modelling complex process dynamics,” *IEEE Transactions on Neural Networks, Special Issue on Dynamic Recurrent Neural Networks: Theory and Applications*, vol. 5, pp. 255–266, March 1994.

[39] F.H.P. Fitzek and M. Reisslein,“MPEG-4 and H.263 video traces for network performance evaluation,” Technical University of Berlin, Germany. [Online]. Available: http://www-tkn.ee.tu-berlin.de/research/trace/trace.html

[40] F.H.P. Fitzek and M. Reisslein, “MPEG-4 and H.263 video traces for network performance evaluation,” TKN Technical Report, Telecommunications Networks Group, Technical University of Berlin, Berlin, October 2000.

[41] J. Walter, “bttvgrab,” [Online]. Available: http://www.garni.ch/bttvgrab/.

VITA

Deepanker Gupta was born in Kota, Rajasthan, India on the 11th of November, 1979. He received his Bachelor of Technology degree in manufacturing science and engineering from the Indian Institute of Technology, Kharagpur, India, in June 2002. In August 2002, he joined the master's degree program in mechanical engineering at Texas A&M University, College Station, Texas. His main area of research is system modeling and controls.

Contact Address:
313 Lincoln Avenue,
Apt # 5,
College Station, Tx 77840.
Contact phone: (979) 571 7126.
Contact email address: deepanker@tamu.edu

The typist for this thesis was Deepanker Gupta.