International Journal of Computer Theory and Engineering, Vol. 2, No. 2 April, 2010
1793-8201

Kernel Least Mean Square Features For HMM-Based Signal Recognition

Seyed Hossein Ghafarian, Hadi Sadoghi Yazdi, Hamidreza Baradaran Kashani

Abstract— In this paper, an attempt is made to propose a new feature extraction method that is capable of capturing nonlinearities in signals. For this purpose, Kernel Least Mean Square KLMS (KLMS) method is used to extract features from signal and in order to evaluate it, Hidden Markov Model (HMM) is used to model extracted feature sequence and to recognize it from other models. In HMM, Gaussian Mixture Model is used. By introducing noise on signal, results showed that recognition rate in the same level of noise is good but in other SNR values it can degrade. It is also compared with Linear Predictive Coding (LPC). Results showed that in low noise level, the proposed feature extraction has better results but in high noise level LPC has better results.

***Index Terms*——Kernel least mean square, feature extraction, nonlinear prediction, linear predictive coding, signal recognition.**

I. Introduction

Prediction of signals has many applications in signal recognition and coding. Linear structure adaptive architectures are suitable for the prediction of signals, such as Linear Prediction Coding (LPC), etc, but they do not exploit their inherent nonlinearity and associated higher order statistics. The prediction of time series is equivalent to modeling the underlying physical mechanism that produces signals. In many physical mechanisms the underlying process has two distinct characteristics: nonlinearity and nonstationarity. In other cases, the underlying mechanisms have chaotic characteristics. In classical example of Speech, signals are produced by a dynamical system which is nonlinear and statistically nonstationary. In these and many other cases, nature of signal leads researchers and engineers to apply nonlinear prediction approaches in signal analysis. By using nonlinear predictor, one can achieve more feasibility and better results, because linear models are optimal just for Gaussian signals, which is not the case of speech signals [1]. Although in one of the classical papers written by D. Gabor using nonlinear predictors such as Volterra is emphasized [2], but nonlinear nature of speech signals has not absorbed enough attention that it deserves during the past years [3].

Various approaches to nonlinear prediction have been introduced. Some of these approaches use neural networks with various architectures, while others use methods like kriging which is an interpolation method. In recent years other tools were also introduced that can be used to model nonlinear systems. One of these analysis tools is Kernel Least Mean Square (KLMS) [4, and 5]. In this paper, our goal is to introduce a new nonlinear approach based on KLMS and compare it to traditional Linear Prediction Coding (LPC).

In the following sections, we survey nonlinear prediction literature especially on speech signals. KLMS and its application are discussed in section 1.2 and our motivation is also discussed in 1.3.

A. Nonlinear prediction: Literature on speech

Speech applications usually use a linear prediction model for vocal tract. This method is based on source-filter model. Usually the filter is linear and based on linear prediction. The source or excitation for filter is left undefined, modeled as noise or described by a simple pulse train. This model has been successfully applied in various applications but has some major drawbacks. It neglects the structure known to be present in speech signals. This results in inferior ability to discriminate speech sounds. Mainly it cannot model nonlinearity in speech production mechanisms. The sources of nonlinearity are nonlinearity of the production of speech and noise, nonlinearity of signal acquisition system, nonlinearities of transmission channel, and nonlinearity of human perception system [6].

Some problems are more difficult to solve with linear techniques and are more tractable with nonlinear techniques. But there are some drawbacks when working with nonlinear techniques.

- Lack of unifying theory between different nonlinear techniques (neural nets, homomorphic, polynomial, etc)
- Computational cost much more than linear techniques.
- Difficulty in analyzing a nonlinear system. Several useful tools like frequency transform domain analysis are not valid. Several efforts in this field do not produce general results.
- In some techniques there is not a closed formulation to derive nonlinear models. So, an iterative procedure must be used and we may have local minima problem.

Assuming that we use a linear predictor and a nonlinear predictor on the same speech signal, it is observed that the prediction residual of nonlinear predictor has smaller energy.

Manuscript received August 24, 2009. (Write the date on which you submitted your paper for review.)

Seyed Hossein Ghafarian is an university lecturer in Mashhad (e-mail: s_h_ghafarian@ um.ac.ir).

Hadi Sadoghi Yazdi is with Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran (e-mail: sadoghi@sttu.ac.ir).

Hamid Reza baradaran is a MSc student in Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran(e-mail:hamidreza.baradaran@gmail.com)

It can also eliminate the pitch periodicity without the use of a long term predictor [6].

There is a large variety of methods in literature and it's difficult to classify those. Nonlinear predictors are part of the more general class of nonlinear autoregressive methods. For practical design of nonlinear autoregressive models various methods proposed. These methods are categorized in two main categories:

Parametric methods:

One of the best examples for parametric methods is polynomial approximation (Volterra series, etc). Other methods are locally linear models, state dependent models, etc. It's obvious that with a high-enough order polynomial, you can approximate any smooth function. But the number of parameters can be very high. Another drawback of polynomials is that polynomials can go to infinity outside of sampling range. Also an important group of parametric methods are based on neural nets, radial basis function approximation, multilayer perceptrons, and recurrent neural nets.

Nonparametric methods:

Various methods based on nearest neighbor methods are proposed including Lorenz method of analogues, nonlinear predictive vector quantization or codebook quantization. Kernel density estimates is another approach in nonparametric methods [6].

Two classical nonlinear parametric models are Volterra and NARMAX models. A Discreet-time time-invariant Volterra model with memory length m_x and order of nonlinearity D is defined by the input-output relationship

$$\eta_k = \sum_{i=1}^{D} \sum_{j_1,\dots,j_i=0}^{m_x} h_i(j_1,\dots,j_i) x_{k-j_1} \dots x_{k-j_i} \tag{1}$$

Where η_k is the model response at time k associated with input sequence $\{x_k\}$, and $h_i(j_1,\dots,j_i)$ is referred to as i-th order Volterra Kernel. The number of parameters in the model is given by unique products in above equation i.e.

$$\frac{(D+m_x)!}{D!m_x!} - 1 \tag{2}$$

The observation $\{y_k\}$ are then assumed to be given by

$$y_k = \eta_k + \varepsilon_k \tag{3}$$

The nonlinear autoregressive moving average model with exogenous inputs (NARMAX) [7] model is a general parametric model. It is defined by

$$\begin{aligned} y_k = f(&y_{k-1},\dots,y_{k-m}, x_k,\dots,x_{k-m_\varepsilon+1}, \\ &\varepsilon_{k-1},\dots,\varepsilon_{k-m_\varepsilon}) + \varepsilon_k \end{aligned} \tag{4}$$

Where $\{\varepsilon_k\}$ is an i.i.d. sequence of errors and m_ε is the associated memory length. In this equation f is a polynomial or rational function.

In both of Volterra and NARMAX models, the number of parameters quickly increases with D, m_x and m_y. As a result, large data sets are needed in order to obtain an accurate estimation of model parameters. One method to deal with this problem is to use a semi-parametric method based on kriging [8]. Kriging is defined as "Optimal interpolation based on regression against observed *z* values of surrounding data points, weighted according to spatial covariance values". J.-P. Costa, et al, in [8] proposed a nonlinear prediction based on kriging. One of the advantages of kriging is that it helps to compensate for the effects of data clustering, assigning individual points within a cluster less weight than isolated data points.

Nearest neighbor methods assumes that behavior of the system is dependent on the state the system is in. So observation in nearest neighbor methods is a whole bunch of state and their subsequent behavior. To predict the output of the system in a new state that hasn't seen, find the most similar state and predict it will do likewise or find some of the closest states and average the behavior of these states in order to predict behavior in new state. Nearest neighbor methods also assumes that behavior of system changes smoothly with states and that there is a way to recognize state from observations.

To deal with nonlinearity, in 1995, Haykin and Li introduced a novel approach for the nonlinear prediction of nonstationary signals. By utilizing distributed nonlinearity built into design of neural network and ability to learn from its environment, it is well suited for this task. They proposed neural network architecture and learning algorithm that is able to learn on the fly. So it can learn to adapt to the incoming time series. The system decomposed into two subsections, nonlinear subsection which is responsible for linear mapping of input time series into a linear space and linear subsection that is a conventional tapped-delay-line (TDL) filter. This combination of nonlinear and linear processing should be able to extract both nonlinear and linear relationships contained in the input signal. The nonlinear subsection consists of Pipelined Recurrent Neural Networks (PRNN). PRNN itself consists of several identical modules, performs a nonlinear mapping from the input space to another space in order to linearize the input signal. PRNN does that by filtering a set of samples of the input signals using feedbacks in PRNN architecture. The TDL filter used to do a linear mapping from the new space to output space. It's important to notice that both of these operations are performed adaptively on a continuous basis. It is shown that nonlinear adaptive predictor outperforms the traditional linear adaptive scheme, significantly [9].

Baltersee and Chambers, in 1998, further worked on PRNN and proposed two more computationally efficient algorithms based on gradient descent (GD) and extended recursive least squares (ERLS) learning algorithm. They reported better performance achievement by ERLS learning algorithm [10].

In 2007, Stavrakoudis et al introduced Pipelined Recurrent Fuzzy neural networks (PRFNN). The overall structure of this model is similar to PRNN i.e. it consists of nonlinear and linear subsections but instead of RNN module it has Recurrent Fuzzy neural network (RFNN) module. RFNN has better qualities compared to RNNs, including superior modeling performance due to local modeling and fuzzy partition of input space, linguistic description in terms of dynamic IF/Then rules, proper structure learning based on training examples. It is shown that PRFNNs are superior to

PRNN in regard to prediction gain [11].

Some of the authors/researchers worked on different other types of neural networks for nonlinear prediction, M. Faúndez-Zanuy, proposed a nonlinear predictive speech encoder based on an adaptive nonlinear combiner with a neural net that weighs the prediction of several nonlinear predictors. It uses several nonlinear predictors as experts and a combiner that produces overall output, a technique known as ensemble averaging, committee machine in pattern recognition. It then uses three different neural nets, Multi-Layer Perceptrons (MLP), Recurrent Elman Nets (Elman) and Radial Basis Functions (RBF) as different nonlinear predictors [12].

B. *Kernel discussion*

In recent years, kernel methods are becoming very popular. This is due to its solid mathematical foundation and considerable experimental success. By kernel methods (KM), it's possible to map data to a high dimensional Euclidean feature space known as Hilbert space. This mapping can be quite general. At the heart of KM is a kernel function. A kernel function is a function that enables KM to operate at the mapped feature space without even computing coordinates at that space. This is done with the aid of famed kernel trick. Any kernel function must satisfy mercer conditions. Many algorithms developed that works based on KM. One of these algorithms is kernel least mean square (KLMS) [4, 5], is also based on LMS algorithm.

C. *Motivation*

In various signal recognition applications what is needed is a more robust modeling of system. Modeling nonlinearities can be very useful in better recognition rate or coding of speech, especially when dealing with speech production system. As discussed briefly in section 1.2, there is some evidence that constructing a nonlinear model for speech and signals can produce better features for speech recognition or coding. As a method for doing this, in this paper Kernel Least Mean Square (KLMS) is chosen. KLMS has strong nonlinear modeling capabilities.

The paper is organized in the following manner. In Section 2, structure of the proposed method is discussed in the feature extraction of signal and recognition. In Section 3, an experimental study of the KLMS–based features in signal recognition and its result has been described. Conclusion is presented in the final Section.

II. The proposed Scheme

Block diagram of the proposed scheme is presented in following figure. This scheme includes three main parts. First part deals with feature extraction using KLMS algorithm. Second part includes HMM models for recognizing of various signals and final part does select model using MLE (Maximum Likelihood Estimation) procedure. We are able to recognize different signals based on nonlinear feature and apply to conventional HMM models. Superiority of this scheme is utilization of nonlinear feature extraction procedure. We compare this feature with LPC (Linear Prediction Coding) based feature and we will show ability of KLMS features. Following subsection explains three stages of the proposed scheme.

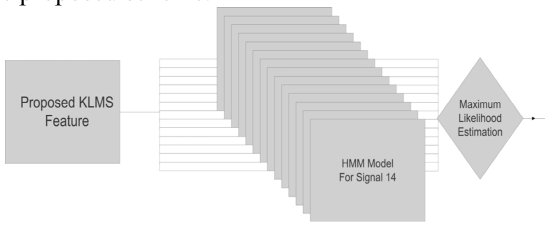

Figure 1:Block Diagram of the proposed scheme

A. *Kernel Least Mean Square Based (KLMS) Features*

The purpose of using KLMS is to map input signal to a high dimensional space, hoping that nonlinear features can be extracted from signal which is linear in this new space. After mapping, linear algorithms can be used on this high dimensional space. This simple idea helps to use linear methods for extraction of nonlinear features. First, we mention KLMS algorithm then feature extraction procedure is discussed.

1) *Kernel Least Mean Square (KLMS) algorithm*

LMS algorithm is an efficient simplification of gradient decent method. By using local estimate of mean square error, it uses a stochastic gradient decent instead of deterministic gradient method. It is used extensively in all areas of adaptive learning. LMS is efficient partly because it avoids estimation of correlation function and matrix inversion. Unfortunately it's not the case for nonlinear algorithms.

In order to LMS can be used in nonlinear cases, at first data points must be mapped in high dimensional space by a mapping function. The basic idea of kernel methods is to map x_i into feature space $\varphi(x_i)$, where the inner product of two vectors can be computed using a positive definite kernel function satisfying mercer conditions:

$$K(x_i, x_j) = < \varphi(x_i), \varphi(x_j) > \quad (5)$$

This allows obtaining nonlinear versions of any linear algorithms expressed in terms of inner product without even knowing mapping function $\varphi(x_i)$. An interesting characteristic of the feature space is that it is a reproducing kernel Hilbert space (RKHS): i.e. the span of functions $\{k(.,x) : x\varepsilon\chi\ \}$ defines a unique functional Hilbert space. Crucial property of this space is the reproducing property

$$f(x) = < k(.,x), f >, \forall f \varepsilon F \quad (6)$$

In particular, a nonlinear mapping can be defined as

$$\varphi(x) = k(.,x)$$

$$< \varphi(x), \varphi(y) > = < k(.,x), k(.,y) > = k(x,y) \quad (7)$$

The LMS algorithm minimizes the empirical risk:

$$\min_{w} R_{emp}[w\varepsilon H_1, Z] = \sum_{i=1}^{n} (y_i - w(u_i))^2 \quad (8)$$

Where H_1 is hypothesis space consists of all the linear operators on U, denoted by $w: U \longrightarrow R$ and

$$((u_1, y_1), \ldots (u_N, y_N)) \varepsilon Z^N, \quad (9)$$

$(u_i, y_i) \in Z = U \times Y$

Using stochastic gradient, it can be shown that

$$w_0 = 0$$

$$e_n^a = y_n - w_{n-1}(u_n) \tag{10}$$

$$w_n = w_{n-1} + \eta e_n^a u_n$$

e_n^a is called priori error and η is the step size. By repeated application of above equation we have,

$$w_n = \eta \sum_{i=1}^{n} e_i^a u_i \tag{11}$$

The input-output operation of this learning system can be solely expressed in terms of inner products

$$\tilde{y} = w_n(\tilde{u}) = \eta \sum_{i=1}^{n} e_i^a < u_i, \tilde{u} > U \tag{12}$$

$$e_n^a = y_n - \eta \sum_{i=1}^{n} e_i^a < u_i, \tilde{u} > U \tag{13}$$

By Mercer's theorem, for any kernel, there exist a mapping ϕ such that

$$k(u_1, u_2) = \phi(u_1)\phi^T(u_2) \tag{14}$$

Now, we can perform LMS on example sequence $\{(\phi(u_1), y_1), ..., (\phi(u_N), y_N)\}$

$$\Omega_0 = 0$$

$$e_n^a = y_n - \Omega_{n-1}(\phi(u_n)) \tag{15}$$

$$\Omega_n = \Omega_{n-1} + \eta e_n^a \phi(u_n)$$

By using (12), (13) and (14)

$$\Omega_{n-1}(\phi(u_n)) = \eta \sum_{i=1}^{n-1} e_i^a k(u_i, u_n)$$

$$e_n^a = y_n - \eta \sum_{i=1}^{n-1} e_i^a k(u_i, u_n) \tag{16}$$

And the final input-output relation of the learning algorithm is

$$\Omega_N = \eta \sum_{i=1}^{N} e_i^a \phi(u_i)$$

$$\tilde{y} = \eta \sum_{i=1}^{N} e_i^a k(u_i, \tilde{u}) \tag{17}$$

It is clear that the well-posedness of KLMS is dependent upon the step size and training data size which also affects the speed of convergence and misadjustment [5].

2) Features Extraction using KLMS

In the following figure, input signal partitioned to several frames. Each frame is applied to KLMS algorithm for extraction of suitable features as shown in Fig 2. Feature vector include combining of obtained KLMS errors from each frames.

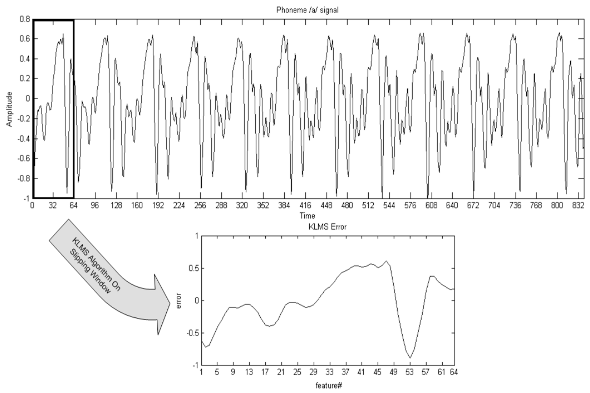

Figure 2:KLMS Feature extraction

Error output in equation (16) is used as extracted feature. In doing this, a slipping window of the signal is obtained and is sent as input to KLMS algorithm. It is assumed that the windowed signal can be represented at least for the purpose of signal recognition by error function (e_n^a). Error function of the KLMS is of the same size of window and is used as an observation vector in hidden Markov model. In obtaining window, a Hanning window is used.

In order to have a better noise immunity, four models trained for four different SNR range values. Each signal has a corresponding model by which it can be recognized much better than the model that has been trained with the same SNR range as the signal. The problem is how to detect signal to noise ratio of the signal. Fortunately there is a vast body of research in recognizing signal to noise ratio. Several methods have been proposed [13], [14].

In equation (17), parameter η is the step size. This parameter must satisfy the following inequality [5]:

$$\eta < N / tr[G_\Phi] \tag{18}$$

Otherwise the algorithm does not converge and also is not stable.

Kernel function is a measure of similarity between input vectors in high dimensional feature space. It must have positive definite kernel properties. The kernel function used in this work is Translation-invariant radial basis function (Gaussian) kernel which is the widely used mercer kernel, i.e.

$$k(x, y) = \exp-(\frac{\|x-y\|^2}{2\partial^2}) \tag{19}$$

Algorithm used for extraction of features from signal is shown is in Algorithm I. As you can see, the algorithm computes e_t^a in equation (16) and $\tilde{y}$ in equation (17) iteratively. After iterating over all input-output pairs, error output from start to end of window (e_1^a to e_t^a) sent to next stage which is HMM. In the next section, issues about HMM is discussed.

Algorithm I:Kernel Feature Extraction

Input: Window (u_i, y_i) size N of Signal(u_i, y_i)

Initialization

f^0 =0,

η :learning step,

∂ :kernel width parameter

loop over input-output(u, y)
{
$e_t^a = y - f^{t-1}(u)$:Equation (16)
$c_{new} = u, h_{new} = \eta e_t^a$
$f^t = h_{new} k_\partial (c_{new}, .) + f^{t-1}$
:Equation(17)
}
$e_{1..t}^a$ as extracted features

B. HMM-based signal recognition

Hidden Markov models have good capabilities in modeling statistical pattern. But it's very dependent on structure and types of models being used. In order to have better modeling capabilities a continuous density Hidden Markov observation is used with three states. In fact Mixture of Gaussian Model with two Gaussian Mixtures is used as observation.

The covariance type for mixture of Gaussian is full. Length of the sequence is 1024. Signals is trained with 10 sequence of features extracted from signal for each model. The signal was subjected to noise with SNR from 20 to 5 dB. Each input in sequence was a vector with window length element. In training maximum number of iteration was chose 5. In each cycle of iteration on expectation maximization in training HMM models, a covariance prior matrix equal to 0.01*I added to Sigma of Mixture of Gaussians (I: Identity Matrix).

C. Maximum Likelihood Estimation

A family D_θ of probability distributions parameterized by an unknown parameter θ (which could be vector-valued) associated with either a known probability density function denoted as f_θ is considered. A sample $x_1, x_2, ..., x_n$ of n values from this distribution is drawn. Using f_θ, probability density associated with observed data ($x_1, x_2, ..., x_n$) can be computed, equal to $f_\theta(x_1, ..., x_n)$.

The likelihood function is $l(\theta) = f_\theta(x_1, ..., x_n)$. The method of maximum likelihood estimates θ by finding the value of θ that maximizes $l(\theta)$, i.e.

$$\hat{\theta} = \arg\max_\theta l(\theta) \quad (20)$$

By computing $l(\theta)$ for each HMM model, using equation (20), $\hat{\theta}$ can be computed and assumed as recognized input.

III. EXPERIMENTAL RESULTS

In order to analyze the proposed KLMS Feature, we applied white Gaussian noise with SNR from 20 down to 6 dB. We had 14 different signals to detect. The signals are on table 2. The signals are also subject to white noise. Results were shown in fig 3. As you can see, by decreasing signal to noise ratio, both KLMS and LPC recognition rate decreased, but KLMS rate at first less decreased and after signal to noise ratio further decreased, KLMS decreased more rapidly.

In order to investigate behavior of the system under different values for various parameters such as sigma in KLMS and also SNR, we trained the system with SNR value equal to 5 and test these models under different values for Signal SNR and KLMS sigma. HMM models built for SNR 5 dB, but recognition is done on signals with different SNR values. As fig 4 shows, there is not much considerable change by changing sigma from around 0.05 to 1.95.

In another experiment, we test the system with the same SNR as the model trained. By decreasing noise level, recognition rate goes better. As can be seen in fig 4, changing sigma value from 0.05 to 1.95 does not have considerable change in recognition rate in high SNR values, but in low SNR values it's obvious that recognition rate changes with sigma. In yet another experiment, we trained every signal model with various SNR levels. Every model trained with fifteen signals each with a different SNR level. Figure 5 shows test results of these models with signals with different SNR level.

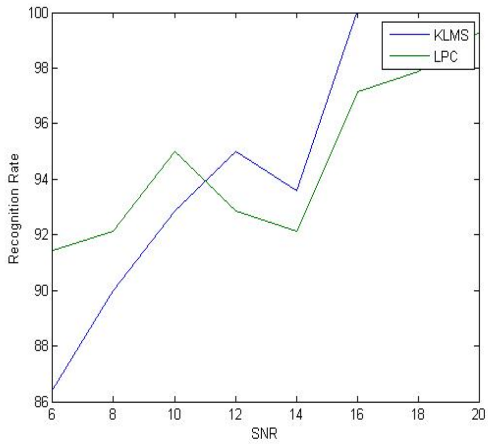

Figure 3:KLMS and LPC Recognition Rate in presence of noise, snr changed from 5 to 20, fig shows that LPC has better result in high noise levels compare to KLMS

By building a confusion matrix (Table 1 and 3), it can be seen that both of KLMS and LPC has confused signals 6 and 7. These Signals both are very similar and essentially are the same with different period. Also in case of KLMS signals 11 and 12 confused in high noise level.

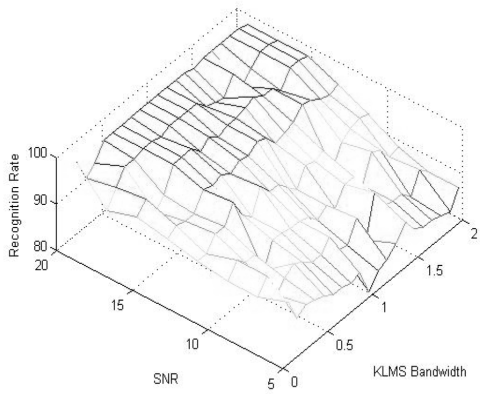

Figure 4:Recognition Rate vs SNR vs KLMS Sigma, as fig shows changing sigma doesnot have considerable recognition rate change in low noise , but there is more variability with sigma in high noise level.

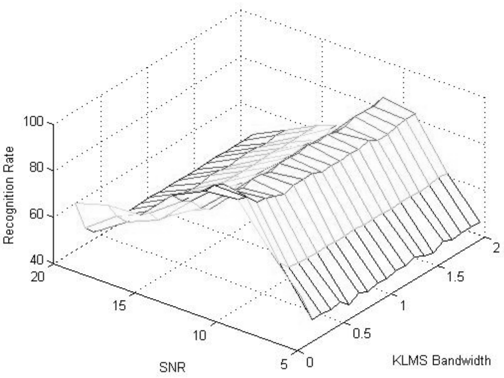

Figure 5:Models Trained for Different SNRs, when models trained with signals with various SNRs,bestresult is in recognizing in middle range of SNR values.

Table I:Signals used in training and recognition

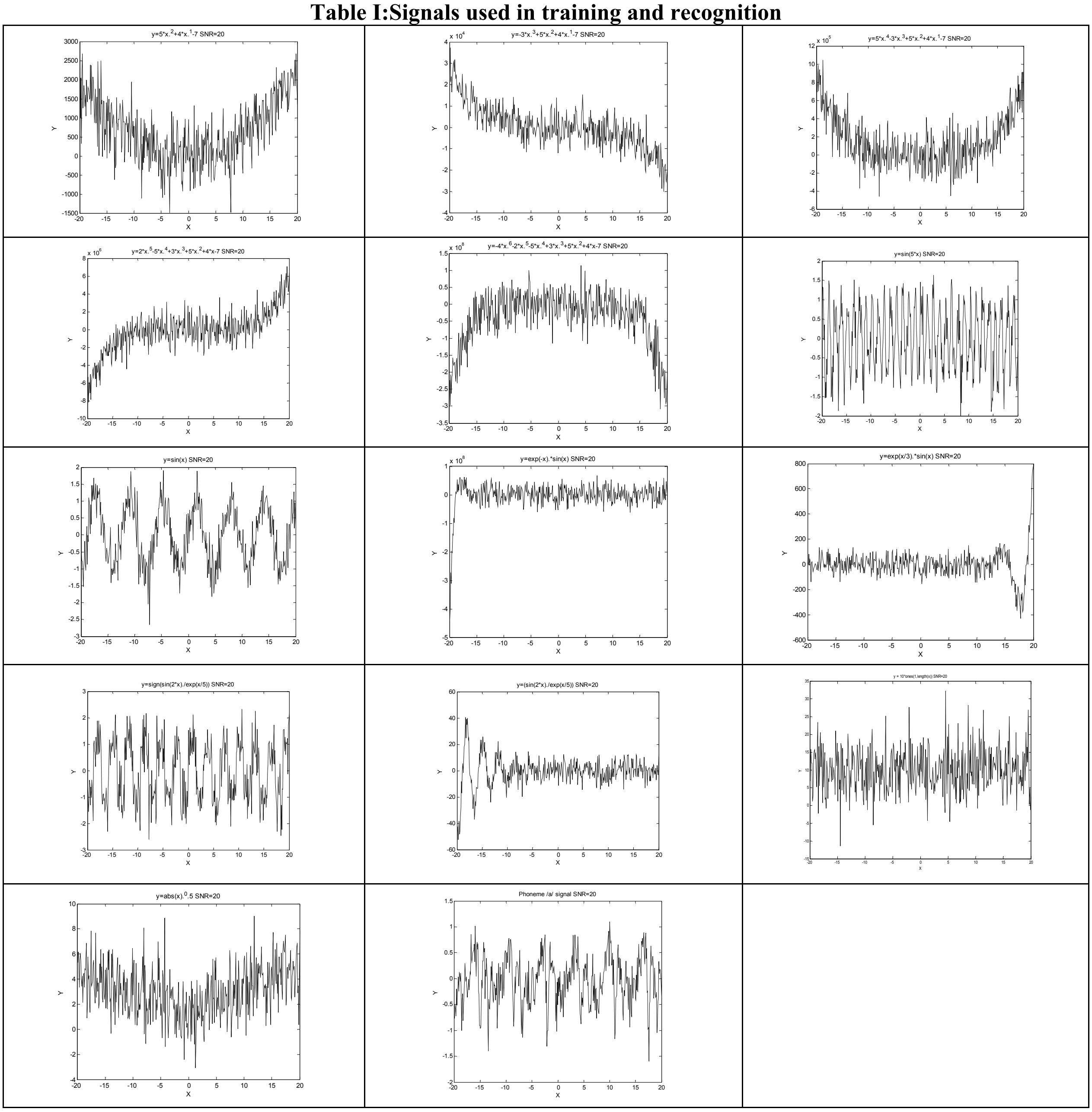

Table II: KLMS-HMM Recognition of training signals

Detected / Signal	1	2	3	4	5	6	7	8	9	10	11	12	13	14

1	10	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	10	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	6	4	0	0	0	0	0	0	0
7	0	0	0	0	0	1	9	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	10	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	10	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	10	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	9	1	0	0
12	0	0	0	0	0	0	0	0	0	0	2	8	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	10	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	10

Table III: LPC-HMM Recognition of training signals

Signal \ Detected	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	10	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	10	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	7	3	0	0	0	0	0	0	0
7	0	0	0	0	0	2	8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	10	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	10	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	10	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	10	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	10	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	10	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	10

IV. Conclusion And Future Works

KLMS has some good characteristics for signal recognition. With its nonlinear capabilities, KLMS can model systems with nonlinear capabilities much better. But since in KLMS we used error output as features, it is very sensitive to noise. Further work must be done in order to find a KLMS feature that in addition to nonlinear modeling capabilities, it can be noise immune.

References

[1] Faúndez-Zanuy, M., McLaughlin, S., Esposito, A., Hussain, A., Schoentgen, J., Kubin, G., Klein, W.B., Maragos, P., Nonlinear speech processing: overview and applications.Control and Intelligent Systems, Vol. 30, no. 1, ACTA Press, pp. 1–10 ,2002

[2] D. Gabor , W. P. L. Wilby, and R. Woodcock, An universal nonlinear filter, predictor and simulator which optimizes itself by a learning process, Proc Inst. Elec. Eng., vol. 108, pp. 422-438, 1961.

[3] N.S. Jayant and P. Noll, Digital Coding of Waveforms, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[4] Puskal P. Pokharel, Weifeng Liu, Jose. C. Principe, KERNEL LMS, p.421, IEEE ,ICASSP 2007.

[5] Weifeng Liu, Puskal P. Pokharel,, and Jose C. Principe, The Kernel Least-Mean-Square Algorithm, IEEE Trans. Signal Processing, vol. 56, no. 2, Feb 2008.

[6] M. Faúndez-Zanuy, G. Kubin, W. B. Kleijn, P. Maragos, S. McLaughlin, A. Esposito, A. Hussain, J, Nonlinear Speech Processing: Overview and Applications.

[7] I. Leontaritis, S. Billings, Input-Output parametric models for nonlinear systems part 2: stochastic nonlinear systems, Int. J. Control 41 (2) (1985) 329-344.

[8] J.-P. Costa*,1, L. Pronzato, E. Thierry, Nonlinear prediction by kriging, with application to noise cancellation ,Signal Processing 80 (2000) 553-566,Elsevier Science.

[9] Simon Haykin, Liang Li, Nonlinear Adaptive Prediction of Nonstationary Signals, IEEE Trans Signal Processing. vol. 43, no. 2, Feb 1995.

[10] Jens Baltersee and Jonathon A. Chambers, Nonlinear Adaptive Prediction of Speech with a Pipelined Recurrent Neural Network, IEEE Trans. Signal Processing. vol. 46, no. 8, Aug 1998.

[11] Dimitris G. Stavrakoudis and John B. Theocharis, Pipelined Recurrent Fuzzy Neural Networks for Nonlinear Adaptive Speech Prediction, IEEE Trans. Systems, Man and Cybernetics Part B: Cybernetics, vol. 37, no. 5, Oct 2007 1305.

[12] M. Faúndez-Zanuy, Speech coding through adaptive combined nonlinear prediction, Speech Communication 48 (2006) 838–847, Elsevier.

[13] N. Yuanfei, G. Jianhua, and W. Yong, 2009, Iterative SNR estimation using a priori information, Digit. Signal Process. 19, 2 (Mar. 2009), 278-286. DOI= http://dx.doi.org/10.1016/j.dsp.2008.07.003.

[14] Yao Ren, M.T. Johnson, An improved SNR estimator for speech enhancement, Acoustics, Speech and Signal Processing, ICASSP, 2008, IEEE Intl Conf on.