Fractional and Kernel Methods with Their Application to Nonlinear Systems

Ph.D Thesis

Submitted By:

Bilal Shoaib

42-FET/PHDEE/F10

Supervised By:

Prof. Dr. Ijaz Mansoor Qureshi

Department of Electronics Engineering

Faculty of Engineering & Technology

International Islamic University, Islamabad

March 27, 2017

Fractional and Kernel Methods with Their Application to Nonlinear Systems

Ph.D Thesis

By

Bilal Shoaib

42-FET/PHDEE/F10

Submitted to the

Department of Electronics Engineering,

International Islamic University, Islamabad

as a partial fulfillment of the requirements

for the award of the degree of

Doctor of Philosophy in Electronics Engineering.

Department of Electronics Engineering

Faculty of Engineering & Technology

International Islamic University, Islamabad

March 27, 2017

A dissertation submitted to the

Department of Electronics Engineering,

International Islamic University, Islamabad

as a partial fulfillment of the requirements

for the award of the degree of

Doctor of Philosophy in Electronics Engineering.

Certificate of Approval

Title of Thesis: Fractional and Kernel Methods with Their Applications to Nonlinear Systems

Name of Student: Bilal Shoaib Khan

Registration No.: 42-FET/PhDEE/F10

Accepted by the department of Electronics Engineering, Faculty of Engineering and Technology, International Islamic University, Islamabad, in partial fulfillment of the requirements for the Doctor of Philosophy Degree in Electronics Engineering.

Viva Voice Committee:

Professor Dr. Aqdas Naveed Malik (Internal Examiner)
Dean Faculty of Engineering and technology,
International Islamic University, Islamabad. _____

Dr. M. M. Talha (External Examiner I)
Principle Scientist
KRL, Islamabad. _____

Dr. Faisal Zafar (External Examiner II)
Principle Scientist,
PAEC, Islamabad. _____

Professor Dr. Ijaz Mansoor Qureshi (Supervisor)
Department of Electrical Engineering
AIR University, Islamabad. _____

Professor Dr. Muhammad Amir
Chairman Department of Electronics Engineering,
International Islamic University, Islamabad. _____

Date: March 27, 2017

Declaration

I hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that no portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Bilal Shoaib

Copyright

March 27, 2017

Dedication

Dedicated to My family, especially to my parents.

Bilal Shoaib

Acknowledgments

I am very grateful to *ALLAH* the *ALMIGHTY* for without His grace and blessing this study would not have been possible.

Foremost, I would like to express my sincere gratitude to my supervisor *Prof. Dr. Ijaz Mansoor Qureshi* for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would like to extend immeasurable appreciation and deepest gratitude for the help and support of my Co-Supervisor *Dr. Ihsanul Haq*, who gave all his knowledge, guidance and support to boost my confidence and learning. His mentoring and encouragement have been specially valuable, and his early insights launched the greater part of this dissertation.

I would also like to acknowledge my friends, students and colleagues especially *Mr. Shafqat-ullah Khan, Mr. Abdul Basit, Mr. Waseem Khan, Mr. Atif Elahi Khan*. All of them encouraged and provided logistic and technical help during this research.

I would also like to thank my wife who has supported me patiently and firmly during completion of this task.

I would like to admit that I owe all my achievements to my truly, sincere and most loving parents and friends who mean the most to me, and whose prayers have always been a source of determination for me.

Abstract

Motivated by the recent development in the field of nonlinear filtering, this dissertation presents a new set of nonlinear fractional and kernel algorithms for the solution of bench mark nonlinear problems. Fractional methods have the ability to handle the nonlinear problems more efficiently as compared with the linear methods, because of the fact that the derivative will be taken by a small time step. A new algorithm modified fractional least mean square algorithm is developed that basically is an extension of fractional least mean square algorithm. A scaling factor is introduced that manually adjust the weightage between the least mean square and fractional least mean square parts of the algorithm according to the nature of the problem. The algorithm is then validated on a set of bench mark nonlinear problems. Another algorithm is developed in which all adjustable parameters of the modified fractional least mean square algorithm is made adaptive by using gradient method accordingly by minimizing the mean square error. A method is also introduced adapt the order of fractional derivative. Secondly the formulation of the kernel fractional affine projection algorithm is introduced with the inclusion of Reimann Louisville fractional derivative to Gradient based stochastic Newton recursive method to minimize the cost function of the kernel affine projection algorithm. This approach extends the idea of fractional signal processing in reproducing kernel Hilbert space. The algorithm is then tested on some well known nonlinear problems. A new identification scheme is established for the Hammerstein nonlinear controlled autoregressive(CAR) system using kernel affine projection algorithm. The proposed scheme is validated in comparison with Affine projection algorithm. A new square root extended kernel recursive least squares algorithm is introduced for the unforced dynamical nonlinear state space model. This algorithm utilizes the numerical properties of the matrix computation by the use of an orthonormal triangularization process that is based on numerically stable givens rotation. It gives a considerable reduction of time and computational complexity. The algorithm is illustrated by discussing its application to stationary, as well as non stationary Mackey Glass series and Lorenz series prediction.

List of Publications

1. **Shoaib, Bilal** and Qureshi, Ijaz Mansoor and others, *A modified fractional least mean square algorithm for chaotic and nonstationary time series prediction*, Chinese Physics B, vol. 23, no. 3, 2014. (**Impact Factor: 1.43**)

2. **Shoaib, Bilal** and Qureshi, Ijaz Mansoor and others, *Adaptive step size modified fractional least mean square algorithm for chaotic time series prediction*, Chinese Physics B, vol. 23, no. 5, 2014.(**Impact Factor: 1.43**)

3. **Shoaib, Bilal** and Qureshi, IM and Haq, IU and Mehmood, Shahid, *Square Root Extended Kernel Recursive Least Squares Algorithm for Nonlinear Channel Equalization*, Research Journal of Applied Sciences, Engineering and Technology, 2013.

4. **Shoaib, Bilal** and Qureshi, Ijaz Mansoor and Khan, Shafqat Ullah and Butt, Sharjeel Abid and others, *Kernel fractional affine projection algorithm*, Applied Informatics, vol. 2, no. 1, 2015.

5. **Shoaib, Bilal** and Qureshi, Ijaz Mansoor and Butt, Sharjeel Abid and Khan, Shafqat Ullah and Khan, Wasim, *Adaptive step size kernel least mean square algorithm for Lorenz time series prediction*, 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 218-221, IEEE, 2015.

6. **Shoaib, Bilal** and Qureshi, Ijaz Mansoor and Khan, Wasim and Khan, Shafqat Ullah and Basit, Abdul, *Numerical solution of nonlinear Boundary value problems in kernel space*, 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 269–273, IEEE, 2016.

7. **Shoaib, Bilal**, Ijaz Mansoor Qureshi, Shafqat Ullah Khan, and Adnan Umer, *A new square root extended kernel recursive least squares algorithm for nonlinear time series prediction*, Submitted to Nonlinear Dynamics.

8. **Shoaib, Bilal**, Ijaz Mansoor Qureshi, Ihsanulhaq, Sharjeel Abid Butt and Shafqat Ullah Khan, *Identification of Hammerstein nonlinear controlled autoregressive system using kernel affine projection algorithm*, Submitted to Canadian journal of physics.

9. Khan, Shafqat Ullah and Qureshi, Ijaz Mansoor and **Shoaib, Bilal** and Basit, Abdul, *Correction of faulty pattern using cuckoo search algorithm and symmetrical element failure technique along with distance adjustment between the antenna array*, 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 633–636, IEEE, 2015.

10. Khan, Shafqat Ullah and Qureshi, Ijaz Mansoor and **Shoaib, Bilal**, *Meta-Heuristic Cuckoo Search Algorithm for the Correction of Faulty Array Antenna*, publications.muet.edu.pk. (**ISI Indexed**)

11. Khan, Shafqat Ullah and Qureshi, Ijaz Mansoor and Naveed, Aqdas and **Shoaib, Bilal** and Basit, Abdul, *Detection of Defective Sensors in Phased Array Using Compressed Sensing and Hybrid Genetic Algorithm*, Journal of Sensors, vol. 2016, 2015. (**Impact factor 1.03**)

12. Khan, Shafqat Ullah and Qureshi, Ijaz Mansoor and Zaman, Fawad and **Shoaib, Bilal** and Ashraf, Kabir, *An application of hybrid nature inspired computational technique to detect faulty element in array antenna*, 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 629-632, IEEE, 2015.

13. Khan, SU and Qureshi, IM and Zaman, F and **Shoaib, B.** and Naveed, A and Basit, A, *Correction of faulty sensors in phased array radars using symmetrical sensor failure technique and cultural algorithm with differential evolution*, The Scientific World Journal, vol. 2014. 2014.

14. Khan, Shafqat Ullah and QURESHI, IJAZ and **Shoaib, Bilal** and Naveed, Aqdas, *Recovery of Failed Element Signal with a Digitally Beamforming Using Linear Symmetrical Array Antenna.*, Journal of Information Science & Engineering, vol. 32, no. 3, 2016.

15. Basit, Abdul and Qureshi, Ijaz Mansoor and Khan, Waseem and Malik, Aqdas Nqveed and **Shoaib, Bilal**, *Beam pattern synthesis for a cognitive frequency diverse array radar to localize multiple targets with same direction but different ranges*, 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 682-688, IEEE, 2016.

16. Khan, Wasim and Qureshi, Ijaz Mansoor and Basit, Abdul and **Shoaib, Bilal**, *Transmit/received beamforming for MIMO log-frequency diverse array radar*, 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 689-693, IEEE, 2016.

17. Khan, Shafqat Ullah and Qureshi, Ijaz Mansoor and Haider, Hassan and Zaman, Fawad and **Shoaib, Bilal**, *Diagnosis of Faulty Sensors in Phased Array Radar Using Compressed Sensing and Hybrid IRLS–SSF Algorithm*, Wireless Personal Communications, pp. 1-20, 2016.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Problem Statement

Fractional and kernel methods have diverse applications in the field of applied science and engineering. Fractional signal processing algorithms are basically nonlinear in nature. A fractional weight adaptation term is added to the algorithm that is derived by minimizing the cost function with the fractional derivative. The theory of fractional calculus is as old as that of classical calculus. However most of its applications are found to be in the last three decades. In fact many scientific areas are currently paying attention to this field, that includes system identification, signal processing, robotics, traffic system, heuristic algorithms, physics as well as as economy trends. In fractional signal processing area, fractional least mean square algorithm is developed previously. The problem arises in this algorithm is that of slow convergence. Divergence phenomena arising in the presence of noise when dealing with the nonlinear systems. Also the problem arises while adjusting the step size of the algorithm manually according to the nature of the problem.

On the other hand kernel methods have numerous applications in nonlinear area. kernel methods have been very much popular in the field of machine learning during the last four to five years. The basic idea behind kernel methods is to transform the input data into some high dimensional feature space where linear operation are performed effectively. Kernel methods use kernel functions to transform the input data, these kernel functions are universal approximators, continuous

and stable. Identification of structured nonlinear systems is still an open issue by using kernel signal processing algorithms. The problem arises is that of divergence in Kernel methods that are based on nonlinear state space model. In this regard the major issues that has to be addressed in this dissertation are given below:

- Unfortunately, the fractional term of the fractional least mean square algorithm must be problem dependent. we do make sure the algorithm does not diverge. So the careful adjustment is required between the linear term and fractional term of the algorithm.
- Manually adjusting the step size of the algorithm is a difficult task. By making it adaptive according to the mean square error reduces the hit and trial effort.
- To find an appropriate way of adjusting the fractional derivative term in the kernel Affine Projection algorithm.
- The reliability and effectiveness of the kernel methods needs to be investigated in the nonlinear system identification problems.
- To find a suitable square root extended kernel recursive least squares algorithm for the unforced nonlinear dynamical model to overcome the divergence phenomena.

1.2 Contribution of the thesis

This dissertation presents newly developed fractional and kernel methods for nonlinear systems. The designed scheme consists of the approximate mathematical modeling of the newly developed fractional and kernel methods. The detail is provided as follows

1. Development of the new fractional least mean square algorithm for nonlinear and non stationary time series prediction. A modification in the Fractional Least Mean Square Algorithm is introduced. We do make sure that the algorithm does not diverge. A parameter β is being intro-

duced that has to be adjusted for optimal results.

2. Development of the new adaptive step size modified fractional least mean square algorithm. In this algorithm the two step sizes and the order of the fractional derivative is made adaptive by using the 5tochastic gradient technique and according to the mean square error. The algorithm is tested on the prediction of stationary as well as non stationary Mackey Glass time series prediction and also the validated on the X-component of Lorenz series prediction.

3. Development of the kernel Fractional affine projection algorithm. The algorithm is validated on the bench mark time series prediction problem, nonlinear channel equalization as well as static function approximation.

4. Application of the kernel Affine projection algorithm to the identification of Hammerstein nonlinear controlled autoregressive system.

5. Development of the new square root extended kernel recursive least squares algorithm for unforced nonlinear dynamical state space model and its application to the prediction of chaotic Mackey Glass and three dimensional lorenz time series prediction.

1.3 Organization of the Thesis

Chapter 2 summarizes the overview of fractional and kernel methods. It starts with the short survey on the history of fractional calculus then provides some basic definition for the important relations and an overview regarding fractional signal processing. It also provides the overview of kernel methods, some basic definitions along with detail about mercer's theorem and some kernel functions.

Chapter 3 provides the detail derivation of the modified fractional least mean square algorithm and adaptive step size modified fractional least mean square algorithm. It also provides the problem setting of the Mackey Glass time series prediction, stationary as well as non stationary.

Chapter 4 Provides the detail derivation and formulation of the kernel fractional affine projection algorithm. It also Provides the Identification algorithm based on kernel Affine projection algorithm for the Hammerstein nonlinear controlled autoregressive system.

Chapter 5 Provides a new square root extended kernel recursive least squares algorithm based on nonlinear unforced dynamical model for nonlinear and nonstationary time series prediction is presented. The mathematical model and detail derivation is also provided for the proposed algorithm.

Chapter 6 summarizes and concludes the dissertation. Some of the future direction and recommendations are also highlighted at the end.

2 Literature Review

2.1 Introduction to Fractional and kernel methods

In this chapter, an overview about the historical development in the field of fractional and kernel signal processing is presented. Basic definition of both these methods along with examples are discussed briefly. Some overview about different kernel functions, the famous kernel trick and then kernel algorithms along with the previous work is also discussed.

2.2 Fractional Methods, An overview

The birth of fractional calculus takes place almost at the same time as that of classical calculus. In 17th centaury first reference was made to the subjective term of the fractional derivative. Theory of fractional calculus is published by a variety of books [1, 2, 3] and they also contain the historical survey of the development in this field. A number of other research articles have been published related to the different aspects of its history [4, 5, 6]. Based on the books and articles mentioned above, a short summary about the history of fractional calculus is presented at the beginning of this thesis. It is controversial about the founder of classical calculus, but is linked with Newton and Leibniz [7, 8]. However the birth of fractional calculus is purely associated with Leibniz. Johann Bernoulli and J. Wallis also made correspondence with Leibniz about the topic of non-

integer order derivatives [9, 10] in 1697 and this work continuous even after the death of Leibniz in 1716. Leonhard Euler for the first time introduced the generalization of factorials of Gamma function [11] in 1783. J. L. Lagrange in 1772 contributed in the development of exponent law for the differential operators of integer order. P.S Laplace has provided the detailed definition of fractional calculus by providing the representation of fractional derivative by an integral, that can be written as

$$\int y(t)t^{-x}dt \tag{2.1}$$

S. F. Lacroix generalize the integer order derivative of function $y(t) = t^m$ to fractional order. It can be given as

$$\frac{d^n}{dt^n}t^m = \frac{m!}{(m-n)!}t^{m-n}, m \geq n. \tag{2.2}$$

Euler gamma function is used by Lacroix to formulate the fractional order derivative as

$$\frac{d^v}{dt^v}t^m = \frac{\Gamma(m+1)}{\Gamma(m-v+1)}t^{m-v}, m \geq v. \tag{2.3}$$

$\frac{1}{2}$ order derivative for the function $y(t) = t$ was also provided by Lacroix and is written as

$$\frac{d^{\frac{1}{2}}}{dt^{\frac{1}{2}}}t^m = \frac{\Gamma(2)}{\Gamma(\frac{3}{2})}t^{\frac{1}{2}} \tag{2.4}$$

In 1823 Neils Herik Abel solved a physical application using fractional operation named as tautochrone problem, and he also describes the solution in detail [10]. J. Lioville provided a unique definition of fractional derivative as a function in the form of an infinite series with limitation to its convergence and applicability to limited class of functions only. Lioville definition is very helpful and is used in many applications to geometrical, physical and mechanical problems [11, 12, 13]
G. F. B. Reiman in 1817 gives his definition for fractional derivative as a generalization of Taylor series. In view of Reiman and Lioville's definition has been developed and is commonly

known as Reimann-Lioville fractional integer. Number of other definitions for fractional derivative or integrals has been available in the literature by some renowned scholars and scientists. M. Riesz [14, 15] developed the integral known as Reisz potential. B. S. Nagy [16] in 1940 developed the technique to approximate the fractional integral and an inequality for the trigonometric sums through the technique known as trigonometric polynomials. The modification in Reimann Lioville fractional integral is done through mellin transform by Erdelyi and Kober [17, 18]. M. Caputo's definition provided in 1960 is worth mentioning and used in variety of scientific applications. The first international conference about the theory of application of fractional calculus is held at New Heaven. First book on fractional calculus is written by oldham and Spanier in 1974 [19]. The first issue of the journal solely related on fractional calculus named "Fractional calculus and Applied Analytic" was published in 1998. The most popular books in this regard are written by Miller and Ross, Samko et al. Podlubuy [20], [21], Ben Auraham and Haulin [22], Carpinteri and Mainardi [23], Ozaktas et al [24], and West at al. [25] etc. In short, fractional calculus covers a wide range of applications in the field of science and engineering.

2.2.1 Basic Definitions

In this section, some basic concepts and theory about the fractional integrals and derivatives is given. Gamma function and Reimann Lioville fractional derivative and integrals are used in this thesis and are discussed briefly.

GAMMA Function

One of the fundamental functions of the fractional calculus is Euler's Gamma function, which provides the generalization of the factorial, $n!$ and permits n to take real as well as complex values.

The definition of the gamma function is given as

$$\Gamma(t) = \int_0^\infty e^{-x} x^{t-1} dx \tag{2.5}$$

By limit representation the Gamma function can also be defined as

$$\Gamma(t) = \lim_{n\to\infty} \frac{n! n^x}{x(x+1)...(x+n)}, Re(x) > 0. \tag{2.6}$$

The two basic properties of Gamma function is given as

$$\Gamma(t+1) = t\Gamma(t) = t(t-1)! = t! \tag{2.7}$$

and the second important property is its simple pole at the points $x = -n, (n = 0, 1, 2, 3, 4, 5.....)$.

Riemann Liouville Fractional Integral and Derivative

Definition of Riemann-Lioville fractional integral of order ν is given as

$$I^\nu f(t) = \frac{1}{\gamma(\nu)} \int_a^t (t-\tau)^{\nu-1} f(\tau) d\tau, \tag{2.8}$$

$$I^0 f(t) = f(t) \tag{2.9}$$

where I^ν is the fractional integral of order ν. The fractional derivative of order ν, $(\nu > 0)$ is written as

$$D^\nu f(t) = (\frac{d}{dt})^n (I^{n-\nu}) f(t), \tag{2.10}$$

where D^{ν} is the fractional derivative of order ν. As an example, let us apply the Reimann Lioville fractional derivative to power function as

$$\begin{aligned} f(t) &= (t-a)^{\alpha} \\ D^{\nu}(t-a)^{\alpha} &= \frac{\gamma(1+\alpha)}{\gamma(\alpha-\nu+1)}(t-a)^{\alpha-\nu}, \end{aligned} \tag{2.11}$$

Where a is some constant and α is the arbitrary real number. Similarly another example of exponential function $e^{\lambda t}$is evaluated as

$$D^{\nu}e^{\lambda t} = t^{-\nu}E_{1,1-\nu}(\lambda) \tag{2.12}$$

2.2.2 Fractional Signal Processing approach

In this, a brief summary has been provided on the application of fractional calculus to signal processing. In recent years, it has been a productive field for research in applied sciences and engineering. Fractional signal processing approach is used in many well known applications as in system identification, equalization of nonlinear channel, nonlinear chaotic time series prediction, control of nonlinear controlled autoregressive systems, Identification of box jenkins system and many more.

A convex combination of fractional least mean square (FLMS) is presented by Masoud Geraunanchizadah [26], which provides an interesting way to improve the performance of adaptive filters. This approach gives better performance in terms of mean square error as a figure of merit for dual channel speech enhancement and also an improved speech quality is observed than the LMS, FLMS, and CCLMS algorithms. Another application of FLMS algorithm is the parameter estimation of input nonlinear controlled autoregressive(INCAR) system [27]. The authors uses FLMS algorithm with different adaptive step size parameters for two examples of INCAR model, and performance is tested and analyzed for different scenarios of signal-to-noise ratios. The opti-

mization problem is also compared with Volterra LMS and Kernel Least mean square algorithm.

A new approach to fractional signal processing is used in [28] based on fractional domains to supress nonstationary noise in Active noise cancellation. The achievement is observed at least twice as compared to the time domain adaptation. A comparative study is made by [29] based on echo cancellation between LMS and Fractional LMS algorithm. A new multi dimensional step size is introduced in [30] for practical active noise control systems. An indirect design approach is established based on step size in relation with disturbance in the signal and with the desired response of the modeling filter.

Another design scheme is established in on single stage fractional least mean square identification and two stage fractional least mean square identification algorithm is introduced for the parameter estimation of CARMA systems. Weight adaptation procedure with different techniques is analyzed and the performance is validated by taking both low and high signal to noise ratios. The main idea behind this technique is the decomposition of CARMA model into a system and noise model [31] .

A fractional weight update part is added to the conventional constant modulas algorithm for blind equalization and output of the linear filter is passed through a nonlinear fractional update term. The algorithm is applied on flat and frequency selective channels. The algorithm is validated against the performance metric considered as the mean squared error for a quadratic shift keying [32].

The strength of fractional signal processing is also explored by developing the identification of Input nonlinear Box Jenkins system. The auxillary model for the fractional LMS algorithm with three values of fractional order has been developed in order to adapt the variables of INBJ system for various scenarios based on different step sizes and noise variances [33]. Another study in [34] also presents the fractional adaptive algorithm for the parameter estimation of Box Jenkin system. Also to validate the scheme, camparison is also made with volterra and kernel least mean square algorithm. Identification scheme based on fractional LMS algorithm is developed in [35] for the parameter estimation of Hammerstein Nonlinear Controlled Autoregressive system with exoge-

nious noise.

2.3 Kernel Methods, an overview

In this section, the summary is given about the theoretical framework on which the kernel methods are build. Discussion is also presented about some basic concepts about kernel functions, nonlinear mapping of the input data into the feature space and theory of reproducing kernel Hilbert space.

2.3.1 Kernels, A basic definition

A kernel is a continuous, symmetric function $K : X \times X \longrightarrow \Re$ operates on the data on the input space. A kernel is said to be positive definite if any set of input data points $\{x_i\}_{i=1}^{N} \in \mathbf{X}$, satisfies the following condition

$$\sum_{i,j=1}^{N} \beta_i \beta_j k(x_i, x_j) \geq 0, \forall \beta_i \in \mathbf{R}. \tag{2.13}$$

For a given set of N data points $[x_1, x_2, x_3, ...x_N]$, the $N \times N$ matrix K formed with elements $K_{i,j} = k(x_i, x(j))$ is called kernel or gram matrix

A kernel matrix is said to be positive definite matrix, if it is constructed using a positive definite kernel function. A matrix is said to be positive definite matrix, if it satisfies the following condition

$$\sum_{i,j=1}^{N} \beta_i \beta_j K_{i,j} \geq 0, \forall \beta_i, \beta_j \in \mathbf{R}. \tag{2.14}$$

but subject to the strictly positive kernel function. Types of kernel functions are discussed in section 2.3.8.

2.3.2 Reproducing kernel Hilbert space

A feature space associated with a positive definite kernel is an inner product in that feature space. Now let us consider a feature mapping from **X** into the space of functions **H**, for a given positive definite kernel k,

$$\begin{aligned} \Phi &: \mathbf{X} \rightarrow \mathbf{H} \\ &: x \rightarrow \varphi(x) \\ &: x \rightarrow k(x, .) \end{aligned} \tag{2.15}$$

The kernel function φ applied on the input vector x as $\varphi(x)$ assigns the value $k(x, x^{'})$ to input point $x^{'}$. In order to construct a feature space associated with a nonlinear kernel function φ, the image of φ must be turned into a vector space. This vector space is defined by taking linear combination as.

$$\begin{aligned} f(.) &= \sum_{i=1}^{m} \beta_i \varphi(x_i) \\ &= \sum_{i=1}^{m} \beta_i k((x_i), .) \end{aligned} \tag{2.16}$$

where x_i is chosen according to the length of the specified input data vector. β_i are the coefficients chosen arbitrary and $i = 1, 2, 3, ..., m$. The interesting property from the above kernel function satisfies

$$\langle k(x, .), f \rangle = f(x) \tag{2.17}$$

In particle the kernel k posesses the reproducing property as

$$\langle k(x,.), k(x^{'},.)\rangle = f(x) \tag{2.18}$$

where $k(x, x^{'})$ is the inner product in the associated feature space and is written as

$$k(\mathbf{x}, \mathbf{x}^{'}) = \langle \varphi(x), \varphi(x^{'})\rangle \tag{2.19}$$

2.3.3 Kernel Trick

The famous kernel trick states that we can transform any linear inner product based algorithm to an alternative algorithm by replacing the inner product with a nonlinear kernel. This inner product based kernel algorithm is same as that of original inner product based linear algorithm, but in the feature space. The beauty of this elegant and simple trick is that, the obtained solution is the nonlinear function of the input data, and it is achieved by performing a convex optimization problem implicitly in the feature space. The advantage of moving data to the higher dimensional feature space in the classification problems is discussed briefly by cover's theorem.

2.3.4 Cover's Theorem

When a complex pattern classification task is performed using RBF function [36, 37], the problem arises by transforming it into a high dimensional space, and again seperating it in the output layer. The justification is found in Cover's theorem on the separability of patterns and is stated as *A complex pattern classification problem in a high dimensional space is more likely to be linearly seperable than in a low dimensional space, provided that the space is not populated densely*.

2.3.5 Mercer's Theorem

The mercer's theorem [38] is stated as

the kernel $k(\mathbf{x}, \mathbf{x}^{'})$ *can be expanded as*

$$k(\mathbf{x}, \mathbf{x}^{'}) = \sum_{i=1}^{\infty} \lambda_i \varphi_i(x), \varphi_i(x^{'}) \tag{2.20}$$

with the positive coefficients $\lambda_i > 0$ for all i.

$\varphi_i(\mathbf{x})$ are called eigenfunction of the expansion, and λ_i are the eigenvalues. This is being the fact that if all the eigenvalues are positive means that the kernel $k(\mathbf{x}, \mathbf{x}^{'})$ is positive definite. this means that a complex problem can be solved efficiently for the weight vector $\mathbf{w}$.

2.3.6 Kernel Properties

Kernel function that satisfies the mercer's theorem are said to be positive definite and their kernel matrices have no non-negative eigen values. The usage of the positive definite kernel ensures that the solution of the optimization problem will be unique and convex. However some kernel functions, which are not strictly positive definite also perform very well in practice. As an example sigmoid kernel is not positive semi definite, but is used in variety of applications. Sometimes the kernels which are not conditionally positive definite can possibly outperforms most classical and well known positive definite kernels in some applications. Kernels can also be classified as Anisotropic Stationary, Isotropic Stationary, Locally stationary, Separate Non stationary, scale dependent or scale invariant.

2.3.7 Choosing the appropriate kernel

Choosing the right kernel depends on the problem at hand, and tuning of its parameter is become a tedious and tricky task. Automatic kernel selection is also possible and is discussed by Howley

and Mickeal G. Madden in [39]. The choice of a particular kernel can be very easy and a straightforward task depending upon the kind of information we want to extract about the data. It also depends on what we are trying to model.

2.3.8 Common kernel Functions

In this section list of some commonly used kernel functions is presented briefly.

Linear kernel

A simplest kernel function is the linear kernel. It is written as the inner product plus a constant c as

$$k(\mathbf{x}, \mathbf{x}^{'}) = \mathbf{x}^{T}\mathbf{x}^{'} + c \tag{2.21}$$

By using this kernel function the kernel algorithms are often equivalent to their non kernel counterparts. As an example, kernel principle component analysis with linear kernel function is same as Principal Component Analysis.

Polynomial kernel

Polynomial kernel is typically known as non stationary kernel and is written as

$$k(\mathbf{x}, \mathbf{x}^{'}) = (\alpha\mathbf{x}^{T}\mathbf{x}^{'} + c)^{d} \tag{2.22}$$

where α, c and d are the slope, constant and polynomial degree respectively.

Gaussian kernel

The Gaussian kernel is an example of radial basis function and is written as

$$k(\mathbf{x}, \mathbf{x}^{'}) = exp(\frac{-\|\mathbf{x} - \mathbf{x}^{'}\|^2}{2\sigma^2}) \tag{2.23}$$

or it can be written as

$$k(\mathbf{x}, \mathbf{x}^{'}) = exp(\alpha\|\mathbf{x} - \mathbf{x}^{'}\|^2) \tag{2.24}$$

The adjustable parameter $\alpha = \frac{1}{2\sigma^2}$ plays a significant role in the overall performance of the kernel and σ^2 is the variance.

Exponential kernel

It is also a radial basis kernel function, but with only the square of the norm left out as

$$k(\mathbf{x}, \mathbf{x}^{'}) = exp(\frac{-\|\mathbf{x}^T - \mathbf{x}^{'}\|}{2\sigma^2}) \tag{2.25}$$

Laplacian kernel

In order to make the kernel function less sensitive for the change in alpha parameters, a laplacian kernel is introduced. It is defined as

$$k(\mathbf{x}, \mathbf{x}^{'}) = exp(\frac{-\|\mathbf{x} - \mathbf{x}^{'}\|}{\sigma}) \tag{2.26}$$

Hyperbolic Tangent kernel

This kernel function typically known as multi layer perceptron or sigmoid kernel. Bipolar sigmoid function is used as an activation function for artificial neurons

$$k(\mathbf{x}, \mathbf{x}^{'}) = \sum_{k=1}^{N} exp(-\alpha(x^k - x^{'k})^2)^d \tag{2.27}$$

Rational Quadric kernel

This kernel function is used as an alternative, where the Gaussian kernel becomes too expensive

$$k(\mathbf{x}, \mathbf{x}^{'}) = 1 - \frac{\|\mathbf{x} - \mathbf{x}^{'}\|^2}{\|\mathbf{x} - \mathbf{x}^{'}\|^2 + c} \tag{2.28}$$

Multiquadric kernel

This kernel is non positive definite kernel function.

$$k(\mathbf{x}, \mathbf{x}^{'}) = \sqrt{\|\mathbf{x} - \mathbf{x}^{'}\|^2 + c^2} \tag{2.29}$$

Inverse Multiquadric kernel

As with the Gaussian function, this kernel also results in a Gram matrix with full rank.

$$k(\mathbf{x}, \mathbf{x}^{'}) = \frac{1}{\sqrt{\|\mathbf{x} - \mathbf{x}^{'}\|^2 + c^2}} \tag{2.30}$$

Circular kernel

This function is an example of isotropic stationary kernel and is positive definite in $textbf R^2$.

$$k(\mathbf{x}, \mathbf{x}^{'}) = \frac{2}{\pi} \arccos(-\frac{\|\mathbf{x} - \mathbf{x}^{'}\|}{\sigma}) - \frac{2}{\pi} \frac{\|\mathbf{x} - \mathbf{x}^{'}\|}{\sigma} \sqrt{1 - (\frac{\|\mathbf{x} - \mathbf{x}^{'}\|}{\sigma})^2} \tag{2.31}$$

Spherical kernel

This kernel function is positive definite in $\mathbf{R}^3$

$$k(\mathbf{x}, \mathbf{x}^{'}) = 1 - \frac{3}{2} \frac{\|\mathbf{x} - \mathbf{x}^{'}\|}{\sigma} + \frac{1}{2} (\frac{\|\mathbf{x} - \mathbf{x}^{'}\|}{\sigma})^3, \ If \|\mathbf{x} - \mathbf{x}^{'}\| < \sigma, zero \ otherwise. \tag{2.32}$$

Power kernel

This kernel is scale invariant and typically known as triangular kernel and is conditionally positive definite

$$k(\mathbf{x}, \mathbf{x}^{'}) = -(\|\mathbf{x} - \mathbf{x}^{'}\|)^d \tag{2.33}$$

log kernel

This kernel is also conditionally positive definite

$$k(\mathbf{x}, \mathbf{x}^{'}) = -\log(\|\mathbf{x} - \mathbf{x}^{'}\|^d + 1) \tag{2.34}$$

Bessel kernel

This kernel is well in the theory of function spaces of fractional smoothness

$$k(\mathbf{x}, \mathbf{x}^{'}) = \frac{J_{\nu+1}(\sigma \|\mathbf{x} - \mathbf{x}^{'}\|)}{\|\mathbf{x} - \mathbf{x}^{'}\|^{-n(\nu+1)}} \tag{2.35}$$

J is the Bessel function of the first kind.

Sequence, cauchy, chi-square, time series, histogram, generalized histogram, wavelet are the other well known kernel function used in various applications.

2.4 Kernel algorithms

In this section, we provide some brief overview about the family of kernel signal processing algorithms. Some applications of kernel algorithms has also been discussed and analyzed as given in the previous literature.

2.4.1 Kernel Least Mean Square Algorithm

Kernel least mean square algorithm is developed by weifeng Liu et. all [40]. The main idea behind this development is to transform the input data into high dimensional feature space by using mercer's theorem and perform linear operations in the transformed space. Some of main discrete steps involve in the development of the algorithms are discussed below.

$$e(i) = d(i) - \mathbf{w}^T(i-1)\varphi(i) \tag{2.36}$$

Here $e(i)$ is the error, $d(i)$ is the desired response,$\mathbf{w}$ is the weight vector and $\varphi(i)$ is the transformed input. Through iterations the repeated application of the above weight-update equation yields

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta e(i)\varphi(i) \tag{2.37}$$

Where η is the step size parameter. If we take w(0)=0, then

$$\mathbf{w}(i) = \eta \sum_{j=1}^{n} e(j)\varphi(j) \tag{2.38}$$

The output of the KLMS filter is described as

$$y = \eta \sum_{j=1}^{i} e(j) < \varphi(j), \varphi(i) >= \eta \sum_{j=1}^{i} e(j) k(\mathbf{x}(j), \mathbf{x}(i)) \tag{2.39}$$

It is clear from the equation 2.39, the output of the KLMS algorithm is the sum of all past errors multiplied by the kernel evaluation of the previously received data. η is the step size parameter for the KLMS, k is the kernel function using Gaussian distribution.

Algorithm 1 Kernel least mean square algorithm

Initialization:

$$\mathbf{w}(0) = 0$$

Choose parameter η, kernel width and kernel k

$iterate\ for\ \ i\ = 1, 2, 3, ... :$

Compute the output

$$y(i) = \eta \sum_{j=1}^{i} e(j) k(\mathbf{x}(j), \mathbf{x}(i))$$

Compute the error

$$e(i) = d(i) - \mathbf{w}^T(i-1)\varphi(i)$$

Normalized Kernel Least Mean Square Algorithm

It is noted that Normalized Least Mean Square Algorithm (NLMS) exhibits better performance then LMS, by using this fact the normalized version of KLMS is also developed by weifeng Liu et.al [41] in 2010 and is defined as

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \frac{\eta}{\epsilon + \|\varphi(i)\|^2} e(i)\varphi(i) \tag{2.40}$$

And by using the definition, the norm of the feature space is defined as

$$\|\varphi(i)\|^2 = \langle \varphi(i), \varphi(i) = k(\mathbf{x}(i), \mathbf{x}(i)) \tag{2.41}$$

A Quantized kernel least mean square algorithm is developed by Badong Chen et. al [42] in 2012, to improve the performance of the kernel least mean square algorithm by introducing a simple vector quantization method. The quantization concept already been used in Linear adaptive filters such as LMS and RLS. The basic idea behind this approach is to reduce the dynamic range requirement and numerical complexity of the adaptive algorithms, and also to reduce the hardware complexity in order to meet the demand of real time high speed implementation. The weight vector and error equation of QKLMS is defined as

$$\begin{aligned} w(0) &= 0 \\ e(i) &= d(i) - w(i-1)^T \varphi(i) \qquad (2.42) \\ w(i) &= w(i-1) + \eta e(i) \mathbf{Q}[\varphi(i)] \qquad (2.43) \end{aligned}$$

Where **Q** is the quantization operator in feature space. The QKLMS is applied on static function approximation and lorenz system prediction and the result are shown in terms of mean square error as the figure of merit by the authors.

In kernel signal processing algorithm, the parameters of the kernel employed is based on statistics of the input to the linear filters. A finite nonlinearity model order is required for practical implementations. The KLMS filter having Gaussian kernel has two design parameters, one is kernel bandwidth and the other is step size. The steady state and transient behavior of the KLMS algorithm is discussed by parreira et. al. [43] for nonlinear Gaussian inputs and a finite order nonlinearity model. A recursive expression is also established for the mean square error and the mean weight error vector. Kernel Least Mean Square Algorithm built from a weighted sum of kernel function multiplied with error vector, that evaluated at each incoming data samples. The computational complexity, memory requirement and the size of the filter increases with time. A new efficient methodology is used to constrain the growth of RBF network without the significant degradation in the performance of the filter. In this method, the elimination step is introduced to

test the Linear dependency of the feature vector corresponding to new input vector with all the previous feature vectors.The output of the KLMS filter is the weighted sum of the kernel function evaluated at each incoming data sample. With the increase in time, the size of the filter increases along with computation and memory requirement. In order to constrain the growth of RBF network, a method is introduced p.p. pokharel et. al. [44], that involves sequential elimination steps on the GRAM matrix.

A method is introduced by yazdi et. al for the solution of ordinary differential equation. The author proposes a method in which a trial solution is developed by keeping in view the structure of the ordinary differential equation. He introduced the term unsupervised KLMS algorithm, because the error vector is adjusted iteratively in an online fashion. without the training process [45]. Another scheme is developed by the same author, the only difference is that, the error vector is optimized via BFGS algorithm. The structure of this method is defined as Here the authors introduces the general form of the trial solution of the ordinary differential equation [46].

In order to prove the stability of the KLMS algorithm, the fundamental energy conversation relation has been developed by richard et. al. in the Reproducing kernel Hilbert space. The procedure is also established for the optimal step size that guarantees fastest convergence and an optimal kernel size for rapid initial convergence [47].

The theoretical value of the steady state excess mean square value is also presented and the work is solely based on Gaussian kernel function. A fixed budget quantized kernel least mean square algorithm is presented by zhao et. al [48] to deal with growing support inherently in online kernel methods. The main idea behind this method is to discard the center of RBF network with the smallest influence on the overall system, The significance measure can be update recursively at every step that is very suitable for online operations. The stability of the KLMS algorithm is checked by calculating the extreme eigenvalues of the kernel matrix, for each set of tuning parameters. A sufficient and necessary condition for the convergence of the KLMS algorithm has been clearly established and the discussion is also presented that examines the variation in the stability limit as

the function of kernel bandwidth, step size and the filter length [47].

In view of handling the complex signals in RKHS, Complex Gaussian KLMS algorithm is bieng established. An efficient and elegant computations of the gradient has been taken place by the extension of writinger's calculus in complex RKHS [49]. Wirtinger's calculus is used to obtain the gradient of the cost function of CGKLMS with different kernels are discussed by paul et. al [50].

Active noise control is a method to attenuate the noise signal that are in a low frequency range. The performance of the Linear Active noise control methods degrades because in a real time systems nonlinearity exists from noise source to connecting points through electrical and acoustic paths. KLMS algorithm is used for active noise control (ANC) in to improve the performance. Prediction of two dimensional hand trajectories from recording the surface of cortical to instantaneous positions of hands is conducted by gunduz et. al. using KLMS algorithm. Because the functional mapping between activity of cortical to behavior might be nonlinear [51, 52].

A state space model based on baysian filtering is developed by Weiming park. In this work the KLMS algorithm is extended to integer and binary valued observations and a forgetting factor is also introduced to improve its tracking capability. This scheme allows systematic development towards the extension of KLMS algorithm by modifying the underlying observation and state space models [53].

By using extended version of wirtinger's calculus, a suitable augmented complex KLMS algorithm is developed by bouboulis et. al. for applications involving non circular data. In order to provide a practical meaning between the primal and dual space, a novel class of complex valued KLMS is established, termed as independent complex KLMS algorithm. A general class of CKLMS algorithm is also developed in a widely linear sense that is suitable for both circular and non circular data. The KLMS filter is approximated to Bayesian filtering and then developed systematically an underlying observation and measurement models [54] [55].

The proportionate factor is introduced for the normalized KLMS algorithm to increase the tracking abilities and convergence speed of the algorithm [56]. In order to get a considerable performance

of the KLMS algorithm, a finite pretuned dictionary model must be established in addition with step size, kernel function and kernel bandwidth. The dictionary size has the considerable impact on the overall performance of the KLMS algorithm. The energy conversion relation is extended to RKHS in [54] , on the basis of the upper bound on the step size for MSE convergence along with finding the theoretical value of the steady state. Excess MSE and optimal kernel size for fast initial convergence is also established. To deal with complex signals in RKHS, a technique introduced in [55], called as complexification. Powerful tool of wirtinger's calculus is used for this purpose that has recently been popular in signal processing community. In order to eliminate predict the nonlinear characteristics of the network traffic, a KLMS based prediction method is proposed. An experiment is conduct by using the data from the network traffic database. The author uses traffic data of a week to predict next's days traffic data. The KLMS gives far better results and the convergence is also consistent in comparison with linear LMS.

The problem of selecting a particular kernel from a pool of predefined kernel function is highlighted. A Mixture kernel Least Mean square algorithm [57]with different kernels is proposed. Gatting method is used to choose the best kernel at each input regime. The authors also proves the accuracy of the Mixture KLMS algorithm from prediction of short term chaotic lorenz time series prediction.

Kernel Least mean square algorithm along with Hidden Markov Model(HMM) is used to propose a new feature extraction method that efficiently handles nonlinearities in signals by Hossein G. Hafarain [58]. KLMS algorithm is used to extract the features from the signal, whereas on the other hand HMM is used to model these extracted feature sequence and to differentiate it from other models.

2.4.2 Kernel Affine Projection Algorithm

In this section we provide a brief introduction of Kernel fractional affine projection algorithm(KFAPA), Its application found in the literature and some basic concepts and derivation of the algorithm.

The stochastic newton cost function for the KFAPA algorithm is developed as

$$J(w) = \frac{1}{2}\sum_{i=0}^{n}(d(i) - \mathbf{w}^T\varphi(i))^2 \tag{2.44}$$

By minimizing the above mentioned cost function with respect to $\mathbf{w}$ and by straight forward manipulation, the stochastic gradient descent becomes

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t\mathbf{\Phi}(i)[\mathbf{d}(i) - \mathbf{\Phi}^T\mathbf{w}(i-1)] \tag{2.45}$$

where $\mathbf{\Phi}(i)$ is a matrix and φ are the corresponding vectors

$$\mathbf{\Phi}(i) = \varphi(1), \varphi(2), \varphi(3), ...\varphi(i) \tag{2.46}$$

and

$$\mathbf{d}(i) = [d(1), d(2), d(3), ...d(i)] \tag{2.47}$$

And stochastic Newton method becomes

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t\mathbf{\Phi}(i)[\mathbf{\Phi}(i)^T\mathbf{\Phi}(i) + \epsilon\mathbf{I}]^{-1}[\mathbf{d}(i) - \mathbf{\Phi}^T\mathbf{w}(i-1)] \tag{2.48}$$

Notice that by using matrix identity

$$[\mathbf{\Phi}(i)\mathbf{\Phi}(i)^T + \epsilon\mathbf{I}]^{-1}\mathbf{\Phi}(i) = \mathbf{\Phi}(i)[\mathbf{\Phi}(i)^T\mathbf{\Phi}(i) + \epsilon\mathbf{I}]^{-1} \tag{2.49}$$

the corresponding weight update equation becomes

$$\mathbf{w}(i) = (1-\eta_t)\mathbf{w}(i-1) + \eta_t\mathbf{\Phi}(i)[\mathbf{\Phi}(i)^T\mathbf{\Phi}(i) + \lambda\mathbf{I}]^{-1}[\mathbf{d}(i) - \mathbf{\Phi}^T\mathbf{w}(i-1)] \tag{2.50}$$

This algorithm is applied on nonlinear chaotic time series prediction, adaptive noise cancellation and nonlinear channel equalization [40, 41]. An improvement to the KAPA is introduced in [59] by using dichotomous coordinate descent(DCD) iterations. The suitable DCD updates were examined for system identification and forward prediction examples. The considerable reduction in computational complexity is achieved by suing DCD-KAPA. Online classification problem for the KAPA is discussed by the adaptive projected subgradient method(APSM). This algorithm is considered to be as a Generalized Affine Projection Algorithm. Sparcification of the resulting series kernel series is obtained by using a closed ball constraint on classification norm. By embedding online memory limitaions and online tracking capabilities, an upper bound on the dimension of linear subspace is imposed. The imposition is known as projection mappings. The architecture is also validated by the nonlinear equalization problem. The siliding window KAPA algorithm is introduced in [60] for nonlinear acoustic echo cancellation. A kernel function is also proposed that is weighted sum of linear kernel along with a Gaussian kernel, that is somehow efficient in handling acoustic applications. A flexible mechanism is also introduced in order to find the weights in the kernel by imposing different forgetting factor mechanisms in the sliding window [61].

Robustness against non gaussian impulse response is achieved by introducing the kernalized version of Affine projection sign algorithm known as kernel affine projection sign algorithm. The combination of affine projection sign algorithm and kernel methods helps in improving the performance of the filter against impulse interference [62]. A variable step size is also introduced that is based on errors and inputs. To gain the advantage of low computational complexity. A specific type of kernel is introduced especially for echo cancellation. A useful regularization mechanism is introduced by finding the weights in RKHS using different forgetting mechanisms in sliding window. nonlinear distortion produced by the loudspeaker parallel and cascade sliding window kernel affine projection algoritm is developed [63].

2.4.3 Kernel Recursive Least Squares Algorithm

Another important kernel algorithm of the kernel family is the kernel recursive least squares(KRLS) algorithm [64, 41]. The brief introduction of the KRLS algorithm is defined here along with the logical steps that are carried out in the development of the algorithm.

The least squares cost function in RKHS is written as

$$min(w) = \sum_{j=1}^{i} |d(j) - \mathbf{w}^T \varphi(j)|^2 + \lambda \|\mathbf{w}\|^2 \tag{2.51}$$

After differentiating the above mentioned cost function with respect to weights w the relation is defined as

$$\mathbf{w} = [\lambda I + \Phi(i)\Phi(i)^T]^{-1}\Phi(i)\mathbf{d}(i) \tag{2.52}$$

By applying the following matrix inversion lemma it is easy to verify that

$$[\lambda I + \Phi(i)\Phi(i)^T]^{-1}\Phi(i) = \Phi(i)[\lambda I + \Phi(i)^T\Phi(i)]^{-1} \tag{2.53}$$

by substituting this result in the weight update equation yields

$$\mathbf{w}(i) = \Phi(i)[\lambda I + \Phi(i)\Phi(i)^T]^{-1}\mathbf{d}(i) \tag{2.54}$$

The weights are finally expressed as a linear combination of the input data as

$$\mathbf{w}(i) = \Phi(i)\mathbf{a}(i) \tag{2.55}$$

where $\mathbf{a}(i)$ are the coefficients and is defined as

$$\mathbf{a}(i) = [\lambda I + \Phi(i)\Phi(i)^T]^{-1}\mathbf{d}(i) \tag{2.56}$$

The final update equation of the coefficients is defined in the algorithm 1 as

Algorithm 2 Kernel recursive least squares algorithm

Initialization

$$\mathbf{Q}(1) = \lambda + k(\mathbf{x}(1), \mathbf{x}(1))^{-1}, \mathbf{a}(1) = \mathbf{Q}(1)d(1)$$

iterate for $i > 1$:

$$\mathbf{h}(i) = [k(\mathbf{x}(i), \mathbf{x}(1)), ..., k(\mathbf{x}(i), \mathbf{x}(i-1))]^T$$
$$\mathbf{z}(i) = \mathbf{Q}(i-1)\mathbf{h}(i)$$
$$r(i) = \lambda + k(\mathbf{x}(i), \mathbf{x}(i)) - \mathbf{z}(i)^T\mathbf{h}(i)$$
$$\mathbf{Q}(i) = \begin{bmatrix} \mathbf{Q}(i-1)r(i) + \mathbf{z}(i)\mathbf{z}(i)^T r(i)^{-1} & -\mathbf{z}(i)r(i)^{-1} \\ -\mathbf{z}(i)^T r(i)^{-1} & r(i)^{-1} \end{bmatrix}$$
$$e(i) = d(i) - \mathbf{h}(i)^T\mathbf{a}(i-1)$$
$$\mathbf{a}(i) = \begin{bmatrix} \mathbf{a}(i-1) - \mathbf{z}(i)r(i)^{-1}e(i) \\ r(i)^{-1}e(i) \end{bmatrix}$$

$$w(0) = 0$$
$$e(i) = d(i) - w(i-1)^T\varphi(i) \quad (2.57)$$
$$w(i) = w(i-1) + \eta e(i)\mathbf{Q}[\varphi(i)] \quad (2.58)$$

Sliding window kernel RLS is developed by Van Vaerenbergh in 2006 for the identification non-linear Hammerstein and weiner system. The adaptive equation for the Sw-KRLS is written as

$$\mathbf{w}(i) = \Phi(i)[\lambda I + \Phi(i)^T\Phi(i)]^{-1}\mathbf{d}(i) \quad (2.59)$$

Where the most recent $k \times k$ inversions is obtained easily by the following identity

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{pmatrix} \tag{2.60}$$

where $\mathbf{A} = \mathbf{\Phi}(i-1)^T\mathbf{\Phi}(i-1) + \lambda\mathbf{I}$, $\mathbf{B} = \mathbf{\Phi}(i-1)^T\varphi(i)$, $\mathbf{C} = \varphi(i)^T\mathbf{\Phi}(i-1)$ and $\mathbf{D} = \varphi(i)^T\varphi(i) + \lambda I$. The biggest challenge in the KRLS algorithm is their linear growing structure, with each new incoming sample, that also result in increasing memory requirement and computational complexity.A new KRLS algorithm is introduced [65] in order to track time varying and nonlinear relationships in data. The Bayesian theorem is used to derive the equations of KRLS. This framework is efficient to handle forgetting in a consistent way and also to perform tracking in a non stationary environment. This method is numerically stable, online, requires a fixed amount of memory and computations per time step, provides regularization in a natural manner and provides confidence interval along each time steps. A sliding window KRLS is another algorithm introduced [66] under the umbrella of kernel family. A combination of sliding window along with L2-norm regularization is introduced. The algorithm is also validated on a set of nonlinear problems including time varying channel identification.

A new enhancement in KRLS has been developed to update the solution of the algorithm. This is done by the addition of one new data point to the support and one already chosen data point is discarded, using a suitable and easy to evaluate pruning criterion. A label update procedure is also introduced that equips the proposed algorithm with tracking capability [67]. This algorithm focusses on the idea of associative memory biologically inspired by the cerebellum, typically found in the brains of the animals. This algorithm converges faster than typical CMAC algorithm. Fixed memory budget KRLS algorithm is introduced that is capable of tracking changes over time and recursive learning a nonlinear mapping. This method combines the strategy of growing and pruning, prune the least significant data point. Another scheme that is developed is known as label update

procedure [68]. For implementing low and fixed amount of computational complexity, a anew algorithm named as KRLS-DCD is developed [69]. This algorithm maintain a constant amount of computations by fixing the order of the filter and while maintaining the dictionary, fast rate of convergence is guaranteed.

A KRLS type neuron is introduced that improves the performance of the KRLS algorithm in terms of convergence, local minima, slow convergence, nonparametric and universal approximation [70]. A standard KRLS algorithm is derived in a baysian perspective in pruning and dictionary updating criteria. A KRLS algorithm and the Kalman filter is combined to predict the current signal. The state or observation model is constructed in the original state space and the hidden state is estimated using the kalman filter. The measurement model is learned by the KRLS algorithm in RKHS [71].

2.4.4 Extended kernel recursive least squares algorithm

The non linear state space model in reproducing kernel hilbert space is developed that results in the formulation of Extended kernel recursive least squares algorithm(EKRLS) [72, 41]. The nonlinear model is written as

$$\begin{aligned} \mathbf{x}(i+1) &= \alpha\mathbf{x}(i) + \mathbf{n}(i) \\ d(i) &= \varphi(i)^T\mathbf{x}(i) + v(i) \end{aligned} \tag{2.61}$$

This EKRLS algorithm is validated by a set of experiments including time series prediction and nonlinear regression problems [41].

A general framework is deloped by Wencui et al. for handling the problem of adaptive filtering of quaternion signals using the quaternion Hilbert space. The HR calculus is also introduced for reproducing kernel Hilbert space is also introduced for RKHS in liu of finding the gradient of the

cost function. Simple rules such as chain rule and product rules are also developed in the quaternion Hilbert space. Along with that the quaternion signal is developed for the KLMS filter [73].
By using the Brock, Dechert and Scheinkman(BDS) test, it is proved by Haghighat et al. that the variable bit rate is nonlinear in nature. Video coding consists of three types of frames namely intra(I frame), predictive (P frame) and bidirectional-predictive frame (B-frame). These frames are arranged in a periodic pattern called group of pictures. the distance between these frames are assumed to be unknown. KLMS algorithm is used to predict the variable bit [74].
Online kernel methods are based normally on a single layer neurons, the structure update method is simply used to decide the number of neurons in the single output layer. An output recurrent feedback loop is introduced by Song et al. in the single layered output layer. Along with that the weight convergence analysis is also provided using the Lyapunov stability theory A variable length kernel sparcification scheme with the stability proof of structure update error is also discussed [75].
In online kernel methods the number of samples are sufficiently large and the final solution is independent of the kernel size. The issue is how to speed up convergence to the neighbourhood of the optimum solution that leads towards smaller network size. This issue is resolved by treating the kernel size as an extra parameter for optimization. This new frame work is developed by Chen et al for the KLMS algorithm. In this algorithm both the weights and kernel size are updated sequentially in order to minimize the mean square error. This algorithm can be incorporated in the Quantized KLMS. This algorithm is validated on a set of experiments, including static function approximation and short term chaotic lorenz time series prediction.
Kernel width also plays an important role in the performance of online kernel algorithms. A small kernel width will leads towards locality of preserved data, while larger kernel width ensure the estimation function smooth. Some optimal methods for the selection of kernel width are developed in the literature that includes target kernel alignment, information theoretic and kernel polarization. However, a consideration is taken place by taking the kernel width as a free parameter. A mechanism is developed by Haijin et al. [76] that updates the kernel weights and kernel width

same at a time. Another scheme is also developed named as dead zone scheme to determine the learning rate in the presence of external disturbance. A cumulative coherence is also developed to select the dictionary online. this algorithm is validated on a set of experiments including Duffing forced- oscillation system, laser generated time series prediction and monthly sunspot time series prediction [77].

A generic function approximation in a Markov game setup having a worst case design methodologies for the nonlinear continuous state space system is developed by Shah et. al. The use of Support vector machine along with KRLS algorithm is introduced for value function approximation. The mean square accuracy, absolute error, absolute torque, low computational complexity and better convergence shows the effectiveness of the algorithm [78].

The kernelized version of Least mean square absolute third algorithm is developed by Zheo et al. to enhance the convergence speed of the KLMS algorithm. This algorithm is robust against noises with different probability density functions. Along with that, a variable step size is introduced using a lorentzian function. The algorithm is validated in terms of prediction of Mackey Glass Time series prediction with different noise probabilities including uniform, Rayleigh, Rectangular and Exponential [79].

In online kernel learning algorithms, the problem is the growing structure of the solution size with each iteration steps. Another problem is the model of the input data in an infinite dimensional RKHS. A new method is introduced by Bouboulis et al. to map the input data into a finite dimensional Euclidean space using the random features of kernel fourier transform. This resulting algorithm does not require any sparcification scheme to curb the growth of the dictionary size or resulting kernel network. The advantages achieved by developing this algorithm is that the solution space is remained fixed and not increases with the iteration steps. Along with that the algorithm is computationally more efficient than the previous version of KLMS and KRLS algorithms. This algorithm is validated by a set of experiments including the identification of a simple nonlinear model followed by a chaotic nonlinear model [80].

To improve the performance and convergence speed of the KLMS algorithm, a multiple delay feedback structure is developed by Wang et. al.. A regularized cost function is minimized using a gradient descent method resulting in a regularized KLMS with multiple delay feedback. Along with that a spherical radius criterion is developed for sparcification purposes. The algorithm is validated by a set of experiments including time series prediction and identification of a highly nonlinear model [81].

In adhoc networks, the input/output measurements obtained sequentially, one per time step is related by an unknown nonlinear function. To cope with this problem, a diffusion based nonlinear cost function is minimized. This algorithm is developed by Chouvardas et. al. and named as diffusion based KLMS algorithm. The theoretical properties of the algorithm is also studied that employs an excellent behavior and no regret bound under uncertain conditions [82].

A fully pipelined implementation of kernel normalized least mean square algorithm is developed by Fraser et. al. for the regression. Ensuring that the pipeline does not stall and independent training tasks are performed for weight optimization to fill L cycles of latency. As compared with the previous developed techniques for L parallel problems, no dependencies exist within the pipeline. This is achieved by developing a deep pipelined modules by a recursive part of the KNLMS. Memory optimization, pipelining, scheduling and mixed-precision processing are combined to achieve a 575$\times$ speed up for parameter optimization [83].

A framework for time adaptive, online, supervised multiregression method is developed in infinite dimension RKHS. Special cases are considered for a fairly large number of nonlinear multiregression model. A continuous, convex and non differentiable function is considered as a loss function that quantify the desired response and output of the system. A sub gradient of the robust loss function is required in an analytic form [84]. Kernel adaptive filters are used to develop a measurement de noising module for filtering out random data before state estimation [85]. This module suppress the noises along with random false or malicious data. Online kernel based learning, the

system is updated when each training sample is obtained, a higher computational speed is required. To overcome this problem, a sparcification method is introduced based on Hessian matrix of the loss function to examine the significance of the new training sample. The kernel ridge regression is introduced to derive the surface as well as atmospheric properties from the hyperspectral infrared sounding spectra. The temperature and humidity profiles are considered from atmospheric sounding interferometer data. The kernel ridge regression outer perform linear regression. Some discrepencies may occur in the presence of clouds and low emisitivities in the desert areas [86, 87]. A comparative study of six different kernel algorithms were made with three different benchmark datasets. The comparison is made on the quality of solution as a foundation of memory and time complexity [88]. An application of KRLS algorithm is presented for the visual servoing of robots. An image Jacobbian expression is derived for nonlinear mapping between target feature and robot joint angles [89]. Two new versions of online kernel regression is developed to update the model accurately. Also avoid the computation load associated with recalculating the whole process every time the new data points are available. The first approach is by sliding window, that maintains the size of the kernel matrix under study. The another developed approach is known as so called "warm" window. It again shrinks the kernel matrix at every entry of the new data [90] A kernel function selection and optimization based on cost function maximization is developed to handle the most challenging and open problem in kernel methods and statistical learning theory. The algorithm is also validated on the application of infrared dim and small target detection based on KRLS [91].

When dealing with a sample flow learning system, an incremental kernel MSE is developed [92] to acheive better sparcity when there is online learning. Two additional adaptive learning algorithm has also been developed named as Complete reduction MSE and Partial reduction MSE. A novel and efficient control strategy based on KLMS algorithm is proposed [93] for the realization to control a nonlinear aircraft system. A Gaussian kernel is used in KLMS algorithm and Lyapunuv stability theory is used to control an aircraft manuever having dynamic and nonlinear nature. This

architecture provide considerable security and accuracy. The technique is validated with an experiment using a nonlinear model of F-18 aircraft.

By the use of coherence criterion method, the reduction in average size of the dictionary is achieved and also residual error is reduced. In infromation security and network trafic management, the important tool is the network trafic prediction. This prediction is done with KLMS algorithm [94]. A new mercer kernel named as survival kernel is developed, that is parameter free and simple in calculation. This survival kernel used in particular KLMS algorithm [95].

A new sparcification method based on Hessian matrix of the specific loss function is applied to examine the new training sample and dictionary updation accordingly with the important measure. The correlation matrix of the KLRS is considered to be as the Hessian matirx [96].

The formulation of the KPCA is considered in the euclidean space. A novel KPCA algorithm is developed [97] that tracks the eigen vectors with kernel in nonlinear space. The transformation problem of the input data to higher dimensional space is acheived. A new alternative to the kernel trick is developed that maps the input data to a reduced dimensional kernel space. This is achieved by the eigenvalue decomposition of the kernel or gram matrix [98]. This alternative to the kernel trick is named as nonlinear projection trick.

The element of the nonlinear mapping is determined by the newly developed chebychev kernel. The proposed kernel is validated for both classification and regression tasks [99]. The improvement is made in multiple kernel learning by introducing a soft margin technique. A kernel stack variable is introduced to each quadratic constraint of MKL, this is connected to single SVM model using a base kernel [100].

Compressive sensing is the technique to recover the signal that are somehow sparse linearly from a smaller number of measurements than required. Many image and videos are modeled as the nonlinear manifold and described as the nonlinear function of some parameter. A new compress sensing technique based on kernel method is developed for applying manifold models in compressive sensing. It uses dictionary learning in feature space in order to build a healthy model for signal

manifold [101].

A kernel signal to noise ratio in the context of feature extraction to general signal processing and machine learning community is developed. The developed scheme maximizes the signal variance while minimizing the noise veriance in the feature space(RKHS). This scheme can efficiently work with non colour noises [102]. The modification in RBF kernel is developed named as Tuned RBF kernel to overcome the costly learning and classification complexities. With this newly developed kernel, an algorithm is introduced named as Tuned Radial Basis function-Pruned Probability density assiciation [103].

A kernel function is developed for time series of exponential decay process. Conventional Gaussian kernel is not appropriate for this class of data. A class of kernel function is developed to study the generalization ability in the sence of universality [104, 105]..

In kernel mean matching(KMM) algorithm, the existance of many parameters is still an issue to adjust. A novel method is developed that automatically tunes the parameter of the KMM. The approach that is used to estimate the parameters of KMM is Normalizaed Mean Square error between the estimated importance weight and the ratio of the estimated training and test sample densities. This method is very useful for KMM to handle different types of kernels along with applicability in real domains [106].

A novel kernel adaptive filtering algorithm with parallel hyperslab projection along affine subspaces is developed by takizawa, that specifically observed data efficiently. The developed algorithm projects the filter into multiple hyperslabs in parallel along the dictionary subspace. The algorithm is named as adaptive projected subgradient method(APSM) [107].

An ECG derived respiration algorithm based on kernel PCA is developed. KPCA is basically the kernalized version of PCA, where the nonlinearities in the data is taken into account with nonlinear mapping of the input data to the higher dimensional feature space. This improvement is also done by tuning the kernel bandwidth of the RBF Gaussian kernel [108].

The kernalized version of least mean mixed-norm algorithm is developed by miao et al. that is

helpful in the environment where the system measurement noise shows distribution having linear combination of short tails and long tails [109]. Kernel particle filter with improved information mutual feedback is also developed by chang et al. for joint tracking and classification for radar surveilance system. Doppler spread, delay spectrum and radar measurement is used to estimate the target state and class respectively [110, 111].

A new online kernel algorithm based on support vector regression(SVR) and Regularized Network(RN) is developed. In offline phase, the SVR algorithm is used to obtain a RKHS model with a few reduced number of parameters. Then in online phase, the RN algorithm is used to update the parameters obtained by the SVR technique [112].

An intelligent and effective scheme is developed by Luo et al. in the wireless sensor networks to support the data sensing and fusion of cyber physical system. This architecture works on the principle of prediction based fusion and data sensing in order to reduce the data transmission and maintaining the required number of sensors for coverage in wireless sensor networks. The guarantee of data confidentiality is also provided. This developed architecture is named as Grey Model KRLS (GM-KRLS). The grey model takes the responsibility of predicting the data of next period having a smaller number of data items, while KRLS algorithm approximates this predicted value to its true value with high accuracy. To complete the transmission system, a very successful Blowfish algorithm in the literature is used for data encoding and decoding during the process of transmission [113].

An improvement in the quantized kernel least mean square algorithm(QKLMS) and gradient descent method is used to update the coefficients of the filter. This method uses both the new training data and the prediction error for coefficient adjustment at the closest center in the dictionary. This method completely utilizes the hidden knowledge in the new training data and also achieves better accuracy [114].

A reconfigurable Parallel FPGA accelerator is designed by Ren et. al for Kernel affine projection algorithm. A new and novel sparcification mechanism is introduced to constrain the network

size. This accelerator allows multiple input data to be processed simultaneously. To accelerate the execution rate, shift registers are used for input data. coefficients and code book are updated by shifting the register constantly. Finally the KAPA based accelerator is implemented with eight data paths on the Xillinx virtex-7 FPGA board. It is observed that it is 404.47 time faster efficiency [115].

A new kernel based multi book tensor partial least squares framework is developed by Ming Hou et al. for the generalized nonlinear tensor regression. The basic idea behind this work is to incorporate multi book regression context in the kernel space or RKHS. A general and effective architecture is also introduced that integrates multibook and single tensor regression scenarios into one general model having both common and discriminative features. A further enhanced predictive power is achieved by combining kernel machines with joint tucker decomposition to address the nonlinear dependencies between tensor blocks and multiple blocks. The author validated the proposed algorithm to multimodal and multiview human motion model estimation problems appears in the field of computer vision [116].

In product designing time forecast, existing problems are heteroscedastic noise and small samples. To handle these problems, a new kernel based method involving Gaussian distribution weight introduced by Zhi-Gen Shang and Hong-Sen Yan. The main idea behind this study is to combine the kernel based regression and Gaussian margin machines to solve the design time forecasting problems. This is achieved by inheriting the Gaussian distribution over weight vector, having the most least information distribution that lie in its corresponding confidence interval. Simplified optimization problem of Gaussian distribution weights and its solution using the kernel regression is also given. The set of experiments are performed to validate the architecture including forecast of designing plastic injection molds, forecasting of data within slump test dataset and yatch Hydrodynamics dataset. These datasets are obtained from the University College of London (UCI) repository [117].

An identification scheme utilizing sparse weighted kernel partial least squares as local models and

online Gustafson-Kessel clustering algorithm for structure identification is developed by Shafieezadeh Abadeh and Ahmad Kalhor. This algorithm forms special elliptical clusters without any orientation that will leads towards more complex structured clusters than the spherical ones. This algorithm also able to determine the number of clusters over time and also reduce the redundancy of model, when similar clusters are merged. A new sparcification procedure is also developed that is based on instant prediction error. This algorithm is validated on a set of experiments including time varying nonlinear system. Two bench mark real world problems are considered including Mackey Glass time series and electrical load prediction problem [118].

A single delayed output feedback is used to develop a novel KLMS with single feedback by Ji Zhao et al.. The past information of the input data is used to accelerate the convergence speed of the algorithm significantly. As compared to other kernel adaptive filtering algorithm having multiple feedback, this algorithm is more compact in nature but similar to momentum least mean square algorithm. In order to reduce the growing network size, a convex combination sparcification scheme is also used [119].

A more general information theoretic criterion is applied to minimum error entropy in RKHS, to develop a kernel minimum error entropy by Badong Chen and Jose C. principe. The development of the algorithm is similar to stochastic information gradient algorithm in kernel space. The computational load of this algorithm is similar to KAPA. To restrain the network size, computational load and memory requirement, a quantized approach is used that leads towards the quantized kernel minimum error entropy. The energy conversation relation for the algorithm is also obtained and a sufficient condition for the convergence of mean square algorithm is proved. The algorithm is validated by an experiment of nonlinear system identification [120].

A novel data driven soft sensor method is developed for nonlinear problems of batch processes. This architecture is achieved by introducing a dual kernel partial least squares algorithm. Dual kernel matrices are established by projecting both output data and input data into two reduced kernel spaces. Then the data predicted in the kernel space is reversely projected to its original space using

online prediction. A very complicated series of biochemical reaction known as Escherichia Coli prediction having characteristics of time variability, nonlinearity and uncertainty is performed by kernel multiway partial least squares algorithm [121]. The other related work in the kernel methods are discussed in [122, 123, 118, 124, 125].

3 Fractional Algorithms with their applications to nonlinear systems

In this chapter, a modified fractional least mean square algorithm is introduced for nonlinear and nonstationary time series prediction. A novel method is introduced to adjust the step sizes and the order of the fractional derivative of MFLMS algorithm on the basis of minimizing the mean square error. The proposed algorithm is validated with the experiments performed against the prediction of stationary as well as nonstationary chaotic Mackey glass and short term lorenz time series.

3.0.5 Prediction problem setting

One of the most celebrated problems in signal processing is that of predicting a future value in nonlinear time series analysis [126]. Consider the time series $x(n), x(n-1), x(n-2), x(n-3), ...x(n-M)$, representing M+1 samples of such a process. In the process of prediction, samples $x(n-1), x(n-2), x(n-3), ...x(n-M)$ are used to predict the current value of $x(n)$. First the specified data is chosen for training upon which the filter taps are adjusted and use these weights to predict the next value of the time series. The taps are adjusted here in this research work according to the stochastic gradient algorithm (LMS), nonlinear feature mapping technique (KLMS) and fractional derivative technique (FLMS and MFLMS). The basic architecture of the predictor of order M will be given in Fig. 1.

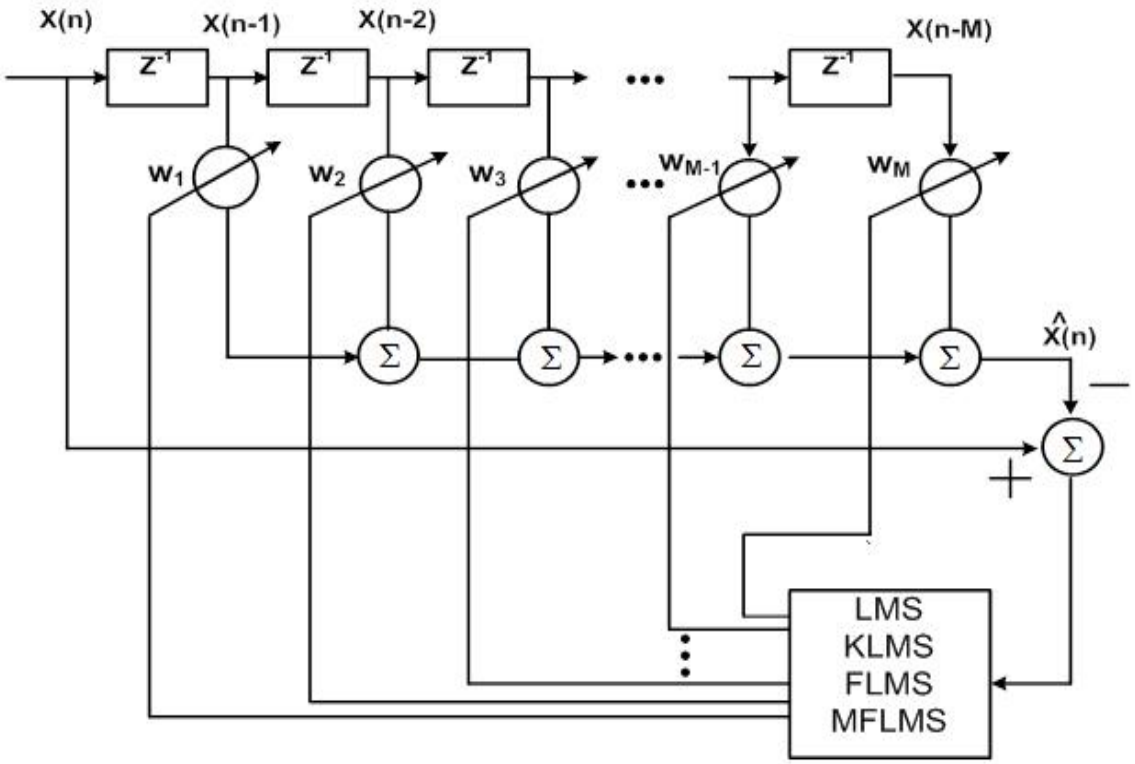

Figure 1: Basic architecture of the predictor

Least Mean Square Algorithm

Widow and Hopf [127] use stochastic gradient technique to formulate the least mean square algorithm. In order to minimize the empirical risk by using the example sequence $[\mathbf{x}(i), d(i)]$ as

$$J(w) = \frac{1}{2}\sum_{i=0}^{n}(d(i) - w^T\mathbf{x}(i))^2 \tag{3.1}$$

The gradient of the above mentioned cost function with respect to w is given as

$$\nabla_w J = -\sum_{i=1}^{n} x(i)(d(i) - w^T\mathbf{x}(i)) \tag{3.2}$$

Finally the weights are adjusted according to the method of steepest descent as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e(n)x(n) \tag{3.3}$$

It is customary to initialize the tap weight vector equal to zero.

Kernel Least Mean Square Algorithm

A poor performance can be expected when the difference or the mapping between **x** and **d** is highly nonlinear. A nonlinear mapping is introduced in [41] as $\varphi(\mathbf{x}(i))$, which gives a powerful model $\mathbf{w}^T\varphi(\mathbf{x}(i))$ than $\mathbf{w}^T\mathbf{x}$. So by using this model and finding **w** through stochastic gradient technique may prove an efficient mechanism towards nonlinear filtering as LMS ensures for linear problems. Using the example sequence$[\varphi(i), d(i)]$, evaluate the algorithm

$$e(i) = d(i) - \mathbf{w}(i-1)^T\varphi(i) \tag{3.4}$$

$$
\begin{aligned}
\mathbf{w}(i) &= \mathbf{w}(i-1) + \mu e(i)\varphi(i) \\
\mathbf{w}(i) &= \mathbf{w}(i-2) + \mu e(i-1)\varphi(i-1) + \mu e(i)\varphi(i) \\
\mathbf{w}(i) &= \mathbf{w}(i-3) + \mu e(i-2)\varphi(i-2) + \mu e(i-1)\varphi(i-1) + \mu e(i)\varphi(i) \\
&= \mathbf{w}(0) + \mu \sum_{j=1}^{n} e(j)\varphi(j) \qquad (3.5)
\end{aligned}
$$

If we take $\mathbf{w}(0) = 0$, then

$$\mathbf{w}(i) = \mu \sum_{j=1}^{n} e(j)\varphi(j) \qquad (3.6)$$

The repeated application of the weight update equation yields

$$y = \mu \sum_{j=1}^{n} e(j)k(\mathbf{x}(j), \mathbf{x}(i)) \qquad (3.7)$$

There is an inability to work directly with the filter weights in the feature space (RKHS). So we take a difference between the output of the network and desired signal as coefficient of the filter [41]. They are adjusted instead of weight vector for KLMS in order to predict the next value of the time series.

3.1 Fractional Least Mean Square Algorithm

A nonlinear and fractional adaptive algorithm has been introduced [128] for identification of the autoregressive (AR) systems. Since its formulation, it has been immensely utilized in various fields effectively, such as, echo cancellation, dual channel speech enhancement [129] and performance analysis of the Bessel beamformer [130]. Before discussing the algorithm, we hereby present some basic concepts about the fractional derivative. The idea stems from the Riemann Liouville

fractional integral and derivative method. The definition of the method is given as

$$I^v f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\nu)^{\nu-1} f(\tau) d\tau \tag{3.8}$$

where I^ν is the fractional integral of order ν. Accordingly the fractional derivative is given as

$$\begin{aligned} (D^\nu f)(t) &= (\frac{d}{dt})^n (I^{n-\nu} f)(t) \\ (D^\nu f)(t) &= \frac{1}{\Gamma(\alpha)} (\frac{d}{dt})^n \int_0^t (t-\tau)^{n-\nu-1} f(\tau) d\tau \end{aligned} \tag{3.9}$$

D^ν is defined as the fractional derivative and n is an integer. In order to add little more detail we present Reimann Lioville fractional derivative applied to power function given as follows. The ν derivative of $f(t) = (t-a)^\alpha$ is,

$$D^\nu (t-a)^\alpha = \frac{\Gamma\ (1+\alpha)}{\Gamma(1+\alpha+\nu)} (t-a)^{\alpha-\nu} \tag{3.10}$$

where a and α are real constants.

3.1.1 Derivation of FLMS Algorithm

First start with the standard LMS FIR filter [126, 131]. The error is minimized according to the cost function given as

$$J(n) = \frac{1}{2} E[|e(n)|^2] \tag{3.11}$$

The adaptive weight mechanism of LMS filter having only the first derivative is given as

$$w_k(n+1) = w_k(n) - \mu(\frac{\partial J(n)}{\partial w_k}) \tag{3.12}$$

We use fractional derivative in addition to the first derivative to Eq. 12.

$$w_k(n+1) = w_k(n) - \mu(\frac{\partial J(n)}{\partial w_k}) - \mu_2(\frac{\partial}{\partial w_k})^\nu J(n) \tag{3.13}$$

Equating 13, we get

$$\begin{aligned} ((\frac{\partial J(n)}{\partial w_k}) &= -e(n)x(n-k) && (3.14)\\ (\frac{\partial}{\partial w_k})^\nu J(n) &= -e(n)x(n-k)D^\nu w_k(n) \\ (\frac{\partial}{\partial w_k})^\nu J(n) &= -e(n)x(n-k)[\frac{1}{\Gamma(2-\nu)} w_k^{1-\nu}(n)] && (3.15) \end{aligned}$$

By substituting 3.14 and 3.15 into 3.13, we have a method for the weight adaptation of k^{th} element for FLMS as

$$w_k(n) = w_k(n-1) + \mu_1 e(n)x(n-k) + \mu_2 e(n)x(n-k)\frac{w_k^{1-\nu}(n)}{\Gamma(2-\nu)} \tag{3.16}$$

where ν is a real number between 0 and 1. To pace up the computation for equations of nonlinear nature in cyclic single step iteration, assume that $w_k^{1-\nu}(n) \cong w_k^{1-\nu}(n-1)$, then 3.16 becomes

$$w_k(n) = w_k(n-1) + \mu_1 e(n)x(n-k) + \mu_2 e(n)x(n-k)\frac{w_k^{1-\nu}(n-1)}{\Gamma\ (2-\nu)} \tag{3.17}$$

3.1.2 Proposed Modified FLMS Algorithm

Here we introduced a mechanism to avoid the computation of Gamma function, probably, one of the most fundamental function of the fractional calculus. By absorbing this function in μ_f and introducing the adjustable gain parameter β, reformulate 3.17 as follows

$$w_k(n) = w_k(n-1) + \beta\mu e(n)x(n-k) + (1-\beta)\mu_f e(n)x(n-k)w_k^{1-\nu}(n) \tag{3.18}$$

where $0 < \beta < 1$, is adjusted manually. μ and μ_f are the step size fractional parameters, respectively, with their values adjusted between 0 and 1. We do make sure that the algorithm does not diverge. β has to be adjusted for optimal results. Obviously if β is taken as 0.9, we are giving dominance to LMS. But if $\beta < 0.5$, we are giving preference to FLMS. Furthermore it is also possible that β may be adapted accordingly with the number of iterations.

3.2 Simulation Results

In this section we present some simulation results of time series prediction using LMS, KLMS, FLMS and MFLMS predictors. To evaluate the performance of these filters, Chaotic Mackey-Glass time series is generated by the following delay differential equation as

$$\frac{dx(t)}{dt} = -bx(t) + \frac{ax(t-\tau)}{1 + x^{10}(t-\tau)} \tag{3.19}$$

With b=0.1,a=0.2 and $\tau = 20$, initial conditions $x(t-\tau) = 0$ for $0 \leq t \leq \tau$. So we have a series x(k) for k=1,2,3,.....,3000, obtained by Eq. 3.19. It is achieved by sampling the continuous curve x(t) with the interval of 1 second. The time embedding length or the order of the filter M is 10 for all the experiments, and a segment of 700 samples is used for training, while another 200 is used for testing. Different cases are discussed here for the analysis of the Mackey-Glass time series prediction. The time series generated is shown in fig. 2.

3.2.1 Case I: Simulation for Stationary Series

The performance of the proposed MFLMS algorithm is tested on the stationary series. Data samples of points 900-1600 are used for the training of the filters and sample points 1600-1801 are used as the test or validation set. The weights are adjusted in 50 epochs for this experiments. These weights are then used to validate the test data. The parameters are adjusted as LMS ($\mu = 0.15$),

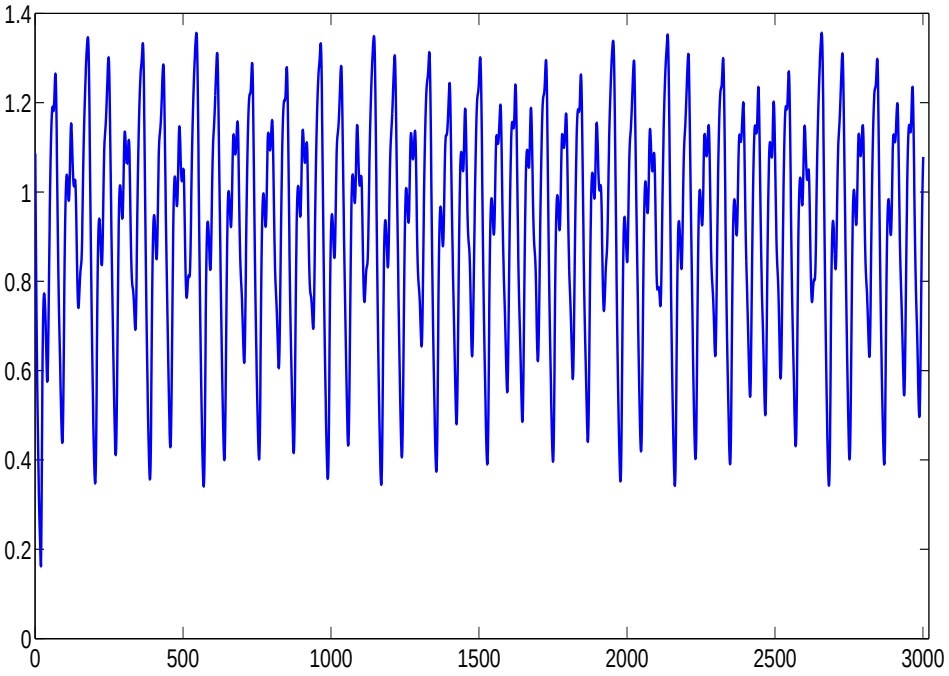

Figure 2: Mackey-Glass time Series

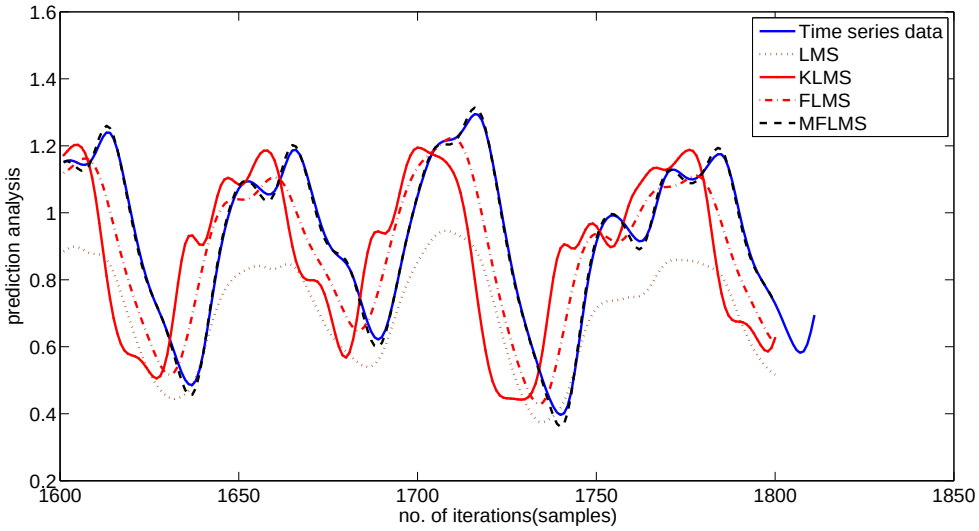

Figure 3: Prediction analysis of Stationary Series without noise

KLMS ($\eta = 0.1$ and Gaussian kernel with a width of 0.1), FLMS ($\mu = 0.019, \mu_f = 0.03$ and $\nu = 0.0972$) and MFLMS ($\mu = 0.08, \mu_f = 0.03, \nu = 0.0972$ and $\beta = 0.48$) for achieving the most optimal results. The results in Fig. 3 show the predictive performance of LMS, KLMS, FLMS, and proposed MFLMS. The superiority of the proposed method is clearly observed and to validate this, the learning curves of these performances are compared in terms of the mean square error displayed in Fig. 4.

3.2.2 Case II: Simulation for Stationary Series with different noise levels

Experiments are performed with a segment of 500 samples used for training and another 200 for testing. All the stationary time series data is corrupted by Gaussian noise with zero mean and different noise variances to further validate the MFLMS algorithm. Throughout this set of simulation the values of μ, μ_f, ν andβ are used same as in case I. Gaussian kernel with a width of 0.1 is used for KLMS predictor. 100 independent Monte Carlo simulations are run with different noise variances σ_n^2 to further validate the MFLMS applicability. The results are summarized in

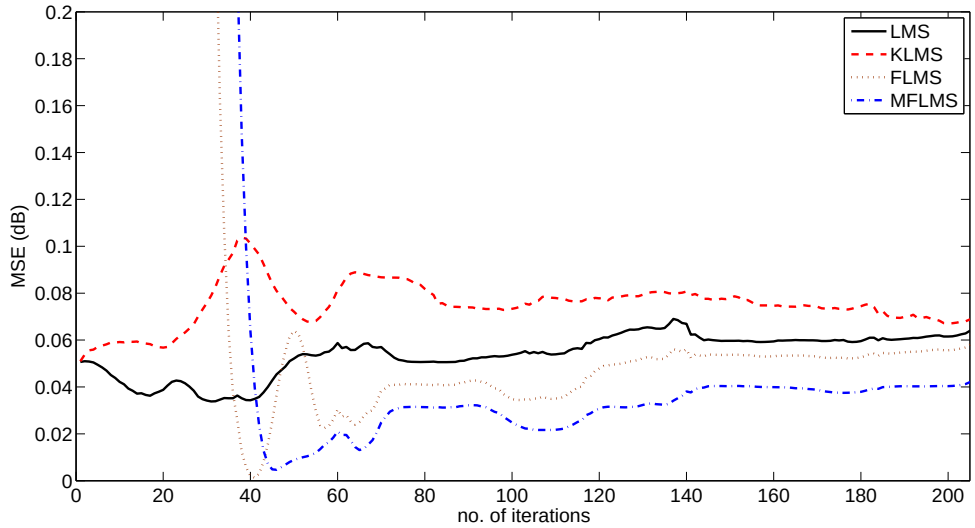

Figure 4: Learning curves of stationary series for LMS, KLMS, FLMS and MFLMS

table 1, which show that MFLMS is the best performer, specifically in terms of low noise variances. Results are displayed using the form mean±standard deviation. As it is easily observed in Table 1, the MFLMS algorithm performs effectively and consistently on both the training and test data.

Table 1: Performance Comparison with Different Noise Levels

Algorithm	**LMS**	**KLMS**	**FLMS**	**MFLMS**
Traning MSE($\sigma = 0.005$)	0.00250±3.85e-005	0.00454±9.26e-005	0.001827±2.93e-005	0.00156±1.28e-005
$TestingMSE(\sigma = 0.005)$	0.00243±0.4.219e-005	0.04927±0.0001069	0.0017738±3.28e-005.	0.001538±1.927e-005
$TraningMSE(\sigma = 0.02)$	0.00303±0.000149	0.005175±0.000358	0.002399±0.000121	0.0021152±6.50e-005
$TestingMSE(\sigma = 0.02)$	0.002970±0.00170	0.005556±0.0004324	0.002356±0.000131	0.0020892±9.49e-005
$TraningMSE(\sigma = 0.04)$	0.004640±0.000297	0.00680±0.000716	0.004117±0.000251	0.0038219±0.0001680
$TestingMSE(\sigma = 0.04)$	0.0046379±0.0003128	0.007162±0.0007767	0.004117±0.000282	0.003822±0.0002391
$TraningMSE(\sigma = 0.1)$	0.015863±0.0009678	0.01832±0.0015963	0.015628±0.0009414	0.015301±0.00086739
$TestingMSE(\sigma = 0.1)$	0.016166±0.001296	0.019066±0.0024241	0.015929±0.0012814	0.0155606±0.0013143
$TraningMSE(\sigma = 0.5)$	0.42356±0.1011	0.20001±0.010242	0.42533±0.1115	0.3209±0.030332
$TestingMSE(\sigma = 0.5)$	0.41752±0.12074	0.29218±0.028178	0.41918±0.12231	0.32013±0.047689

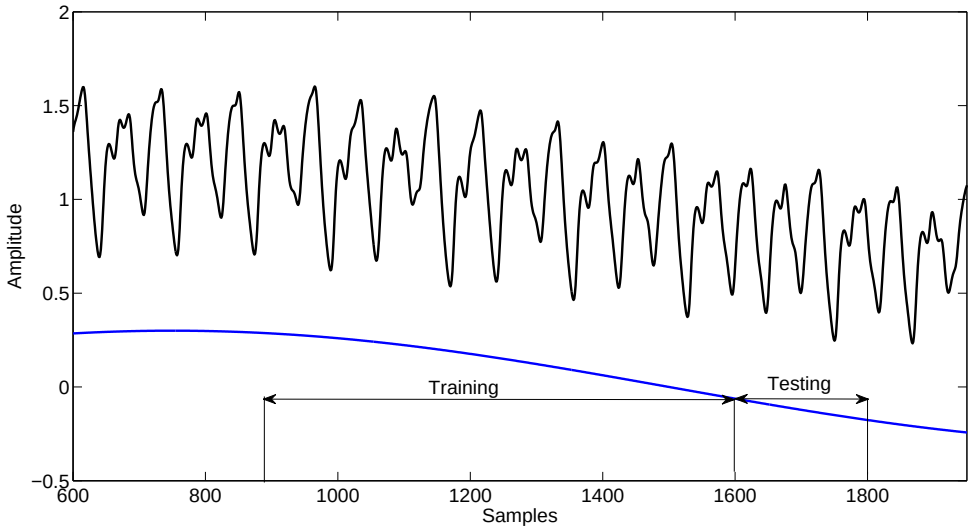

Figure 5: Modified Time Series

3.2.3 Case III: Simulation for Nonstationary Series

To examine the behavior of predictors for nonstationary series, we used a modified time series with sinusoid [132]. This new series is generated by adding a sinusoid ($y = 0.3\sin(2\pi ft), f = 1/3000$ and t is a continuous curve between $1 : 3000$) with an amplitude 0.3 and a frequency of 3000 samples per seconds to 3.24. The modified time series is displayed in Fig. 5, with clear indication of training and testing samples. Training data is taken from the samples 900-1600 as shown in Fig. 5, testing is done on the samples 1600-1800. After adjusting the best possible values of the filter parameters, the predictors are then trained with the aim to efficiently handle the time varying mean of the test data. The filter parameters are chosen as LMS ($\mu = 0.04$), KLMS ($\eta = 0.2$ and Gaussian kernel with a width of 0.1), FLMS ($\mu = 0.4, \mu_f = 0.03$ and $\nu = 0.0972$ and MFLMS ($\mu = 0.01, \mu_f = 0.03, \nu = 0.0972$ and $\beta = 0.35$ accordingly. The results displayed in Fig. 6 show the predictive performance of the algorithms. The authentication towards performance is observed in Fig. 7, which displays the learning curves of the mean square error performance comparison for LMS, KLMS, FLMS, and MFLMS.

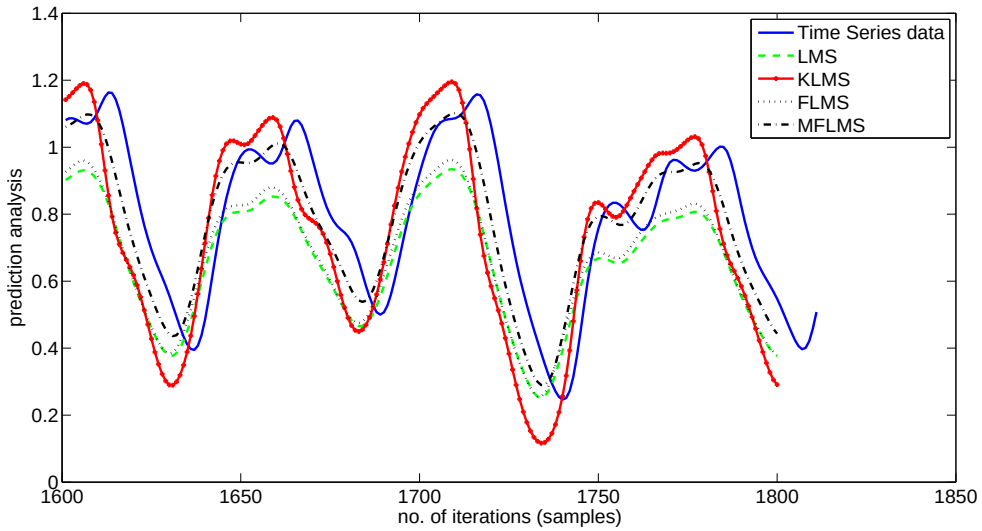

Figure 6: Prediction Analysis of modified Time Series

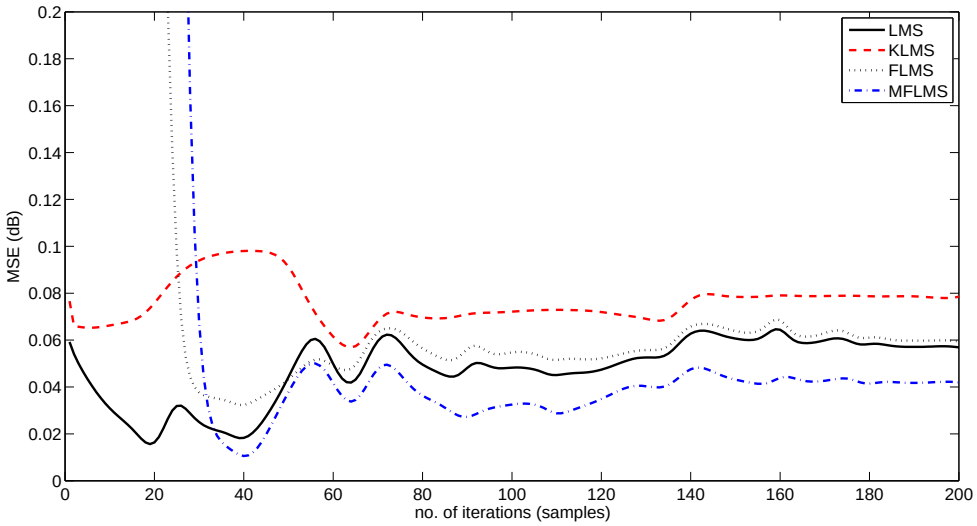

Figure 7: Learning curves of Nonstationary Series

Table 2: Performance Comparison of Modified Series with Different Noise Levels

Algorithm	LMS	KLMS	FLMS	MFLMS
$TraningMSE(\sigma = 0.005)$	0.0075878±0.00018345	0.0042825±6.8525e-005	0.0095947±0.0002324	0.0096714±0.000123
$TestingMSE(\sigma = 0.005)$	0.003413±6.7908e-005	0.039595±0.00022386	0.004302±8.8553e-005.	0.004523±5.32e-005
$TraningMSE(\sigma = 0.02)$	0.0082071±0.00078202	0.0048971±0.00028805	0.01018±0.0010086	0.010158±0.00055381
$TestingMSE(\sigma = 0.02)$	0.0039908±0.00029233	0.040221±0.0010017	0.0048459±0.000381	0.0050118±0.0002199
$TraningMSE(\sigma = 0.04)$	0.010862±0.0016886	0.0066764±0.00070018	0.012715±0.0020468	0.012111±0.0010296
$TestingMSE(\sigma = 0.04)$	0.0060947±0.0006786	0.042194±0.0023461	0.0068924±0.000811	0.0067999±0.00049113
$TraningMSE(\sigma = 0.1)$	0.027542±0.0059527	0.017685±0.0017916	0.029103±0.0065121	0.026047±0.0033939
$TestingMSE(\sigma = 0.1)$	0.019771±0.0025419	0.05688±0.005641	0.020757±0.002782	0.019248±0.0017778
$TraningMSE(\sigma = 0.5)$	0.4559±0.11013	0.21233±0.015882	0.45755±0.11098	0.37931±0.058416
$TestingMSE(\sigma = 0.5)$	0.40591±0.082598	0.34762±0.03449	0.40716±0.083371	0.34391±0.049129

3.2.4 Case IV: Simulation for Nonstationary Series with different noise levels

In table II, we summarize the algorithm behavior towards nonstationary series prediction, setting different noise levels parameters and testing each case on 100 Monte Carlo simulations. The performance of the proposed MFLMS attains the achievable performance. In this experiment, all the filter parameters are alloted as given in case III.The results are summarized in table II after 100 Monte Carlo simulations using the form mean±standard deviation. The results are displayed with different noise variances σ_n^2 to further validate the MFLMS applicability towards nonstationary series prediction.

3.3 Adaptive Step Size Modified Fractional Least Mean Square Algorithm

A mechanism is introduced here in this section of the dissertation to sidestep the computation of Gamma function, probably, one of the most major function of the fractional calculus. By gripping this function in the term μ_f and presenting the adjustable gain parameter β, reformulate 3.17 and

is written as

$$w_k(n) = w_k(n-1) + \beta\mu e(n)x(n-k) + (1-\beta)\mu_f e(n)x(n-k)w_k^{1-f}(n) \tag{3.20}$$

where $0 < \beta < 1$, to be adjusted manually. We do make sure that the algorithm does not diverge. β has to be adjusted for optimal results. Obviously if β is taken as 0.9, we are giving dominance to LMS. But if $\beta < 0.5$, we are giving preference to FLMS.

3.3.1 Proposed Adaptive Step Size Modified Fractional Least Mean Square Algorithm

The variable step size algorithm is introduced here to eliminate the "guesswork" exists in the selection of step size parameters. In order to speed up the performance and convergence of modified FLMS algorithm, the step size parameters μ, μ_f and f is made adaptive by using the gradient based approach. Let us consider the goal of reducing the squared estimation error at each instant of time by adapting these parameters.

Adaptation of parameter μ

. First we consider the goal to adapt the step size μ in (3.20) in order to reduce the squared estimation error at each instant of time as

$$\mu(n+1) = \mu(n) - \frac{\alpha}{2}\widehat{\nabla}_\mu(n) = \mu(n) - \frac{\alpha}{2}[\frac{\partial e(n)}{\partial \mu(n)}e^*(n) + \frac{\partial e^*(n)}{\partial \mu(n)}e(n)] \tag{3.21}$$

Here we consider the real case only and α is a small positive constant.

$$\frac{\partial e(n)}{\partial \mu(n)} = \frac{\partial}{\partial \mu(n)}[d(n) - \mathbf{w}^T(n)\mathbf{x}(n)] = -\frac{\partial \mathbf{w}^T(n)}{\partial \mu(n)}\mathbf{x}(n) = -\mathbf{\Psi}^T(n)\mathbf{x}(n) \tag{3.22}$$

putting the value of (23) in (22) we have

$$\mu(n+1) = \mu(n) - \frac{\alpha}{2}[-2e(n)\frac{\partial e(n)}{\partial \mu(n)}] = \mu(n) + \alpha e(n)\mathbf{\Psi}^T(n)\mathbf{x}(n). \tag{3.23}$$

Now we formulate the vector $\mathbf{\Psi}(n)$, denotes the weight vector with respect to step size parameter μ according to (3.20) as

$$\begin{aligned}\frac{\partial w_k(n+1)}{\partial \mu(n)} &= \frac{\partial w_k(n)}{\partial \mu(n)} + \beta e(n)x(n-k) + \beta\mu(n)x(n-k)\frac{\partial e(n)}{\partial \mu(n)} \\ &+ (1-\beta)\mu_f(n)\frac{\partial e(n)}{\partial \mu(n)}x(n-k)w_k^{1-f}(n) \\ &+ (1-\beta)\mu_f(n)e(n)x(n-k)\frac{\partial w_k^{1-f}(n)}{\partial \mu(n)}\end{aligned} \tag{3.24}$$

And then

$$\begin{aligned}\Psi_k(n+1) &= \Psi_k(n) + \beta e(n)x(n-k) + \beta\mu(n)x(n-k)(\mathbf{\Psi}^T(n)\mathbf{x}(n)) \\ &+ (1-\beta)\mu_f(n)(\mathbf{\Psi}^T(n)\mathbf{x}(n))x(n-k)w_k^{1-f}(n) \\ &+ (1-\beta)\mu_f(n)e(n)x(n-k)(1-f)w_k^{-f}(n)\Psi_k(n)\end{aligned} \tag{3.25}$$

In order to find an estimate of the particular value of the parameter μ, that minimizes the cost function (3.36). we follow the process of adaptation in (3.20) to (3.25)

Adaptation of parameter μ_f

. Now we consider the goal to adapt the fractional step size parameter μ_f. The procedure is as follows.

$$\mu_f(n+1) = \mu_f(n) - \frac{\alpha_f}{2}\widehat{\nabla}_{\mu_f}(n) = \mu_f(n) - \frac{\alpha_f}{2}[-2e(n)\frac{\partial e(n)}{\partial \mu_f(n)}] \tag{3.26}$$

Where α_f is a small positive constant.

$$\frac{\partial e(n)}{\partial \mu_f(n)} = -\frac{\partial \mathbf{w}^T(n)}{\partial \mu_f(n)}\mathbf{x}(n) = -\mathbf{\Psi}_f^T(n)\mathbf{x}(n) \tag{3.27}$$

And

$$\mu_f(n+1) = \mu_f(n) + \alpha_f e(n)(\mathbf{\Psi}_f^T(n)\mathbf{x}(n)) \tag{3.28}$$

Formulate $\mathbf{\Psi}(n)$ according to (3.20) as

$$\begin{aligned}\frac{\partial w_k(n+1)}{\partial \mu_f(n)} &= \frac{\partial w_k(n)}{\partial \mu_f(n)} + \beta\mu(n)\frac{\partial e(n)}{\partial \mu_f(n)}x(n-k) + (1-\beta)e(n)x(n-k)w_k^{1-f}(n) \\ &+ (1-\beta)\mu_f(n)\frac{\partial e(n)}{\partial \mu(n)}x(n-k)w_k^{1-f}(n) \\ &+ (1-\beta)\mu_f(n)e(n)x(n-k)\frac{\partial w_k^{1-f}(n)}{\partial \mu_f(n)}\end{aligned} \tag{3.29}$$

$$\begin{aligned}\Psi_{k,f}(n+1) &= \Psi_{k,f}(n) + \beta\mu(n)(\mathbf{\Psi}^T(n)\mathbf{x}(n)) + (1-\beta)e(n)x(n-k)w_k^{1-f}(n) \\ &- (1-\beta)\mu_f(n)(\mathbf{\Psi}^T(n)\mathbf{x}(n))x(n-k)w_k^{1-f}(n) \\ &+ (1-\beta)\mu_f(n)e(n)x(n-k)(1-f)w_k^{-f}(n)\Psi_{k,f}(n)\end{aligned} \tag{3.30}$$

Adaptation of parameter f (the order of fractional derivative)

. The proposed method for adapting the order of fractional derivative by using the gradient descent approach is given as

$$f(n+1) = f(n) - \frac{\alpha_{fr}}{2}\widehat{\nabla}_f(n) = f(n) + \alpha_{fr}e(n)\frac{\partial e(n)}{\partial f(n)} \tag{3.31}$$

Where α_{fr} is a small positive constant

$$\frac{\partial e(n)}{\partial f(n)} = -\frac{\partial \mathbf{w}^T(n)}{\partial f(n)}\mathbf{x}(n) = -\mathbf{\Phi}_f^T(n)\mathbf{x}(n) \tag{3.32}$$

And now formulate the adaptation rule for f as

$$f(n+1) = f(n) + \alpha_{fr} e(n)(\mathbf{\Phi}_f^T(n)\mathbf{x}(n)) \tag{3.33}$$

Differentiating (3.20) with respect to $f(n)$, we get

$$\begin{aligned} \Phi_{k,f}(n+1) &= \Phi_{k,f}(n) - \beta\mu(n)(\mathbf{\Phi}^T(n)\mathbf{x}(n)) + (1-\beta)e(n)x(n-k)w_k^{1-f}(n) \\ &- (1-\beta)\mu_f(n)(\mathbf{\Phi}^T(n)\mathbf{x}(n))x(n-k)w_k^{1-f}(n) \\ &+ (1-\beta)\mu_f(n)e(n)x(n-k)(-w_k^{1-f}(n)\ln w_k(n)) \end{aligned} \tag{3.34}$$

The order of the fractional derivative is adjusted adaptively through the method described in (3.31) to (3.34)

3.4 Prediction problem setting

One of the most celebrated problems in signal processing is that of predicting a future value in nonlinear time series analysis. Prediction involves the method of taking advantage of time series history to estimate the future values in the series. A time series can be defined as a sequence of vectors or scalers depending upon time.

Let us consider a time series $T.S. = \{x(n_0), x(n_1), x(n_2), ..., x(n_{k-1}), x(n_k), x(n_{k+1}), x(n_{k+2}), ...\}$

To predict the future value at a certain time, a tapped delay line is created through a process called embedding. The output is a matrix containing a tapped delay line shifted by one time sample in its

each column and will be written as

$$\begin{pmatrix} x(n_k) & x(n_{k+1}) & ... & x(n_{k+N-1}) \\ x(n_{k-1}) & x(n_k) & ... & x(n_{k+N-2}) \\ x(n_{k-2}) & x(n_{k-1}) & ... & x(n_{k+N-3}) \\ ... & ... & ... & ... \\ x(n_{k-M-1}) & x(n_{k-M}) & ... & x(n_{k+N-M}) \end{pmatrix} \tag{3.35}$$

The number of columns represents the input patterns that the predictor used for training or testing. M is the number of events in the past or the order of the filter. Now let us consider the first input pattern, which contain M events. This vector is then delivered to the predictor for estimating the future value using the weight vector. This weight vector is updated using a law based on the function of mean square error. I-e

$$e = |x(n_{k+1} - \widehat{x}(n_{k+1}))|^2 \tag{3.36}$$

The above mentioned cost function (3.36) is used to adapt the parameter of LMS, MFLMS and AMFLMS. The formulation of KLMS is little bit different, it uses error vector obtained during training to test the validation set or desired data samples. The architecture of the predictor is given in Fig. 8.

3.5 Experimental Results

We proceed here in this section to demonstrate the performance of the proposed algorithm in the paper. First we evaluate the algorithm by considering the example of Mackey glass series under stationary as well as nonstationary situation and also with different noise levels. Next Lorenz time series is considered, only the performance is evaluated on X component of the series. We mainly

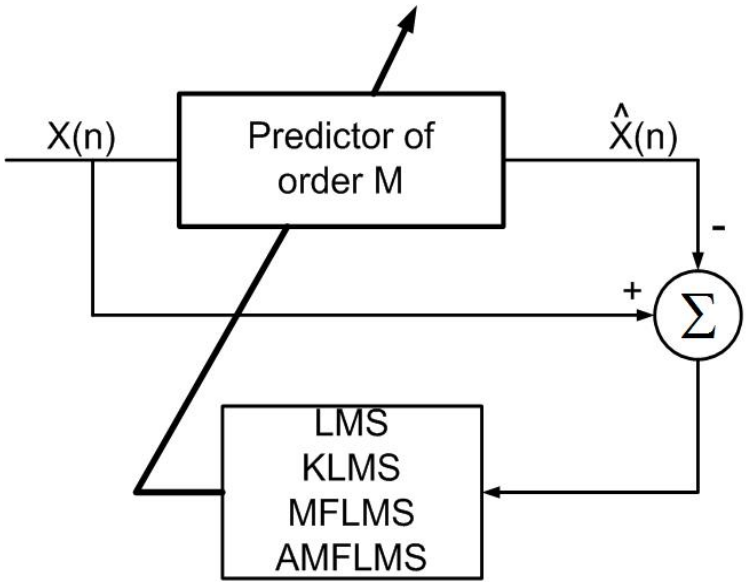

Figure 8: Architecture of the predictor

focus on the comparison of different algorithms with gradient based approach.

3.5.1 Mackey Glass Time series

The experimental results of prediction using LMS, KLMS, MFLMS and AMFLMS predictors are discussed. To counter the performance of the proposed filter, Chaotic Mackey-Glass time series is generated by the following delay differential equation (3.37) as

$$\frac{dx(t)}{dt} = -bx(t) + \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} \tag{3.37}$$

With b=0.1,a=0.2 and $\tau = 20$, initial conditions $x(t-\tau) = 0$ for $0 \leq t \leq \tau$. So we have a series x(k) for k=1,2,3,.....,3000, obtained by 3.24. It is achieved by sampling the continuous curve x(t) with the interval of 1 second. The time embedding length or the order of the filter M is 10 for all the experiments, and a segment of 700 samples is used for training, while next 200 is used for testing. Three cases are debated here for the inspection of the Mackey-Glass time series prediction. The generated time series is shown in Fig. 9.

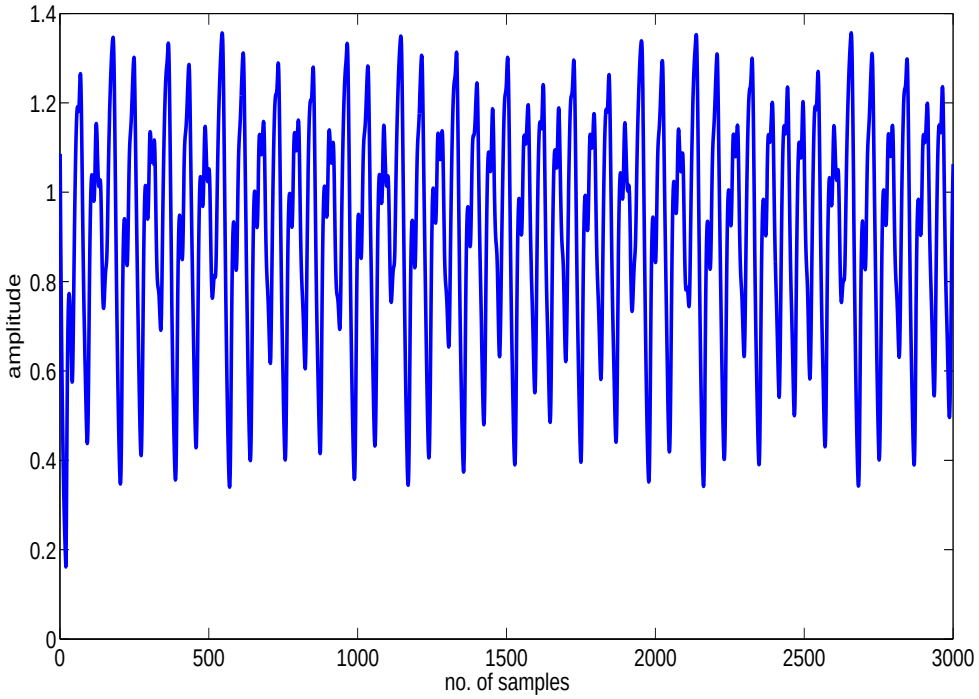

Figure 9: Mackey-Glass time Series

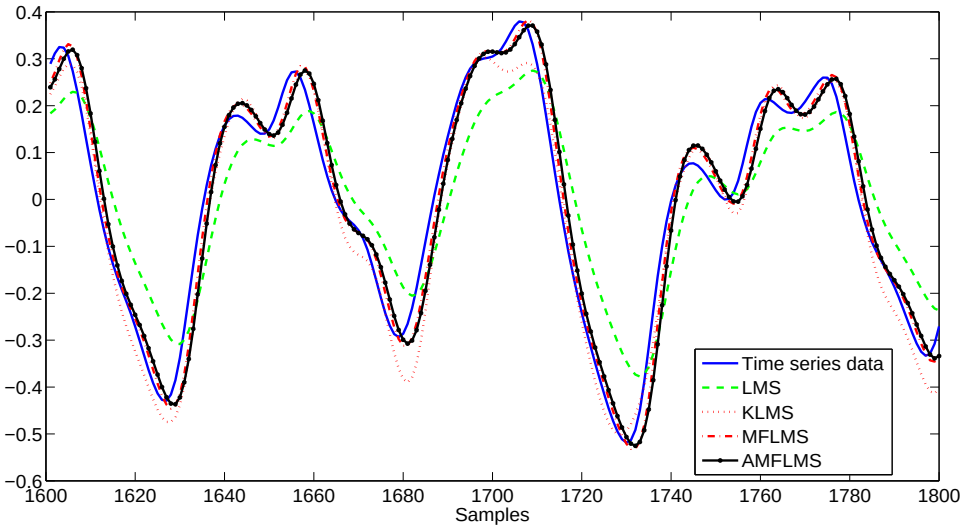

Figure 10: Prediction analysis of Stationary time series

Case I: Simulation for Stationary Series

The experiment is performed to test the proposed AMFLMS algorithm on the stationary series. Sample points 1100-1600 are used for training and also to adjust the weights of the filter. These weights are then used to validate the test samples 1600-1800. The parameters of the filters are adjusted as: LMS ($\mu = 0.015$), KLMS($\eta = 0.2$) and Gaussian kernel with a width of 0.1 is used, MFLMS($\mu = 0.08, \mu_f = 0.03, f = 0.0972 and \beta = 0.48$) and AMFLMS ($\beta = 0.48$). The predictive performance of LMS, KLMS, MFLMS and the proposed AMFLMS is depicted in fig.10, and to validate the superiority, learning curves of of MSE are plotted in Fig. 11. The mean behavior of the step size parameter μ and μ_f is depicted in Fig. 12 for the stationary series during training. One can easily observe that value of both these parameters goes up very fast initially and then settled. This behavior clearly explains fast convergence. In Fig. 13, the value of the fractional order derivative is also plotted. In Fig. 13 the value of ν is plotted with the whole iteration. This figure clearly mentions the variation in the value of ν has been remained fixed after 100 iterations of time.

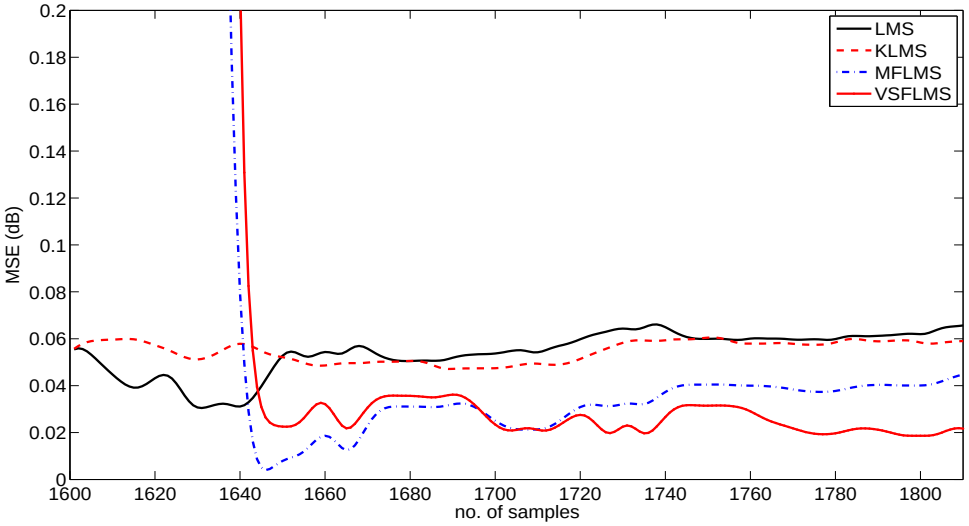

Figure 11: Learning curves of stationary time series

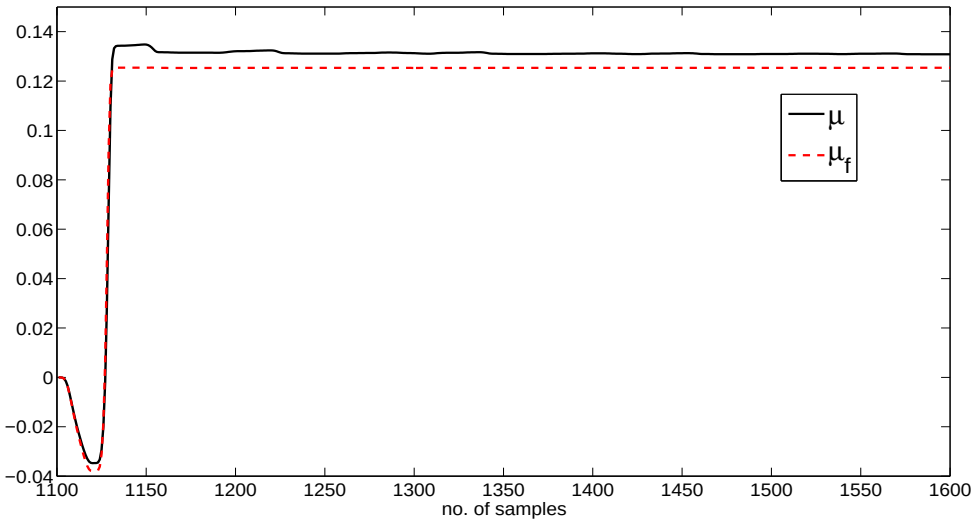

Figure 12: Behavior of the step size parameters during training

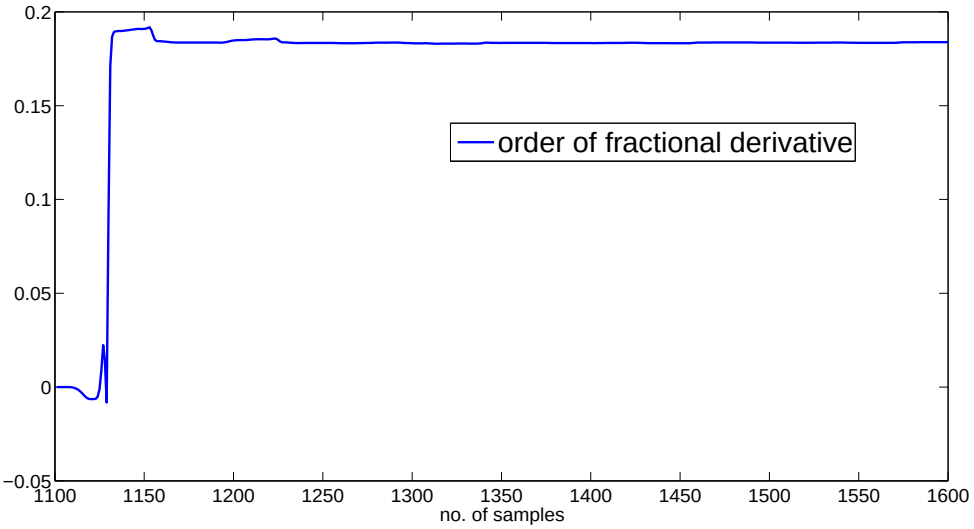

Figure 13: Fractional order derivative during training in stationary environment

Case II: Simulation with Stationary Series having different noise levels

All the stationary of time series data is corrupted by Gaussian noise with zero mean and different noise variances σ_n^2 to further validate the proposed algorithm. The values of the step size parameters are adjusted accordingly as in case 1. In this experiment 200 independent monte carlo simulations are to test the proposed algorithm, and the results are displayed using the form (mean $\pm$ standard deviation) in Table 1.

Table 3: Performance Comparison of Stationary Series with Different Noise Levels

Algorithm	**LMS**	**KLMS**	**MFLMS**	**AMFLMS**
Traning MSE($\sigma = 0.005$)	0.0024785±1.7523-005	0.004514±6.1332e-018	0.001795±2.1904e-018	0.001625±1.7523e-018
$TestingMSE(\sigma = 0.005)$	0.0024785±4.5714e-019	0.0049036±6.1332e-18	0.001795±2.1904e-018.	0.064462±2.8037e-017
$TraningMSE(\sigma = 0.02)$	0.0024785±-4.571e-19	0.06970±1.4101e-17	0.02247±-3.657e-018	0.0017952±2.2857e-019
$TestingMSE(\sigma = 0.02)$	0.0024075±4.571e-19	0.074906±1.4628e-17	0.03756±3.868e-18	0.067227±1.48932e-019
$TraningMSE(\sigma = 0.04)$	0.0046267±0.00028867	0.009670±0.0005167	0.002137±0.000398	0.0038219±0.0001680
$TestingMSE(\sigma = 0.04)$	0.0046379±0.0003128	0.007162±0.0007767	0.004117±0.000282	0.003822±0.0002391
$TraningMSE(\sigma = 0.1)$	0.015863±0.0009315	0.018072±0.0019643	0.015315±0.00091788	0.012115±0.000808
$TestingMSE(\sigma = 0.1)$	0.016209±0.0014763	0.018942±0.0021438	0.073852±0.00484	0.01213±0.0013113
$TraningMSE(\sigma = 0.5)$	0.47252±0.24589	0.20107±0.01235	0.3209±0.058172	0.4908±0.34772
$TestingMSE(\sigma = 0.5)$	0.4487±0.18748	0.29321±0.032418	0.34689±0.090635	0.46899±0.09916

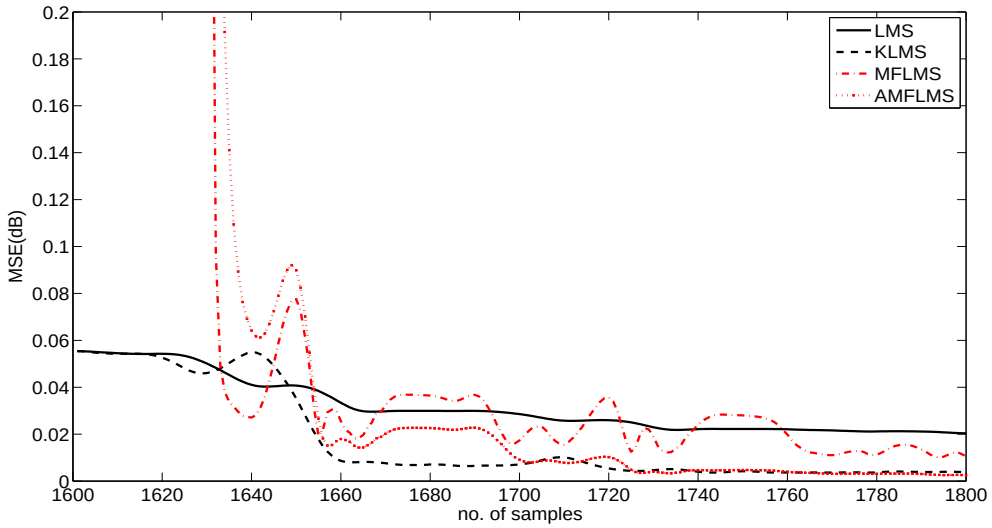

Figure 14: Learning curves of Time Series with noise level 0.1

It is clear from the Table. 1, that the proposed AMFLMS algorithm gives best results in low noise levels but its performance is slightly degraded in the presence of high noise levels, as mentioned in the table for noise level $\sigma_n^2 = 0.5$. Noise $\sigma_n^2 = 0.1$ is added to the stationary series and MSE is plotted in fig.14. The performance of AMFLMS is still better.

Case III: Experiment with Nonstationary Series

In order to examine the behavior and to further validate the performance of the proposed algorithm, experiments are performed on the nonstationary time series [132] data. We generate a modifies series by adding a sinusoid ($y = 0.3\sin(2\pi ft)$, $f = 1/3000$ and t is a continuous curve $1 : 3000$) with an amplitude of 0.3 and a frequency of 3000 samples per second to stationary Mackey Glass time series as shown in Fig. 15 with the indication of training and testing samples. Next we perform the experiment to test the proposed algorithm on nonstationary series with time varying mean. Training data is taken from the samples 1100-1600. After adjusting the parameters of the filters, the predictors are then given the test data of sample points 1600-1800, and the predictive performance of these are shown in Fig.16. The filter parameters are selected are as: LMS ($\mu =$

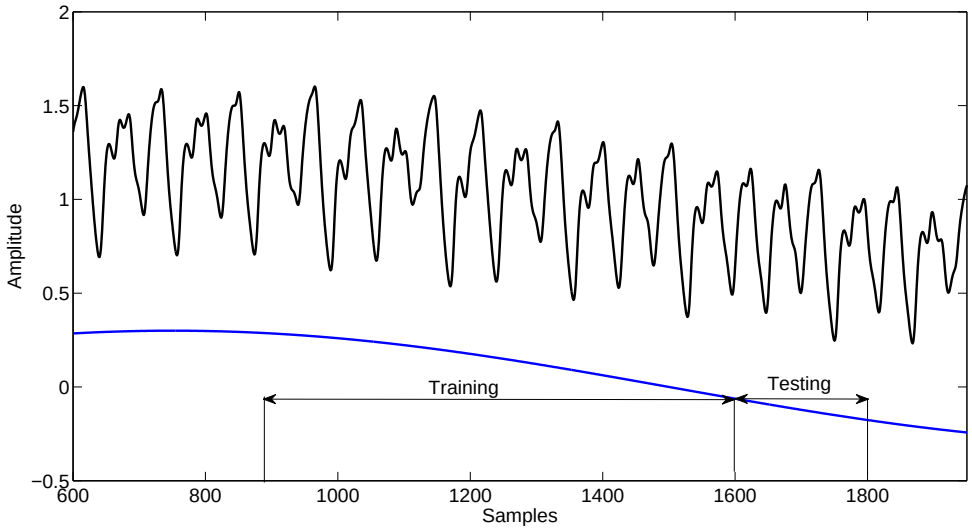

Figure 15: Modified Time Series

0.02), KLMS ($\eta = 0.2$ and Gaussian kernel with a width of 0.1), MFLMS ($\mu = 0.01, \mu_f = 0.03, f = 0.0972, \beta = 0.45$) and AFLMS $\beta = 0.45$. The performance comparison of the proposed algorithm with LMS, KLMS and MFLMS is detailed in terms of mean square error in Fig. 17. The result shows the superiority of the proposed AMFLMS filter.

In order to further validate the performance of the proposed algorithm, the experiment is performed with less number of training samples for modified non stationary series. 200 samples(1600-1800) are taken to train the algorithms, while another 300 sample poins(1600-1900) are are used as a test or validation set for LMS, KLMS, MFLMS amd AMFLMS. The results are displayed in terms of mean square error(MSE) in Fig. 19. and it depicts that the performance of the proposed AMFLMS algorithm is better in comparison with others. The behavior of the three proposed adaptive parameters μ, μ_f and f are displayed in Fig. 18 during training.

Case IV: Simulation for Nonstationary Series with different noise levels

In table II, we summarize the behavior of the filters towards nonstationary time series prediction with different noise variances σ_n^2. After 200 independent runs of Monte Carlo simulations, the

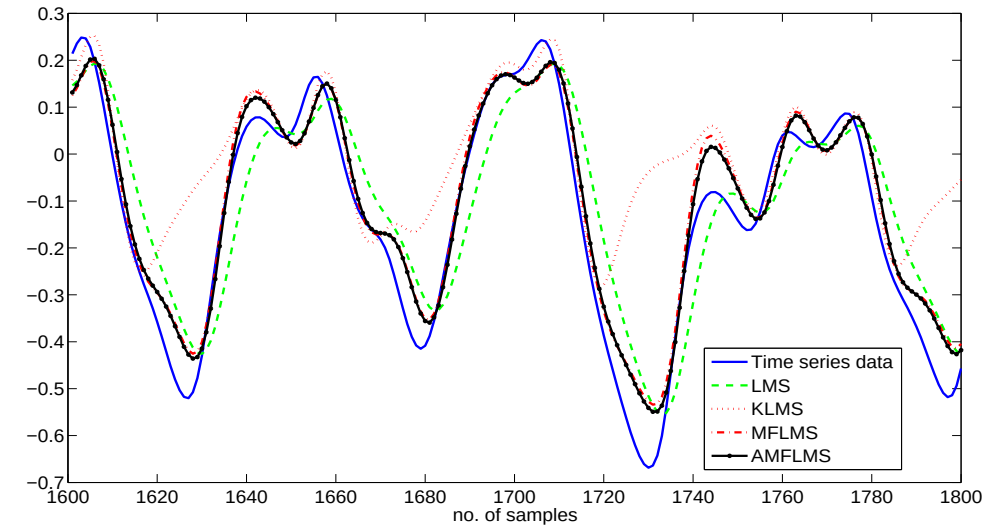

Figure 16: Prediction Analysis of modified Time Series

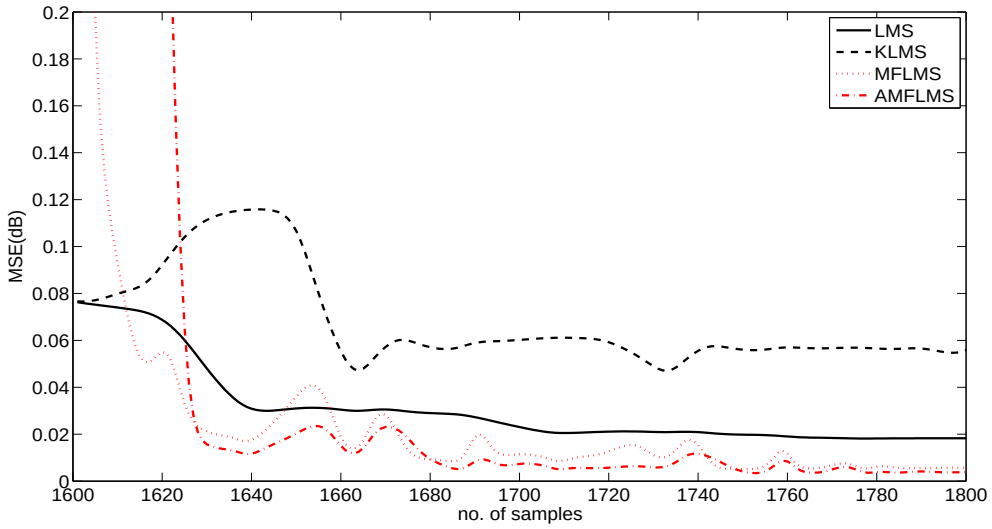

Figure 17: Learning curves of Nonstationary Series

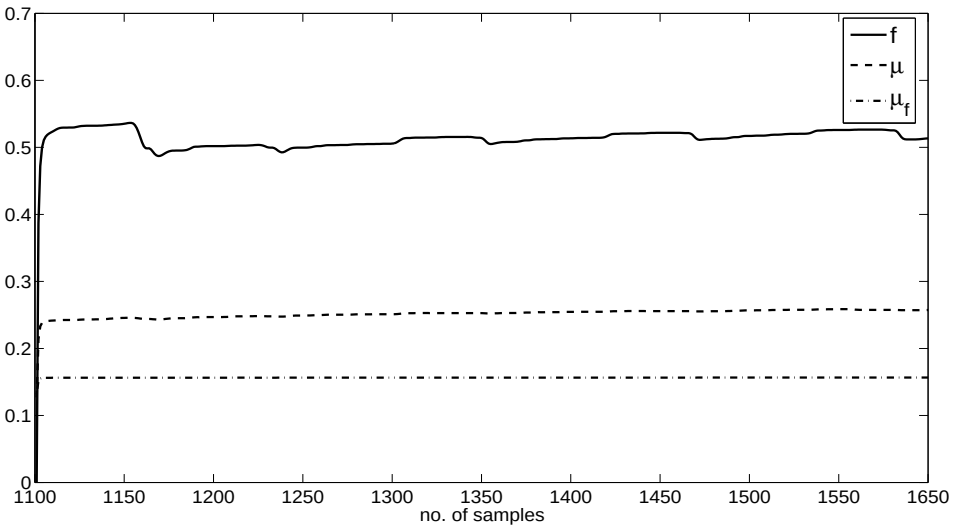

Figure 18: Behavior of three adjustable parameters during training of nonstationary series

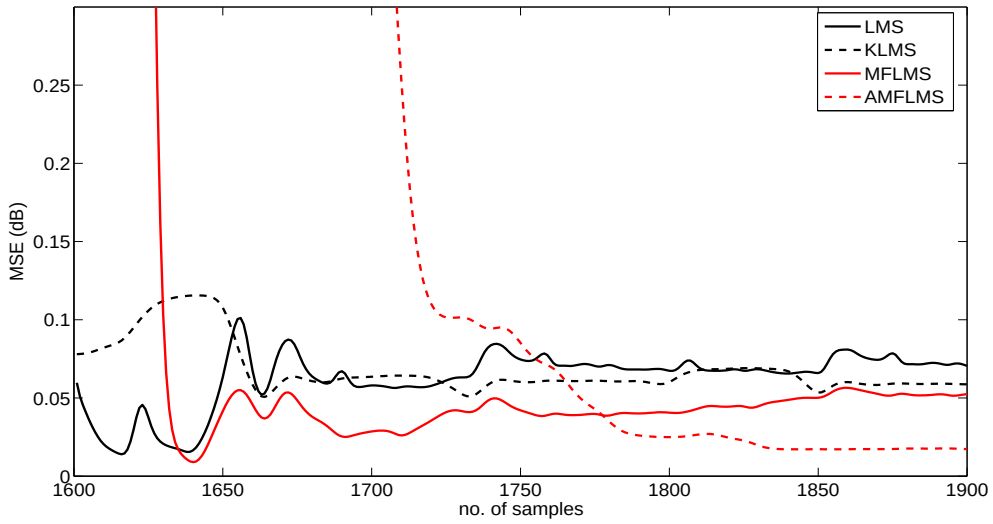

Figure 19: Learning curves of modified series with different training and test samples

results are displayed using the form (mean $\pm$ standard deviation). The results shows that the proposed algorithm achieves better performance in terms of low as well as high noise levels.

Table 4: Performance Comparison of Nonstationary Series with Different Noise Levels

Algorithm	**LMS**	**KLMS**	**MFLMS**	**AMFLMS**
$TraningMSE(\sigma = 0.005)$	0.0024999±-2.9452e-005	0.004534±6.0666e-005	0.001818±2.3034e-005	0.001646±9.033e-006
$TestingMSE(\sigma = 0.005)$	0.002487±3.5334e-005	0.0049278±6.8589e-005	0.067229±0.00021486.	0.064473±0.0001984
$TraningMSE(\sigma = 0.02)$	0.0030239±0.00013662	0.005153±0.00038711	0.0023982±0.000117	0.0022105±5.4614e-005
$TestingMSE(\sigma = 0.02)$	0.002944±0.00017584	0.0055061±0.00040318	0.067354±0.0010987	0.064812±0.00095545
$TraningMSE(\sigma = 0.04)$	0.0046267±0.00028867	0.0070108±0.0007506	0.0041104±0.000254	0.0039353±0.00019148
$TestingMSE(\sigma = 0.04)$	0.0045764±0.00033865	0.0073131±0.00084083	0.0067554±0.002234	0.065667±0.00019555
$TraningMSE(\sigma = 0.1)$	0.015673±0.0009315	0.018072±0.0019643	0.015461±0.0009336	0.015315±0.0009178
$TestingMSE(\sigma = 0.1)$	0.016209±0.0014763	0.018942±0.0021438	0.073457±0.0057177	0.073852±0.00484
$TraningMSE(\sigma = 0.5)$	0.47252±0.24589	0.20107±0.01235	0.47498±0.24988	0.329±0.058172
$TestingMSE(\sigma = 0.5)$	0.4487±0.18748	0.29321±0.032418	0.49428±0.25345	0.34689±0.090365

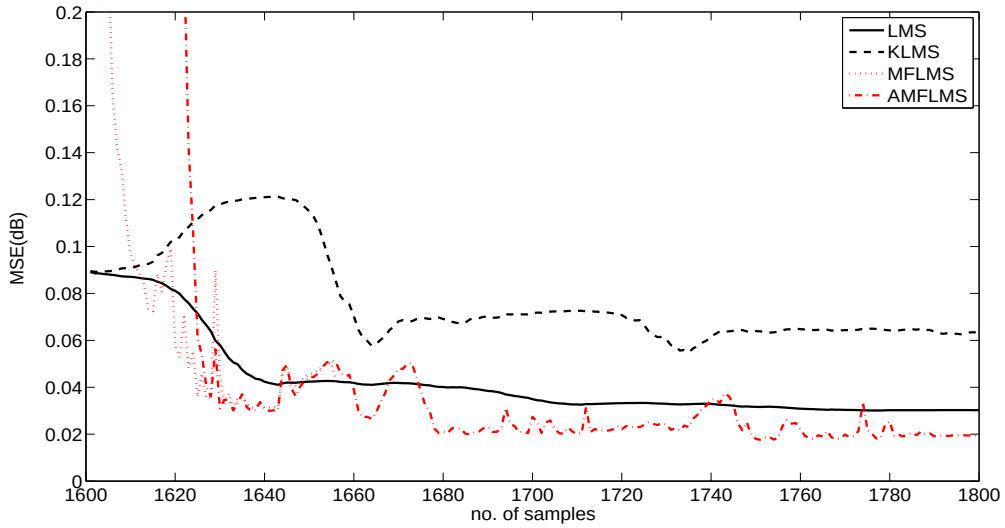

Figure 20: Learning curves of modified series with noise 0.15

Noise with variance of $\sigma_n^2 = 0.15$ is added to the modified time series and the performance of LMS, KLMS, MFLMS and AMFLMS is observed in terms of MSE as the figure of merit in Fig. 20.

3.5.2 Lorenz Time Series

The Lorenz time series represents a dynamical system with chaotic flow, also prominent for its unique butterfly shape. The Lorenz system is nonlinear, three dimensional and deterministic and described by the following set of differential equations.

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y(t) - x(t)) \\ \frac{dy}{dt} &= -x(t)z(t) + \gamma x(t) - y(t) \\ \frac{dz}{dt} &= x(t)y(t) - \beta z(t) \end{aligned} \tag{3.38}$$

The values of the unknown parameters upon which the Lorenz system exhibits chaotic behavior are set as $\sigma = 10$, $\gamma = 28$ and $\beta = \frac{8}{3}$. Consider the initial values as $x(0) = 1, y(0) = 1, z(0) = 1$, and a sampling period of 0.01 second is fixed to obtain the sample data using first order approximation method. State trajectory of the Lorenz system is depicted in Fig. 21.

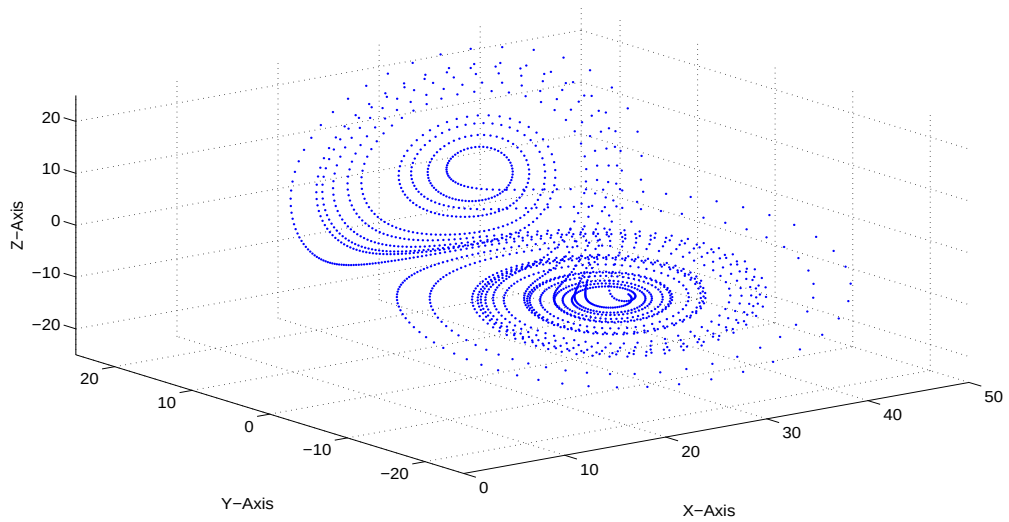

Figure 21: State trajectory of the lorenz system

Then another experiment is performed with training sample points 1100-1600 and test sample

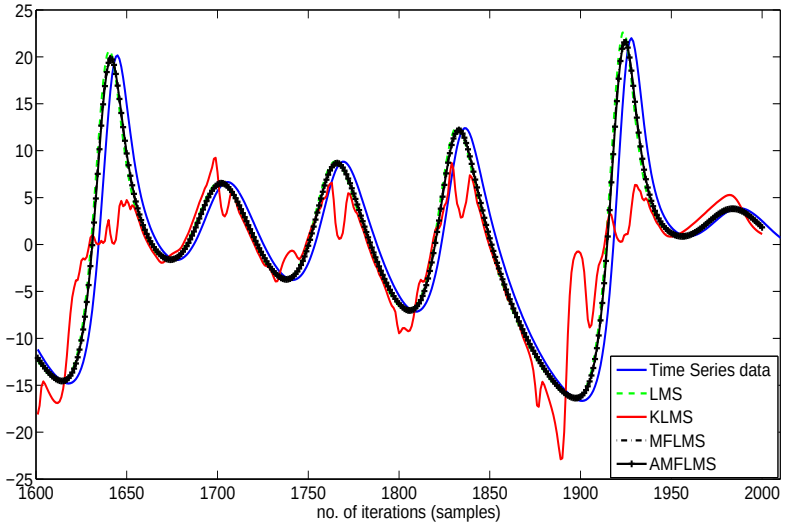

Figure 22: Prediction analysis of X component of Lorenz time series during training

points 1600-2000 to evaluate the performance of the filters as depicted in Fig. 22. To further validate the performance of the proposed algorithm, mean square error is plotted in Fig. 23.

3.6 Conclusion

Modified and Adaptive step size modified fractional LMS algorithm employing a time varying step size parameters in the weight adaptation equation is introduced. These parameters are adjusted according to the square of a time averaged error through gradient based technique. As a result, the proposed algorithm can effectively handle the prediction of chaotic series. One application of predicting a chaotic Mackey Glass and another of lorenz series is presented, which illustrate that our proposed algorithm achieves better performance in terms of mean square error(MSE). The performance of the proposed algorithm is also tested and analyzed for both the stationary as well as non stationary time series and also with different noise levels in comparison with that of LMS, KLMS and MFLMS. This new algorithm (AMFLMS) hopefully is another contribution in the field of nonlinear fractional signal processing.

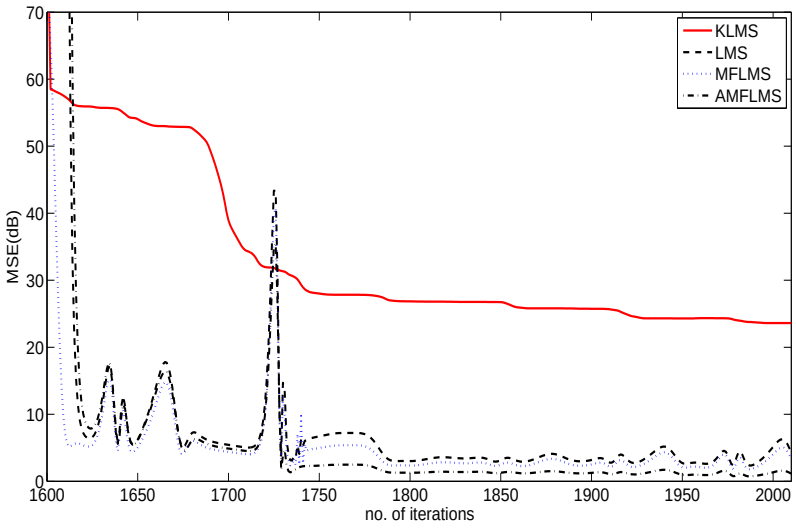

Figure 23: Learning curves for x-component of Lorenz Series

4 Kernel Fractional and Kernel Affine Projection Algorithms with their applications to nonlinear systems

The main aim of this chapter is to develop a Kernel Fractional Affine Projection algorithm (KFAPA). A method is introduced to adjust the Reimann-luiville fractional derivative term in the kernel affine projection algorithm. The proposed algorithm is then applied on the prediction of only the X-component of the three dimensional chaotic lorenz time series, nonlinear channel equalization, forecasting of carbon dioxide concentration in the atmosphere and static function approximation. Along with that a new Identification scheme is presented for the Hammerstein nonlinear controlled autoregressive system using kernel affine projection algorithm. This identification scheme is validated on a set of experiments with different noise variances.

4.1 Kernel Fractional Affine Projection Algorithm

In this section of the dissertation, the main aim is to develop a Kernel Fractional Affine Projection algorithm (KFAPA). A method is introduced to adjust the Reimann-luiville fractional derivative

in the weight adaptation equation of the kernel Affine Projection Algorithm. To formulate the KFAPA, we introduce a cost function that includes a fractional term, this cost function is minimized according to the mean square algorithm using gradient based smoothed nonlinear recursive method. The proposed algorithm is then validated on a set of experiments including prediction of only the X-component of the three dimensional, deterministic and chaotic lorenz time series, nonlinear channel equalization, prediction of carbondioxide concentration in the atmosphere and static function approximation.

4.1.1 Affine Projection Algorithm

Affine projection algorithm [133] uses smoothed newton's recursion to formulate the algorithm. In order to minimize the cost function, we hereby uses the following example sequence [**x**(i),d(i)] as

$$J(w) = \frac{1}{2}\sum_{i=0}^{n}(d(i) - \mathbf{w}^T\mathbf{x}(i))^2 \tag{4.1}$$

The gradient with respect to weigh vector **w** is

$$\nabla_w J = -\sum_{i=1}^{n} x(i)(d(i) - \mathbf{w}^T\mathbf{x}(i)) \tag{4.2}$$

The weights are adjusted according to stochastic Newton method as

$$\mathbf{w}(i) = \mathbf{w}(i-1) - \eta_t(\nabla_w^2 J)^{-1}\nabla_w J \tag{4.3}$$

Tap weight vector customary be initialized to zero. η_t is the small positive stepsize. Here we perform a line search along the gradient descent direction to compute the weight vector. The

corresponding steepest descent and newton's recursion becomes

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t \mathbf{x}(i)[\mathbf{x}^T(i)\mathbf{x}(i) + \epsilon\mathbf{I}]^{-1}[d(i) - \mathbf{x}(i)^T\mathbf{w}(i-1)] \tag{4.4}$$

ϵ is the small positive constant, which prevents division by zero errors. To smoothen the Newton's recursion and to increase the convergence speed we proceed as

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t[\mathbf{x}(i)\mathbf{x}^T(i) + \epsilon\mathbf{I}]^{-1}\mathbf{x}(i)[d(i) - \mathbf{x}(i)^T\mathbf{w}(i-1)] \tag{4.5}$$

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t[\mathbf{x}(i)\mathbf{x}^T(i) + \epsilon\mathbf{I}]^{-1}[\mathbf{x}(i)d(i) - \mathbf{x}(i)\mathbf{x}(i)^T\mathbf{w}(i-1)] \tag{4.6}$$

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t[\mathbf{R}_u + \epsilon\mathbf{I}]^{-1}[\mathbf{r}_{d\mathbf{u}} - [\mathbf{R}_u + \epsilon\mathbf{I}]\mathbf{w}(i-1)] \tag{4.7}$$

Where the matrix $\mathbf{R}$ is strictly positive definite and invertible.

4.1.2 Kernel Affine Projection Algorithm

Poor performance is examined, when the difference of mapping between **x** and **d** is highly non-linear. A nonlinear mapping is introduced in [40] as $\varphi(\mathbf{x}(i))$, which gives a powerful model $\mathbf{w}^T\varphi(\mathbf{x}(i))$ than $\mathbf{w}^T\mathbf{x}$. So by using this model and finding **w** through stochastic newton method may prove an efficient mechanism towards nonlinear filtering as APA ensures for linear problems. Using the sequence$[\varphi(i), d(i)]$ to parameter weight vector **w** as

$$J(w) = \frac{1}{2}\sum_{i=0}^{n}(d(i) - \mathbf{w}^T\varphi(i))^2 \tag{4.8}$$

By minimizing the cost function (4.8), the stochastic gradient descent becomes

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t\mathbf{\Phi}(i)[\mathbf{d}(i) - \mathbf{\Phi}^T\mathbf{w}(i-1)] \tag{4.9}$$

And stochastic Newton method becomes

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t[\mathbf{\Phi}(i)\mathbf{\Phi}(i)^T + \epsilon\mathbf{I}]^{-1}\mathbf{\Phi}(i)[\mathbf{d}(i) - \mathbf{\Phi}^T\mathbf{w}(i-1)] \tag{4.10}$$

Notice that by using searl's matrix identity

$$[\mathbf{\Phi}(i)\mathbf{\Phi}(i)^T + \epsilon\mathbf{I}]^{-1}\mathbf{\Phi}(i) = \mathbf{\Phi}(i)[\mathbf{\Phi}(i)^T\mathbf{\Phi}(i) + \epsilon\mathbf{I}]^{-1} \tag{4.11}$$

so the corresponding weight update equation becomes

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t\mathbf{\Phi}(i)[\mathbf{\Phi}(i)^T\mathbf{\Phi}(i) + \lambda\mathbf{I}]^{-1}[\mathbf{d}(i) - \mathbf{\Phi}^T\mathbf{w}(i-1)] \tag{4.12}$$

Therefore KAPA only needs a $K \times K$ matrix inversion which can be computed easily by sliding window trick.

4.1.3 Proposed kernel Fractional Affine Projection Algorithm

Here we introduced a mechanism to update the weights of kernel affine projection algorithm with the inclusion of fractional derivative term.

$$\mathbf{w}(i) = \mathbf{w}(i-1) - \eta_t(\nabla_w^2 J)^{-1}\nabla_w J - \nabla_w^\nu J \tag{4.13}$$

and the cost function with fractional derivative is written as

$$\begin{aligned} (\frac{\partial}{\partial w(i)})^\nu J &= -e(i)\varphi(i)D^\nu\mathbf{w}(i) \\ (\frac{\partial}{\partial w(i)})^\nu J &= -e(i)\varphi(i)[\frac{1}{\Gamma(2-\nu)}\mathbf{w}^{1-\nu}(i)] \end{aligned} \tag{4.14}$$

It is more appropriate to use the previous value of weight vector $\mathbf{w}$, that is $\mathbf{w}$(i-1). The weight update equation of the kernel fractional affine projection algorithm is

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta_t \mathbf{\Phi}(i)[\mathbf{\Phi}(i)^T \mathbf{\Phi}(i) + \lambda \mathbf{I}]^{-1}[\mathbf{d}(i) - \mathbf{\Phi}^T \mathbf{w}(i-1)] + \mu_t(\mathbf{e(i)}^T \varphi(i)) \frac{\mathbf{w}^{1-\nu}(i-1)}{\Gamma(2-\nu)} \tag{4.15}$$

Where η_t and μ_t are the small positive step size, typically lie between 0 and 1. In practice, we do not have access to the transformed weights $\mathbf{w}$ in the feature space, so the updated weights have to be evaluated through expansion coefficients as

$$\mathbf{w}(i) = \sum_{j=1}^{i} \mathbf{a}_j(i)\varphi(j), \forall i > 0. \tag{4.16}$$

Now to evaluate the $\mathbf{a}_j(i)$, setting the initial guess $\mathbf{w}(\mathbf{0}) = \mathbf{0}$, and adopted procedure as

$$\begin{aligned} toremovenumbering(beforeeachequation)\mathbf{w}(0) &= 0 \\ \mathbf{w}(0) &= \eta d(1)\varphi(1) = \mathbf{a}_1(1)\varphi(1) \\ \vdots &= \vdots \\ \mathbf{w}(i-1) &= \sum_{j=1}^{i-1} \mathbf{a_j}(i-1)\varphi(j) \end{aligned}$$

similarly the $K \times K$inversion in (4.17) is evaluated by the following searl's identity

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} (\mathbf{A} - \mathbf{BD^{-1}C})^{-1} & -\mathbf{A^{-1}B}(\mathbf{D} - \mathbf{CA^{-1}B})^{-1} \\ -\mathbf{D^{-1}C}(\mathbf{A} - \mathbf{BD^{-1}C})^{-1} & (\mathbf{D} - \mathbf{CA^{-1}B})^{-1} \end{pmatrix} \tag{4.17}$$

where $\mathbf{A} = \mathbf{\Phi}(i-1)^T\mathbf{\Phi}(i-1) + \epsilon\mathbf{I}$, $\mathbf{B} = \mathbf{\Phi}(i)^T\varphi(i)$, $\mathbf{C} = \varphi(i)^T\mathbf{\Phi}(i)$ and $\mathbf{D} = \varphi(i)^T\varphi(i) + \epsilon$

And the fractional part of (17) is efficiently evaluated as

$$to remove numbering(before each equation)(\mathbf{e(i)}^T\varphi(i))\mathbf{w}^{1-\nu}(i) = (\mathbf{e(i)}^T\varphi(i))(\sum_{j=1}^{i}\mathbf{a}_j(i)\varphi(j))^{1-\nu}$$

ν is the order of fractional derivative and the term $(\mathbf{e(i)}^T\varphi(i))\mathbf{w}^{1-\nu}(i)$ in 4.18 is evaluated using Eq. 20.

4.1.4 Experimental Results

This section presents experimental results to reveals the performance of the proposed algorithm. The performance of KFAPA is validated by the prediction of X-component of Lorenz time series, equalization of nonlinear channel, CO_2 concentration in atmosphere and static function approximation is presented.

4.1.5 Time series Prediction

A demanding problem in signal processing is that of predicting a future value in the chaotic time series. The prediction method takes time series history to estimate the future value. A time series is being defined as a sequence of vectors or scalers depends upon time.

Let us consider a time series $T.S. = \{u(n_0), u(n_1), u(n_2), ..., u(n_{k-1}), u(n_k), u(n_{k+1}), u(n_{k+2}), ...\}$

To predict future value at a certain time, through the process known as embedding, a delay line is formed. The output contains a tapped delay line and then form a matrix shifted by one time sample

in each column and is written as

$$\begin{pmatrix} u(n_k) & u(n_{k+1}) & ... & u(n_{k+N-1}) \\ u(n_{k-1}) & u(n_k) & ... & u(n_{k+N-2}) \\ u(n_{k-2}) & u(n_{k-1}) & ... & u(n_{k+N-3}) \\ ... & ... & ... & ... \\ u(n_{k-M-1}) & u(n_{k-M}) & ... & u(n_{k+N-M}) \end{pmatrix} \tag{4.18}$$

Columns in the matrix represents the pattern of the input for training or testing. M is known as the order of the filter. The first input pattern is delivered to predictor for estimating the future value by using the weight vector. This weight vector is then updated using a law based on the function of mean square error as given below.

$$e = |u(n_{k+1}) - \widehat{u}(n_{k+1})|^2 \tag{4.19}$$

The above mentioned cost function 4.19 is used to adapt the parameters of the predictors.

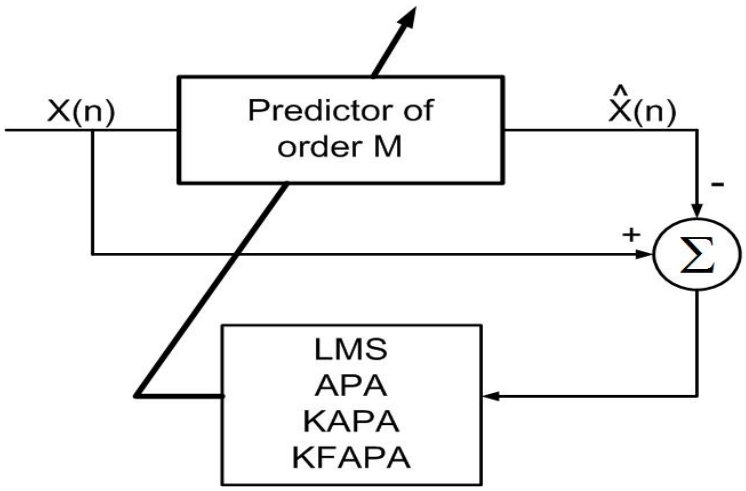

Figure 24: Architecture of the predictor

Lorenz Time Series

Lorenz time series exhibits chaotic flow. Lorenz system is three dimensional, nonlinear, and deterministic, expressed as the following nonlinear partial differential equations.

$$\begin{aligned} to remove numbering(before each equation)\frac{dx}{dt} &= \sigma(y(t) - x(t)) \\ \frac{dy}{dt} &= -x(t)z(t) + \gamma x(t) - y(t) \\ \frac{dz}{dt} &= x(t)y(t) - Bz(t) \end{aligned} \tag{4.20}$$

Parameters of the Lorenz system on which its behavior becomes chaotic are $\sigma = 10$, $\gamma = 28$ and $B = \frac{8}{3}$. Set the initial values as $x(0) = 1$, $y(0) = 1$, $z(0) = 1$, a sampling period is taken 0.01 second, also it is fixed to obtain the sample data using first order approximation method. The three dimensional state trajectory of the Lorenz system is shown in Fig. 25. The next experiment

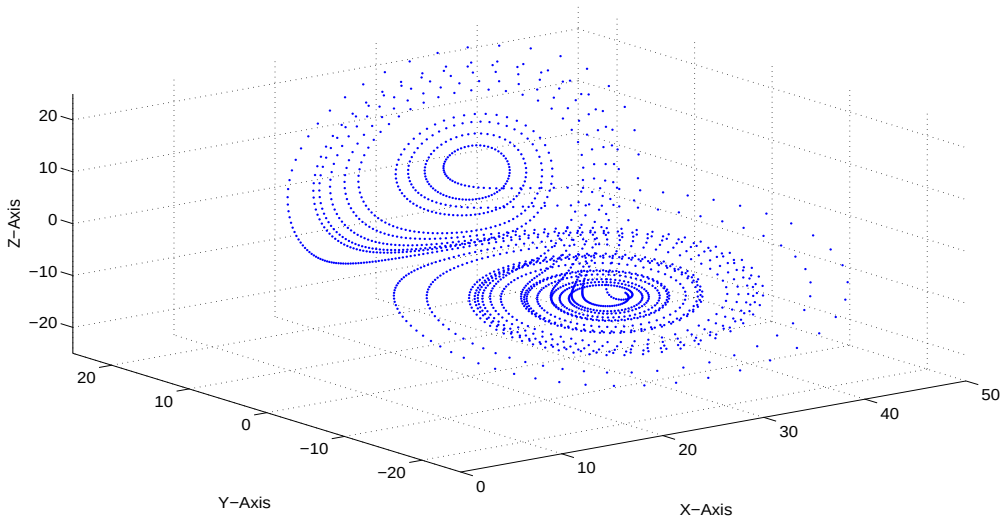

Figure 25: State trajectory of the lorenz system

is performed with training sample points 500-1000 of the X-component of the lorenz series, and test sample points 1000-1200 to evaluate the performance of the proposed algorithm. The time

embedding length or the order of the filter M is 5 for this experiment. To validate the performance of the proposed algorithm, learning curves in terms of mean square error (MSE) is plotted in Fig. 26.

Table 5: Performance Comparison of LMS, APA, KAPA and KFAPA for X-Component of lorenz series prediction with Different Noise Levels

Algorithm	**LMS**	**APA**	**KAPA**	**KFAPA**
Traning MSE($\sigma = 0.05$)	0.02250±1.75e-005	0.01454±5.26e-004	0.041827±1.93e-065	0.02156±1.39e-005
$TestingMSE(\sigma = 0.05)$	0.01583±0.22e-005	0.07820±0.00306	0.017738±2.28e-003.	0.091538±1.367e-004
$TraningMSE(\sigma = 0.02)$	0.01903±0.003149	0.025175±0.000198	0.001399±0.000189	0.0020052±6.50e-005
$TestingMSE(\sigma = 0.02)$	0.002970±0.00170	0.005556±0.0004324	0.001356±0.000131	0.0027892±9.49e-004
$TraningMSE(\sigma = 0.04)$	0.004349±0.0003927	0.00680±0.0007169	0.004117±0.000251	0.0048219±0.0001680
$TestingMSE(\sigma = 0.04)$	0.0049979±0.0004128	0.007162±0.0009767	0.005117±0.000382	0.005822±0.0003391
$TraningMSE(\sigma = 0.1)$	0.015863±0.0009678	0.010932±0.0025963	0.026628±0.00094904	0.045301±0.00086739
$TestingMSE(\sigma = 0.1)$	0.016166±0.007296	0.019066±0.0044241	0.035729±0.00128194	0.0555606±0.0023143
$TraningMSE(\sigma = 0.5)$	0.42356±0.10011	0.50001±0.010242	0.82530±0.2115	0.4209±0.030332
$TestingMSE(\sigma = 0.5)$	0.51752±0.22074	0.69218±0.028178	0.91918±0.32231	0.52013±0.047689

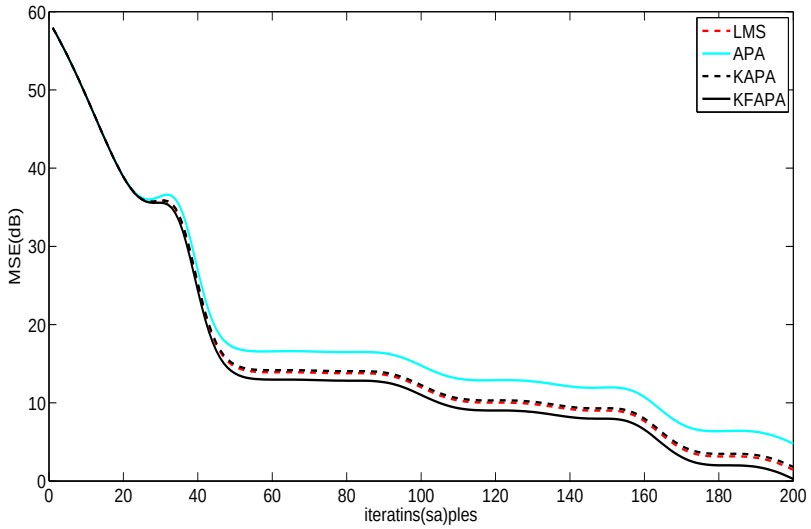

Figure 26: MSE curves for x-component of Lorenz Series

The X-component of the Lorenz time series is corrupted with white noise having different variances, the algorithms including the proposed one is tested and the results are displayed in the tabular form. Mean square error is observed after 1000 Monte Carlo simulations and also with different noise variances added to the X-component of the lorenz series is listed in Table 5. Throughout this experiment Gaussian kernel is used and the kernel width is set as 0.1.

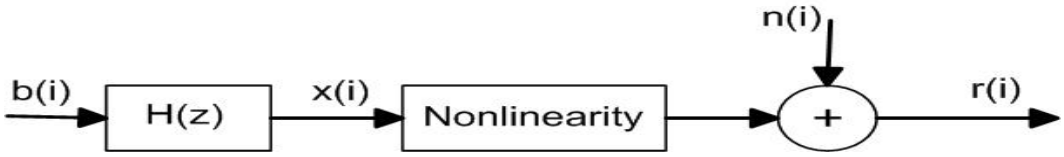

Figure 27: Architecture of nonlinear channel

Nonlinear channel equalization

Nonlinear channel model considered here in this experiment as a test bench, consists of serial connection of linear filter and a memoryless nonlinearity. This type is commonly used to model digital communication channels and digital magnetic recording channels. A binary signal $[b(1), b(2), b(3), ..., b(k)]$is fed into a nonlinear channel, while adding static nonlinearity and additive white Gaussian noise the signal will be observed as $[r(1), r(2), r(3), ..., r(k)]$. The channel model is defined as$h(i) = b(i) + 0.5b(i-1)$ and output is $r(i) = h(i) - 0.9h(i)^2 + n(i)$. The nonlinear channel model is shown in Fig. 27. Where $n(i)$ is the additive white Gaussian noise having variance of 0.01. We aim here in this experiment to reproduce the original signal with low error rate. The time embedding length or the order of the filter is 5. 5000 symbols are used to train the coefficients of the nonlinear channel and the mean square error during training is displayed in the Fig. 28. Table 6, shows the comparison of APA, KAPA and the proposed KFAPA for non-

Table 6: Performance Comparison of APA, KAPA and KFAPA in nonlinear channel equalization

Algorithm	**MSE (dB)**
APA	0.6 ± 0.2
KAPA	0.55 ± 0.05
KFAPA	0.4 ± 0.1

linear channel equalization after 1000 montecarlo simulations. The performance of the proposed algorithm is also tested by inserting an abrupt change at iteration 500. It can be easily observed from Fig. 29, that the proposed algorithm is able to recover efficiently in comparison with its counterparts and the improvement of 0.1 dB is acheived.

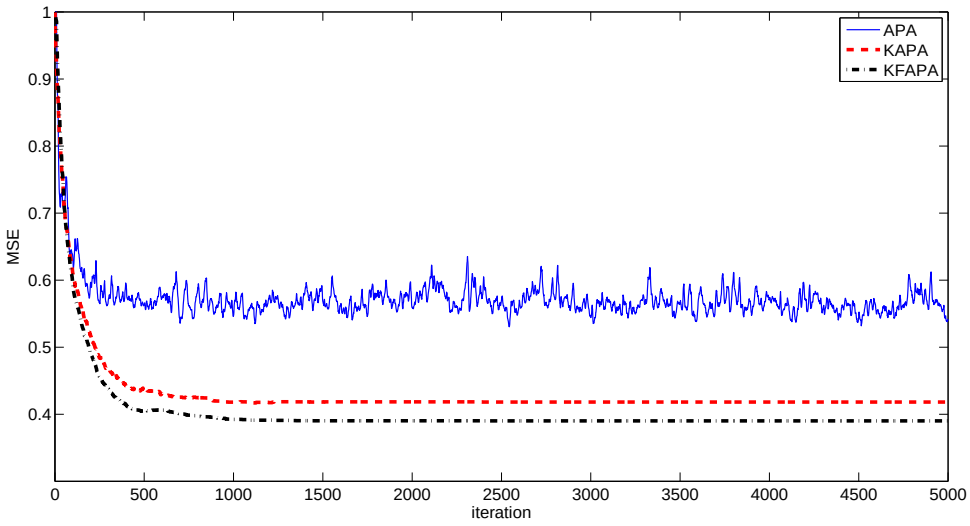

Figure 28: Learning curve of nonlinear channel equalization

4.1.6 Atmospheric Carbon dioxide concentration forecasting

In this experiment, the data consists of monthly average CO_2 concentrations(in parts per million by volume ppmv) in atmosphere collected at Mauna Loa observatory Hawai, between 1958 and 2008, with 600 total observations. the first 400 points are used for training while the other 200 for testing. The kernel function handles long term rising, seasonal effect, periodicity and some irregularities. The kernel function is

$$k(\mathbf{x},\mathbf{x}^{'}) = k_1(\mathbf{x},\mathbf{x}^{'}) + k_2(\mathbf{x},\mathbf{x}^{'}) + k_3(\mathbf{x},\mathbf{x}^{'}) + k_1(\mathbf{x},\mathbf{x}^{'}) \tag{4.21}$$

$k_1(\mathbf{x},\mathbf{x}^{'})$ is used to model the rising trends and is defined as

$$k_1(\mathbf{x},\mathbf{x}^{'}) = a_1^2 exp(-\frac{(\mathbf{x}-\mathbf{x}^{'})^2}{2a_2^2}) \tag{4.22}$$

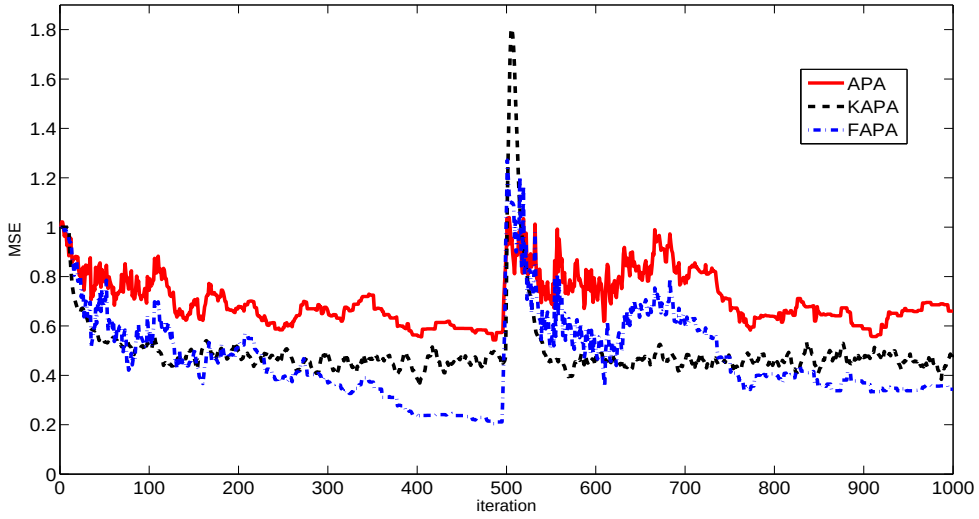

Figure 29: MSE curves for KFAPA of nonlinear channel equalization with an abrupt change at iteration 500

a_1 is the amplitude and a_2 is the kernel width. A seasonal effect is modeled through periodic kernel with a time period of 1 year. A gaussian kernel is used to put decay away from the exact periodicity.

$$k_2(\mathbf{x}, \mathbf{x}^{'}) = a_3^2 exp(-\frac{(\mathbf{x}-\mathbf{x}^{'})^2}{2a_2^4} - \frac{\sin^2 \pi((\mathbf{x}-\mathbf{x}^{'}))}{a_5^2}) \tag{4.23}$$

$a_3, a_4 and a_5$ are magnitude, smoothing factor and the periodic component. To handle the irregularities in the observed data set $k_3(\mathbf{x}, \mathbf{x}^{'})$ is defined as

$$k_3(\mathbf{x}, \mathbf{x}^{'}) = a_6^2(1 + \frac{(\mathbf{x}-\mathbf{x}^{'})^2}{2a_8^2 a_7^2})^{-a_8^2} \tag{4.24}$$

a_6 is magnitude, a_7 and a_8 are the smoothing factor and shape parameter respectively. To model the noise component $k(\mathbf{x}, \mathbf{x}^{'})$ is defined as

$$k_4(\mathbf{x}, \mathbf{x}^{'}) = a_9^2(\frac{(\mathbf{x}-\mathbf{x}^{'})^2}{2a_{10}^2} + a_{11}^2 \delta(\mathbf{x}-\mathbf{x}^{'}) \tag{4.25}$$

Table 7: kernel function parameter values

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
66	0.075	0.40	0.0576	1.0878	0.6600	0.4167	0.78	0.18	3.7509	0.1900

a_9 and a_{10} is magnitude and smoothing factor of the coloured noise. a_{11} is the magnitude of white noise. The values of these kernel function parameters are listed in Table 7. The Co_2 concentration in atmosphere is modeled as a function of time and as shown in Fig. 30. The prediction of carbon

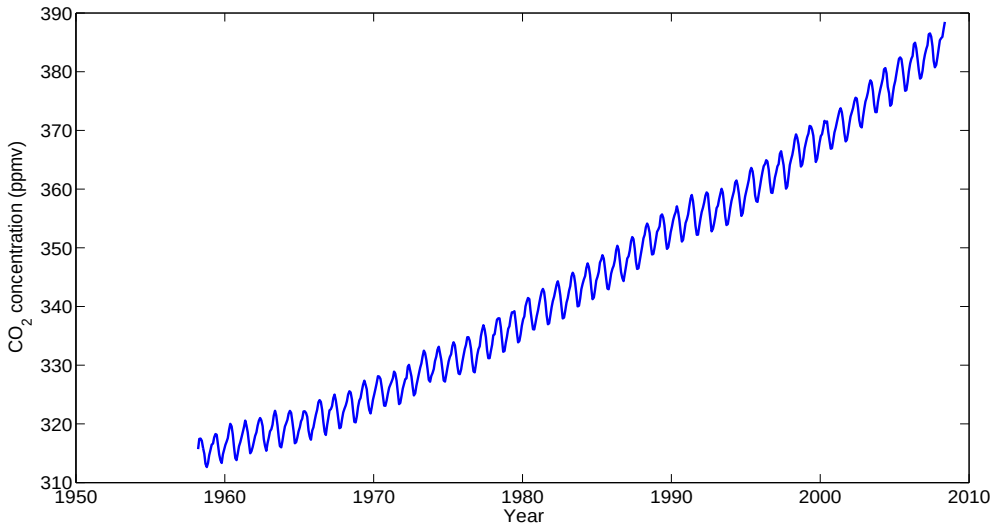

Figure 30: Carbon dioxide concentration trend from year 1958 to 2008

dioxide concentration using the proposed kernel fractional affine projection algorithm is depicted in Fig. 31, while the prediction error in terms of mean square is plotted in Fig. 32.

4.1.7 Static function approximation

This example explains the static function approximation, where the desired output data is generated by Eq. (4.26).

$$d(i) = cos(\omega(x(i) - \tau) + v(i) \tag{4.26}$$

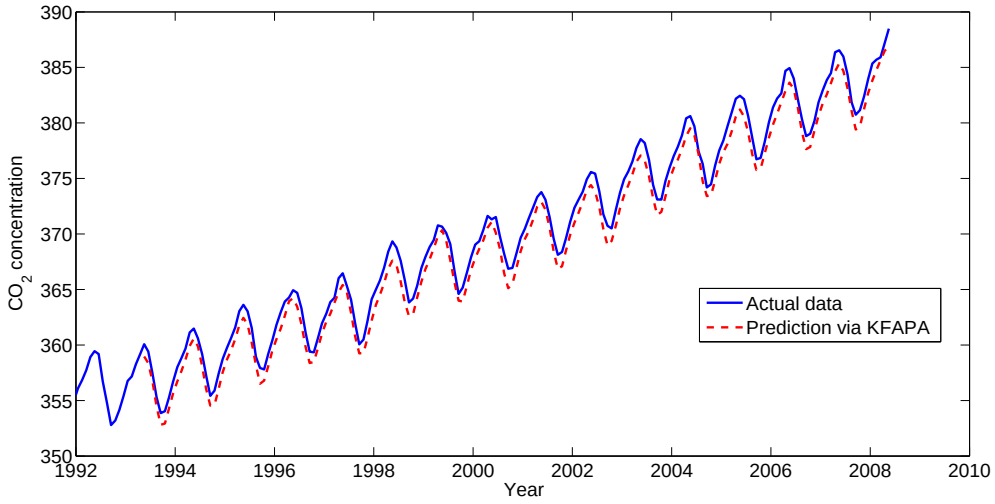

Figure 31: Forecasting prediction result of KFAPA for carbondioxide concentration

where $\tau \geq 0$, input $\mathbf{x}(i)$ is uniformly distributed over $[\tau, \tau + 2]$, and [v(i)] is a zero mean Gaussian noise with variance σ_v^2. In this experiment N=2000 samples are generated with $\sigma_v^2 = 0.01,\ \omega = 2$ and $\tau = 1.0$. 500 samples are used for training and another 200 is used for testing. The test pattern is shown in Fig. 33.

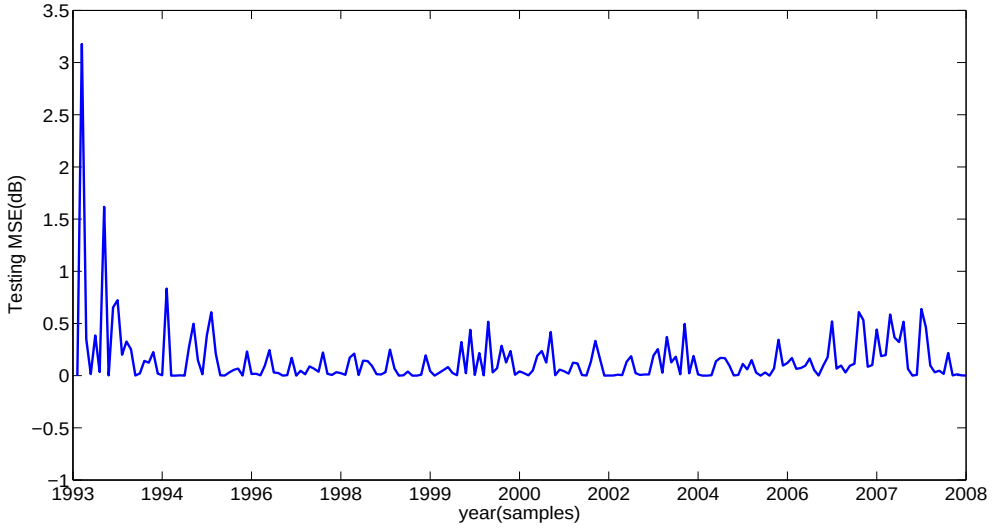

Figure 32: Testing Mean Square Error for KFAPA of carbondioxide concentration

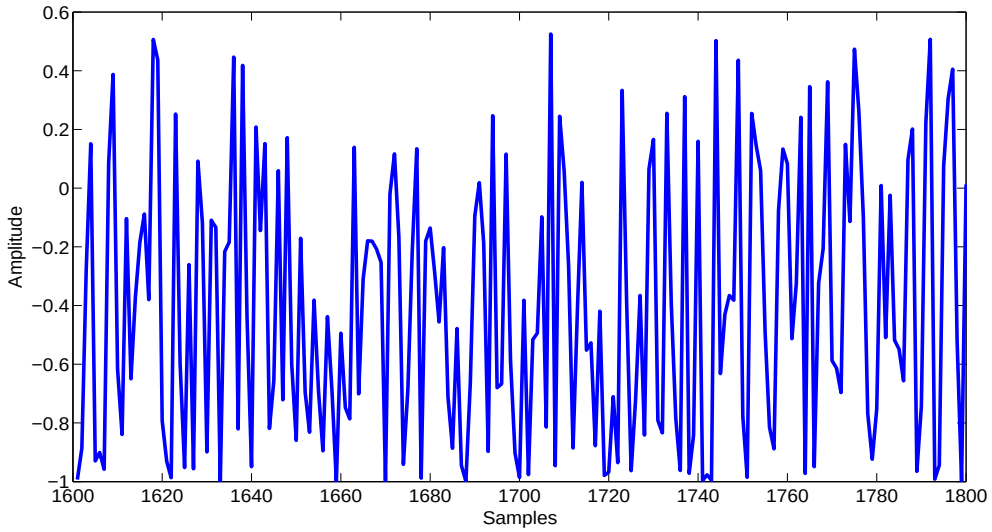

Figure 33: Test samples

4.2 Identification of Hamerstein Nonlinear controlled Auto-regressive systems using kernel affine projection algorithm

Identification of linear and nonlinear systems through parameter estimation has gained considerable importance in control engineering and signal processing. Hammerstein systems are a class of nonlinear models consist of linear time invariant block followed by the static nonlinear blocks. These models are very much in common and are widely used in the industry, e.g., valve saturation, signal analysis, nonlinearities arrises in the chemical processes, deadzone nonlinearities, actuator saturation [134, 135, 136, 137, 138] etc. For Hammerstein nonlinear CAR models research community has shown great interest in developing the new methods for the identification of these models, including gradient based identification method [139], parameter estimation using recursive method [140] for Hammerstein CAR system, Hirarical identification schemes [141, 142, 143], multi innovation method [144, 145, 146] etc. Feng Deng and chen [147] developed an iterative least squares and recursive least squares based identification method for the identification for ARMA/CARMA model. The idea he introduces is to replace the unmeasurable noise term by their estimates and compute those by parameter estimation and also prove its convergence. Half substitution method was also introduced by Voros [148].

Recently a new scheme is developed for parameter estimation method for the CARMA systems using single and two stage fractional least mean square algorithm. He splits the information vector into two parts and proved that the parameters are estimated efficiently [31].

A lot of other work has been performed on the parametric model identification of the Hammerstein system. Some of them assume the nonlinearity as a two- segment peicewise or a multiple segment piecewise linear function. Some assume hard nonlinearities or a discontinuous function. The Hammerstein parametric identification problem can also be efficiently deal with the iterative identification methods[16], Generally it is assumed that the nonlinearity $\bar{u} = f(u)$ is basically a polynomial of known order at the input and is expressed as the sum of unknown coefficients c_j and

the sum of nonlinear basis functions of known basis $f = (f_1, f_2, f_3), ..., f_{n_c}$. Here in this thesis, we also follow this method. Other existing methods for parametric identification of Hammerstein system includes the subspace method, the over parameterization method, the iterative least squares method, blind identification method, the iterative identification method and so on etc.

In this section of the thesis, we developed kernel affine projection based identification algorithm for the parameter estimation of the hammerstein nonlinear controlled auto regressive system. This approach handles efficiently the nonlinearities arising in the information vector due to some known nonlinear basis function. Comparison with other adaptive strategies has also been made with the proposed identification scheme. The results are presented in terms of parameter error and also with the mean square error based on these estimated parameter. The other adaptive strategies include Least mean square algorithm, Kernel least mean square algorithm, Modified fractional least mean square algorithm and affine projection algorithm.

4.2.1 System description and Estimation algorithm

Let us introduce some brief description about the system model. The governing equation for nonlinear Hammerstein system described by a CAR model is

$$A(z)y(t) = B(z)\bar{u}(t) + \nu(t) \tag{4.27}$$

Input $\bar{u}(t)$ is the output of the nonlinear block and is assumed to be a nonlinear function with known basis$f_1, f_2, f_3, ...f_m$ of the system input $u(t)$. The output of the nonlinear block is described as

$$\bar{u}(t) = f(u(t)) = \sum_{i=1}^{m} c_i f_i(u(t)) \tag{4.28}$$

$y(t)$ is the system output, $\nu(t)$ is the stochastic white noise. $A(z), B(z)$ are the n degree polynomial in z^{-1} which is unit backward shift operator $z^{-1}y(t) = y(t-1)$ with

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + ... + a_n z^{-n} \tag{4.29}$$

$$B(z) = b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + ... + b_n z^{-n} \tag{4.30}$$

The above mentioned model (4.29) represents the nonlinear CAR model. The objective here in this thesis is to identify the model by estimating the unknown parameters a_i, b_i, c_i using the available input and output$u(t), y(t)$. From 4.27,4.28 we have

$$A(z)y(t) = B(z)f(u(t)) + \nu(t) \tag{4.31}$$

and is easy to get

$$\begin{aligned}
y(t) &= -\sum_{i=1}^{n} a_i y(t-i) + \sum_{i=1}^{n} b_i \bar{u}(t-i) + \nu(t) \\
&= -\sum_{i=1}^{n} a_i y(t-i) + \sum_{i=1}^{n} b_i \sum_{j=1}^{m} c_j f_j(u(t-i)) + \nu(t) \\
&= -\sum_{i=1}^{n} a_i y(t-i) + \sum_{i=1}^{n} \sum_{j=1}^{m} c_j b_i f_j(u(t-i)) + \nu(t) \\
&= -\sum_{i=1}^{n} a_i y(t-i) + \sum_{i=1}^{n} [c_1 b_i f_1(u(t-i)) \\
&+ c_2 b_i f_2(u(t-i)) + ... + c_m b_i f_m(u(t-i))] + \nu(t) \\
&= -\sum_{i=1}^{n} a_i y(t-i) + c_1 b_1 f_1(u(t-1)) + c_1 b_2 f_1(u(t-2)) ..... + c_1 b_n f_1(u(t-n)) \\
&+ c_2 b_1 f_2(u(t-1)) + c_2 b_2 f_2(u(t-2)) + ... + c_2 b_n f_2(u(t-n)) \\
&+ c_3 b_1 f_3(u(t-1)) + c_3 b_2 f_3(u(t-2)) + ... + c_3 b_n f_3(u(t-n)) + ... + c_m b_1 f_m(u(t-1)) \\
&+ c_m b_2 f_m(u(t-2)) + ... + c_m b_n f_m(u(t-n)) + \nu(t)
\end{aligned} \tag{4.32}$$

define the parameter vector θ and information vector $\varphi(t)$ as

$$\theta = \begin{bmatrix} \mathbf{a} \\ c_1\mathbf{b} \\ c_2\mathbf{b} \\ \vdots \\ \vdots \\ c_m\mathbf{b} \end{bmatrix}, \qquad \varphi(\mathbf{t}) = \begin{bmatrix} \psi_{\mathbf{0}}(\mathbf{t}) \\ \psi_{\mathbf{1}}(\mathbf{t}) \\ \psi_{\mathbf{2}}(\mathbf{t}) \\ \vdots \\ \vdots \\ \psi_{\mathbf{m}}(\mathbf{t}) \end{bmatrix} \tag{4.33}$$

The parameter vector is defined as

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ \vdots \\ a_n \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ \vdots \\ b_n \end{bmatrix} \tag{4.34}$$

$$\psi_{\mathbf{0}} = \begin{bmatrix} -y(t-1) \\ -y(t-2) \\ -y(t-3) \\ \vdots \\ \vdots \\ -y(t-n) \end{bmatrix}, \qquad \psi_{\mathbf{j}} = \begin{bmatrix} f_j(u(t-1)) \\ f_j(u(t-2)) \\ f_j(u(t-3)) \\ \vdots \\ \vdots \\ f_j(u(t-n)) \end{bmatrix}, \qquad \mathbf{j} = \mathbf{1}, \mathbf{2}, \mathbf{3}, ..., \mathbf{m} \tag{4.35}$$

Then we have

$$y(t) = \varphi^{\mathbf{T}}(\mathbf{t})\theta + \nu(\mathbf{t}) \tag{4.36}$$

we make the following assumption on the noise sequence $\nu(t)$

$$\begin{aligned} E[\nu(t)] &= 0 \\ E[\nu^2(t)] &= \sigma_v^2 \end{aligned}$$

4.2.2 Design methodology

In this section, a brief introduction of the stochastic gradient based iterative Linear and fractional algorithm is presented as well as proposed iterative kernel affine projection algorithm is given for the identification of nonlinear CARMA system.

Least Mean Square Identification Algorithm

The Least Mean Square(LMS) algorithm developed by widrow and Hopf [127] is used in many areas of signal processing applications including beamforming, channel equalization, Adaptive control, machine learning, system identification etc. By taking $y(t) = \varphi(t)\theta^T + v(t)$ as the reference signal and $y(t) = \hat{\theta}^T\varphi(t)$ as the estimated output of the algorithm. The cost function is minimized for the parameter identification of the CARMA system as

$$J(\hat{\theta}) = \frac{1}{2}\|y(t) - \hat{\theta})^i\varphi(t)\|^2 \tag{4.37}$$

$$\begin{aligned} \Delta J(\hat{\theta}) &= e(t)\frac{d}{d\hat{\theta}}e(t) \\ &= -e(t)\varphi(t) \end{aligned} \tag{4.38}$$

The Iterative weight updating formula for the LMS based identification of HNCAR method is given as

$$\begin{aligned} \hat{\theta}(t+1) &= \hat{\theta}(t) - \mu e(t) \Delta J(\hat{\theta}) \\ \hat{\theta}(t+1) &= \hat{\theta}(t) + e(t)\varphi(t) \end{aligned} \tag{4.39}$$

Affine Projection Identification Algorithm

The Affine Projection Algorithm uses Newton's smoothed recursion to further enhance the performance of the Gradient based LMS algorithm[131]. The weight updation equation for the Affine projection Algorithm is

$$\mathbf{w}(t+1) = \mathbf{w}(t) + (\nabla_w^2 J(w))^{-1} \nabla J(w) \tag{4.40}$$

The final relation becomes

$$\begin{aligned} \mathbf{w}(t) &= \mathbf{w}(t-1) + \eta[\mathbf{x}(t)\mathbf{x}^T(t) + \epsilon\mathbf{I}]\mathbf{x}(t)e(t) \\ \mathbf{w}(t) &= \mathbf{w}(t-1) + \eta[\mathbf{R}_{\mathbf{x}} + \epsilon\mathbf{I}]\mathbf{x}(t)e(t) \end{aligned} \tag{4.41}$$

Accordingly on a similar basis, the parameter estimation for the identification of CAR model is developed as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \eta[\mathbf{R}_{\varphi}(t) + \epsilon\mathbf{I}]^{-1}\varphi(t)e(t) \tag{4.42}$$

Here in 4.42$\mathbf{R}_{\varphi}(t)$ is the information correlation matrix and η is the step size parameter.

4.2.3 Kernel Affine Projection Identification Algorithm for identificaton of nonlinear systems

A poor performance can be expected when the difference or the mapping between **x** and **d** is highly nonlinear. A nonlinear mapping is introduced in as $\varphi(\mathbf{x}(t))$, which gives a powerful model $\mathbf{w}^T\varphi(\mathbf{x}(t))$ than $\mathbf{w}^T\mathbf{x}$. So by using this model and finding **w** through stochastic newton method may prove an efficient mechanism towards nonlinear filtering as APA ensures for linear problems. Using the example sequence $[\varphi(i), y(t)]$ to estimate the parameter vector θ as

$$J(\hat{\theta}) = \frac{1}{2}\|(y(t) - \hat{\theta})^T\varphi(t))\|^2 \tag{4.43}$$

By minimizing the above mentioned cost function (4.43) with respect to $\hat{\theta}$ and by straight forward manipulation, the stochastic gradient descent becomes

$$\hat{\theta}(t) = \bar{\theta}(t-1) + \eta_t\mathbf{\Phi}(t)[\mathbf{y}(t) - \mathbf{\Phi}^T\hat{\theta}(t-1)] \tag{4.44}$$

Where η_t is the small positive constant and the stochastic Newton method becomes

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \eta_t[\mathbf{\Phi}(t)\mathbf{\Phi}(t)^T + \epsilon\mathbf{I}]^{-1}\mathbf{\Phi}(t)[\mathbf{y}(t) - \mathbf{\Phi}^T\hat{\theta}(t-1)] \tag{4.45}$$

ϵ is the smoothing factor.

$$[\mathbf{\Phi}(t)\mathbf{\Phi}(t)^T + \epsilon\mathbf{I}]^{-1}\mathbf{\Phi}(t) = \mathbf{\Phi}(t)[\mathbf{\Phi}(t)^T\mathbf{\Phi}(t) + \epsilon\mathbf{I}]^{-1} \tag{4.46}$$

so the corresponding parameter update equation for the estimation of CAR system becomes

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \eta_t\mathbf{\Phi}(t)[\mathbf{\Phi}(t)^T\mathbf{\Phi}(t) + \lambda\mathbf{I}]^{-1}[\mathbf{y}(t) - \mathbf{\Phi}^T\hat{\theta}(t-1)] \tag{4.47}$$

Therefore KAPA only needs a $K \times K$ matrix inversion which can be easily computed by sliding window trick. K mainly depends the nature of the problem on which the parameter of the CAR model is estimated. $\mathbf{\Phi}(t)$ is the transformed information matrix. To initialize the kernel Affine projection identification algorithm, we take $\hat{\theta}(0) = 0$ or by using some small real value $\hat{\theta}(0) = 10^{-6}1_{n_0}$ with 1_{n_0} is being an n_0-dimensional column vector having values of the element 1.

Computational Steps

1. Input and output data[u(i),y(i)] is collected and form y(i) and $\varphi(t)$.

2. Initialize the weight vector $\hat{\theta}(0) = 10^{-6}1_{n_0}$.

3. Compute the estimate $\hat{\theta}(i)$.

4. Compare the current parameter vector $\hat{\theta}(t)$ with $\hat{\theta}(t-1)$, for some preset small value ϵ or some preset number of cycles are fixed, i-e, if it is satisfied then terminate the procedure and obtain the current $\hat{\theta}(t)$, otherwise increment by 1 and estimate it again.

Now let us assume that $\hat{\theta}_k$ is the k-th element of θ. Assuming that $c_1 = 1$, $\mathbf{a}(t) = [\hat{a}_1(t), \hat{a}_2(t), \hat{a}_3(t), ..., \hat{a}_n(t)]^T$ and $\mathbf{b}(t) = [\hat{b}_1(t), \hat{b}_2(t), \hat{b}_3(t), ..., \hat{b}_n(t)]^T$, $\mathbf{a}$ and $\mathbf{b}$ can be read from first and second n entries of $\hat{\theta}$. The estimate $\hat{c}_j(t)$ of c_j, $j = 2, 3, 4, ..., m$ may be computed as

$$\hat{c}_j(t) = \hat{\theta}_{jn+l}, \quad j = 2, 3, 4, 5, ..., m, \quad l = 1, 2, 3, 4, ..., n \tag{4.48}$$

It is evident from the above equation that for each c_j, there exists n estimates $\hat{c}_j(t)$ for $l = 1, 2, 3, 4, ..., n$. Alternative way is to take their average as the estimate of c_j as (4.49).

$$\hat{c}_j(t) = \frac{1}{n}\sum_{l=1}^{n} \frac{\hat{\theta}_{jn+l}(t)}{\hat{b}_l(t)}, \quad j = 2, 3, 4, 5, ..., m \tag{4.49}$$

4.3 Experimental Results

In this section, we present the results of simulations of two examples of hammerstein nonlinear CAR model by using proposed kernel affine projection adaptive identification algorithm. consider the following example

$$\begin{aligned}
A(z)y(t) &= B(z)\bar{u}(t) + \nu(t) \\
A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} \;=\; 1 - 1.35z^{-1} - 0.75z^{-2} \\
B(z) &= b_1 z^{-1} + b_2 z^{-2} \;=\; z^{-1} + 1.68z^{-2} \\
\bar{u}(t) &= f(u(t)) = c_1 u(t) + c_2 u^2(t) + c_3 u^3(t) \\
&= u(t) + 0.50u^2(t) + 0.20u^3(t) \\
\theta &= [\theta_1,\ \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8]^T \\
\theta &= [a_1, a_2, c_1 b_1, c_1 b_2, c_2 b_1, c_2 b_2, c_3 b_1, c_3 b_2]^T \\
\theta &= [1.35, -0.75, 1, 0.50, 0.20, 1.68, 0.84, 0.336]^T
\end{aligned} \tag{4.50}$$

The input $u(t)$ is taken as the persistent excitation signal sequence with zero mean and unit variance σ_u^2. The sequence is generated using a periodic signal switches between ± 0.5 with a period of 2π and the noise $\nu(t)$ is taken as a white noise with a constant variance σ_v^2.

$$u(t) = \sum_{j=1}^{t} a_j cos(2\pi\ f\ t + \phi_j) \tag{4.51}$$

The parameter estimation errors $\delta := \|\hat{\theta} - \theta\|/\|\theta\|$ versus t are shown in Figure 34 without noise, While Figure 35 shows the Output mean square error of the actual CAR system described above by using the estimated parameters without noise. Figure 36, is the enhancement of Figure 35. The output error at different iterations of time without noise is reported in Table 8. The results in Fig. 34, describes the parameter estimation error δ against t number of iterations without noise.

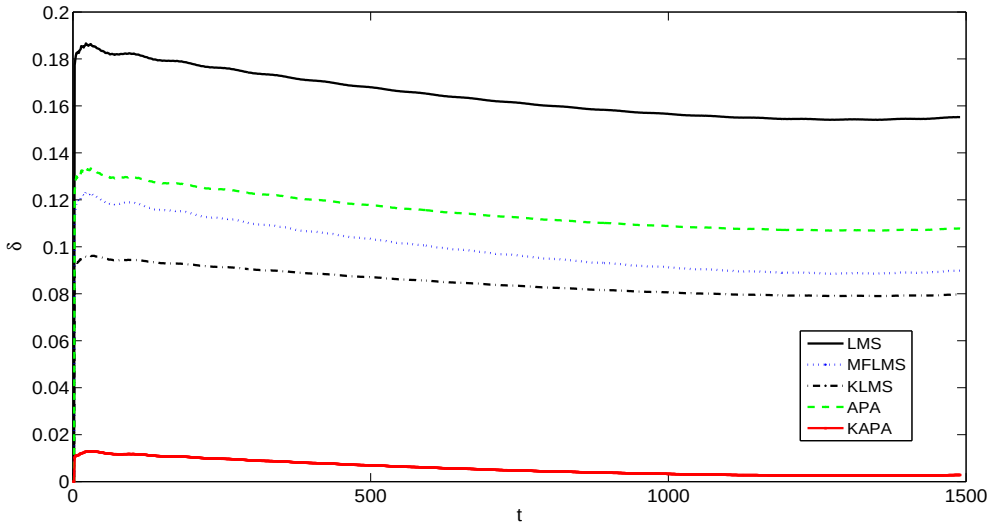

Figure 34: The parameter estimation error δ versus t

The step size parameters used in this experiment for adaptive online identification algorithms are taken as 0.01 for LMS, $\beta = 0.5,\ \mu = 0.0005,\ \mu_f = 0.003 \quad and \quad f = 0.392$ for MFLMS, 0.3 for KLMS, $\eta = 0.24 \quad and \quad \epsilon = 0.01$ for APA and $eta_t = 0.003 and \epsilon = 0.1$ for Kernel affine projection identification algorithm. Based on these estimated parameters, the output error of the Hammerstein nonlinear CAR model in terms of mean square as shown in Fig. 35. Fig. 36 is the enhancement of Fig. 36 . In Table 8, the percentage error of the estimated parameters is displayed at different iterations of time t.

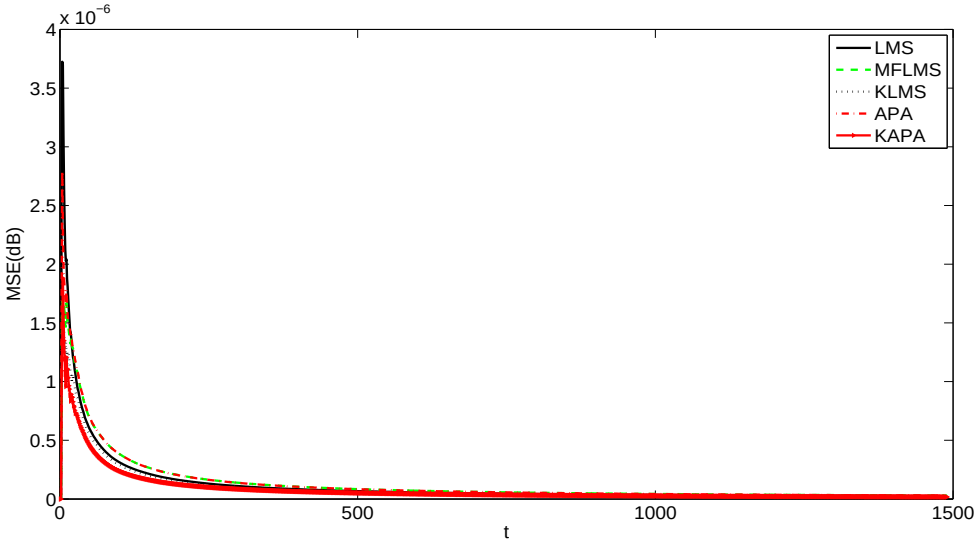

Figure 35: MSE of CAR model using estimated weights versus t

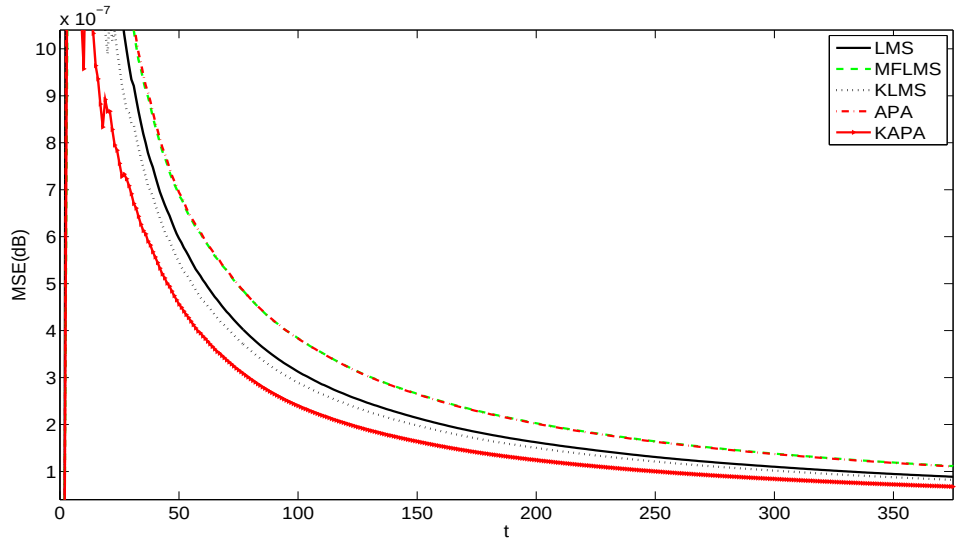

Figure 36: (Enhanced Figure 2)MSE of CAR model using estimated weights versus t

Table 8: Parameter Estimates θ without noise

t	**Algorithm**	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	δ
	LMS	1.881765	-1.045425	1.3939	0.69695	0.27878	2.341752	1.170876	4683504	39.39
	MFLMS	1.86678	-1.0371	1.3828	0.6914	0.27656	2.323104	1.161552	0.4646208	38.28
250	KLMS	1.866645	-1.037025	1.3827	0.69135	0.27654	2.322936	1.161468	0.4645872	38.27
	APA	1.8819	-1.0455	1.394	0.697	0.2788	2.34192	1.17096	0.468384	39.4
	KAPA	1.83762	-1.0209	1.3612	0.6806	0.27224	2.286816	1.143408	0.4573632	36.12
	LMS	1.718955	-0.954975	1.2733	0.63665	0.25466	2.139144	1.069572	0.4278288	27.33
	MLMS	1.703025	-0.946125	1.2615	0.63075	0.2523	2.11932	1.05966	0.423864	26.15
500	KLMS	1.71153	-0.95085	1.2678	0.6339	0.25356	2.129904	1.064952	0.4259808	26.78
	APA	1.730295	-0.961275	1.2817	0.64085	0.25634	2.153256	1.076628	0.4306512	28.17
	KAPA	1.688715	-0.938175	1.2509	0.62545	0.25018	2.101512	1.050756	0.4203024	25.09
	LMS	1.565865	-0.869925	1.1599	0.57995	0.23198	1.948632	0.974316	0.3897264	15.99
	MFLMS	1.569375	-0.871875	1.1625	0.58125	0.2325	1.953	0.9765	0.3906	16.25
1000	KLMS	1.58679	-0.88155	1.1754	0.5877	0.23508	1.974672	0.987336	0.3949344	17.54
	APA	1.592325	-0.884625	1.1795	0.58975	0.2359	1.98156	0.99078	0.396312	17.95
	KAPA	1.56762	-0.8709	1.1612	0.5806	0.23224	1.950816	0.975408	0.3901632	16.12
	LMS	1.487025	-0.826125	1.1015	0.55075	0.2203	1.85052	0.92526	0.370104	10.15
	MFLMS	1.475955	-0.819975	1.0933	0.54665	0.21866	1.836744	0.918372	0.3673488	9.33
1250	KLMS	1.472445	-0.818025	1.0907	0.54535	0.21814	1.832376	0.916188	0.3664752	9.07
	APA	1.508895	-0.838275	1.1177	0.55885	0.22354	1.877736	0.938868	0.3755472	11.77
	KAPA	1.460025	-0.811125	1.0815	0.54075	0.2163	1.81692	0.90846	0.363384	8.15
	LMS	1.43424	-0.7968	1.0624	0.5312	0.21248	1.784832	0.892416	0.3569664	6.24
	MFLMS	1.42776	-0.7932	1.0576	0.5288	0.21152	1.776768	0.888384	0.3553536	5.76
1500	KLMS	1.44423	-0.80235	1.0698	0.5349	0.21396	1.797264	0.898632	0.3594528	6.98
	APA	1.41507	-0.78615	1.0482	0.5241	0.20964	1.760976	0.880488	0.3521952	4.82
	KAPA	1.364715	-0.758175	1.0109	0.50545	0.20218	1.698312	0.849156	0.3396624	1.09
True values		1.35	-0.75	1.00	0.50	0.20	1.68	0.84	0.336	

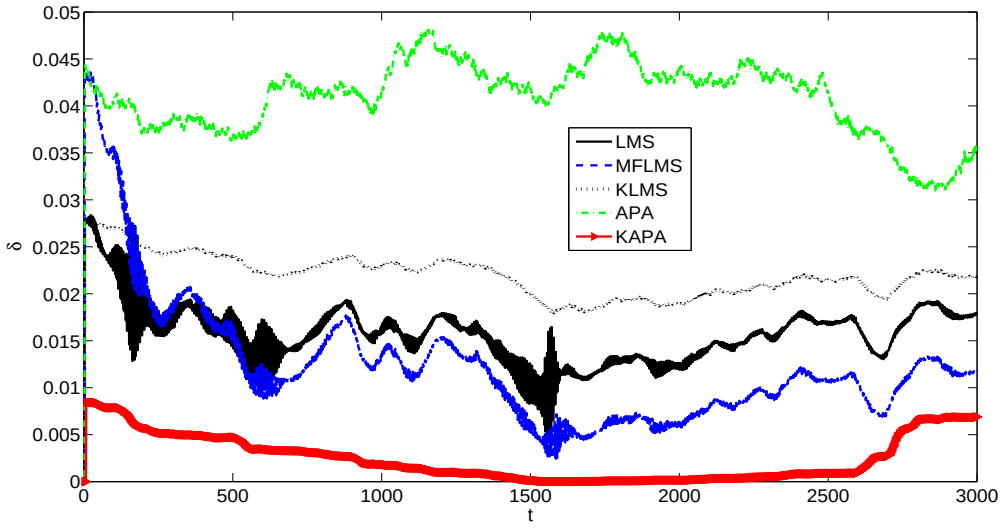

Figure 37: The parameter estimation error δ versus t with noise variance $\sigma_v^2 = 0.50$

Table 9: Parameter Estimates $\hat{\theta}$ with noise variance $\sigma_v^2 = 0.50^2$

t	**Algorithm**	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	$\delta(percentageerror)$
	LMS	1.897155	-1.053975	1.4053	0.70265	0.28106	2.360904	1.180452	0.4721808	40.53
	MFLMS	1.877175	-1.042875	1.3905	0.69525	0.2781	2.33604	1.16802	0.467208	39.05
500	KLMS	1.86435	-1.03575	1.381	0.6905	0.2762	2.32008	1.16004	0.464016	38.1
	APA	1.87866	-1.0437	1.3916	0.6958	0.27832	2.337888	1.168944	0.4675776	39.16
	KAPA	1.84815	-1.02675	1.369	0.6845	0.2738	2.29992	1.14996	0.459984	36.9
	LMS	1.75203	-0.97335	1.2978	0.6489	0.25956	2.180304	1.090152	0.4360608	29.78
	MLMS	1.7442	-0.969	1.292	0.646	0.2584	2.17056	1.08528	0.434112	29.2
1000	KLMS	1.757565	-0.976425	1.3019	0.65095	0.26038	2.187192	1.093596	0.4374384	30.19
	APA	1.769715	-0.983175	1.3109	0.65545	0.26218	2.202312	1.101156	0.4404624	31.09
	KAPA	1.716525	-0.953625	1.2715	0.63575	0.2543	2.13612	1.06806	0.427224	27.15
	LMS	1.599075	-0.888375	1.1845	0.59225	0.2369	1.98996	0.99498	0.397992	18.45
	MFLMS	1.580715	-0.878175	1.1709	0.58545	0.23418	1.967112	0.983556	0.3934224	17.09
1500	KLMS	1.5795	-0.8775	1.17	0.585	0.234	1.9656	0.9828	0.39312	17.89
	APA	1.59435	-0.88575	1.181	0.5905	0.2362	1.98408	0.99204	0.396816	18.1
	KAPA	1.57383	-0.87435	1.1658	0.5829	0.23316	1.958544	0.979272	0.3917088	16.58
	LMS	1.513215	-0.840675	1.1209	0.56045	0.22418	1.883112	0.941556	0.3766224	12.09
	MFLMS	1.50039	-0.83355	1.1114	0.5557	0.22228	1.867152	0.933576	0.3734304	11.14
2000	KLMS	1.49526	-0.8307	1.1076	0.5538	0.22152	1.860768	0.930384	0.3721536	10.76
	APA	1.48473	-0.82485	1.0998	0.5499	0.21996	1.847664	0.923832	0.3695328	9.98
	KAPA	1.451925	-0.806625	1.0755	0.53775	0.2151	1.80684	0.90342	0.361368	7.55
	LMS	1.41939	-0.78855	1.0514	0.5257	0.21028	1.766352	0.883176	0.3532704	5.14
	MFLMS	1.405755	-0.780975	1.0413	0.52065	0.20826	1.749384	0.874692	0.3498768	4.13
2500	KLMS	1.41426	-0.7857	1.0476	0.5238	0.20952	1.759968	0.879984	0.3519936	4.76
	APA	1.43073	-0.79485	1.0598	0.5299	0.21196	1.780464	0.890232	0.3560928	5.98
	KAPA	1.405215	-0.780675	1.0409	0.52045	0.20818	1.748712	0.874356	0.3497424	4.09
	LMS	1.40265	-0.77925	1.039	0.5195	0.2078	1.74552	0.87276	0.349104	3.9
	MFLMS	1.38726	-0.7707	1.0276	0.5138	0.20552	1.726368	0.863184	0.3452736	2.76
3000	KLMS	1.376865	-0.764925	1.0199	0.50995	0.20398	1.713432	0.856716	0.3426864	1.99
	APA	1.37376	-0.7632	1.0176	0.5088	0.20352	1.709568	0.854784	0.3419136	1.76
	KAPA	1.3634865	-0.7574925	1.00999	0.504995	0.201998	1.6967832	0.8483916	0.33935664	0.999
True values		1.35	-0.75	1.00	0.50	0.20	1.68	0.84	0.336	

In figure 37, parameter estimation error is displayed with noise variance $\sigma_v^2 = 0.50$. The step size parameters used in the proposed algorithm are 0.4 for LMS, $\beta = 0.5, \mu = 0.2, \mu_f = 0.3 \quad and \quad f = 0.392$ for MFLMS, 0.2 for KLMS, $\eta = 0.02 \quad and \quad \epsilon = 0.1$ for APA and $eta_t = 0.9 \quad and \quad \epsilon = 0.1$ for Kernel affine projection identification algorithm. In Table 9, the percentage error values in tabular form at different iteration time t with noise variance $\sigma_v^2 = 0.5^2$ is dispalyed.

4.3.1 Example 2

In the next example we take here another nonlinear CAR model. The numerical simulation has been carried out in the similar pattern as in the previous example.

$$\begin{aligned}
A(z)y(t) &= B(z)\bar{u}(t) + \nu(t) \\
A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} \;=\; 1 - 1.60z^{-2} + 0.80z^{-3} \\
B(z) &= b_1 z^{-1} + b_2 z^{-2} \;=\; z^{-1} + 0.65z^{-2} \\
\bar{u}(t) &= f(u(t)) = c_1 u(t) + c_2 u^2(t) + c_3 u^3(t) \\
&= u(t) + 0.50u^2(t) + 0.25u^3(t) \\
\theta &= [\theta_1,\ \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8]^T \\
\theta &= [a_1, a_2, c_1 b_1, c_1 b_2, c_2 b_1, c_2 b_2, c_3 b_1, c_3 b_2]^T \\
\theta &= [-1.60, 0.80, 1.00, 0.65, 0.50, 0.325, 0.250, 0.1625]^T
\end{aligned} \tag{4.52}$$

The proposed KAPA identification method is used to identify the parameters for the nonlinear Hammerstein CAR model in comparison with the MFLMS, LMS, KLMS, APA identification schemes using sufficiently large number of iterations, i-e, $t = 3000$ in Example 2. In Fig. 38, the parameter estimation error is shown with noise variance $\sigma_v^2 = 0.20^2$. The step size parameters used in this experiment for the adaptive identification algorithms are as 0.3 for LMS, $\beta = 0.5, \mu = 0.5, \mu_f = 0.09 \;\; and \;\; f = 0.392$ for MFLMS, 0.7 for KLMS, $\eta = 0.08 \;\; and \;\; \epsilon = 0.03$ for APA and $\eta_t = 0.6 \;\; and \;\; \epsilon = 0.1$ for Kernel affine projection identification algorithm. In Table 10, the percentage error is displayed of the estimated parameters at various iterations of time t with noise variance$\sigma_v^2 = 0.20^2$.

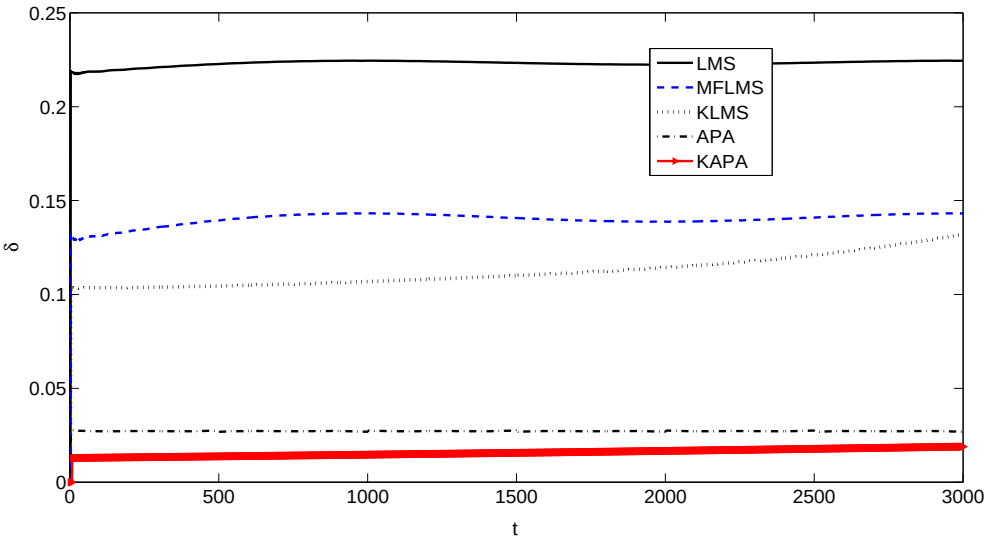

Figure 38: The parameter estimation error δ versus t with noise variance $\sigma_v^2 = 0.20$

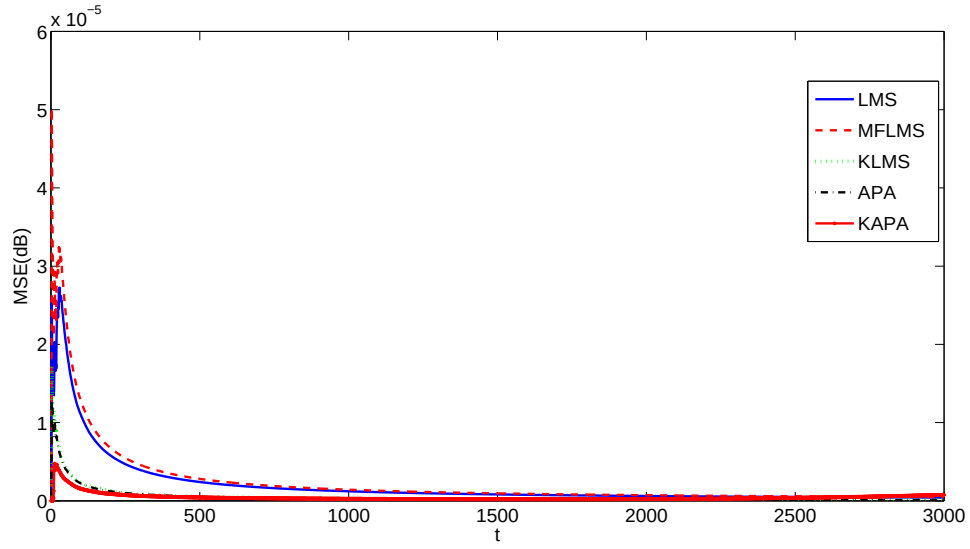

Figure 39: MSE of CAR model using estimated weights versus t with noise variance $\sigma_v^2 = 0.20$

Table 10: Parameter Estimates $\hat{\theta}$ of CAR model in Example 2 with noise variance $\sigma_v^2 = 0.20^2$

t	**Algorithm**	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	$\delta(percentageerror)$
	LMS	-2.24416	1.12208	1.4026	0.91169	0.7013	0.455845	0.35065	0.2279225	40.26
	MFLMS	-2.22256	1.11128	1.3891	0.902915	0.69455	0.4514575	0.347275	0.22572875	38.91
500	KLMS	-2.19728	1.09864	1.3733	0.892645	0.68665	0.4463225	0.343325	0.22316125	37.33
	APA	-2.2064	1.1032	1.379	0.89635	0.6895	0.448175	0.34475	0.2240875	37.9
	KAPA	-2.14576	1.07288	1.3411	0.871715	0.67055	0.4358575	0.335275	0.21792875	34.11
	LMS	-2.07296	1.03648	1.2956	0.84214	0.6478	0.42107	0.3239	0.210535	29.56
	MLMS	-2.06768	1.03384	1.2923	0.839995	0.64615	0.4199975	0.323075	0.20999875	29.23
1000	KLMS	-2.09216	1.04608	1.3076	0.84994	0.6538	0.42497	0.3269	0.212485	30.76
	APA	-2.0992	1.0496	1.312	0.8528	0.656	0.4264	0.328	0.2132	31.2
	KAPA	-2.01856	1.00928	1.2616	0.82004	0.6308	0.41002	0.3154	0.20501	26.16
	LMS	-1.84864	0.92432	1.1554	0.75101	0.5777	0.375505	0.28885	0.1877525	15.54
	MFLMS	-1.83616	0.91808	1.1476	0.74594	0.5738	0.37297	0.2869	0.186485	14.76
1500	KLMS	-1.83648	0.91824	1.1478	0.74607	0.5739	0.373035	0.28695	0.1865175	14.78
	APA	-1.8544	0.9272	1.159	0.75335	0.5795	0.376675	0.28975	0.1883375	15.9
	KAPA	-1.80624	0.90312	1.1289	0.733785	0.56445	0.3668925	0.282225	0.18344625	12.89
	LMS	-1.76688	0.88344	1.1043	0.717795	0.55215	0.3588975	0.276075	0.17944875	10.43
	MFLMS	-1.76	0.88	1.1	0.715	0.55	0.3575	0.275	0.17875	10
2000	KLMS	-1.75088	0.87544	1.0943	0.711295	0.54715	0.3556475	0.273575	0.17782375	9.43
	APA	-1.75968	0.87984	1.0998	0.71487	0.5499	0.357435	0.27495	0.1787175	9.98
	KAPA	-1.7312	0.8656	1.082	0.7033	0.541	0.35165	0.2705	0.175825	8.2
	LMS	-1.69776	0.84888	1.0611	0.689715	0.53055	0.3448575	0.265275	0.17242875	6.11
	MFLMS	-1.69568	0.84784	1.0598	0.68887	0.5299	0.344435	0.26495	0.1722175	5.98
2500	KLMS	-1.6832	0.8416	1.052	0.6838	0.526	0.3419	0.263	0.17095	5.2
	APA	-1.70688	0.85344	1.0668	0.69342	0.5334	0.34671	0.2667	0.173355	6.68
	KAPA	-1.6784	0.8392	1.049	0.68185	0.5245	0.340925	0.26225	0.1704625	4.9
	LMS	-1.6656	0.8328	1.041	0.67665	0.5205	0.338325	0.26025	0.1691625	4.1
	MFLMS	-1.67984	0.83992	1.0499	0.682435	0.52495	0.3412175	0.262475	0.17060875	4.99
3000	KLMS	-1.664	0.832	1.04	0.676	0.52	0.338	0.26	0.169	4
	APA	-1.6824	0.8412	1.0515	0.683475	0.52575	0.3417375	0.262875	0.17086875	5.15
	KAPA	-1.61616	0.80808	1.0101	0.656565	0.50505	0.3282825	0.252525	0.16414125	1.01
True values		-1.60	0.80	1.00	0.65	0.50	0.325	0.250	0.1625	

Another experiment is performed with noise variance $\sigma_v^2 = 0.50^2$ same as in Example 2, and the results are displayed in Fig. 40 and Figure 8. The stepsize parameters are adjusted as 0.07 for LMS, $\beta = 0.5,\ \mu = 0.9,\ \mu_f = 0.9\ \ and\ \ f = 0.392$ for MFLMS, 0.5 for KLMS, $\eta = 0.4\ \ and\ \ \epsilon = 0.01$ for APA and $\eta_t = 0.9\ \ and\ \ \epsilon = 0.1$ for Kernel affine projection identification algorithm. In table 11, the percentage error at different iterations of time t is presented for noise variance $\sigma_v^2 = 0.50^2$.

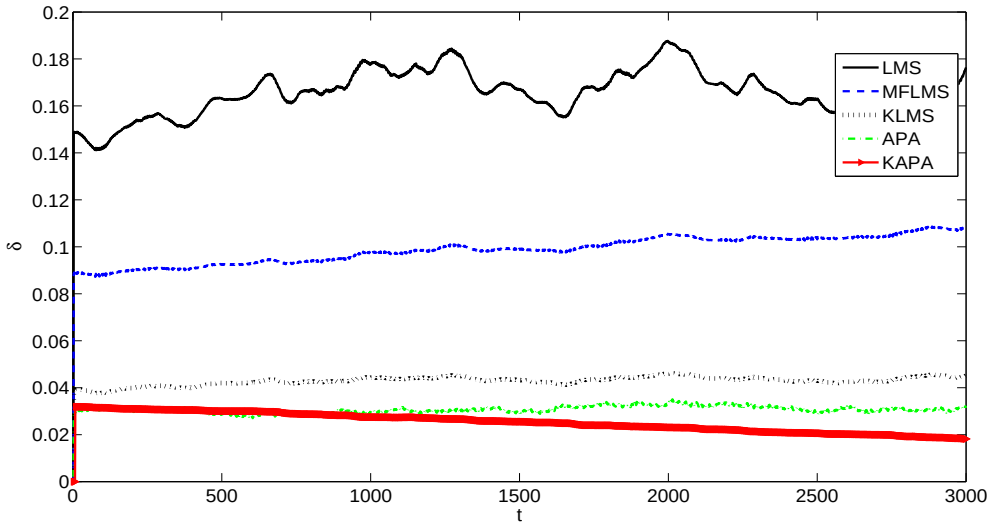

Figure 40: The parameter estimation error δ versus t with noise variance for example 2 $\sigma_v^2 = 0.50$

4.4 Conclusion

In this chapter, a new kernel fractional affine projection algorithm is presented. Affine projection and kernel affine projection algorithms has also been discussed. One application of predicting a chaotic . three dimensional lorenz system is presented, that demonstrates the performance of the proposed algorithm in comparison with LMS, APA, KAPA and in terms of mean square error as a figure of merit. Proposed algorithm is also tested on nonlinear channel equalization, Atmospheric Carbon dioxide forecasting and static function approximation. Parametric identification of Hammerstein Nonlinear CAR systems through kernel affine projection algorithm is another contribution

Table 11: Parameter Estimates $\hat{\theta}$ with noise variance $\sigma_v^2 = 0.50^2$ for example 2

t	**Algorithm**	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	δ
LMS	-0.94928	0.47464	0.5933	0.385645	0.29665	0.1928225	0.148325	0.09641125	40.67	
	MFLMS	-0.97632	0.48816	0.6102	0.39663	0.3051	0.198315	0.15255	0.0991575	38.98
500	KLMS	-0.9816	0.4908	0.6135	0.398775	0.30675	0.1993875	0.153375	0.09969375	38.65
	APA	-1.00656	0.50328	0.6291	0.408915	0.31455	0.2044575	0.157275	0.10222875	37.09
	KAPA	-1.00832	0.50416	0.6302	0.40963	0.3151	0.204815	0.15755	0.1024075	36.98
	LMS	-1.1664	0.5832	0.729	0.47385	0.3645	0.236925	0.18225	0.1184625	27.1
	MFLMS	-1.15232	0.57616	0.7202	0.46813	0.3601	0.234065	0.18005	0.1170325	27.98
1000	KLMS	-1.17344	0.58672	0.7334	0.47671	0.3667	0.238355	0.18335	0.1191775	26.66
	APA	-1.18608	0.59304	0.7413	0.481845	0.37065	0.2409225	0.185325	0.12046125	25.87
	KAPA	-1.21072	0.60536	0.7567	0.491855	0.37835	0.2459275	0.189175	0.12296375	24.33
	LMS	-1.35856	0.67928	0.8491	0.551915	0.42455	0.2759575	0.212275	0.13797875	15.09
	MFLMS	-1.36176	0.68088	0.8511	0.553215	0.42555	0.2766075	0.212775	0.13830375	14.89
1500	KLMS	-1.36368	0.68184	0.8523	0.553995	0.42615	0.2769975	0.213075	0.13849875	14.77
	APA	-1.35696	0.67848	0.8481	0.551265	0.42405	0.2756325	0.212025	0.13781625	15.19
	KAPA	-1.39152	0.69576	0.8697	0.565305	0.43485	0.2826525	0.217425	0.14132625	13.03
	LMS	-1.4384	0.7192	0.899	0.58435	0.4495	0.292175	0.22475	0.1460875	10.1
	MFLMS	-1.42128	0.71064	0.8883	0.577395	0.44415	0.2886975	0.222075	0.14434875	11.17
2000	KLMS	-1.41184	0.70592	0.8824	0.57356	0.4412	0.28678	0.2206	0.14339	11.76
	APA	-1.40832	0.70416	0.8802	0.57213	0.4401	0.286065	0.22005	0.1430325	11.98
	KAPA	-1.432	0.716	0.895	0.58175	0.4475	0.290875	0.22375	0.1454375	10.5
	LMS	-1.50416	0.75208	0.9401	0.611065	0.47005	0.3055325	0.235025	0.15276625	5.99
	MFLMS	-1.50928	0.75464	0.9433	0.613145	0.47165	0.3065725	0.235825	0.15328625	5.67‘
2500	KLMS	-1.49712	0.74856	0.9357	0.608205	0.46785	0.3041025	0.233925	0.15205125	6.43
	APA	-1.48592	0.74296	0.9287	0.603655	0.46435	0.3018275	0.232175	0.15091375	7.13
	KAPA	-1.51712	0.75856	0.9482	0.61633	0.4741	0.308165	0.23705	0.1540825	5.18
	LMS	-1.53632	0.76816	0.9602	0.62413	0.4801	0.312065	0.24005	0.1560325	3.98
	MFLMS	-1.55216	0.77608	0.9701	0.630565	0.48505	0.3152825	0.242525	0.15764125	2.99
3000	KLMS	-1.54496	0.77248	0.9656	0.62764	0.4828	0.31382	0.2414	0.15691	3.44
	APA	-1.54896	0.77448	0.9681	0.629265	0.48405	0.3146325	0.242025	0.15731625	3.19
	KAPA	-1.58288	0.79144	0.9893	0.643045	0.49465	0.3215225	0.247325	0.16076125	1.07
True values		-1.60	0.80	1.00	0.65	0.50	0.325	0.250	0.1625	

that is presented in the current chapter. The experimental analysis of the proposed algorithm can give the consistent parameter estimates. Also the proposed method is validated in comparison with LMS, FLMS, MFLMS and Affine Projection algorithms. Future guidelines leads towards the new identification algorithms using kernel methods.

5 A Square Root Extended Kernel Recursive Least Squares Algorithm

5.1 Problem Statement

The problem in applying the EKRLS algorithm is that of numerical instability (divergence) that can arise due to the way in which Ricatti equation is formulated. To encounter this divergence problem, we introduce a new square root variant of EKRLS. The positive definite character of the state error correlation matrix of EKRLS is preserved by using the square root of the correlation matrix for updating and using the fact that any square matrix and its hermitian transpose is always positive definite. The respective SREKRLS algorithm uses Givens rotation for updating process. Thus it gives considerable improvement in numerical accuracy, time and computational complexity as compared to EKRLS. Moreover, it is suitable for parallel implementation. A prearray of numbers has to be triangularized by a rotation or sequence of rotation in order to yield another set of numbers (post array). The quantities needed for the next prearray can then be read off from the current entries of the prearray, and the same procedure is repeated. The effectiveness of the proposed algorithm is tested on the prediction of nonlinear Mackey glass time series prediction for stationary, as well as, nonstationary case. Moreover, we have used the algorithm for prediction of x-component of the three dimension Lorenz series. The performance of the proposed algorithm is

compared with that of EKRLS algorithm.

5.2 Extended Recursive Least Squares Algorithm (ERLS).

A more general linear state space model for sequential estimation containing process equation and measurement equation is given below

$$\begin{aligned} \mathbf{x}(i) &= \mathbf{A}\mathbf{x}(i+1) + \mathbf{n}(i) \\ d(i) &= \mathbf{u}(i)^T\mathbf{x}(i) + \nu(i) \end{aligned}$$

Here **A** is the state transition matrix, **n**(i) is the state noise, and v(i) is the observation or process noise. Both the noises are white and Gaussian distributed. The Extended RLS algorithm is summarized as follows

Algorithm 3 Extended Recursive Least squares algorithm

For i=1,2,3,4..... compute
Step 1: Input parameters of the Algorithm
Require: Input data matrix :$\mathbf{U} = [\mathbf{u}(1), \mathbf{u}(2), \ldots\ldots\ldots, \mathbf{u}(k)]$
Require: Desired data vector :$\mathbf{d}(i) = [d(1), d(2), \ldots\ldots\ldots, d(i)]$
q_1 is the covariance of process noise
q_2 is the covariance of measurement noise
Prescribed Parameters:
Forgetting Factor=β
Initialize
$\mathbf{w}(0) = 0$
$\mathbf{P}(0) = \lambda\mathbf{I}, \mathbf{A} = \alpha\mathbf{I}$
Compute for $i \geq 1$

$$\begin{aligned} e(i) &= d(i) - \mathbf{u}(i)^T\mathbf{w}(i-1) \\ \mathbf{k}(i) &= \mathbf{A}\mathbf{P}(i-1)\mathbf{u}(i)/r(i) \\ \mathbf{r}(i) &= q_2 + \mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i) \\ \mathbf{w}(i) &= \mathbf{A}\mathbf{w}(i-1) + \mathbf{k}(i)e(i) \\ \mathbf{P}(i) &= \mathbf{A}\mathbf{P}(i-1)\mathbf{A}^T - \mathbf{k}(i)\mathbf{k}^T(i)r(i) + q_1\mathbf{I} \end{aligned}$$

5.3 Extended Kernel Recursive Least Squares Algorithm

Starting from the special unforced dynamical nonlinear state space model (5.1),

$$\begin{aligned} \mathbf{x}(i) &= \mathbf{x}\mathbf{x}(i+1) \qquad ProcessEquation \\ d(i) &= \varphi(i)^T\mathbf{x}(i) + \nu(i) \qquad MeasurementEquation \end{aligned} \tag{5.1}$$

Here the measurement noise is white with zero mean. There is no process noise and so it is unforced dynamical model. By keeping in view the above model, the recursive equations of the EKRLS will be written as

$$\begin{aligned} r(i) &= \beta^i + \varphi(i)^T\mathbf{P}(i-1)\varphi(i) \\ \mathbf{k}(i) &= \mathbf{A}\mathbf{P}(i-1)\varphi(i)/r(i) \\ (i) &= d(i) - \varphi(i)^T\mathbf{w}(i-1) \\ \mathbf{w}(i) &= \mathbf{A}\mathbf{w}(i-1) + \mathbf{k}(i)e(i) \\ \mathbf{P}(i) &= \mathbf{A}[\mathbf{P}(i-1) - \mathbf{P}(i-1)\varphi(i)\varphi(i)^T\mathbf{P}(i-1)/r(i)] \end{aligned} \tag{5.2}$$

5.4 Proposed Square Root Extended kernel Recursive Least squares algorithm

We are talking the same unforced dynamical model as taken for EKRLS. The important terms are arranged in the matrix form which includes the following terms as

1. A scaler term $\beta^i + \varphi(i)^T\mathbf{P}(i-1)\varphi(i)$

2. A vector term of length $1 \times n$ $\varphi(i)^T\mathbf{P}(i-1)\mathbf{A}$

3. A vector term of length $n \times 1$ $\mathbf{A}\mathbf{P}(i-1)\varphi(i)$

4. A matrix of dimension $n \times n$ $\mathbf{AP}(i-1)\mathbf{A}$

These four terms contain the complete information of the Ricatti difference equation 5.3, we may arrange these terms in the matrix form by keeping in view the dimensionality of each of these terms, which in the combined form will be written as

$$\left[\begin{array}{c} \mathbf{H}(i) \end{array} \right] = \left[\begin{array}{ll} \beta^i + \varphi(i)^T \mathbf{P}(i-1)\varphi(i) & \varphi(i)^T \mathbf{P}(i-1)\mathbf{A} \\ \mathbf{AP}(i-1)\varphi(i) & \mathbf{AP}(i-1)\mathbf{A} \end{array} \right] \tag{5.3}$$

Its factored form yields

$$\left[\begin{array}{ll} \beta^{i/2} & \varphi(i)^T \mathbf{P}^{1/2}(i-1) \\ \mathbf{0} & \mathbf{AP}^{1/2}(i-1) \end{array} \right] \left[\begin{array}{ll} \beta^{i/2} & \mathbf{0}^T \\ \mathbf{P}^{T/2}(i-1)\varphi(i) & \mathbf{P}^{T/2}(i-1)\mathbf{A}^T \end{array} \right] \tag{5.4}$$

$$\mathbf{H}(i) = \mathbf{H}^{1/2}(i)\mathbf{H}^{T/2}(i) \tag{5.5}$$

The right side of 5.5 is interpreted as the product of a matrix with its transposed form. Using matrix factorization Lemma, which states for real case that given two $\mathbf{N}$ by $\mathbf{M}$ matrices $\mathbf{A}$ and $\mathbf{B}$ with dimension $\mathbf{N} \leq \mathbf{M}$, $\mathbf{AA}^H = \mathbf{BB}^H$if and only if their exists an orthogonal matrix $\mathbf{\Theta}$ such that $\mathbf{B} = \mathbf{A\Theta}$. In our case, $\mathbf{A} = \mathbf{H}^{1/2}(i)$. Thus we get

$$\left[\begin{array}{ll} \beta^{i/2} & \varphi(i)^T \mathbf{P}^{1/2}(i-1) \\ \mathbf{0} & \mathbf{AP}^{1/2}(i-1) \end{array} \right] \mathbf{\Theta} = \left[\begin{array}{ll} b_{11}(i) & \mathbf{0}^T \\ \mathbf{b}_{21}(i) & \mathbf{B}_{21}(i) \end{array} \right] \tag{5.6}$$

The scalar $b_{11}(i)$, the vector term $\mathbf{b}_{21}(i)$ and the matrix $\mathbf{B_{22}(i)}$ constitute the nonzero block elements of the matrix, say $\mathbf{B}$, that forms as the result of operation by orthogonal rotation matrix $\theta(i)$. To evaluate the unknown elements in 5.6, we proceed in a way by squaring both sides of the above

equation.

$$\begin{bmatrix} \beta^{i/2} & \varphi(i)^T\mathbf{P}^{1/2}(i-1) \\ \mathbf{0} & \mathbf{A}\mathbf{P}^{1/2}(i-1) \end{bmatrix} \begin{bmatrix} \beta^{i/2} & \mathbf{0}^T \\ \mathbf{P}^{T/2}(i-1)\varphi(i) & \mathbf{P}^{T/2}(i-1)\mathbf{A}^T \end{bmatrix} \begin{bmatrix} b_{11}(i) & \mathbf{0}^T \\ \mathbf{b}_{21}(i) & \mathbf{B}_{22}(i) \end{bmatrix} \begin{bmatrix} b_{11}(i) & \mathbf{0}^T\mathbf{b}_{21}(i)^T \\ \mathbf{0} & \mathbf{B}_{22}(i)^T \end{bmatrix} \tag{5.7}$$

By recognizing the fact that $\theta(i)$ is an orthogonal matrix and therefore its product $\theta(i)\theta(i)^T$ equals unity. And the rest of the terms will be formulated as follows

$$\begin{aligned} |b_{11}(i)|^2 &= \beta^{i/2}\beta^{i/2} + \varphi(i)^T\mathbf{P}^{1/2}(i-1)\mathbf{P}^{T/2}(i-1)\varphi(i) \\ |b_{11}(i)|^2 &= r(i) \end{aligned} \tag{5.8}$$

$$\mathbf{b}_{21}b_{11}(i) = \mathbf{A}\mathbf{p}^{1/2}(i-1)\varphi(i) \tag{5.9}$$

$$\begin{aligned} \mathbf{B}_{22}\mathbf{B}_{22}(i)^T &= \mathbf{A}\mathbf{p}(i-1)\mathbf{A}^T - \mathbf{b}_{21}\mathbf{b}_{21}^T \\ \mathbf{B}_{22}\mathbf{B}_{22}(i)^T &= \mathbf{A}\mathbf{p}(i-1)\mathbf{A}^T \\ &- (\mathbf{A}\mathbf{p}^{1/2}(i-1)\varphi(i)\varphi(i)^{T/2}(i-1)\mathbf{A}^T)/r^{1/2}(i)r^{1/2}(i) \end{aligned} \tag{5.10}$$

By satisfying the above equations

$$b_{11}(i)^T = r(i)^{1/2} \tag{5.11}$$

$$\mathbf{b}_{21}(i) = \mathbf{k}(i)r(i)^{1/2} \tag{5.12}$$

$$\mathbf{B}_{22}(i) = \mathbf{p}^{1/2}(i) \tag{5.13}$$

The term $r(i)^{1/2}$ is the variance with a power half. The second equation gives the gain along with the variance term. The third and the last term gives the Chollesky factor of the state error

correlation matrix.

$$\begin{bmatrix} \beta^{i/2} & \varphi(i)^T \mathbf{P}^{1/2}(i-1) \\ \mathbf{0} & \mathbf{A}\mathbf{P}^{1/2}(i-1) \end{bmatrix} \mathbf{\Theta}_\theta = \begin{bmatrix} r(i)^{1/2} & \mathbf{0}^T \\ \mathbf{k}(i) r(i)^{1/2} & \mathbf{P}^{1/2}(i) \end{bmatrix} \tag{5.14}$$

The elements $\beta^{1/2}$ and **0** in the prearray induces the creation of two variables: the variance r(i) and the gain $\mathbf{k}(i)$. These two variables are obtained simply by squaring and dividing $r^{1/2}(i)$, respectively. The set of equations (5.6) through (5.14) give the explicit representation of the SREKRLS. The true form of the rotation matrix Θ is like an identity matrix except for four points, where the pair of columns k and l will intersect with the pair of rows k and l, with diagonal points as cosines and off diagonal as sine and minus sign.

$$\prod_{k=0}^{l} \Theta(kl) \tag{5.15}$$

Detailed summary of the algorithm is given in Algorithm 4, expresses the computation from prearray to pasture

5.5 Simulations and Results

In this section we present experimental results of prediction of Mackey glass time series and x-component of the three dimension Lorenz series prediction using the proposed square root Extended kernel recursive least squares algorithm.

5.5.1 Mackey Glass Time series prediction.

In this experiment we consider stationary series and secondly a nonstationary time series for the prediction of Mackey Glass time series prediction.The Mackey glass time series is generated by

Algorithm 4 Square Root Extended Kernel Recursive Least Squares (SREKRLS) Algorithm (unforced dynamical model).

For k=1,2,3,4..... compute

Step 1: Input parameters of the Algorithm

Require: Input data matrix :$\mathbf{\Phi} = [\varphi(1), \varphi(2), \ldots\ldots\ldots, \varphi(k)]$

Require: Desired data vector :$\mathbf{d}(k) = [d(1), d(2), \ldots\ldots\ldots, d(k)]$

Prescribed Parameters:

Forgetting Factor=β

Initialize

$\mathbf{w}(0) = 0$

$\mathbf{P}(0) = \lambda\mathbf{I}, \mathbf{A} = \alpha\mathbf{I}$

Step 2: Obtaining Θ_θ(k) for updating

$$\begin{bmatrix} \beta^{i/2} & \varphi(i)^T\mathbf{P}^{1/2}(i-1) \\ \mathbf{0} & \mathbf{A}\mathbf{P}^{1/2}(i-1) \end{bmatrix} \mathbf{\Theta} = \begin{bmatrix} b_{11}(i) & \mathbf{0}^T \\ \mathbf{b}_{21}(i) & \mathbf{B}_{21}(i) \end{bmatrix}$$

Step 3: Obtaining $\mathbf{w}(i), e(i) and \mathbf{k}(i)$

$e(i) = d(i) - \varphi(i)^T\mathbf{w}(i-1)$

$\mathbf{k}(i) = (\mathbf{k}(i)r(i)^{1/2})r(i)^{1/2}$

$\mathbf{w}(i) = \mathbf{A}\mathbf{w}(i-1) + \mathbf{k}(i)e(i)$

the following delay differential equation.

$$\frac{dx(t)}{dt} = -bx(t) + \frac{ax(t-\tau)}{1 + x^{10}(t-\tau)} \tag{5.15}$$

With b=0.1,a=0.2 and $\tau = 20$, initial conditions $x(t-\tau) = 0$ for $0 \leq t \leq \tau$. So we have a series x(t) for t=1,2,3,.....,5000, obtained by the above equation. It is achieved by sampling the continuous curve x(t) with the interval of 1 second. The cost function for the time series prediction is given as

$$e = \| \ y_{des} - y_{obt} \ \|^2 \tag{5.16}$$

Here $y_{des} = \mathbf{w}^T\mathbf{x}_{tr}$, $\mathbf{w}$ and $\mathbf{x}_{tr}$ are the trained weights and the training samples respectively. $y_{obt} = \mathbf{w}^T\mathbf{x}_{te}$, $\mathbf{w}$ are the trained weights and $\mathbf{x}_{te}$ are the testing samples.

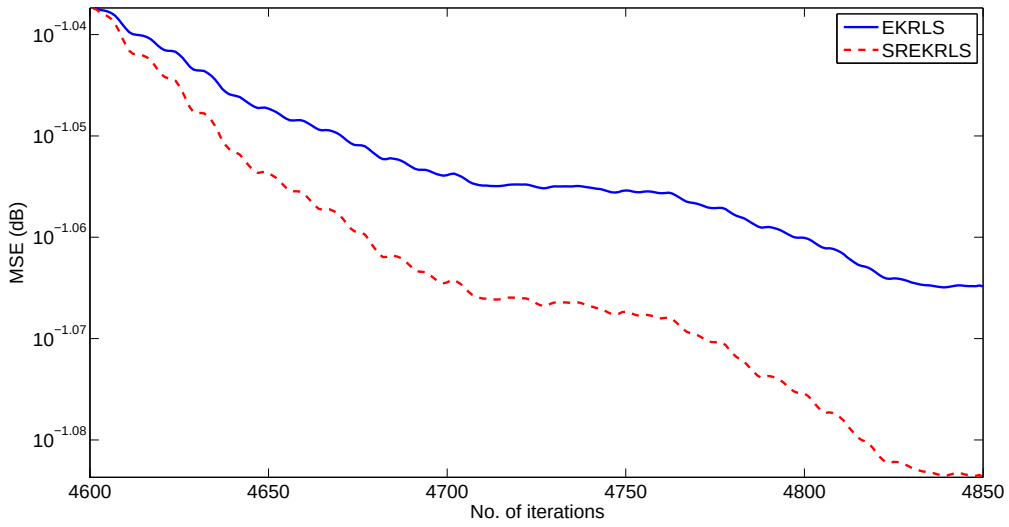

Figure 41: MSE curve of EKRLS and SREKRLS for stationary Mackey Glass Time series Prediction with noise variance 0.09.

Stationary series prediction.

The order of the filter is taken as 10 in this experiment, I-e. $x(t) = [x(t-10), x(t-9), x(t-8), , x(t-1)]^T$previous 10 sample points are consider as the input to predict the present one $x(t)$. 3000 sample points are used for training while the next 250 samples are used for testing the proposed algorithm along with EKRLS algorithm. The proposed algorithms performance is examined while training on the stationary series with sample points 1501-4500 and another sample points 4600-4850 is used for testing purpose. The weights are adjusted during training and then these trained weights are used for testing, the learning curves in terms of mean square error are generated to validate the proposed algorithm. In order to further validate the performance of the proposed algorithm noise is added to the time series with different variances. 500 Monte Carlo simulations are run to adjust the weights of the algorithm and then these weights are used to predict the test samples. The additive noise with zero mean having a variance of 0.09 is added to the series and the results are displayed in Fig. 41. The parameters of the algorithms used are adjusted as EKRLS $\mathbf{A} = \alpha\mathbf{I}, \epsilon = 0.01, \mathbf{P}(0) = \lambda\mathbf{I} with \mathbf{A}\lambda = 1/\epsilon, \beta = 0.01795, \alpha = 0.3$ and Gaussian kernel with

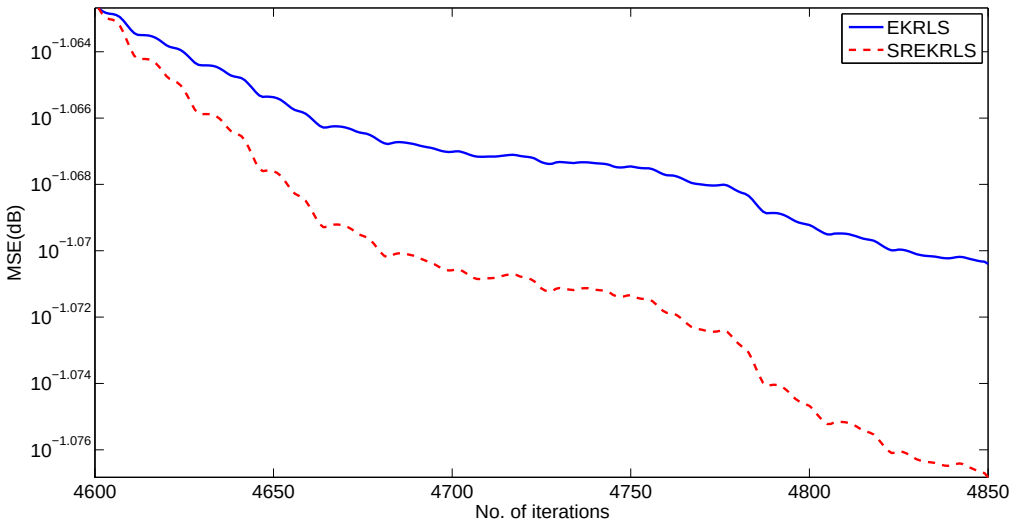

Figure 42: MSE curve of EKRLS and SREKRLS for stationary Mackey Glass Time series Prediction with noise variance 0.07.

a width of 0.2, and the proposed SREKRLS $\mathbf{A} = \alpha\mathbf{I}, \epsilon = 0.01, \mathbf{P}(0) = \lambda\mathbf{I} with \lambda = 1/\epsilon, \beta = 0.01795, \alpha = 0.3$and Gaussian kernel with a width of 0.3.

The AWGN noise with a variance of 0.07 is added to the original series and the prediction performance of the proposed algorithm is improved by 0.01 dB as shown in Fig. 42.

In order to further validate the performance of the proposed algorithm, the noise variance is increased to 0.2 and results are shown in Fig. 43, improvement of 0.07 is dB is observed from the EKRLS algorithm. The results shows that the proposed algorithm is more robust against noise. The results shown in Fig. 43 and Fig. 44 also shows the improvement of 0.04 and 0.05 dB with the noise noise variance of 0.2 and 0.1 respectively.

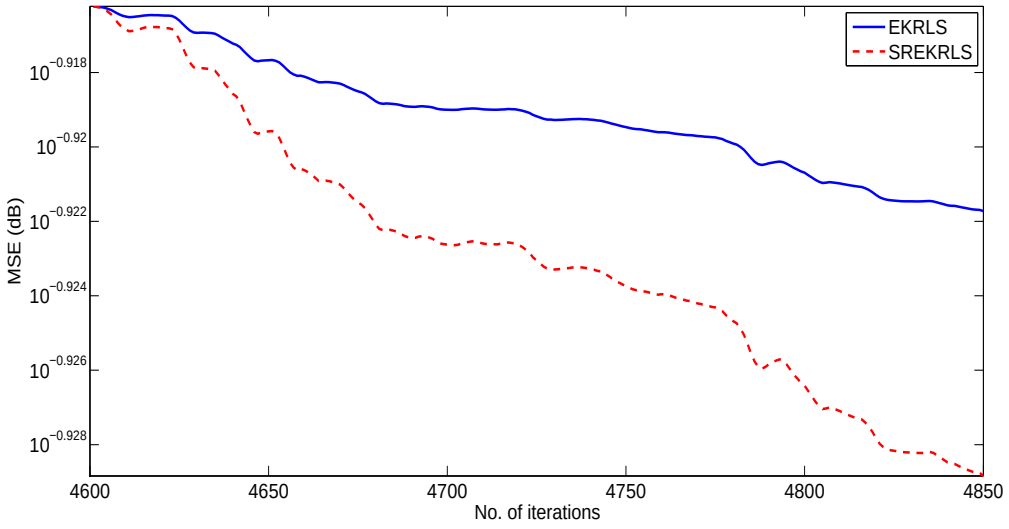

Figure 43: MSE curve of EKRLS and SREKRLS for stationary Mackey Glass Time series Prediction with noise variance 0.2.

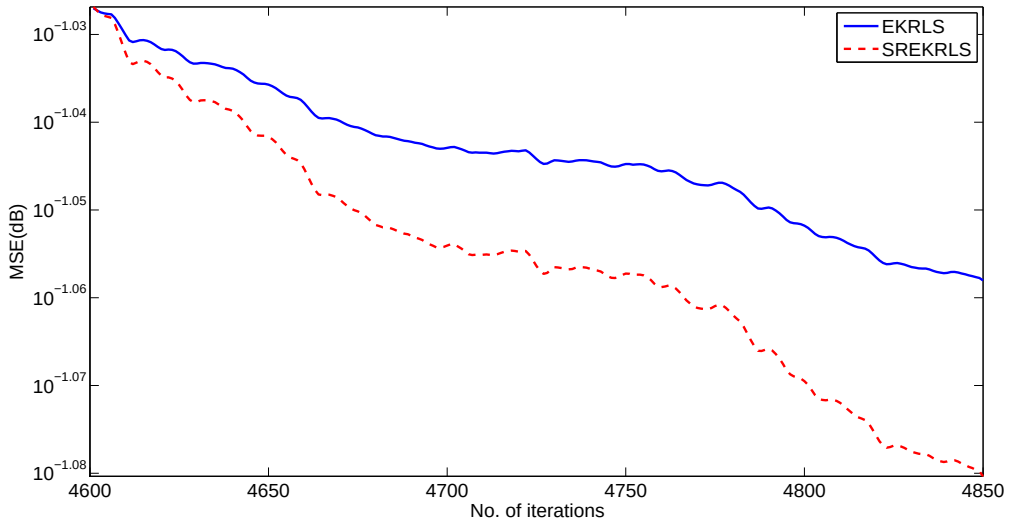

Figure 44: MSE curve of EKRLS and SREKRLS for stationary Mackey Glass Time series Prediction with noise variance 0.1.

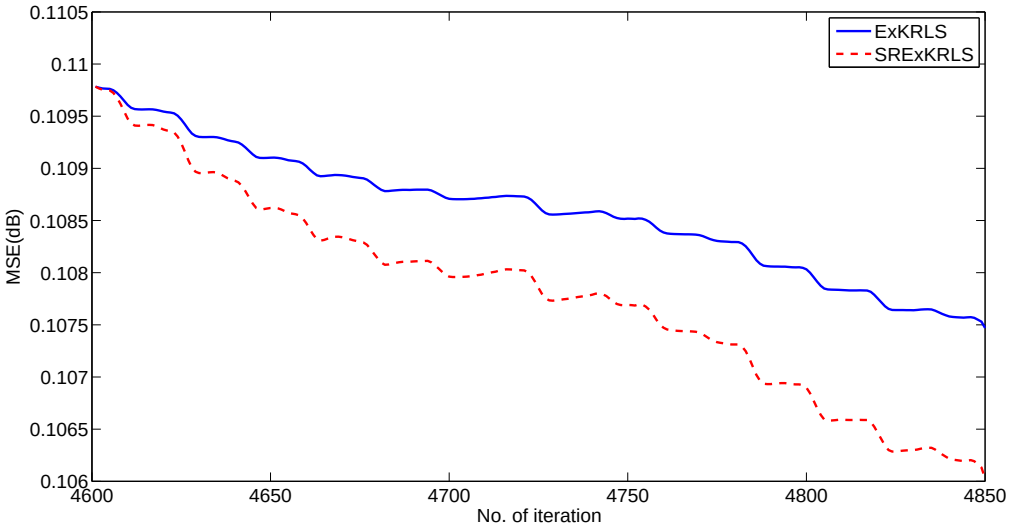

Figure 45: MSE curve of EKRLS and SREKRLS for nonstationary Mackey Glass Time series Prediction with noise variance 0.09.

NonStationary Time Series Prediction.

To further examine and validate the behavior of the proposed algorithm with EKRLS, A nonstationary series is generated by adding a sinusoid $(y = 0.3\sin(2ft), f = 1/5000)$ having an amplitude of 0.3 with a frequency of 5000 samples per second is added to the original mackey glass time series. All the parameters and samples are taken same as in part a (Stationary series).

In order to validate the performance of the proposed algorithm noise is added with different variance to the nonstationary time series and the results are displayed in Fig. 45, Fig. 46, Fig. 47 and fig. 48 respectively. In 45, the performance of the proposed SRExKRLS algorithm is validated for the prediction nonstationary time series in comparison with Extended kernel recursive least squares algorithm. The AWGN noise with a variance of 0.09 is also added to the nonstationary series.

The results of MSE is shown in 46 for the nonstationary time series prediction. The improvement of 0.25dB is also observed. The result is obtained with the addition of AWGN noise having a

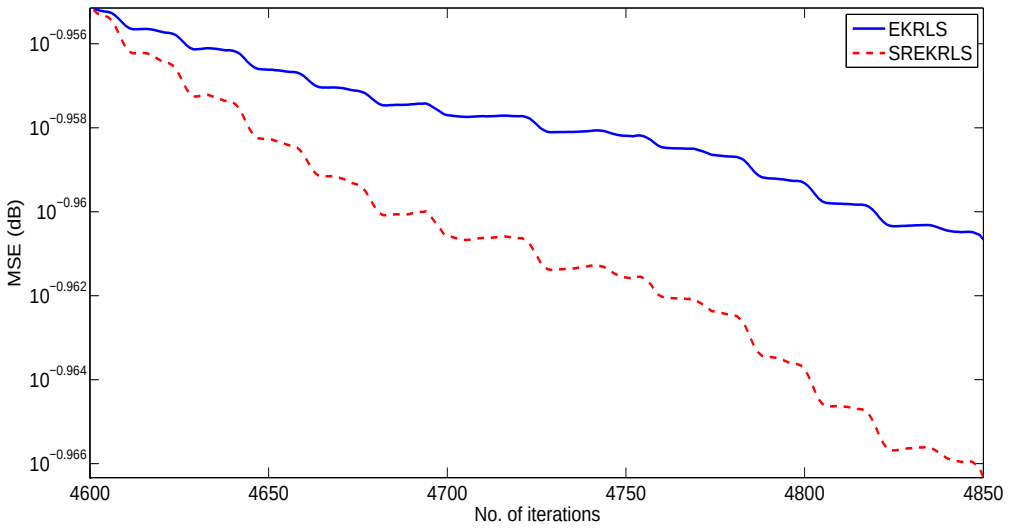

Figure 46: MSE curve of EKRLS and SREKRLS for nonstationary Mackey Glass Time series Prediction with noise variance 0.07.

variance of 0.07.

In 47 the results are further shown with the increase in the noise variance. The obtained results clearly observed that the performance of ExKRLS is still better.

The result in 48 shows the comparison of EKRLS and SREKRLS for the nonstationary time series prediction with the addition of noise having variance of 0.1.

5.5.2 Lorenz Time Series Prediction.

Lorenz series is a three dimensional, nonlinear and deterministic equation, it is expressed as the set of following partial differential equations

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y(t) - x(t)) \\ \frac{dy}{dt} &= -x(t)z(t) + \gamma x(t) - y(t) \\ \frac{dz}{dt} &= x(t)y(t) - Bz(t) \end{aligned} \tag{5.17}$$

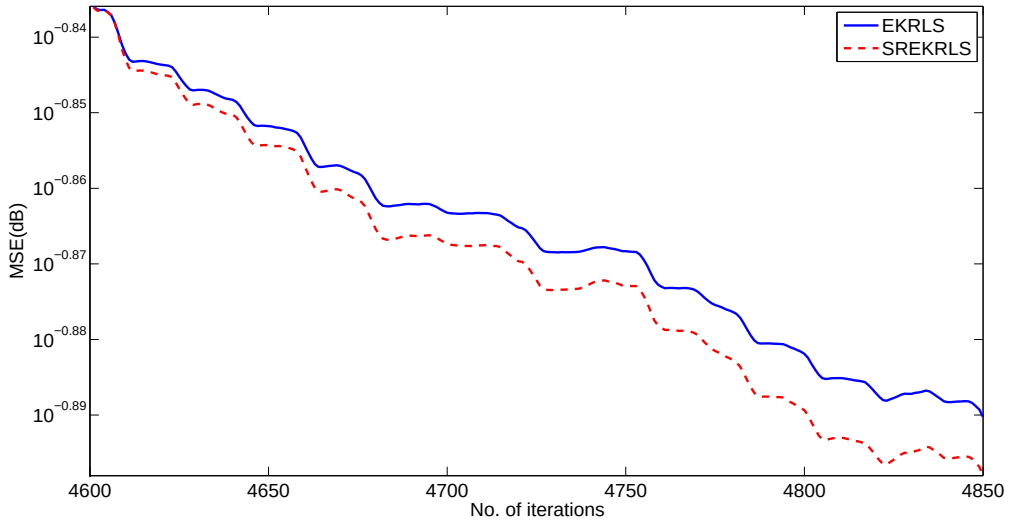

Figure 47: MSE curve of EKRLS and SREKRLS for nonstationary Mackey Glass Time series Prediction with noise variance 0.2.

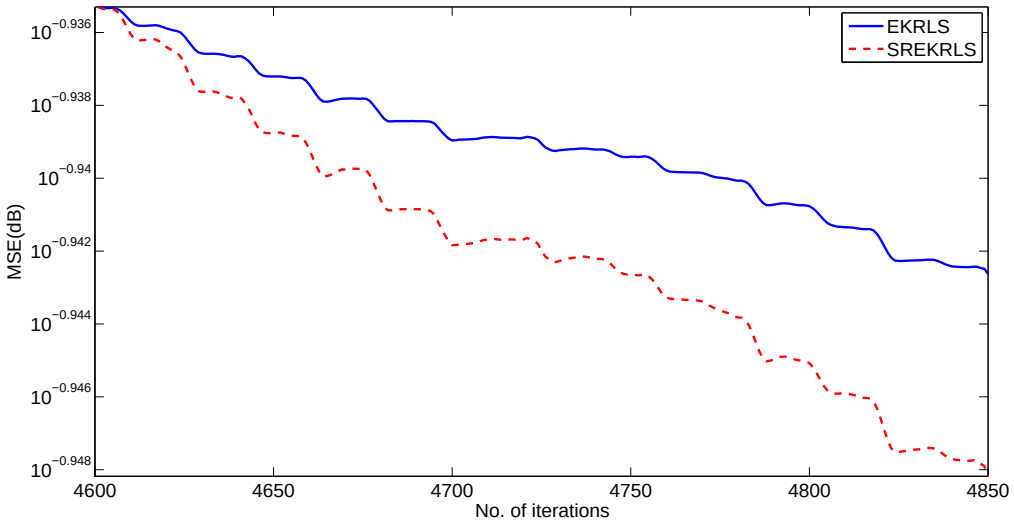

Figure 48: MSE curve of EKRLS and SREKRLS for nonstationary Mackey Glass Time series Prediction with noise variance 0.1.

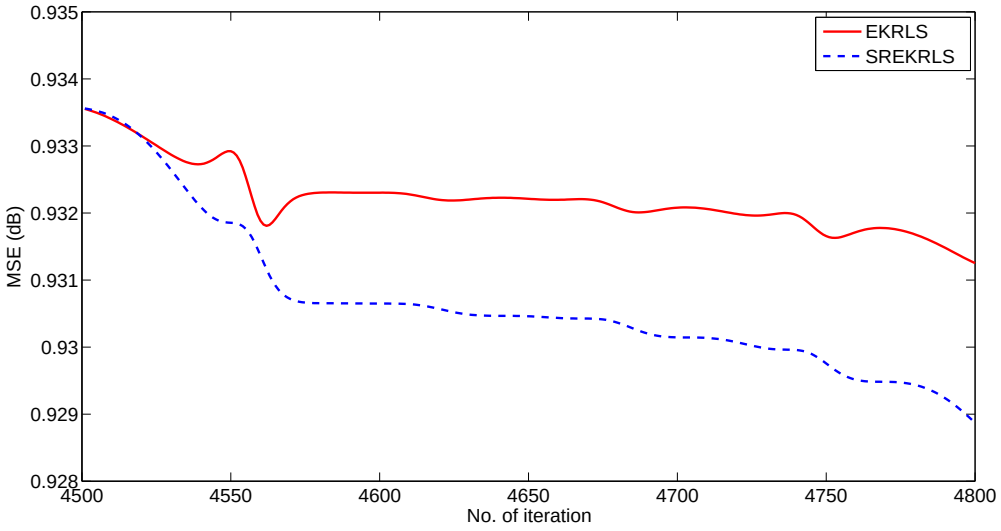

Figure 49: MSE curve of EKRLS and SREKRLS for X-component of Lorenz Time series Prediction without noise.

The Lorenz chaotic behavior on the parameters, $\sigma = 10$, $\gamma = 28$ and $B = \frac{8}{3}$. By using first order approximation method with a sampling period of 0.001 seconds and the initial values are set as x(0)=1, y(0)=1, z(0)=1. This forms a three dimension Lorenz Series. Here we consider only the x-component of the Lorenz series and generate a train set of 5000 samples I-e, 1500-4500 for training the proposed algorithm and its counterpart while the next 300 symbols are used as the test set. The learning curves in terms of mean square error is observed and displayed in Figure 49 without noise. The order of the filter is chosen as 3, I-e previous 3 samples are used to predict the current one. The parameters of the algorithms used as EKRLS $\mathbf{A} = \alpha\mathbf{I}, \epsilon = 0.01, \mathbf{P}(0) = \lambda\mathbf{I} with \mathbf{A}\lambda = 1/\epsilon, \beta = 0.01795, \alpha = 0.3$ and Gaussian kernel with a width of 0.2, and the proposed SREKRLS $\mathbf{A} = \alpha\mathbf{I}, \epsilon = 0.01, \mathbf{P}(0) = \lambda\mathbf{I} with \lambda = 1/\epsilon, \beta = 0.01795, \alpha = 0.3$ with Gaussian kernel having a width of 0.4.

In Fig. 49, the performance of the proposed algorithm is tested without the addition of noise

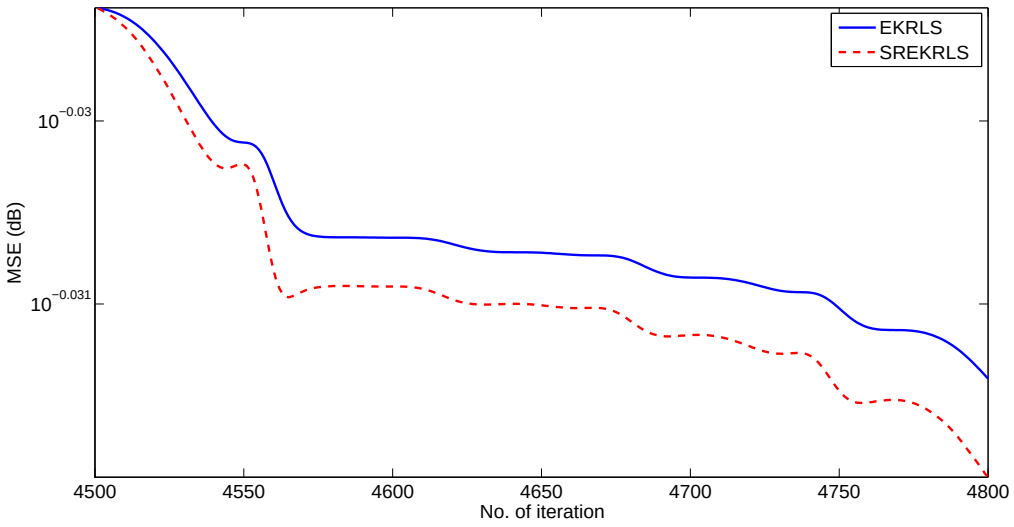

Figure 50: MSE curve of EKRLS and SREKRLS for X-component of Lorenz Time series Prediction with noise variance 0.09.

and the improvement of 0.02 dB is observed from EKRLS algorithm. The noise is further added to the x-component of the Lorenz series with a variance of 0.09 and the result is displayed in Fig. 50.

Fig. 50 and Fig. 51 shows the results of the performance of the proposed algorithm in the presence of noise that is added to series generated to the x-component of the Lorenz series having noise variances of 0.09 and 0.9 respectively.

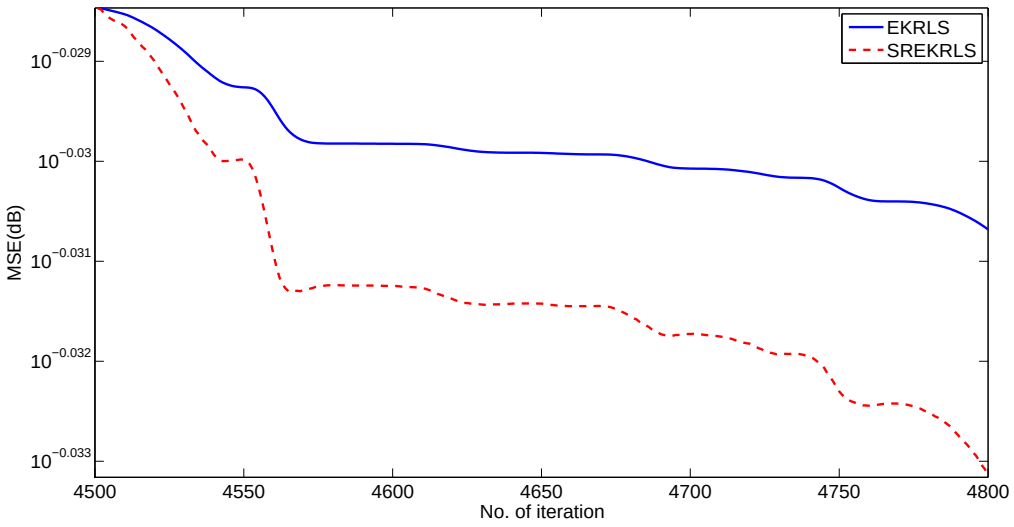

Figure 51: MSE curve of EKRLS and SREKRLS for X-component of Lorenz Time series Prediction with noise variance 0.9.

5.6 Conclusion

In this chapter of the dissertation, a new square root extended kernel recursive least square algorithm is proposed. Extended kernel recursive least square algorithm is also discussed. One application of prediction of stationary as well as nonstationary Mackey Glass time series is presented and also along with different noise levels. The learning curves in term of mean square error are observed, that demonstrates the proposed algorithm achieve better prediction performance, in comparison with EKRLS. Another experiments for the prediction of x-component of Lorenz time series is performed to further test the validity of the proposed algorithm. This new formulation (SREKRLS) hopefully gives a new era towards nonlinear systolic arrays that are useful in making VLSI applications using nonlinear input data.

6 Conclusion and Future Work

In this section of the dissertation, the conclusion is presented on the basis of previous chapters.

A successful design methodology and mathematical model for the development of modified fractional least mean square algorithm has been given using the strength of fractional signal processing. The algorithm has been made problem dependent. If there is no uncertain change in the nature of the problem, the dominance will be given to the LMS, while in the other case the dominance will be given to the Fractional LMS. Prediction problem setting for the nonlinear and non stationary time series is also presented. The weights are trained and then test for the prediction problem. Also the comparison is made with different noise variances and the results are shown in tabular form also.

Next the second modification is taken place by making adaptive the step sizes and the order of the fractional derivative. Stochastic gradient method has been discussed thoroughly. The proposed algorithm is tested for the stationary as well as non stationary Mackey Glass time series prediction. In addition the proposed algorithm is again tested on the prediction of only the x-component of the time series. The figure of merit is the Mean Square Error(MSE). The considerable improvement has been shown in comparison with its competitor algorithms.

A new kernel fractional affine projection algorithm is discussed in this disertation. Affine projection and kernel affine projection algorithms has also been discussed in accordance with the detail mathematical derivation. One application of predicting a chaotic three dimensional lorenz system is presented, that demonstrates the performance of the proposed algorithm in comparison with

LMS, APA, KAPA and in terms of mean square error as a figure of merit. Proposed algorithm is also tested on nonlinear channel equalization. This new formulation is another contribution in the field of nonlinear signal processing.

Parametric identification of Hammerstein Nonlinear Controlled Autoregressive systems through kernel affine projection algorithm is also developed. The experimental analysis of the proposed algorithm can give the consistent parameter estimates and better MSE results. Also the proposed method is validated in comparison with LMS, FLMS, MFLMS and Affine Projection algorithms.

A new square root extended kernel recursive least square algorithm is another development in the dissertation. Extended kernel recursive least square algorithm is also discussed. One application of prediction of stationary as well as nonstationary Mackey Glass time series is presented and also along with different noise levels. The learning curves in term of mean square error are observed, that demonstrates the proposed algorithm achieve better prediction performance, in comparison with EKRLS. Another experiments for the prediction of x-component of Lorenz time series is performed to further test the validity of the proposed algorithm.

6.1 Direction of Future Work

- One can look for other nonlinear application for Fractional Methods. Also different other fractional algorithm will be developed accordingly. Real time application can also be solved using the fractional methods.

- The automatic tuning of the parameter introduced in the Modified fractional least mean square algorithm can be made adaptive according to the nature of the problem. Also different nonlinear applications can tested in this regard.

- Different kernel function can be used in the kernel methods to further test the validity of the kernel algorithms. Fractional terms can also be added to the kernel algorithms.

- Kernel domain can be used to obtain the numerical solution of nonlinear Partial differential equations and other types of different highly nonlinear differential equations.

- The Extended kernel recursive least squares algorithm can be enhanced for nonlinear state space model having a state transition matrix. The application of the proposed filter can also be enhanced to tracking applications.

- Identification of different Auto regressive moving average model can be done using different kernel signal processing algorithms.

Bibliography

[1] K. S. Miller and B. Ross, "An introduction to the fractional calculus and fractional differential equations," 1993.

[2] A. Kilbas, O. Marichev, and S. Samko, "Fractional integral and derivatives (theory and applications)," *Gordon and Breach, Switzerland*, vol. 1, no. 993, p. 1, 1993.

[3] H. M. Srivastava and J. J. Trujillo, "Theory and applications of fractional differential equations," 2006.

[4] B. Ross, "Origins of fractional calculus and some applications," *Int. J. Math. Statist. Sci*, vol. 1, no. 1, pp. 21–34, 1992.

[5] B. Ross, "The development of fractional calculus 1695–1900," *Historia Mathematica*, vol. 4, no. 1, pp. 75–89, 1977.

[6] B. Ross, "Fractional calculus and its applications.," 1975.

[7] W. Arnsperger, *Christian Wolff's Relations to Leibniz*. E. Felber, 1897 in German.

[8] W. Guerrier, *Leibniz in his relations wit Russia and Peter the great: A historical account of this relationship, along with the leters and memoranda related there to,*. Commissioner director Acad. of Wissenschaften: St. Peterburg, Eggers & Cie., Leipzig M. & Voss., 1873 (in German).

[9] R. A. Reiff, *History of infinite series*. H. Laupp, 1889 in German.

[10] H. K. Srensen, "The mathematics of niels henrik abel: continuation and new approaches in mathematics during the 1820s," *Aarhus: RePoSS: Research Publications on Science Studies*, vol. 11, 2010.

[11] J. Liouville, *Memory on the change of independent variable, in the calculation of any differential indices*. 1835 (in French).

[12] J. Liouville, "Memory on the development of functions or part of functions in series whose various terms are subject to satisfy the same differential equation of the second order, containing a variable parameter," *Journal of pure mathemetics and its applications*, pp. 253–265, 1836 (in French).

[13] J. Liouville, "Note on the theory of arbitrary constants," *Journal of pure mathemetics and its applications*, pp. 342–349, 1838 (in French).

[14] M. Riesz, "The integral of riemann-liouville and the cauchy problem," *Acta mathematica*, vol. 81, no. 1, pp. 1–222, 1949 (in French).

[15] M. Riesz, "On the conjugate functions," *Mathematische Zeitschrift*, vol. 27, no. 1, pp. 218–244, 1928.

[16] l. d. S. Nagy, "On the set of positive functions in l2," *Annals of Mathematics*, pp. 1–13, 1938.

[17] A. Erdlyi, "On fractional integration and its application to the theory of hankel transforms," *The Quarterly Journal of Mathematics*, no. 1, pp. 293–303, 1940.

[18] A. Erdlyi and H. Kober, "Some remarks on hankel transforms," *The Quarterly Journal of Mathematics*, no. 1, pp. 212–221, 1940.

[19] K. B. Oldham and J. Spanier, "The fractional calculus: integrations and differentiations of arbitrary order," *Mathematics in Science and Engineering*, 1974.

[20] I. Podlubny, *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, vol. 198. Academic press, 1998.

[21] M. T. Barlow, K. Hattori, T. Hattori, and H. Watanabe, "Restoration of isotropy on fractals," *Physical review letters*, vol. 75, no. 17, p. 3042, 1995.

[22] D. Ben-Avraham and S. Havlin, *Diffusion and reactions in fractals and disordered systems*. Cambridge University Press, 2000.

[23] A. Carpinteri and F. Mainardi, *Fractals and fractional calculus in continuum mechanics*, vol. 378. Springer, 2014.

[24] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu-Batlle, *Fractional-order systems and controls: fundamentals and applications*. Springer Science & Business Media, 2010.

[25] B. West, M. Bologna, and P. Grigolini, *Physics of fractal operators*. Springer Science & Business Media, 2012.

[26] S. G. Osgouei and M. Geravanchizadeh, "Speech enhancement using convex combination of fractional least-mean-squares algorithm," in *Telecommunications (IST), 2010 5th International Symposium on*, pp. 869–872, IEEE, 2010.

[27] N. I. Chaudhary, M. A. Z. Raja, J. A. Khan, and M. S. Aslam, "Identification of input nonlinear control autoregressive systems using fractional signal processing approach," *The Scientific World Journal*, vol. 2013, 2013.

[28] S. Aldirmaz, L. Durak, and A. Serbes, "Fractional fourier domain lms-based adaptive filtering algorithms in active noise control," in *2008 IEEE 16th Signal Processing, Communication and Applications Conference*, 2008.

[29] S. K. Dubey and N. K. Rout, "Flms algorithm for acoustic echo cancellation and its comparison with lms," in *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on*, pp. 852–856, IEEE, 2012.

[30] M. S. Aslam and M. A. Z. Raja, "A new adaptive strategy to improve online secondary path modeling in active noise control systems using fractional signal processing approach," *Signal Processing*, vol. 107, pp. 433–443, 2015.

[31] M. A. Z. Raja and N. I. Chaudhary, "Two-stage fractional least mean square identification algorithm for parameter estimation of carma systems," *Signal Processing*, vol. 107, pp. 327–339, 2015.

[32] S. Shah, R. Samar, S. Naqvi, and J. Chambers, "Fractional order constant modulus blind algorithms with application to channel equalisation," *Electronics Letters*, vol. 50, no. 23, pp. 1702–1704, 2014.

[33] N. I. Chaudhary and M. A. Z. Raja, "Design of fractional adaptive strategy for input nonlinear box–jenkins systems," *Signal Processing*, vol. 116, pp. 141–151, 2015.

[34] M. A. Z. Raja and N. I. Chaudhary, "Adaptive strategies for parameter estimation of box–jenkins systems," *IET Signal Processing*, vol. 8, no. 9, pp. 968–980, 2014.

[35] N. I. Chaudhary and M. A. Z. Raja, "Identification of hammerstein nonlinear armax systems using nonlinear adaptive algorithms," *Nonlinear Dynamics*, vol. 79, no. 2, pp. 1385–1397, 2015.

[36] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE transactions on electronic computers*, no. 3, pp. 326–334, 1965.

[37] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[38] S. K. Rakse and S. Shukla, "Spam classification using new kernel function in support vector machine," *International Journal on Computer Science and Engineering*, vol. 2, no. 5, p. 2010, 1819.

[39] T. Howley and M. G. Madden, "An evolutionary approach to automatic kernel construction," in *International Conference on Artificial Neural Networks*, pp. 417–426, Springer, 2006.

[40] W. Liu, P. P. Pokharel, and J. C. Principe, "The kernel least-mean-square algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 543–554, 2008.

[41] W. Liu, J. C. Principe, and S. Haykin, *Kernel adaptive filtering: a comprehensive introduction*, vol. 57. John Wiley & Sons, 2011.

[42] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012.

[43] W. D. Parreira, J. C. M. Bermudez, C. Richard, and J.-Y. Tourneret, "Stochastic behavior analysis of the gaussian kernel least-mean-square algorithm," *IEEE transactions on signal processing*, vol. 60, no. 5, pp. 2208–2222, 2012.

[44] P. P. Pokharel, W. Liu, and J. C. Principe, "Kernel least mean square algorithm with constrained growth," *Signal Processing*, vol. 89, no. 3, pp. 257–265, 2009.

[45] H. S. Yazdi, M. Pakdaman, and H. Modaghegh, "Unsupervised kernel least mean square algorithm for solving ordinary differential equations," *Neurocomputing*, vol. 74, no. 12, pp. 2062–2071, 2011.

[46] H. S. Yazdi, H. Modaghegh, and M. Pakdaman, "Ordinary differential equations solution in kernel space," *Neural Computing and Applications*, vol. 21, no. 1, pp. 79–85, 2012.

[47] C. Richard and J.-C. M. Bermudez, "Closed-form conditions for convergence of the gaussian kernel-least-mean-square algorithm," in *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 1797–1801, IEEE, 2012.

[48] S. Zhao, B. Chen, P. Zhu, and J. C. Príncipe, "Fixed budget quantized kernel least-mean-square algorithm," *Signal Processing*, vol. 93, no. 9, pp. 2759–2770, 2013.

[49] P. Bouboulis and S. Theodoridis, "The complex gaussian kernel lms algorithm," in *International Conference on Artificial Neural Networks*, pp. 11–20, Springer, 2010.

[50] T. K. Paul and T. Ogunfunmi, "Study of the convergence behavior of the complex kernel least mean square algorithm," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 9, pp. 1349–1363, 2013.

[51] H. Bao and I. M. Panahi, "Active noise control based on kernel least-mean-square algorithm," in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pp. 642–644, IEEE, 2009.

[52] A. Gunduz, J.-P. Kwon, J. C. Sanchez, and J. C. Principe, "Decoding hand trajectories from ecog recordings via kernel least-mean-square algorithm," in *2009 4th International IEEE/EMBS Conference on Neural Engineering*, pp. 267–270, IEEE, 2009.

[53] I. M. Park, S. Seth, and S. Van Vaerenbergh, "Bayesian extensions of kernel least mean squares," *arXiv preprint arXiv:1310.5347*, 2013.

[54] J. Chen, W. Gao, C. Richard, and J.-C. M. Bermudez, "Convergence analysis of kernel lms algorithm with pre-tuned dictionary," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7243–7247, IEEE, 2014.

[55] P. Bouboulis and S. Theodoridis, "Extension of wirtinger's calculus to reproducing kernel hilbert spaces and the complex kernel lms," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 964–978, 2011.

[56] F. Albu and K. Nishikawa, "The kernel proportionate nlms algorithm.," in *EUSIPCO*, pp. 1–5, 2013.

[57] R. Pokharel, S. Seth, and J. C. Principe, "Mixture kernel least mean square," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pp. 1–7, IEEE, 2013.

[58] S. H. Ghafarian, H. S. Yazdi, and H. B. Kashani, "Kernel least mean square features for hmm-based signal recognition," *International Journal of Computer Theory and Engineering*, vol. 2, no. 2, p. 283, 2010.

[59] F. Albu, D. Coltuc, K. Nishikawa, and M. Rotaru, "An efficient implementation of the kernel affine projection algorithm," in *Proc. of ISPA*, pp. 342–346, Citeseer, 2013.

[60] K. Slavakis and S. Theodoridis, "Sliding window generalized kernel affine projection algorithm using projection mappings," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, pp. 1–16, 2008.

[61] J. M. Gil-Cacho, M. Signoretto, T. van Waterschoot, M. Moonen, and S. H. Jensen, "Nonlinear acoustic echo cancellation based on a sliding-window leaky kernel affine projection

algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1867–1878, 2013.

[62] S. Wang, J. Feng, and K. T. Chi, "Kernel affine projection sign algorithms for combating impulse interference," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 11, pp. 811–815, 2013.

[63] J. M. Gil-Cacho, T. van Waterschoot, M. Moonen, and S. H. Jensen, "Nonlinear acoustic echo cancellation based on a parallel-cascade kernel affine projection algorithm," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 33–36, IEEE, 2012.

[64] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on signal processing*, vol. 52, no. 8, pp. 2275–2285, 2004.

[65] S. Van Vaerenbergh, M. Lázaro-Gredilla, and I. Santamaría, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1313–1326, 2012.

[66] S. Van Vaerenbergh, J. Via, and I. Santamaría, "A sliding-window kernel rls algorithm and its application to nonlinear channel identification," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5, pp. V–V, IEEE, 2006.

[67] S. Van Vaerenbergh, I. Santamaría, W. Liu, and J. C. Príncipe, "Fixed-budget kernel recursive least-squares," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1882–1885, IEEE, 2010.

[68] M. N. Tehrani, M. Shakhsi, and H. Khoshbin, "Kernel recursive least squares-type neuron for nonlinear equalization," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, pp. 1–6, IEEE, 2013.

[69] M. Lázaro-Gredilla, S. Van Vaerenbergh, and I. Santamaría, "A bayesian approach to tracking with kernel recursive least-squares," in *2011 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, IEEE, 2011.

[70] M. Naseri Tehrani, M. Shakhsi Dastgahian, and H. Khoshbin Ghomash, "Kernel recursive least squares-type neuron for nonlinear equalization," in *Electrical Engineering (ICEE), 2013 21st Iranian Conference on*, 2013.

[71] D. Wingate and S. Singh, "Kernel predictive linear gaussian models for nonlinear stochastic dynamical systems," in *Proceedings of the 23rd international conference on Machine learning*, pp. 1017–1024, ACM, 2006.

[72] W. Liu, I. Park, Y. Wang, and J. C. Príncipe, "Extended kernel recursive least squares algorithm," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3801–3814, 2009.

[73] W. Xu and D. Xu, "The modified hr calculus to reproducing kernel hilbert space and the quaternion kernel least mean square algorithm," in *2015 Eighth International Conference on Internet Computing for Science and Engineering (ICICSE)*, pp. 50–55, IEEE, 2015.

[74] N. Haghighat, H. Kalbkhani, M. G. Shayesteh, and M. Nouri, "Variable bit rate video traffic prediction based on kernel least mean square method," *IET Image Processing*, vol. 9, no. 9, pp. 777–794, 2015.

[75] Q. Song, X. Zhao, H. Fan, and D. Wang, "Robust recurrent kernel online learning," 2016.

[76] H. Fan and Q. Song, "A sparse kernel algorithm for online time series data prediction," *Expert Systems with Applications*, vol. 40, no. 6, pp. 2174–2181, 2013.

[77] H. Fan, Q. Song, and S. B. Shrestha, "Kernel online learning with adaptive kernel width," *Neurocomputing*, vol. 175, pp. 233–242, 2016.

[78] H. Shah and M. Gopal, "Kernel recursive least squares function approximation in game theory based control," *Procedia Technology*, vol. 23, pp. 264–271, 2016.

[79] L. Lu, H. Zhao, and B. Chen, "Klmat: A kernel least mean absolute third algorithm," *arXiv preprint arXiv:1603.03564*, 2016.

[80] P. Bouboulis, S. Pougkakiotis, and S. Theodoridis, "Efficient klms and krls algorithms: A random fourier feature perspective," *arXiv preprint arXiv:1606.03685*, 2016.

[81] S. Wang, Y. Zheng, and C. Ling, "Regularized kernel least mean square algorithm with multiple-delay feedback," *IEEE Signal Processing Letters*, vol. 23, no. 1, pp. 98–101, 2016.

[82] S. Chouvardas and M. Draief, "A diffusion kernel lms algorithm for nonlinear adaptive networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4164–4168, IEEE, 2016.

[83] N. J. Fraser, D. J. Moss, J. Lee, S. Tridgell, C. T. Jin, and P. H. Leong, "A fully pipelined kernel normalised least mean squares processor for accelerated parameter optimisation," in *2015 25th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–6, IEEE, 2015.

[84] K. Slavakis, P. Bouboulis, and S. Theodoridis, "Adaptive multiregression in reproducing kernel hilbert spaces: The multiaccess mimo channel case," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 2, pp. 260–276, 2012.

[85] Z. Chen and R. C. Qiu, "Measurement denoising using kernel adaptive filters in the smart grid," in *Global High Tech Congress on Electronics (GHTCE), 2012 IEEE*, pp. 106–110, IEEE, 2012.

[86] V. Vovk, "Kernel ridge regression," in *Empirical inference*, pp. 105–116, Springer, 2013.

[87] M. Welling, "Kernel ridge regression," *Max Wellings Classnotes in Machine Learning (http://www. ics. uci. edu/welling/classnotes/classnotes. html)*, pp. 1–3, 2013.

[88] S. Van Vaerenbergh and I. Santamarıa, "A comparative study of kernel adaptive filtering algorithms," in *2013 IEEE Digital Signal Processing (DSP) Workshop and IEEE Signal Processing Education (SPE)*, 2013.

[89] A. Kale, M. L. Meena, and M. Gopal, "Kernel machines for uncalibrated visual servoing of robots," in *2013 IEEE International Symposium on Intelligent Control (ISIC)*, pp. 364–369, IEEE, 2013.

[90] M. Herrera and R. Filomeno Coelho, "Windowing strategies for on-line multiple kernel regression," in *International Workshop on Advances in Regularization, Optimization, Kernel Methods and Support Vector Machines, ROKS-2013, Leuven, Belgium*, pp. 105–106, 2013.

[91] B. Zhu, Z. Cheng, and H. Wang, "A kernel function optimization and selection algorithm based on cost function maximization," in *2013 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 259–263, IEEE, 2013.

[92] Y.-P. Zhao, K.-K. Wang, J. Liu, and R. Huerta, "Incremental kernel minimum squared error (kmse)," *Information Sciences*, vol. 270, pp. 92–111, 2014.

[93] Z. Mazloomi, H. T. Shandiz, and H. Faramarzpour, "Kernel least mean square algorithm in control of nonlinear systems," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, pp. 1–5, IEEE, 2013.

[94] J. Zhao, T. Wang, W. Ma, and H. Qu, "The kernel-lms based network traffic prediction," in *Information Science and Control Engineering 2012 (ICISCE 2012), IET International Conference on*, pp. 1–5, IET, 2012.

[95] B. Chen, N. Zheng, and J. C. Principe, "Survival kernel with application to kernel adaptive filtering," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pp. 1–6, IEEE, 2013.

[96] H. Fan, Q. Song, and Z. Xu, "Recurrent online kernel recursive least square algorithm for nonlinear modeling," in *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pp. 1574–1579, IEEE, 2012.

[97] T. Tanaka, Y. Washizawa, and A. Kuh, "Adaptive kernel principal components tracking," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1905–1908, IEEE, 2012.

[98] N. Kwak, "Nonlinear projection trick in kernel methods: An alternative to the kernel trick," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 12, pp. 2113–2119, 2013.

[99] X. Wei and Z. Pan, "Chebyshev kernel with orthogonal features," in *Computer Science and Information Processing (CSIP), 2012 International Conference on*, pp. 941–944, IEEE, 2012.

[100] X. Xu, I. W. Tsang, and D. Xu, "Soft margin multiple kernel learning," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 5, pp. 749–761, 2013.

[101] F. P. Anaraki and S. M. Hughes, "Kernel compressive sensing," in *2013 IEEE International Conference on Image Processing*, pp. 494–498, IEEE, 2013.

[102] L. Gómez-Chova and G. Camps-Valls, "Learning with the kernel signal to noise ratio," in *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, IEEE, 2012.

[103] S.-Y. Kung and P.-y. Wu, "On efficient learning and classification kernel methods," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2065–2068, IEEE, 2012.

[104] Z. Noumir, P. Honeine, and C. Richard, "Kernels for time series of exponential decay/growth processes.," in *MLSP*, pp. 1–6, 2012.

[105] C. Saidé, R. Lengellé, P. Honeine, C. Richard, and R. Achkar, "Dictionary adaptation for online prediction of time series data with kernels," in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 604–607, IEEE, 2012.

[106] Y.-Q. Miao, A. K. Farahat, and M. S. Kamel, "Auto-tuning kernel mean matching," in *2013 IEEE 13th International Conference on Data Mining Workshops*, pp. 560–567, IEEE, 2013.

[107] M.-a. Takizawa and M. Yukawa, "An efficient data-reusing kernel adaptive filtering algorithm based on parallel hyperslab projection along affine subspaces," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3557–3561, IEEE, 2013.

[108] D. Widjaja, C. Varon, A. Dorado, J. A. Suykens, and S. Van Huffel, "Application of kernel principal component analysis for single-lead-ecg-derived respiration," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 1169–1176, 2012.

[109] Q. Miao and C. Li, "Kernel least-mean mixed-norm algorithm," in *Automatic Control and Artificial Intelligence (ACAI 2012), International Conference on*, pp. 1285–1288, IET, 2012.

[110] C. Chang and R. Ansari, "Kernel particle filter," in *Proceedings: 2003 International Conference on Image Processing, ICIP-2003*, 2003.

[111] Y. Guo, D. Peng, H. Chen, and A. Xue, "A kernel particle filter algorithm for joint tracking and classification," in *Information Fusion (FUSION), 2012 15th International Conference on*, pp. 2386–2391, IEEE, 2012.

[112] O. Taouali, I. Zakraoui, I. Elaissi, and H. Messaoud, "A new online kernel method identification on rkhs space," in *2013 International Conference on Electrical Engineering and Software Applications*, 2013.

[113] X. Luo, D. Zhang, L. T. Yang, J. Liu, X. Chang, and H. Ning, "A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems," *Future Generation Computer Systems*, vol. 61, pp. 85–96, 2016.

[114] Y. Zheng, S. Wang, J. Feng, and K. T. Chi, "A modified quantized kernel least mean square algorithm for prediction of chaotic time series," *Digital Signal Processing*, vol. 48, pp. 130–136, 2016.

[115] X. Ren, Q. Yu, B. Chen, N. Zheng, and P. Ren, "A reconfigurable parallel fpga accelerator for the kernel affine projection algorithm," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pp. 906–910, IEEE, 2015.

[116] M. Hou, Q. Zhao, B. Chaib-draa, and A. Cichocki, "Common and discriminative subspace kernel-based multiblock tensor partial least squares regression," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[117] Z.-G. Shang and H.-S. Yan, "Product design time forecasting by kernel-based regression with gaussian distribution weights," *Entropy*, vol. 18, no. 6, p. 231, 2016.

[118] S. Shafieezadeh-Abadeh and A. Kalhor, "Evolving takagi–sugeno model based on online gustafson-kessel algorithm and kernel recursive least square method," *Evolving Systems*, vol. 7, no. 1, pp. 1–14, 2016.

[119] J. Zhao, X. Liao, S. Wang, and K. T. Chi, "Kernel least mean square with single feedback," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 953–957, 2015.

[120] B. Chen, Z. Yuan, N. Zheng, and J. C. Príncipe, "Kernel minimum error entropy algorithm," *Neurocomputing*, vol. 121, pp. 160–169, 2013.

[121] X. Wang, P. Wang, X. Gao, and Y. Qi, "On-line quality prediction of batch processes using a new kernel multiway partial least squares method," *Chemometrics and Intelligent Laboratory Systems*, 2016.

[122] J. Lee, K. Chang, C.-H. Jun, R.-K. Cho, H. Chung, and H. Lee, "Kernel-based calibration methods combined with multivariate feature selection to improve accuracy of near-infrared spectroscopic analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 147, pp. 139–146, 2015.

[123] T. Song, D. Li, L. Cao, and K. Hirasawa, "Kernel-based least squares temporal difference with gradient correction," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 771–782, 2016.

[124] F. Liu, W. Yuan, Y. Ma, Y. Zhou, and H. Liu, "New enhanced robust kernel least mean square adaptive filtering algorithm," in *Estimation, Detection and Information Fusion (ICEDIF), 2015 International Conference on*, pp. 282–285, IEEE, 2015.

[125] B. Chen, R. Wang, N. Zheng, and J. C. Principe, "On initial convergence behavior of the kernel least mean square algorithm," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–5, IEEE, 2015.

[126] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[127] B. Widrow, M. Lehr, F. Beaufays, E. Wan, and M. Bilello, "Learning algorithms for adaptive signal processing and control," 1993.

[128] R. M. A. Zahoor and I. M. Qureshi, "A modified least mean square algorithm using fractional derivative and its application to system identification," *European Journal of Scientific Research*, vol. 35, no. 1, pp. 14–21, 2009.

[129] M. Geravanchizadeh and G. Sina, "19th iranian conference on electrical engineering, may 17–19, 2011," *Tehran, Iran*, 2011.

[130] P. Akhtar and M. Yasin, "Performance analysis of bessel beamformer and lms algorithm for smart antenna array in mobile communication system," in *International Multi Topic Conference*, pp. 52–61, Springer, 2012.

[131] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.

[132] E. Chng, S. Chen, and B. Mulgrew, "Gradient radial basis function networks for nonlinear and nonstationary time series prediction," *IEEE transactions on neural networks*, vol. 7, no. 1, pp. 190–194, 1996.

[133] S. Hayldn, "Adaptive filter theory," 1996.

[134] I. Hunter and M. Korenberg, "The identification of nonlinear biological systems: Wiener and hammerstein cascade models," *Biological cybernetics*, vol. 55, no. 2-3, pp. 135–144, 1986.

[135] M. J. Korenberg and I. W. Hunter, "The identification of nonlinear biological systems: Wiener kernel approaches," *Annals of biomedical engineering*, vol. 18, no. 6, pp. 629–654, 1990.

[136] K. Fruzzetti, A. Palazoğlu, and K. McDonald, "Nolinear model predictive control using hammerstein models," *Journal of process control*, vol. 7, no. 1, pp. 31–41, 1997.

[137] Y.-Y. Cao and Z. Lin, "Robust stability analysis and fuzzy-scheduling control for nonlinear systems subject to actuator saturation," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 1, pp. 57–67, 2003.

[138] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *IEEE transactions on multimedia*, vol. 2, no. 3, pp. 141–151, 2000.

[139] F. Ding, Y. Shi, and T. Chen, "Gradient-based identification methods for hammerstein nonlinear armax models," *Nonlinear Dynamics*, vol. 45, no. 1-2, pp. 31–43, 2006.

[140] W. Fan, F. Ding, and Y. Shi, "Parameter estimation for hammerstein nonlinear controlled auto-regression models," in *2007 IEEE International Conference on Automation and Logistics*, 2007.

[141] J. Ding, F. Ding, X. P. Liu, and G. Liu, "Hierarchical least squares identification for linear siso systems with dual-rate sampled-data," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2677–2683, 2011.

[142] H. Han, L. Xie, F. Ding, and X. Liu, "Hierarchical least-squares based iterative identification for multivariable systems with moving average noises," *Mathematical and Computer Modelling*, vol. 51, no. 9, pp. 1213–1220, 2010.

[143] L. Xiang, L. Xie, Y. Liao, and R. Ding, "Hierarchical least squares algorithms for single-input multiple-output systems based on the auxiliary model," *Mathematical and Computer Modelling*, vol. 52, no. 5, pp. 918–924, 2010.

[144] Y. Liu, Y. Xiao, and X. Zhao, "Multi-innovation stochastic gradient algorithm for multiple-input single-output systems using the auxiliary model," *Applied Mathematics and Computation*, vol. 215, no. 4, pp. 1477–1483, 2009.

[145] L. Han and F. Ding, "Identification for multirate multi-input systems using the multi-innovation identification theory," *Computers & Mathematics with Applications*, vol. 57, no. 9, pp. 1438–1449, 2009.

[146] F. Ding, P. X. Liu, and G. Liu, "Auxiliary model based multi-innovation extended stochastic gradient parameter estimation with colored measurement noises," *Signal Processing*, vol. 89, no. 10, pp. 1883–1890, 2009.

[147] X.-w. CHEN and F. DING, "Comparison of iterative and recursive identification for systems with colored noises [j]," *Journal of System Simulation*, vol. 21, p. 009, 2008.

[148] F. Chang and R. Luus, "A noniterative method for identification using hammerstein model," *IEEE Transactions on Automatic control*, vol. 16, no. 5, pp. 464–468, 1971.