International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Performance Comparison of Machine Learning Approach for Dynamic System Identification and Control

Rakesh Kumar Pattanaik[1], **Mihir Narayan Mohanty** *[2]

Submitted: 05/02/2024 **Revised:** 13/03/2024 **Accepted:** 19/03/2024

Abstract: Mostly the industrial control for dynamic system is the challenge in recent research. The problem is too complex due to non-linear and dynamic nature. To tackle this problem popular model is chosen as Functional link Artificial Neural Network (FLANN). However, the training is performed with kernel based least mean square (K-LMS) algorithm. Further three different kernels are experienced for the proposed model. Finally, the mixed kernel is proposed for LMS based training to the FLANN model. It is capable of performing at a higher level for faster convergence while maintaining its robust characteristics. However, because of its useful function approximation properties, it has been selected as an alternate method for identifying nonlinear systems. The proposed ANNs model has been demonstrated to be applicable to the modelling of complicated dynamical systems. A comparison is made among different kernel approached as well as with the earlier methods. The results of various strategies, such as Sliding Mode, RBFN, and k-LMS-based FLANN, have been compared in a performance analysis.

Keywords: *Dynamic System, Control, Identification, RBF, FLANN, Artificial Neural Network.*

1. Introduction

In the field of engineering, one of the major challenges is the identification of unknown complex systems [1]. Many different statistical and cutting-edge approaches have been developed to address the issue of system identification. As a result of its versatility, the adaptive filter has become a go-to tool for addressing a wide range of statical system identification issues [2, 3, 4, and 5]. Kernel adaptive filtering, which emerged at the intersection of machine learning and statistical signal processing, has become one of the most popular areas of research in adaptive signal processing in recent years. The adaptive filter's coefficients are continuously and automatically adjusted to optimize the model's performance. Adaptive filters can improve accuracy by transforming raw data into a format with more, or perhaps infinitely more, dimensions. The least mean square (LMS) approach, developed by Windrow and Hoff, is widely used for identifying nonlinear systems since it produces a mean square error [6, 7]. The goal is to develop a kernel least-squares model that acts as an adaptive filter to identify and control of nonlinear systems. Errors in estimates can be adjusted by adjusting filter coefficients according to the input vector and the desired output. Further more research is done on least mean M-estimate (LMM) algorithm and the hyperbolic secant LMS (HSLMS) algorithm and introduced by Zhou et al., applied in various contexts which are two examples of LMS algorithms with modifications that result in the M-estimation function [8, 9, 10, 11]. In the past decade, researchers have focused their attention on single-kernel adaptive filters, testing and analyzing their effectiveness in both theoretical and experimental settings on a wide range of real-valued nonlinear system identification issues. Due to its ease of use and reliability, the KLMS algorithm developed in has gained a lot of popularity in recent years. Recently, adaptive techniques based on a complex kernel have been presented for identifying nonlinear systems whose inputs and outputs take on a complex value is presented in [12,13]. An MVC-based cost-function-driven sparsity-induced KAF algorithm for nonlinear sparse system identification (SSI) problem is presented in [14]. In this work a sliding mode control method is used to estimate the parameters of nonlinear dynamic system for control application. Sliding mode control is presented as a means of synthesizing an adaptive learning algorithm within a neuron whose weights are generated by first-order dynamical filters whose parameters may be altered, so allowing for the characterization of dynamical processes in terms of such neurons. It is demonstrated that the sliding mode control method has quick convergence and robustness qualities [15]. The well-known Widrow-Hoff delta rule [16] is a least-mean-square learning error reduction procedure that, under certain conditions, imposes a linear, asymptotically stable dynamics on the underlying discrete-time error dynamics. The authors analyzed that the Delta Rule may be revised using concepts from quasi-sliding mode control [17]. This work demonstrates that a weight-switching adaptation method imposes a linear-learning error dynamics that is asymptotically stable over time. Recently, an alternative perspective on neuron-based adaptive learning has been given, which takes into account a distinct type of issues described on analogue adaptive neurons. Adaptive weight adjustment in a continuous time frame, as opposed to discrete time frames, is required in this context. Sliding mode control has been used to the problem of designing learning algorithms for adaptable analogue neurons in continuous time [15]. Time-varying neuron weight

[1] *ITER, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, India*
ORCID ID =0000-0002-4930-5134
[2] *ITER, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, India*
ORCID ID: *https://orcid.org/0000-0003-1252-949X*
** Corresponding Author Email: mihir.n.mohanty@gmail.com*

adaptation using a continuous time sliding mode control technique is briefly discussed in [17]. Since the resulting sliding mode control solution is inherently dynamic, it has inspired the introduction of a new type of neuron, the "dynamical filter weights neuron," in which the weights are all implemented as first-order, linear, time-varying dynamical systems. By allowing weight adjustment methods to be done on the time-varying 'gains and 'time constants' of the weights, the 'dynamical filter weights' simplify the expression of dynamical processes in terms of a set of dynamical-filter-weights neurons. The superior performance of NNs taught with different paradigms for approximating nonlinear functions [17] is due to their capacity to learn by optimizing an appropriate error function (multilayer perceptions, radial basis functions, etc.) have become increasingly popular in the past decade have proven to be an effective method for identifying and controlling systems in a learning environment. For the control of nonlinear plants, sliding mode control approach has proven to be superior in a variety of industrial applications. Sliding mode control method is used to control a servo motor as a nonlinear system control application is presented in [18]. This method is used to create dynamic equations that can be controlled in real-time. Sliding mode control (SMC) is a robust control mechanism, however it has problems due to chattering, which limits its practical uses. Authors have proposed different sliding mode control approaches for control problems. A Neural network based adaptive sliding mode control method with the combination of radial-basis-function is presented in [19]. Similarly, another approach called as non-singular fast terminal sliding mode control approach is developed for a class of nonlinear systems with unknown uncertainty bounds is introduced in [20, 21].

Various Nonlinear system identification and control related models are presented in [22-27]. For the challenging problem space of identifying and controlling nonlinear, dynamic systems, a robust Fusion Kernel based Functional link Artificial Neural Network model (K-FLANN) is developed. The objective is to create a kernel least mean square model that can identify and control nonlinear systems by learning as an adaptive filter. A collection of filter coefficients that can be altered based on an input vector and the desired result controls estimation errors [28,29]. The proposed method has been demonstrated to accurately identify and control nonlinear systems. As a result, here are the most important contributions of this paper.

[1] The construction of Least mean square learning algorithm in a neuron is first proposed using a sliding mode control method.

[2] The weights are made up of first-order dynamical filters with adjustable weights, this makes it possible to represent dynamical processes as a set of these neurons.

[3] After parameterization, the model can be used in a control application.

[4] A fusion kernel based on (Gaussian kernel and a cosine kernel) is verified to identify the nonlinear dynamic model.

[5] Furthermore, the fusion kernel method is applied to a Functional link neural Artificial network to form a Robust algorithm for nonlinear dynamic system identification and control model.

The rest of the paper is organised as follows:

The framework of the revised Sliding Mode Model is discussed in Section II. Radial basis function neural network is discussed in greater detail in Section III. In section IV, Procedure of kernel least square algorithm is discussed. The methodology of the Proposed Mixed Kernel based Functional –link Artificial neural network is presented in the section V. Results are discussed in the section VI. Section VII concludes the work.

2. Architecture of the Modified Sliding Mode Model

First-order, linear, dynamic filters with tunable weights have been implemented using models of a single neuron [30]. It is defined as:

$$\dot{y}_i(n) = a_i(n) y_i(n) + k_i(n) x_i(n), i = 1,2,\dots\dots\dots n$$
(1)

The input vector $x(n)$ is made more robust by adding an un-measurable norm-bounded perturbation vector $\eta(n)$, and the resulting vector is used to train a new vector $\xi(n)$ as

$$\xi_i(n) = x_i(n) + \eta_i(n)$$ and

$$\|\xi_i(n)\| = \sqrt{\xi_1^2(n) + \xi_2^2(n) + \cdots\cdots + \xi_n^2(n)} \leq V_\xi$$
(2)

As a result, the adoption parameters can be adjusted together with the weight parameters.

$$a(n) = \frac{-(W_1 sign(e(n)) - W_2 e(n))}{\|\xi(\mathrm{n})\|^2 + \|y(n)\|^2} y(n)$$

(3)

and

$$k(n) = \frac{-(W_1 sign(e(n)) - W_2 e(n))}{\|\xi(n)\|^2 + \|y(n)\|^2} \xi(n)$$

(4)

The time -varying scholar function is denoted as $a_i(n)$ and $k_i(n)$, i=1, 2….m, with input x_i .Fig.1 shows the figure of dynamical-filter weight neuron model. Where $a(n)$, $k(n)$ represented as:

$a(n)$ =tanh $(a_1(n), a_2(n), \ldots, a_n(n))$

$k(n)$ =tanh $k_1(n), k_2(n), \ldots\ldots, k_n(n))$

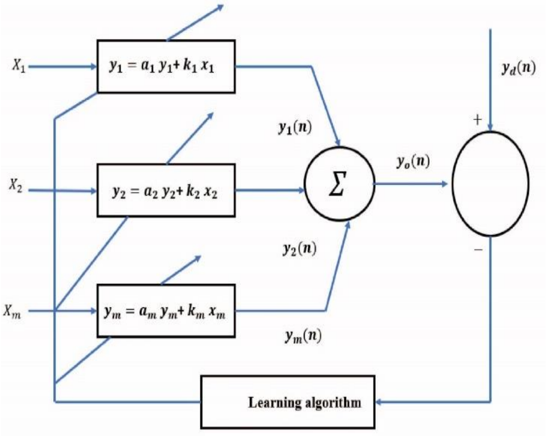

Fig.1 The Process of dynamic filter weight neural network

Chattering is a known issue with sliding mode controls that can make it difficult to predict nonlinear signal parameters, to address this, the authors propose a new parameter adjustment technique that works in the following iterative fashion.

$$a(n+1) = a(n) - \frac{(W_1 \tanh(\beta e(n)) + W_2 e(n))}{\|\xi(\mathrm{n})\|^2 + \|y(n)\|^2} y(n)$$

(5)

and

$$k(n+1) = k(n) - \frac{(W_1 \tanh(\beta e(n)) + W_2 e(n))}{\|\xi(n)\|^2 + \|y(n)\|^2} \xi(n)$$

(6)

The trigonometric function as $\tanh(\beta e(n))$ is considered as the function for the machine learning model to develop the FLANN. It is robust and sensitive to noise; β is considered as the control parameter and smoothen the system input.

Differential equations are used to compute the derivatives y1, y2, ..., ym, which are then used to build the filter. The n-th filtered output is then calculated as

$$\frac{y_{m+1} - y_m}{\Delta n} = -a_m y_m - k_m x_m$$

or

$$y_{m+1} = (-a_m \Delta n + 1) y_m - \Delta t . x_m . k_m$$

(7)

3. Radial Basis Function Neural Network Model

The RBF model's basic structure is depicted in Fig.1. It has three layers: input, hidden, and output.

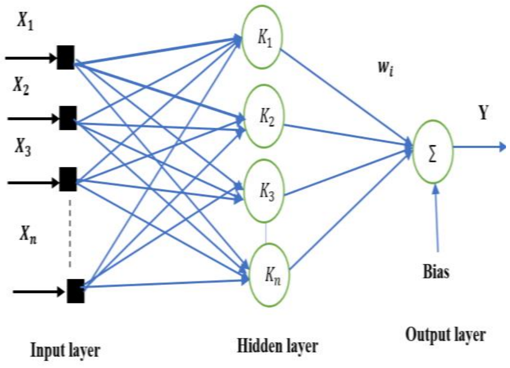

Fig.2 Architecture of basic RBF model

Form Fig.2**,** input vector be $P \epsilon R^{M0}$ and process of the mapping to RBF, $T = R^{M0} \rightarrow R^1$ is mentioned as:

$$y = \sum_{i=1}^{m_1} W_i\, k_i(\|X - X_i\|) + b \quad (8)$$

where m_1=4 neurons within the hidden layer, $P_i \epsilon R^{M0}$ *are* denoted as RBF centers. W_i is the weight connecting between the hidden layer and output neuron.

3.1 Gaussian Kernel

Because of its adaptability [16], the Gaussian kernel is chosen

$$k_i(\|X - X_i\|) = \exp\left(\frac{-\|X - X_i\|^2}{\ddot{\Upsilon}^2}\right) \quad (9)$$

Here, the Gaussian kernel dispersion is denoted by the $\ddot{\Upsilon}$. In order to calculate distances, the kernels are employed. Where the Euclidean norm is considered. An effective distance matrix is employed in the procedure.

3.2 Cosine Kernel

The cosine kernel is represented as:

$$k_i(\|X, X_i\|) = \alpha_1 k_{i1}(X.X_i) + \alpha_2 k_{i2}(\|X - X_i\|) \quad (10)$$

4. Procedure of Mk-LMS Algorithm

Using a kernel function, the input can be turned into a space with many more dimensions. The adaptive filtering technique known as Kernel Least Mean Square (KLMS) can be thought of as a feature space implementation of the Least Squares (LS) algorithm and is performed in real time online [28]. At instant k, the input data vector is assumed to be $x(n) \in, R$ and the target response is represented by $d(n) \in R$. The algorithm's core idea is to use a Mercer's kernel to transform the input data set $x(n)$ into a high-dimensional feature space. Finally, least-squares methods are utilized to formulate $d(n)$. Mercer's kernel is defined as a positive-definite function where k is a constant. $k: x \times x \longrightarrow R$.

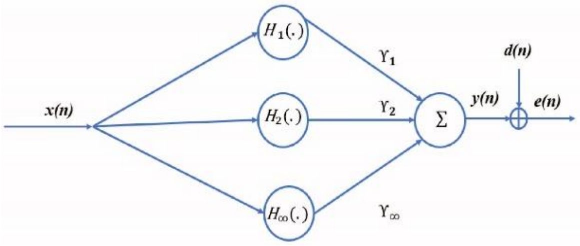

Fig.3 Block diagram representation of kernel identification process.

The block diagram representation of the kernel learning process is depicted in Fig.3**,** To train a linear adaptive system to reduce mean square error, Widrow invented the LMS algorithm in 1960, which is both simple and elegant. For a given quadratic cost function $J_W(n)$. where n is the time index and w is the tap-weight vector. With accurate measurements of the gradient vector $\nabla J_W(n)$ and a suitably chosen step-size parameter η , it is proven that the weight vector updated by the steepest-descent algorithm converges on the best Wiener solution on average. To update the weight vector, the LMS algorithm does not rely on the actual gradient, but rather on an instantaneous estimate calculated as $\nabla \hat{J}_W(n) = -2e(n)u(n)$ leading to the following stochastic gradient descent update rule.

$$w(n+1) = w(n) + 2\,\eta e(n)u(n) \qquad (11)$$

Implementing the linear LMS algorithm given by (2) in the kernel feature space is the main notion. For this let us assume that map the point x(n) in input space to $\emptyset(x(n))$ in the kernel feature space with $< \emptyset(x(n)), \emptyset(x(m)) \geq = k(x(n)), (x(m))$, where {<.,. >} in kernel Hilbert space stands for the inner product. For the most popular kernels, this feature space transformation is nonlinear, and the resulting feature space may have unlimited dimensions, depending on the kernel utilised. Using this space, we may define the weight vector ϒ such that y(n) = (ϒ (n), ∅ (x (n))). ϒ (n) is ϒ at time n, desired response is represented as d(n). Input vector for the nonlinear filtering technique is depicted in Fig.3**.** When u(n) is changed into u(n), it becomes a feature vector with an endless number of elements. whose constituent parts are added together by the infinite equivalent weight vector in different dimensions. This nonlinear filter has only a single weight layer, but it is a universal approximator [32] due to the theoretically unlimited size of the feature space. Now, due to the linear structure for cost function $J_\Upsilon(n) = E[d(n) - y(n)]^2$ can be w.r.to ϒ. Using the stochastic instantaneous estimate of the gradient vector, this may be calculated in the same way as in (10), leading to

$$\Upsilon(n+1) = \Upsilon(n) + 2\,\eta e(n)\emptyset x(n) \qquad (12)$$

The convergence, speed, and amount of mis adjustment of the adaptation algorithm are all determined by a step-size parameter, η, which is the same as before [31,32]. The one exception here is that in (3), ϒ is in the infinite dimensional feature space, making direct updating for ϒ impossible.

$$\Upsilon(n) = \Upsilon(0) + 2\eta \sum_{i=0}^{n-1} e(i)\emptyset(x(i)) \qquad (13)$$

For simplicity, let's set ϒ (0) = 0 (therefore e (0) = d (0)). The ultimate form of ϒ (n) is as follows:

$$\Upsilon(n) = 2\eta \sum_{i=0}^{n-1} e(i)\emptyset(x(i)) \qquad (14)$$

This is where we'll use the kernel technique. With the input ∅ (u(n)), and the output ϒ (n) from (14), the solution at n is given as

$$y(n) =< \Upsilon(n), \emptyset(x(n)) \geq \sum_{i=0}^{n-1} e(i) < \emptyset(x(i)), \emptyset(x(n)) >$$

$$= \eta \sum_{i=0}^{n-1} e(i)k(x(i), x(n)) \qquad (15)$$

The equation (15) is known as Kernel LMS algorithm.

5. Methodology of the Proposed Algorithm

Single layer neural networks can be thought of as an alternate strategy to deal with the difficulties posed by multi-layer neural networks. However, due to its linear character, the single layer neural network frequently fails to map large nonlinear issues. Consequently, it is extremely difficult to use a single-layer feed-forward artificial neural network to solve such issues. In order to bridge the gap between the simplicity of a single-layer network and the complexity and computational demands of multi-layer networks, the FLANN design is developed. Fig.4 depicts the FLANN design, which uses functional expansion of the input vector to avoid linear mapping by employing a single-layer feed-forward neural network. Pao proposes FLANN, a unique single-layer ANN structure that can produce arbitrary-complex decision regions by producing nonlinear decision boundaries. The functional link performs an operation on a portion of the complete pattern, producing a family of linearly independent functions, and then evaluating those functions with the pattern as input. FLANN is used because it speeds up learning and requires less computing power.

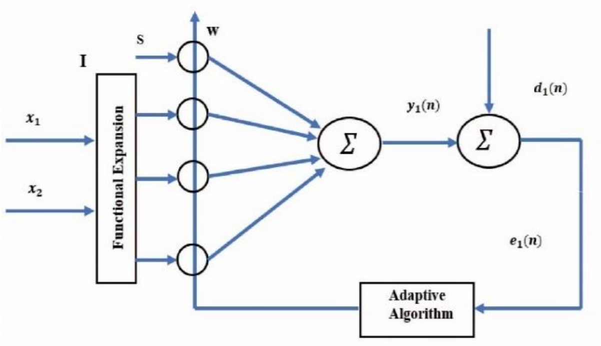

Fig.4. Architecture of Functional link Artificial Neural Network (FLANN)

5.1 Learning Algorithm

It is possible to describe the method for learning in [33]. For the sake of argument, let us say X is a $N \times 1$-dimensional input vector representing N data points, and k represents the kth data point is given by

$$X_k = x_k, 1 \leq k \leq N \quad (16)$$

Because of this nonlinear expansion, the final matrix has $N \times M$ dimensions, with N being the number of entries. Since each element is nonlinearly expanded into M new elements, the resulting matrix contains N by M rows and columns. The equation presented in [10, 14] is used to do the functional expansion of the element X_k using power series expansion.

$$s_i = \begin{cases} x_k for i = 1 \\ x_k^l for 1 = 2,3, \ldots m \end{cases} \quad (17)$$

where l = 1,2,3, … M. For trigonometric expansion, the

$$\begin{cases} x_k for i = 1 \\ \tanh(l\pi x_k) for i = 2,4,..M \\ \tanh(l\pi x_k) for i = 3,5,..M+1 \end{cases} \quad (18)$$

If l =1, 2, 3... M. The expanded components of the input vector E are denoted by the $N \times (M+1)$ matrix S. There is no bias in the input Consequently, the S matrix has dimensions of $N \times Q$, where Q = M+2, as the extra unity value is used to pad the matrix. Let's say that the elements of the weight vector, W having Q element. The results are displayed

$$Y = \sum_{i=1}^{Q} s_i \, \mathrm{wi} \quad (19)$$

In matrix notation the output can be,

$$Y = S.W^T \quad (20)$$

The e(n) error signal at the n-th iteration can be calculated as

$$e(n) = d(n) - y(n) \quad (21)$$

The cost function at the kth iteration, denoted by n, is as follows:

$$\xi(n) = \frac{1}{2} \sum_{l=1}^{p} e_l^2 \, (n) \quad (22)$$

where P is the total number of nodes in the output stage. Least-squares methods can be used to revise the weight vector, as

$$w(n+1) = w(n) - \frac{\mu}{2} \Delta(n) \quad (23)$$

where Δn is an instantaneous estimate of the gradient $\xi(n)$ of with respect to the weight vector $w(n)$.

$$\Delta n = \partial\xi/\partial w = -2e\, n\, y\,(n)/\,\partial w = -2en\partial\,[w\,(n)\,s(n)] \,/\, \partial w \quad (24)$$

Substituting the value of (21) in (22) we get

$$w(n+1) = w(n) - \mu e(n)\, s(n) \quad (25)$$

Where μ is the step-size parameter that can modify the convergence field. Each function in the functional expansion of a mean-square function is linearly independent and satisfies the orthogonality condition since trigonometric functions are used to describe mean-square functions. When studying a function with two variables, the outer product terms paired with the trigonometric polynomials produce superior outcomes.

5.2 Mixed Kernel (Gaussian +Cosine) Algorithm

The literature claims that the cosine distance matrix outperforms Euclidean distance measuring [15] as stated as:

$$k_i(P.P_i) = a_1 k_{i1}(X.X_i) + a_2 k_{i2}(\|X.X_i\|) \quad (26)$$

Where $k_i(X.X_i)$ and $k_{i2}(\|X.X_i\|)$ are the cosine Euclidean distance for $a_1$anda_2 .In equation (4) the values of $a_1$and a_2 can be considered as a dynamic adaptive variable and given as:

$$a_1 \equiv \frac{|a_1(n)|}{|a_1(n)| + |a_2(n)|} \quad (27)$$

$$a_2 = \frac{|a_2(n)|}{|a_1(n)| + |a_2(n)|} \quad (28)$$

Were the mixing weight formulating as, $a_1(n)$+$a_2(n)$=1. Therefore, the new kernel formula is:

$$k_i(X.X_i) = \frac{|a_1(n)|k_{i1}(X.X_i) + |a_2(n)|k_{i2}(\|X.X_i\|)}{|a_1(n)| + |a_2(n)|} \quad (29)$$

The n^{th} iteration mapping for a specific epoch can be denoted as:

$$y = \sum_{L=1}^{m_1} W_L(n)\, k_L(X.X_L) + b(n)m \quad (30)$$

Where W_L(n)= specific weights at each iteration and b(n)=bias, at each iteration adapted. The cost function ε(n) calculated as:

$$\varepsilon(n) = \varepsilon\big(a_1(n), a_2(n)\big) = \frac{1}{2}\big(\hat{y}(n) - y(n)\big)^2 \quad (31)$$

New equation with updated weight and bias:

$$W_i(n+1) = W_i(n) + ή e(n) k_i(X.X_i) \quad (32)$$

$$b_i(n+1) = b_i(n) + ή e(n) \quad (33)$$

6. Result And Discussion

Comparisons of results have been made between FLANN, RBF, and sliding mode are presented in this section. A nonlinear sinusoidal signal is considered for example and its response is presented.

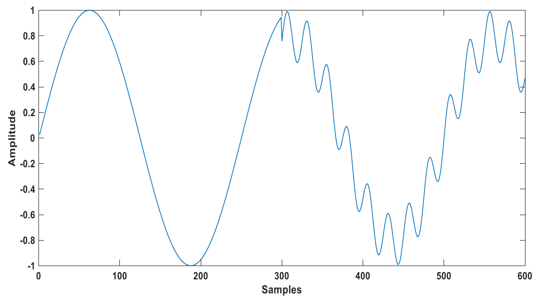

Fig.5.Sinusoidal signal for nonlinear system identification

From the Fig.5 the input response of the nonlinear signal is verified. The total number of samples taken is 600. The learning process of the system indentation includes training and testing. Initially 70% of the samples are used to train the model. After the model learn the system behavior then testing is implemented to verify the superiority of the system.

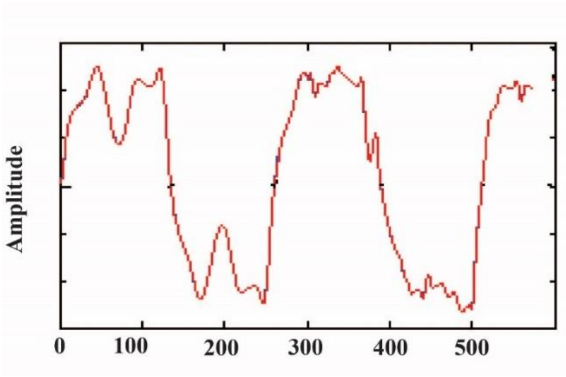

Fig.6. Output response of the nonlinear plant

The output response of the nonlinear signal is depieced in Fig.6. The idea is to design a perfect model that can track the nonlinear models output or in other words the parameters of the given nonlinear plants must be estimated. To verify, three nonlinear model has been developed. Initially sliding model control model is designed and the tracking output as actual verses estimated output depicted in Fig.7. From the figure sliding mode model learned perfectly and provided a good tracking result. The parameter of the nonlinear signal has been estimated by the sliding mode model.

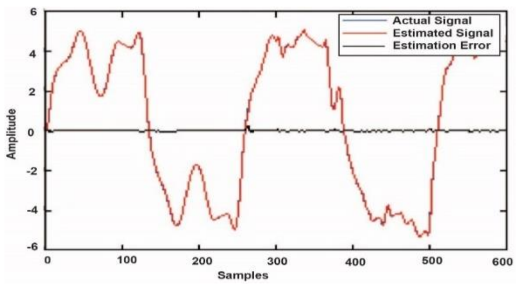

Fig.7. Identification of nonlinear plant with sinusoidal signal Using sliding mode

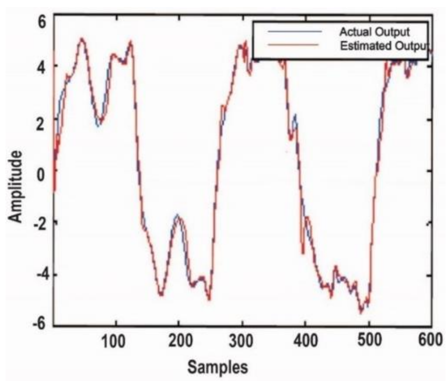

Fig.8 Identification of RBF nonlinear plant with estimated output

After verifying the sliding mode model Radial basis function model is designed. The actual verses estimated output is seen from the Fig.8. from the figure the parameters of the nonlinear model are estimated properly.

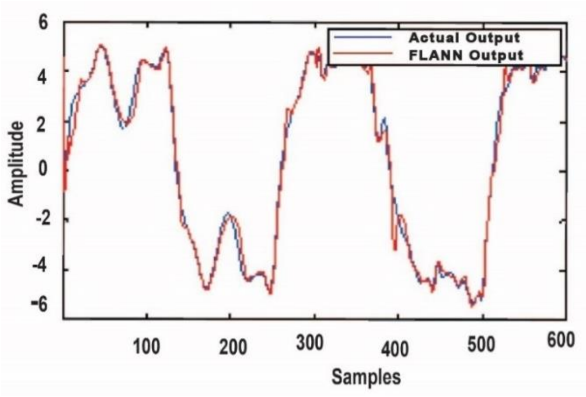

Fig.9 Sinusoidal signal-based FLANN for nonlinear system identification

After verifying the RBF model Another model is designed as FLANN model. In Fig.9 the actual verses estimated result is depicted. The model has a perfect tracking ability as compare to the RBF model. The actual response is sheen in blue line and FLANN estimated response is seen from the red line.

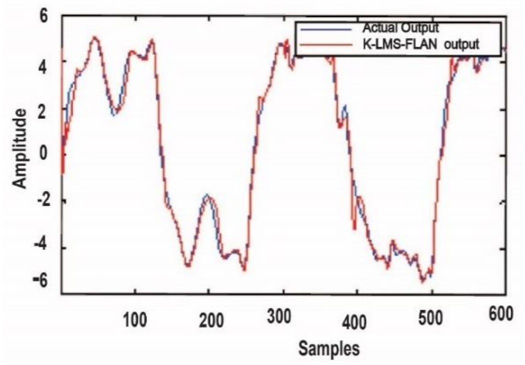

Fig.10. Proposed K-LMS based FLANN model for nonlinear system identification

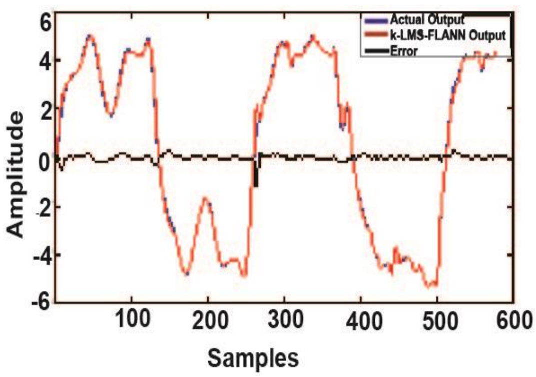

Fig.11 Error between actual and estimated model.

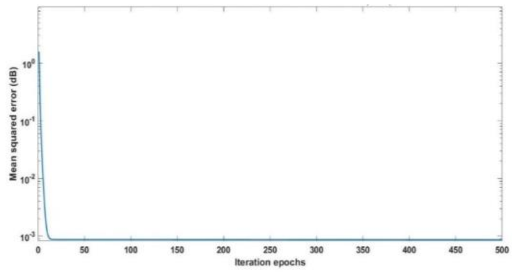

Fig.12. Cost function between actual and predicted model

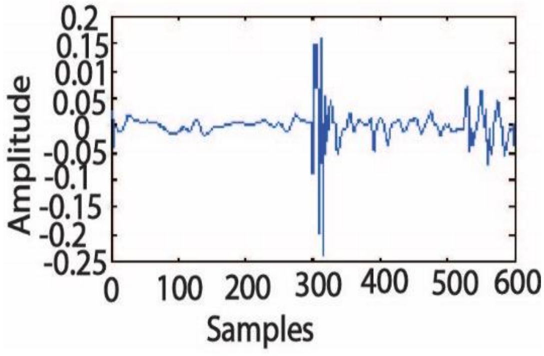

Fig.13 Error plot between actual and estimated model

Finally, a Kernel -Least Mean Square based FLANN model is designed to estimate the parameter of nonlinear signal. From figure Fig.9 it is visible that the model perfectly tracks the actual model's output. As compare to the previous models the estimated output is closer to actual output, from figure 11 and 14 the error between actual estimated model is depicted. Mean spurred error is considered as a stranded parameter to verify the superiority of the proposed model. The MSE plot is seen from the Fig.12.From the plot the errors are gradually decease with time. Table 1 compiles the mean squared errors (MSE) of all the cases. Specifically, it demonstrates how well the proposed evaluation scheme works.

Table 1: Performance of Nonlinear models

Models	***MSE***
SLIDING MODE	0.0127
FLANN	0.1003
RBFN	0.0201
Proposed Method (k-LMS_FLANN)	0.0012

From the Table 1 the comparison of different model is done based on MSE. The sliding mode model achieve 0.0127 of MSE and the FLANN model achieve 0.1003 MSE.As compare to sliding mode model FlANN model is less accurate. RBFn model achieves 0.0201 MSE. As compare to FLANN model RBF model has less MSE and more accurate model then FLANN for parameter estimation. Similarly, from the comparison between Sliding mode control model and RBFn model, the Siding mode control model has less MSE error and it is best as compare to FLANN and RBF model in case of parameter estimation of nonlinear systems. Finally, after applying Kenel least mean as mixed kernel method to the FLANN, the model become robust and there is drastically improvement. In table 2 the performance of proposed kernel is compared against different kernel and the result is analyzed.

Table :2 Performance different kernel methods

Kernels Methods	MSE
Cosine Kernel	0.035
Gaussian Kerel	0.052
Proposed Mixed Kernel	0.0012

Initial stage cosine kernel is applied to the proposed nonlinear model. Where the model achieves 0.035 MSE, then the gaussian kernel is applied to the model and archives 0.52 MSE. From this observation it is clear that implementing a kernel can made a model robust and accurate. Finally, a fusion kernel is designed by combine cosine kernel with gaussian. The proposed kernel is named as Mixed kernel. So, a mixed kernel is proposed with FLANN as a nonlinear model for identification and control of nonlinear plants. The result achieve by the Proposed K-LMS FLANN is better in compare to existing base models as well as different kernel models. The MSE of the model is 0. 0012.From the comparison it is proved that the proposed model is efficient and has a good parameter estimation ability from other nonlinear models.

7. Conclusion

A significant challenge in modern control systems is the detection of nonlinear plants with complicated structures. For this a perfect model for estimating the parameter of the nonlinear plant has to be developed. In this work initially a sliding mode control model is taken to verify its ability tom estimate the parameters of a nonlinear sinusoidal signal. Later RBF and FLANN models are verified. Finally, a Kernel least mean square Functional Link Artificial Neural network model as a nonlinear model is proposed. A stranded parameter as MSE is taken for the comparison between different model. From the comparison it is proved that proposed K-LMS FLANN which archive 0.0012 MSE is better compare to the base models. This model is accurate and can be applicable in complex nonlinear system identification problems. In future more complex nonlinear systems, Practial nonlinear and dynamics plant shall be verified to the propose model. There is room for experimentation and refinement in the FLANN method. Additionally, a recurrent neural network model can be used to evaluate the ANN.

Author contributions

Rakesh Kumar Pattanaik is the research scholar pursuing his Ph. D. work. He is working in the area of Signal Processing and Machine Learning. His contribution in this paper is to execute the program and experiments with the guide. Dr. Mihir Narayan Mohanty has given the concept and finalized the programs executed and verified the manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] L.Ljung, “Perspectives on system identification”. Annual Reviews in Control, vol-3, no.1, 2010, pp.1-12.

[2] A. K. Tangirala, “Principles of system identification: theory and practice.” Crc Press.2018.

[3] W. R. Cluett, “Principles of System Identification: Theory and Practice [Bookshelf].” IEEEControl Systems Magazine, Vol. 37, no-2, 2017, pp.181-184.

[4] J. Benesty, & Y. Huang, “Adaptive signal processing: applications to real-world problems”, Springer Science & Business Media, 2013.

[5] G. John, & M. R. Charles, “Algorithms for statistical signal processing,” Prentice Hall, London,2002.

[6] A. Feuer, & E. Weinstein, “Convergence analysis of LMS filters with uncorrelated Gaussian data,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol-33, no.1,1985, pp-222-230.

[7] Y. Zou, S. C. Chan, & T. S., Ng. “Least mean M-estimate algorithms for robust adaptive filtering in impulse noise,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol-47, no-12, 2000, pp-1564-1569.

[8] Y.Zhou, S. C. Chan, & K. L. Ho, “New sequential partial-update least mean M-estimate algorithms for robust adaptive system identification in impulsive noise,” IEEE Transactions on Industrial Electronics, vol-58, no.9, 2010, pp- 4455-4470.

[9] P. Song, & H. Zhao “Affine-projection-like M-estimate adaptive filter for robust filtering in impulse noise,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol-66, no.12, 2019, pp- 2087-2091.

[10] G.Wang & H. Zhao, “Robust adaptive least mean M-estimate algorithm for censored regression,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021.

[11] L. Lu, L. Chen, Z. Zheng, Y. Yu & X. Yang, “Behavior of the LMS algorithm with hyperbolic secant cost,” Journal of the Franklin Institute, vol-357, no.3, 2020, pp- 1943-1960.

[12] W. Gao, J. Chen, C. Richard, J. C. M Bermudez, & J. Huang, Convergence analysis of the augmented complex klms algorithm with pre-tuned dictionary, IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP,2015, pp. 2006-2010. IEEE.

[13] L. Lu, H. Zhao, & B. Champagne, “Distributed Nonlinear System Identification in $\alpha $-Stable Noise.” IEEE Signal Processing Letters ,2018, vol-25, no-7, pp- 979-983.

[14] S. Jain, & S. Majhi, “Zero-Attracting Kernel Maximum Versoria Criterion Algorithm for Nonlinear Sparse System Identification”. *IEEE Signal Processing Letters*,2022, vol-29, pp-1546-1550.

[15] H. Sira-Ramirez, E.Colina-Morles, and E. Rivas- "Francklin 'Slidingmode-based adaptive learning in dynamical-filter-weights neurons', International Journal of Control,73:8,678-685, 2000.

[16] B. Widrow, and M. A., Lehr, 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proceedings IEEE, vol-78, pp-1415-1442,1990.

[17] H. Sira-Ramlirez and E. Colina-Morles, "A sliding mode strategy for adaptive learning in adalines.IEEE Transactions on Circuits and Systems Ð I: Fundamental Theory and Applications", 1995.vol-42, pp-1001-1012.

[18] D. Shang, X. Li, M. Yin, & F. Li, "Dynamic modeling and fuzzy compensation sliding mode control for flexible manipulator servo system. Applied Mathematical Modelling", 2022, 107, 530-556.

[19] X. Su, Y. Xu & X. Yang, "Neural network adaptive sliding mode control without overestimation for a maglev system. Mechanical Systems and Signal Processing, 2022, vol- 168, pp- 108661.

[20] N. Wang & F. Hao, "Event-triggered sliding mode control with adaptive neural networks for uncertain nonlinear systems. Neurocomputing," 2021, vol- 436, pp-184-197.

[21] H. Rabiee, M. Ataei & M. Ekramian, "Continuous nonsingular terminal sliding mode control based on adaptive sliding mode disturbance observer for uncertain nonlinear systems". Automatica, 2019, vol-109, pp-108515.

[22] R. K. Pattanaik, & M. N. Mohanty, "Nonlinear system identification for speech model using linear predictive coefficients based radial basis function. Journal of Information and Optimization Sciences," 2022. Vol-43no-5, 1139-1150.

[23] R. K. Pattanaik, S. K. Mohapatra, M. N. Mohanty, & B. K. Pattanayak, "System identification using neuro fuzzy approach for IoT application." *Measurement: Sensors*, 2022, pp- 100485.

[24] R. K. Pattanaik, B. K. Pattanayak, & M. N. Mohanty, "Use of multilayer recursive model for non-linear dynamic system identification." *Journal of Statistics and Management Systems*, vol-25, no-6,2022, pp-1479-1490.

[25] A.G. Parlos, K.T. Chong, A.F. Atiya, "Application of recurrent multilayer perceptron in modelling of complex process dynamics," *IEEE Trans. Neural Networks*, 2000, vol-5, pp. 255–266.

[26] B. Prasad, R. Kumar, & M. Singh, "Performance Analysis of Heat Exchanger System Using Deep Learning Controller". IJEER, 2022, vol-10, no-2, pp- 327-334.

[27] Kasiselvanathan, M., Lakshminarayanan, S., Prasad, J., Gurumoorthy, K. B., & Devaraj, S. A, "performance Analysis of MIMO System Using Fish Swarm Optimization Algorithm'". *International Journal of Electrical and Electronics Research*, 2022, Vol-10, no-2, pp- 167-170.

[28] R. K. Pattanaik, M. Sarfraz, & M. N. Mohanty, "Nonlinear System Parameterization and Control using Reduced Adaptive Kernel Algorithm. International Journal of Industrial Engineering & Production Research," 2022, Vol-33, no-4, pp-1-16.

[29] R. K., Pattanaik, & M. N. Mohanty, (2022, "Nonlinear System Identification Using Robust Fusion Kernel-Based Radial basis function Neural Network. *International Conference on Emerging Smart Computing and Informatics* (ESCI) IEEE,2022, pp. 1-5.

[30] N. Sahu and P. K. Dash "Development and implementation of a novel adaptive filter for nonlinear dynamic system identification". In International Journal of Automation and Control (IJAAC), 2011, Vol-5, No-2, pp-171-188.

[31] A. H. Sayed, Fundamentals of Adaptive Filtering, John Wiley, New Jersey, 2003.

[32] Simon S. Haykin, Adaptive Filter Theory, Prentice Hall, Upper Saddle River, NJ 07458, USA, fourth edition, 2002.

[33] Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Neural network design (2nd Edition), PWS Publishing Co. Boston, MA, USA©1996.