Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Efficiency of Kolmogorov-Arnold Networks for Nonlinear Systems Identification

Soumia Mohammed Djaouti[1], Rochdi Bachir Bouiadjra[1], Laouni Mahmoudi[1]
[1]*LISYS Lab, University of Mascara, 29000, Algeria*

ARTICLEINFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

The identification of nonlinear systems remains a critical challenge in control engineering, requiring models that balance accuracy. This study explores Kolmogorov-Arnold Networks (KAN) as a novel solution, leveraging their inherent ability to decompose multivariate functions into hierarchical univariate operations, a property aligned with the Kolmogorov-Arnold representation theorem. KANs excel in capturing complex nonlinearities through adaptive activation functions and sparse subnetwork compositions, offering superior interpretability and parameter efficiency compared to conventional architectures. We demonstrate KAN's efficacy by identifying three nonlinear systems spanning low to high complexity and nonlinear dynamics. The identified models are further validated in Model Reference Adaptive Control (MRAC), showcasing closed-loop stability and tracking performance. Experimental results reveal that KANs achieve lower identification error and lower parameters than MLP and LSTM, while maintaining comparable computational overhead.

Keywords: Kolmogorov-Arnold representation, Kolmogorov-Arnold Networks, Nonlinear systems identification, MLP, LSTM, MRAC.

INTRODUCTION

Nonlinear systems pervade various domains, including engineering and economics. These systems exhibit complex behaviors that cannot be described adequately by linear models. The nonlinearities inherent in these systems give rise to bifurcations, chaos, and limit cycles, which are very important in understanding and controlling their dynamics. Consequently, description and analysis are very important tasks in nonlinear system study, as applications range from the design of robust control approaches to complex behavior forecasting in systems [1].

The identification of nonlinear systems is a core issue in systems theory and control engineering [2-3]. Modeling means setting up a mathematical model describing the system behavior, whereas identification is the procedure to find the parameters or the structure of the model concerning measured data. A good model is very important for simulation, identification, and control design. However, all these tasks are substantially more complicated for nonlinear systems than for their linear counterparts [4].

Numerous methodologies have been created to tackle the difficulties associated with nonlinear system modeling and identification. Traditional methods include linearization around operating points, Volterra series expansions, and describing function analysis. While these techniques can be effective for certain classes of nonlinearities or within limited operating ranges, they often struggle to capture highly complex or global nonlinear behaviors. Moreover, these methods often require significant prior knowledge about the system's structure [4].

In the last few decades, neural networks, has grown as one of the most powerful tools for nonlinear system identification [5]. Neural networks can learn complex nonlinear mappings from input-output data without requiring any explicit knowledge of the underlying dynamics of the system [6]. The multilayer perceptron and recurrent neural networks, particularly LSTM [7-8], are among the various neural network architectures that have

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

become widely accepted for their ability to manage both static and dynamic systems. However, these architectures may suffer from such issues as vanishing/exploding gradients, interpretation difficulties, and high computational cost.

This paper aims at the effectiveness of the Kolmogorov-Arnold Networks regarding nonlinear systems identification problems. The KAN represents a completely new approach to function approximation, according to the Kolmogorov-Arnold representation theorem [9]: A continuous multivariate function can be expressed as a finite superposition of continuous functions of one variable. KANs leverage this theorem in representing complicated functions as simpler functions nested together, offering a potentially more efficient and interpretable representation of nonlinearities. We examine the use of KANs to identify two nonlinear systems that vary widely in complexity.

To benchmark the proposed approach, we evaluate how well KAN-based identification performs in comparison to models identified by MLPs and LSTM networks. The evaluation focuses on the accuracy of system identification using metrics such as MSE, MAE, andR^2, as well as the computational efficiency of KANs, which achieve competitive performance with fewer parameters than traditional neural networks. Two of the four identified nonlinear systems are further validated in a Model Reference Adaptive Control (MRAC) framework, demonstrating promising closed-loop tracking results. The experiments highlight KANs ability to capture complex nonlinear dynamics while maintaining interpretability and efficiency, positioning it as a viable alternative for nonlinear system modeling and control.

The remainder of this document is organized in the following way: We review related works in the fields of nonlinear identification and control in Section 2. Section 3 provides a detailed account of the architecture and corticale foundations of KANs. Section 4 demonstrates the application of KANs for the identification of two nonlinear systems that vary in complexity. Section 5 discusses the experimental results, which include comparative analyses with MLPs and LSTMs, as well as MRAC validation. The paper ends with a presentation of possible future research directions.

RELATED WORKS

Artificial Neural networks have become a cornerstone in nonlinear systems identification due to their universal approximation capabilities and adaptability to complex dynamical behaviors [10]. Among the most widely used architectures are MLPs, RBF networks, and LSTM networks, each offering unique advantages for modeling nonlinear systems.

MLP networks are frequently employed for input-output mapping in dynamical systems, demonstrating effectiveness in capturing complex nonlinear behaviors [11-12]. However, their performance is highly dependent on factors such as model architecture, learning rate, data quality, and training algorithms [13]. To mitigate these challenges, researchers have proposed regularized MLPs with radial activation functions [10] and hierarchical models for specific applications like species identification [11]. Additionally, Wavelet Neural Networks (WNNs) and Fuzzy Wavelet Neural Networks (FWNNs) have been explored, combining the flexibility of neural networks with the localized approximation capabilities of wavelets and fuzzy logic [14].

RBF networks have also proven effective in nonlinear systems identification, particularly due to their global approximation capabilities and robustness in dynamic system modeling [15-16]. They have been applied with success in fault detection and identification (FDI) through neuro-augmented observer designs [17] and multi-kernel fusion techniques to enhance performance [18]. Furthermore, RBF networks have been utilized in applications such as logistics UAV flight status prediction, showcasing their ability to handle nonlinear systems with high precision [19].

LSTM networks, a specific type of recurrent neural networks (RNNs), excel in modeling temporal dependencies and addressing the vanishing gradient problem, making them ideal for complex nonlinear dynamics [20]. They have been applied in diverse domains, including chemical processes, power systems, and underwater thrusters [21-23]. Innovative approaches, such as combining LSTM autoencoders with Normalizing Flows frameworks, have been proposed to extract temporal features and map them to system parameters [24]. Stacked LSTM networks have also been employed for multi-timestep predictions in large-scale dynamical systems, demonstrating improved accuracy over long prediction horizons [22].

While MLP, RBF, and LSTM networks have demonstrated significant success in nonlinear systems identification, our approach leverages Kolmogorov Artificial Networks (KANs), which offer a novel framework for capturing complex nonlinear relationships. Unlike traditional neural networks, KANs are designed to approximate functions with fewer parameters and greater interpretability, potentially addressing some of the limitations associated with MLPs, RBFs, and LSTMs. This makes KANs a promising alternative for nonlinear systems identification, particularly in scenarios requiring high accuracy and computational efficiency.

KOLMOGOROV-ARNOLD NETWORKS

This section starts with a review of the Kolmogorov-Arnold representation theorem, the foundation on which KANs are constructed. This approach is different from the traditional Multi-Layer Perceptron (MLP) in that it is focused on the unique properties of KANs.

The design of KANs is carried out upon, highlighting how they differ from MLPs. KANs use learnable activation functions on weights rather than fixed activation functions on neurons, which allows for greater flexibility and adaptability in learning.

3.1 Kolmogorov-Arnold Representation theorem

According to this theorem, any multivariate continuous function on a bounded domain can be represented as a finite composition of continuous single-variable functions combined through addition. This allows for a structured representation of complex functions [25], [9], [26-28].

Specifically, for a smooth function $f:[0,1]^n \to \mathbb{R}$, it can be represented in a form that involves a combination of simpler functions, which can enhance interpretability and approximation capabilities [25], [9]:

$$f(x_1, x_2, \cdots, x_n) = \sum_{q=1}^{2n+1} \Phi_q\left(\sum_{p=1}^{n} \phi_{q,p}(x_p)\right) \tag{1}$$

Where $\phi_{q,p}: [0, 1] \to \mathbb{R}$and $\Phi_q: \mathbb{R} \to \mathbb{R}$ are, respectively, the inner and the outer functions.

This original representation is limited to a depth-2 and width$-(2n + 1)$ structure, which may not capture all smooth functions effectively. However, extending this representation to deeper networks can potentially overcome limitations and improve expressiveness [9].

The theorem has been foundational in developing Kolmogorov-Arnold Networks (KANs), which leverage this representation to enhance neural network architectures and their training methodologies.

3.2 Kolmogorov-Arnold Network Architecture

The Kolmogorov-Arnold Networks (KANs) are developed based on the Kolmogorov-Arnold representation theorem, which permits the representation of multivariate continuous functions as compositions of simpler univariate functions combined by addition.

KANs can be structured with varying depths and widths, providing flexibility and the potential to improve the expressive power of the network beyond the original depth-2 and width$-(2n + 1)$ representation.

According to the Kolmogov-Arnold theorem, the inner functions create a KAN layer with n nodes as inputs and $2n + 1$ nodes as outputs, while the outer functions establish a KAN layer with $2n + 1$ nodes as inputs and 1 node as output. So to build a deeper KAN with L layers, the authors in [9] have merged inner and outer functions to spline-based univariate functions $\phi_{l,j,i}$.

$$\Phi_l = \{\phi_{l,j,i}\}, i = 1, \cdots, n_l, j = 1, \cdots, n_{l+1}, l = 0, \cdots, L-1 \tag{2}$$

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

where n_l the number of nodes in the l-th layer. The i-th input in the l-th layer is denoted by $x_{l,i}$ and the connexion between i-th node in layer l and j-th node in layer l+1 is:

$$x_{l+1,j} = \sum_{i=1}^{n_l} \phi_{l,j,i}\left(x_{l,i}\right), \;\; j = 1, \cdots n_{l+1} \tag{3}$$

The compact form of (3) is:

$$\mathrm{x}_{\mathrm{l}+1} = \begin{pmatrix} \phi_{\mathrm{l},1,1}(\cdot) & \phi_{\mathrm{l},1,2}(\cdot) & \cdots & \phi_{\mathrm{l},1,\mathrm{n_l}}(\cdot) \\ \phi_{\mathrm{l},2,1}(\cdot) & \phi_{\mathrm{l},2,2}(\cdot) & \cdots & \phi_{\mathrm{l},2,\mathrm{n_l}}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{\mathrm{l},\mathrm{n}_{\mathrm{l}+1},1}(\cdot) & \phi_{\mathrm{l},\mathrm{n}_{\mathrm{l}+1},2}(\cdot) & \cdots & \phi_{\mathrm{l},\mathrm{n}_{\mathrm{l}+1},\mathrm{n_l}}(\cdot) \end{pmatrix} \tag{4}$$

$$= \Phi_{\mathrm{l}}(\mathrm{x}_{\mathrm{l}})$$

Where $\mathrm{x}_{\mathrm{l}} = \left[\mathrm{x}_{\mathrm{l},1}, \mathrm{x}_{\mathrm{l},2}, \cdots, \mathrm{x}_{\mathrm{l},\mathrm{n_l}}\right]^{\mathrm{T}}$.

A KAN network constitutes a succession of L layers. When provided with an input vector $x_0 \in \mathbb{R}^{n_0}$ the resultant output can be expressed as

$$KAN(\boldsymbol{x}) = (\Phi_{L-1} \circ \Phi_{L-1} \circ \cdots \circ \Phi_1 \circ \Phi_0)(\boldsymbol{x_0}) \tag{5}$$

A KAN architecture with 2 inputs and 1 output and 2 layers with respectively 5 and 2 nodes is illustrated in figure 1.

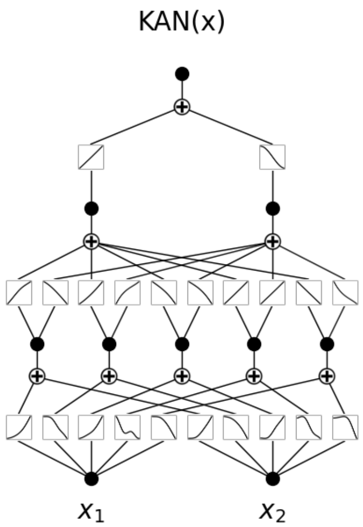


Figure.1 KAN architecture

3.2.1 Learning parameters in KAN

The one-dimensional functions $\phi_{l,j,i}$ is approximated using B-splines functions. A spline is defined as a continuous curve distinguished by a specified set of control points or knots. Spline functions are commonly utilized to interpolate or approximate data points in a manner that is both smooth and continuous. A spline is characterized by its order k, which relates to the degree of the polynomial functions employed for interpolation or approximation of the curve between the specified control points. The number of intervals, denoted as G, indicates the count of segments or subintervals that exist between consecutive control points.

The one-dimensional functions, $\phi_{l,j,i}$ is assumed to be the sum of the basis function $b(x)$ and the spline function $spline(x)$ [9]:

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

$$\phi(x) = w_b b(x) + w_s spline(x) \tag{6}$$

where $b(x) = x/(1 + e^{-x})$ and $spline(x)$ is the sum of B-spline functions recursively determined by the Boor-Cox formula: $spline(x) = \sum_i c_i B_i(x)$.

Learnable parameters of KAN comprise the B-spline control points, shortcut weights, B-spline weights, and bias.

KOLMOGOROV-ARNOLD NETWORKS FOR NONLINEAR SYSTEMS IDENTIFICATION

4.1 Nonlinear systems identification

Nonlinear system identification is the process of building mathematical models to describe the dynamic behavior of systems that exhibit nonlinearities, enabling accurate identification, control, and analysis of complex real-world processes. The identification process is simply illustrated by the figure 2.

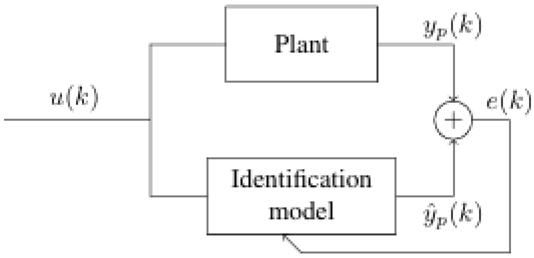


Figure. 2 Bloc diagram of identification

Below is a detailed explanation of the most common structures used in nonlinear system identification [3], [29].

4.1.1 Nonlinear Autoregressive with Linear Exogenous Inputs (NARX-LIN)

This model combines a nonlinear autoregressive (NAR) component with a linear exogenous input contribution. It is suitable for systems where the nonlinearity is primarily driven by past outputs, and the input influence can be approximated linearly. It is simpler to identify and interpret, requiring fewer parameters compared to fully nonlinear models.

$$y_p(k+1) = \sum_{i=0}^{m-1} \beta_i u(k-i) + f\big(y_p(k), y_p(k-1), \cdots, y_p(k-n+1)\big) \tag{7}$$

4.1.2 Nonlinear Autoregressive with Nonlinear Exogenous Inputs (NARX-NLN)

This model combines a nonlinear autoregressive (NAR) component with a nonlinear exogenous input contribution. It is suitable for systems that capture nonlinear effects of both past outputs and inputs. It is simpler to identify and interpret, requiring fewer parameters compared to fully nonlinear models [30].

$$y_p(k+1) = f\big[y_p(k), \cdots, y_p(k-n+1)\big] + g[u(k), \cdots, u(k-m+1)] \tag{8}$$

with f and g being unknown nonlinear functions that represent the plant dynamics.

4.1.3 Fully Nonlinear Autoregressive with Exogenous Inputs (NARX-FULL)

This model represents a fully nonlinear NARX structure, where a single nonlinear function captures the combined influence of past outputs and inputs. It is the most general and flexible, suitable for highly nonlinear and coupled systems, but requires significant computational resources and large datasets for training.

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

$$y_p(k+1) = f\left[y_p(k), \cdots, y_p(k-n+1), u(k), \cdots, u(k-m+1)\right] \tag{9}$$

4.2 Nonlinear systems identification using KAN

According to section 4.1, the identification problem involves establishing an appropriately parameterized identification model and modifying the model parameters to maximize a performance function that depends on the error between the outputs of the identification model and the system. The structure of the identification model is chosen to be the same as that of the system. As the non-linear functions in equations (7-9) are unknown, we aim to estimate them using a Kolmogorov-Arnold network (KAN). In the section below, several plants corresponding to the models are subjected to KAN identification. The identification process using KAN is simply illustrated by figure 3.

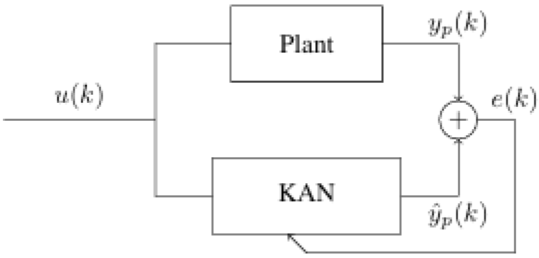


Figure. 3 Bloc diagram of identification using KAN

4.3 KAN Identification for Model Reference Adaptive Control

The Model Reference Adaptive Control (MRAC) framework serves as an excellent validation platform for our Kolmogorov-Arnold Network (KAN) identification approach, where it achieves trajectory tracking by synthesizing a control law $u(k)$ that drives the plant output $y_p(k)$ to match the reference model output $y_m(k)$, ensuring $\lim_{k\to\infty}\left|y_p(k) - y_m(k)\right| = 0$. The accuracy of this control critically depends on the precision of the plant model identification - any mismatch between the identified model and true plant dynamics directly degrades tracking performance and may undermine closed-loop stability. In this architecture, the Kolmogorov-Arnold Network (KAN) serves as the fundamental modeling engine, tasked with learning the plant's complex nonlinear dynamics in real-time. The KAN's approximation capability is paramount: its ability to accurately capture the plant's input-output mapping determines the control system's ultimate tracking accuracy and robustness. Higher identification fidelity translates to smaller asymptotic errors and improved disturbance rejection. The control architecture is depicted in Figure. 4 where $q^{-d} \cdot x(k) = x(k-d)$. For foundational MRAC theory, see [2], [3].

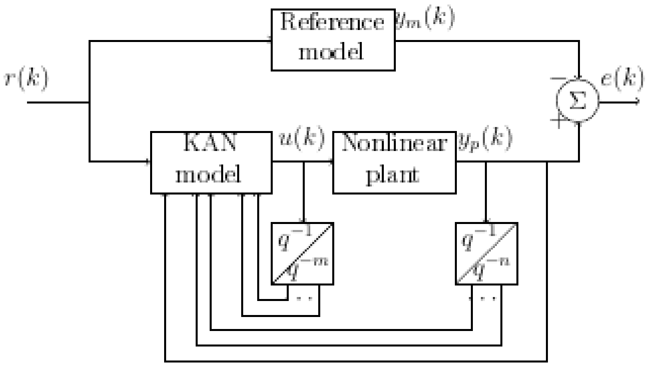


Figure. 4 Bloc diagram of KAN-based MRAC

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

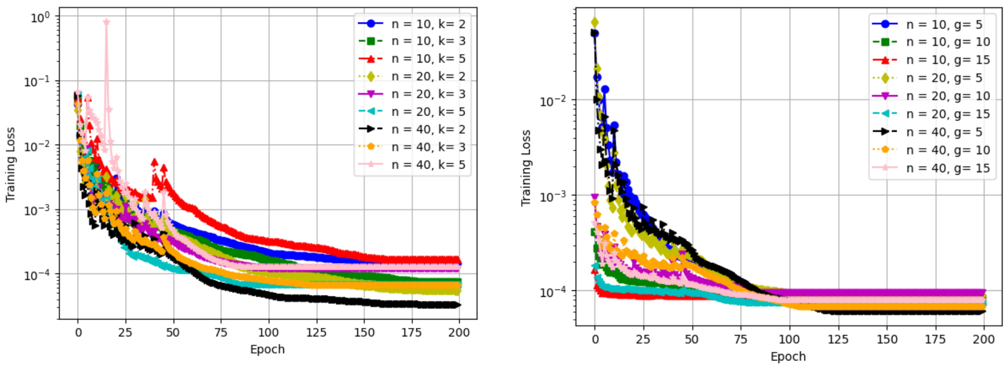
Research Article

EXPERIMENTAL RESULTS

5.1 Experiments settings

This section discusses the results of identifying nonlinear systems using NARX structure selection. The number of nodes, hidden layers, grids and the B-spline function order in the KAN are adjustable parameters. In order to find the optimal parameters of the KAN, we have carried out experiments with varying values for nodes number, grid size and B-spline order to identify models in section 4.1. For example, the results for the ARX-NLN cited below are depicted in figure 5. The best performance is observed in configurations that combine a high node countwith a small grid size, and quadratic B-spline functions. However, when more trainable parameters are used, this improved performance may come at the expense of increased computational times and longer training periods.

To measure the ability of the KAN to predict output NARX models we use a variety of metrics, including MSE, MAE and R^2-score analysis.



(a) Loss values for different nodes number and B-splines order

(b) Loss values for different nodes number and grid size

Figure. 5. Hyperparameters grid-search

5.2 KAN Identification Results

Three plants of nonlinear systems matched to NARX forms are included below to demonstrate the ability of KAN as identifier.

5.2.1 NARX-LIN Model:

In this case a NARX-LIN model is considered, and the plant to be to be identified is given by the following second-order difference equation:

$$y_p(k+1) = f\left(y_p(k), y_p(k-1)\right) + u(k) \tag{10}$$

where

$$f\left(y_p(k), y_p(k-1)\right) = \frac{y_p(k)y_p(k-1)\left(y_p(k)+2.5\right)}{1+y_p^2(k)+y_p^2(k-1)}$$

The nonlinear function $f(\cdot)$ is assumed unknown and a KAN identifier is used to identify the output denoted by $\hat{y}_p$ and given by the difference equation:

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

$$y_p(k+1) = KAN\left(y_p(k), y_p(k-1)\right) + u(k) \tag{11}$$

The architecture of the KAN is [2, 5, 5, 1]. A dataset of 1000 values is created for $\{y_p(k), y_p(k)\}$ as input and $\{y_p(k+1) - u(k)\}$ as output using (10). In training phase, the KAN model parameters are maintained and $u(k)$ is taken as an i.i.d. random signal uniformly distributed in the interval [-1, 1]. To validate the KAN model, we use a sinusoidal input $u(k) = sin(2\pi k/25)$.The training loss curves for the three models under consideration are shown in Figure.6 With an MSE of 0.000121 (after 500 epochs), we can see how much better KAN-model is.

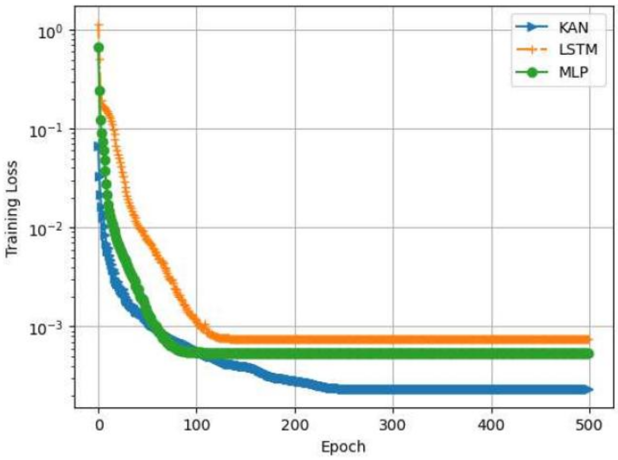


Figure. 6 NARX-LIN Model: Training loss curves with KAN, LSTM and MLP

Additionally, Figure.7 shows, for the input $u(k) = sin(2\pi k/25)$,the identification errors between plant's outputs and the identification model when (a) KAN (b) LSTM and (c) MLP identifiers are used. We can plainly see how well the KAN model is at confirming the identification of the sinusoidal input$u(k)$.

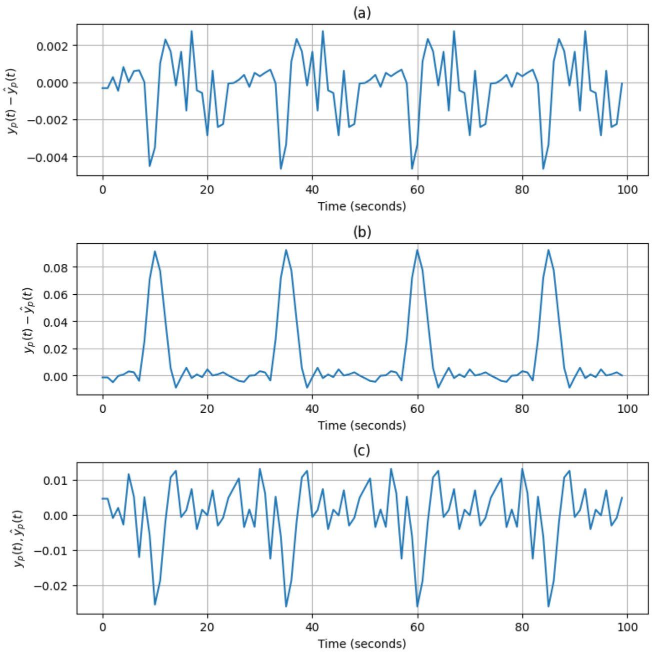


Figure. 7 NARX-LIN Model: (a) KAN: Identification error, (b) LSTM: Identification error, (c) MLP: Identification error.

The outcomes of NARX-LIN Model are compiled in Table 1. It shows the R^2-score, Mean Squared Error (MSE), Mean Absolute Error (MAE), and the number of trainable parameters for each model. KANs perform better than LSTM and MLP, according to an analysis of the error metrics; the KAN (2-depth) with the fewest trainable parameters performs the best. Its superior ability to forecast the ARX-NLN model with a higher R^2-score is indicated by its lower MSE and MAE values.

Table 1. NARX-LIN Model: Results summary of Identification metrics

Model	Nbr. Para.	MSE $\times 10^{-5}$	MAE $\times 10^{-4}$	R^2-Score
KAN [2,5,5,1]	**451**	**0.3087**	**12.650**	**0.99999**
LSTM [2, 50, 50, 1]	30651	88.763	144.29	0.99931
MLP [2, 20, 20, 1]	501	8.1329	66.60	0.99993

5.2.2 NARX-NLN Model:

In this experiment, a NARX-NLN plant described by (8) is considered:

$$y_p(k+1) = f\left(y_p(k)\right) + g\big(u(k)\big) \tag{12}$$

where: $f\left(y_p(k)\right) = y_p(k)/(1 + y_p^2(k))$ and $g\big(u(k)\big) = u^3(k)$.

Two KAN identifiers are used to estimate f and g and the output of the identification model is given by:

$$\hat{y}_p(k+1) = KAN_f\left(y_p(k)\right) + KAN_g\big(u(k)\big) \tag{13}$$

The architectures of KAN_f and KAN_g are [1, 10, 1]. Two datasets of 1000 values are created for each KAN and a random signal uniformly distributed in the interval [-1, 1] is employed during the training phase. The KANs are trained for the following parameters: $G = 5, k = 3$, max epochs = 500. To validate KAN identifiers, we use a sinusoidal signal as input defined by $u(k) = \sin(2\pi\, k/25) + \sin(2\pi\, k/10)$.

Figure 8 presents the training loss curves of KAN_f and KAN_g for the two models that are being examined. We can see how much better the KAN_g model is, which isn't the case for KAN_f model. This is due to the strong nonlinearity in f and the simplicity of the architecture of KAN_f.

Figure 9 display, for the input $u(k) = \sin(2\pi\, k/25) + \sin(2\pi\, k/10)$, the identification error when (a) KAN, (b) LSTM and (c) MLP identifiers are applied. The KAN model's ability to validate the identification of the sinusoidal input $u(k)$ is clearly seen. Table 2 compiles the results. It displays the number of trainable parameters for each model together with MSE, MAE, and R^2-score. These metrics show that kans outperform LSTM and MLP.

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

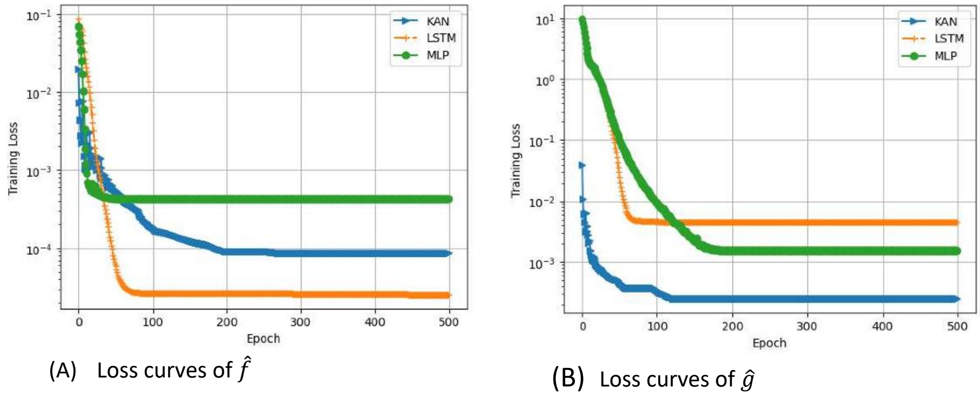


Figure.8 NARX-NLN Model: Loss curves of $\hat{f}$ and $\hat{g}$ with KAN, LSTM and MLP

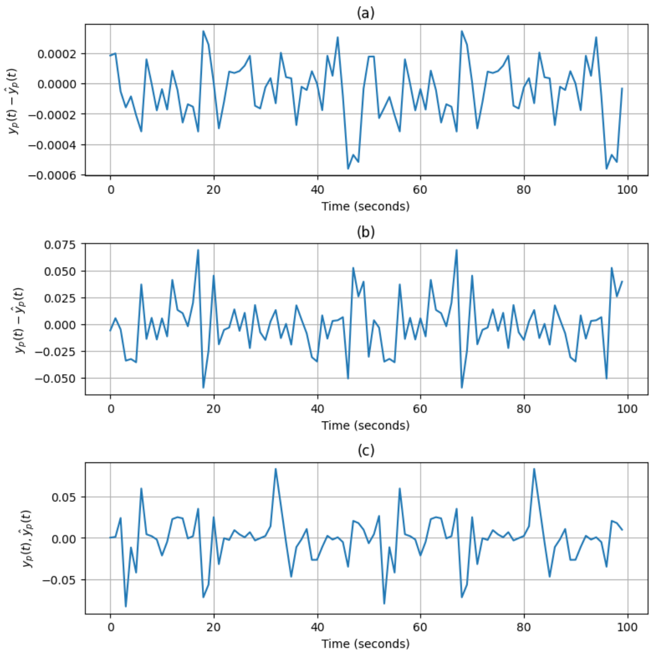


Figure.9 NARX-NLN model: (a) KAN: identification error, (b) LSTM: identification error, (c) MLP: identification error

Table 2. NARX-NLN Model: Results summary of Identification metrics

Model	Nbr. Para.	MSE $\times 10^{-5}$	MAE $\times 10^{-4}$	R^2-Score
KAN [1,10,1]	**231**	**0.0186**	**3.646**	**1.0**
LSTM [1, 50, 50, 1]	30651	343.76	505.99	0.99951
MLP [1, 20, 20, 1]	481	89.23	242.72	0.99987

5.2.3 NARX-FULL Model:

In the last experiment, we considered a plant that belongs to the NARX-FULL form. The plant in this instance is thought to be of the following form:

$$y_p(k+1) = f\left(y_p(k),\ y_p(k-1), y_p(k-2), u(k), u(k-1)\right) \tag{14}$$

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

With:

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2}$$

We will apply a KAN identifier to predict the output plant denoted by $\hat{y}_p$ and given by the difference equation (15):

$$\hat{y}_p(k+1) = KAN\left(y_p(k),\ y_p(k-1), y_p(k-2), u(k), u(k-1)\right) \quad (15)$$

where KAN has the architectures shape [5, 40, 1]. Firstly $u(k)$ is considered to be a random signal uniformly distributed in the interval [-1, 1], and 1000 values of $[y_p(k+1), y_p(k),\ y_p(k-1),\ y_p(k-2)\ , u(k), u(k-1)]$arecreated using (15) in order to train the KAN model. The following parameters are then used to train the KAN: $G = 5, k = 3$, max epochs = 500.

Figure 10 shows the training loss curves for MLP, LSTM, and KAN.

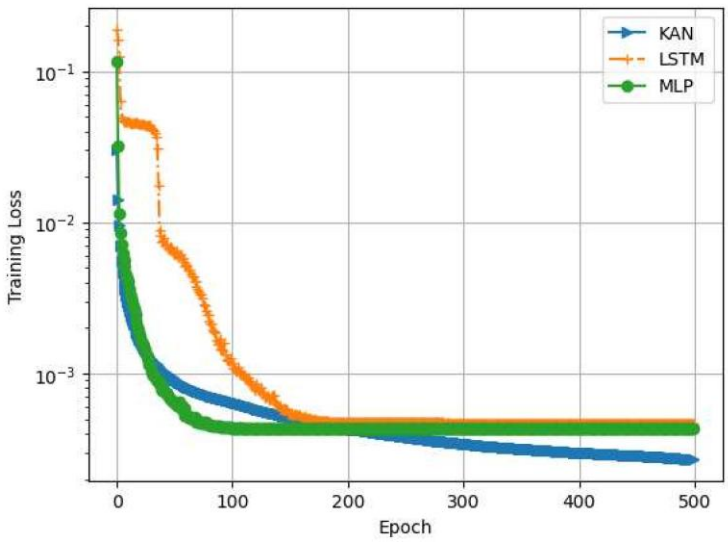


Figure. 10 NARX-FULL Model: Training loss curves with KAN, LSTM and MLP

Figure 11 displays, for the input $u(k) = 0.8 sin\,(2\pi\, k/250) + 0.2 sin\,(2\pi\, k/25)$, the identification error when (a) KAN, (b) LSTM and (c) MLP identifiers are utilized. A discernible superiority of the KAN model is evident in validating the identification for the sinusoidal input $u(k)$. Table 4 encapsulates the results derived from NARX-FULL Models. Upon examination of the error metrics, it becomes evident that KANs surpass the performance of LSTM and MLP, with the KAN (2-depth) exhibiting superior performance despite possessing the fewest trainable parameters.

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

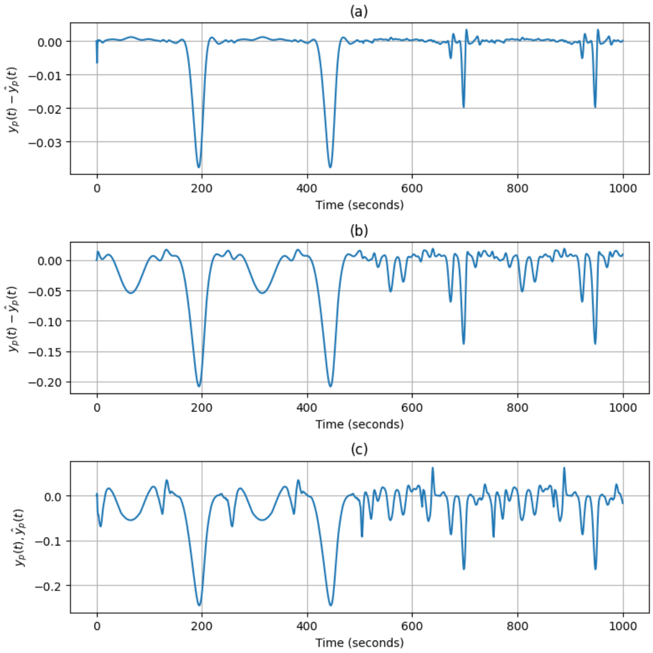


Figure. 11 NARX-FULL Model: (a) KAN: Identification error, (b) LSTM: Identification error, (c) MLP: Identification error

Table 3. NARX-FULL Model: Results summary of Identification metrics

Model	Nbr. para.	MSE $\times 10^{-5}$	MAE $\times 10^{-4}$	R^2-Score
KAN [5,40,1]	**2651**	**1.413**	**16.43**	**0.9999**
LSTM [5, 50, 50, 1]	30651	43.59	107.28	0.99809
MLP [5, 50, 50, 1]	2901	130.87	253.76	0.99458

5.3 Validation of KAN's identifier in MRAC

The formerly identified plants corresponding to NARX-LIN and NARX-NLN models are the subject of the application of KAN-based MRAC.

5.3.1 NARX-LIN Model

We hereby investigate the control of a plant belonging to the NARX-LIN Model, which is defined by the difference equation (13). The function f is assumed to be unknown and will be estimated by a KAN. A reference model is chosen to be stable and it is given by the following second-order difference equation:

$$y_m(k+1) = 0.6y_m(k) + 0.3y_m(k-1) + r(k) \tag{16}$$

Applying the control law

$$u(k) = -f\left(y_p(k), y_p(k-1)\right) + 0.6y_p(k) + 0.3y_p(k-1) + r(k) \tag{17}$$

to the system yields the following difference equation $e(k+1) = 0.6e(k) + 0.3e(k-1)$ where $e(k) = y_p(k) - y_m(k)$ and $\lim_{k\to\infty} e(k) = 0$.

However, the fact that f is unknown, its estimated $\hat{f}$ is replaced by a KAN network and therefore the control law

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

becomes:

$$u(k) = -KAN\left(y_p(k), y_p(k-1)\right) + 0.6y_p(k) + 0.3y_p(k-1) + r(k) \tag{18}$$

We should point out that the identification phase, i.e. estimation of $\hat{f}$ is carried out off-line, and it's done like in the previous section. When the reference input is $r(k) = \sin(2\pi k/25)$, we get the response of the plant depicted in figure 12. This leads us to conclude that KAN is able to track the response of the reference model with a MSE=0.0051. To evaluate the robustness of the KAN-based Model Reference Adaptive Control (MRAC) system, a disturbance was introduced to the output of the plant, denoted as $d(k)$, at the discrete time step $k = 60$. The disturbance signal was selected as a random signal uniformly distributed within the interval $[-0.5,0.5]$. In Figure 13, it can be inferred that the plant continues to track the reference model subsequent to the introduction of the disturbance signal $d(k)$.

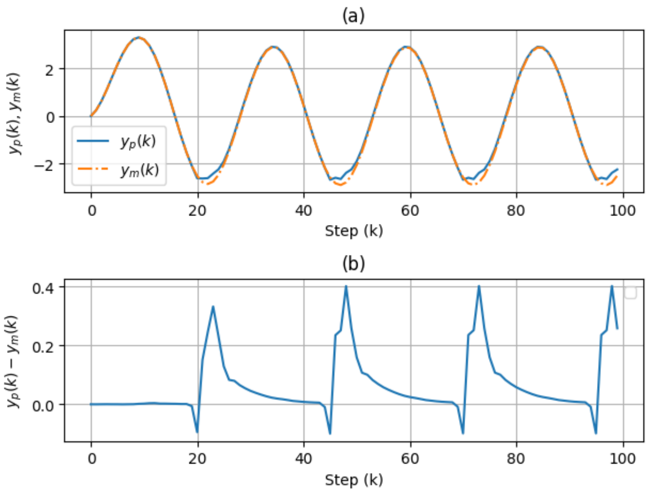


Figure. 12 NARX-LIN Model: (a) Output's plant y_p and reference model y_m with KAN-MRAC. (b) Tracking error

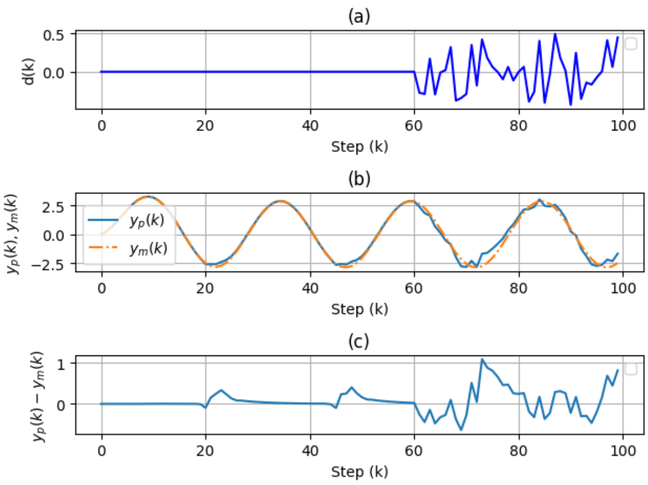


Figure. 13NARX-LIN Model: (a) disturbance signal $d(k)$ (b) Output's plant y_p and reference model y_m with KAN-MRAC. (c) Tracking error

5.3.2 NARX-NLN Model

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

We are now interested by the control of a NARX-NLN plant given by (9) where f and g are assumed to be unknown functions. A reference model is chosen to be stable and it is given by the following first-order difference equation:

$$y_m(k+1) = 0.6y_m(k) + r(k) \tag{19}$$

The aim of the control is again to have a tracking error $e(k) = y_p(k) - y_m(k)$ such that $\lim_{k\to\infty} e(k) = 0$. The proposed control law is given by:

$$u(k) = g^{-1}\left(-f\left(y_p(k)\right) + 0.6y_p(k) + r(k)\right) \tag{20}$$

Applying the control law $u(k)$ to the plant yields the following difference equation $e(k+1) = 0.6e(k)$ and $\lim_{k\to\infty} e(k) = 0$.However, the fact that f and $g^{-1}$are unknown, they are replaced by two KAN networks and therefore the control law becomes:

$$u(k) = KAN_{g^{-1}}\left(-KAN_f\left(y_p(k)\right) + 0.6y_p(k) + r(k)\right) \tag{21}$$

The same parameters used in previous identification of the model for training KAN_f and KAN_g are maintained in the case of training KAN_f and $KAN_{g^{-1}}$ in (21). The outputs $y_p(k)y_m(k)$ and the tracking error $e(k)$ are illustrated in figure 14 with the reference input $r(k) = \sin(2\pi k/25) + \sin(2\pi k/10)$. We can plainly see how well the KAN-based MRAC method is at confirming the tracking of output reference model with MSE = 0.000679.

We therefore draw the conclusion that, the KAN-based MRAC approach is feasible to provide a control input for an unidentified plant in order to achieve nearly flawless model tracking.

In order to test the robustness of KAN-based MRAC approach the same disturbance signal is injected to the plant at the discrete time step $k = 60$. Figure 15 illustrates the system's response, tracking error, and the effectiveness of the disturbance rejection.

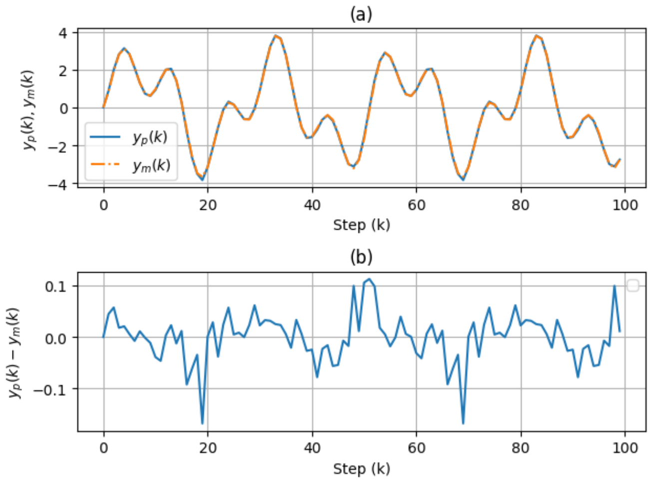


Figure. 14. NARX-NLN Model: (a) Output's plant y_p and reference model y_m with KAN-MRAC. (b) Tracking error

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

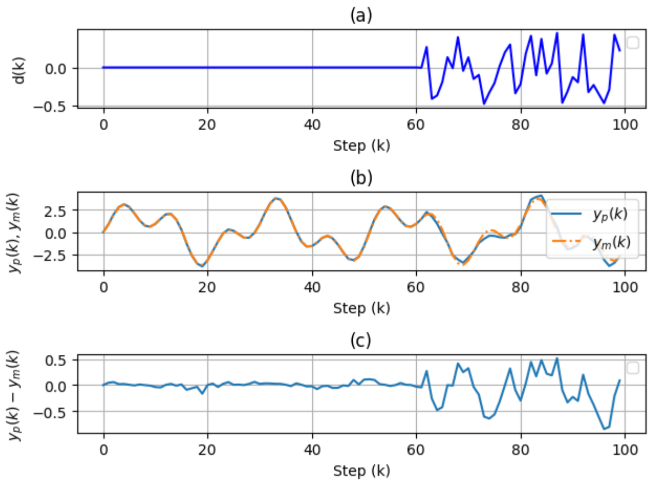


Figure. 15. NARX-NLN Model: (a) disturbance signal $d(k)$ (b) Output's plant y_p and reference model y_m with KAN-MRAC. (c) Tracking error

CONCLUSION

This research illustrates how Kolmogorov-Arnold Networks (KANs) can effectively tackle the difficulties of nonlinear system identification. KANs offer a new method for approximating intricate nonlinear functions by superimposing simpler univariate functions, drawing on the theoretical basis of the Kolmogorov-Arnold representation theorem. This architecture provides a significant alternative to conventional neural network models such as MLPs and LSTMs, as it boosts interpretability and computational efficiency. The experimental findings from four nonlinear systems of different complexities demonstrate that KANs attain competitive accuracy regarding MSE, MAE, and R^2, while utilizing fewer parameters. This highlights their promise for efficient and scalable system identification.Furthermore, the successful application of KAN-identified models in a Model Reference Adaptive Control (MRAC) framework highlights their practical utility in real-world control systems. The closed-loop tracking performance validates that KANs can reliably capture and replicate nonlinear dynamics, making them suitable for advanced control applications. The combination of theoretical rigor, interpretability, and computational efficiency positions KANs as a promising tool for future research in nonlinear system identification and control. Future work could explore the extension of KANs to higher-dimensional systems, robustness under noise and uncertainties, and integration with other adaptive control strategies to further broaden their applicability. Overall, this study contributes a significant step forward in the field of data-driven modeling and control, showcasing KANs as a versatile and efficient framework for tackling nonlinear system challenges.

REFRENCES

[1] Khalil, H. K. (2002). Nonlinear Systems. Prentice Hall, 3rd edition.
[2] Ljung, L. (1999). System Identification: Theory for the User. Prentice Hall, 2nd edition.
[3] Narendra, K. S., Parthasarathya, K. (1989). Identification and Control of Dynamical Systems using Neural Networks, IEEE Transaction on Neural Networks, 1(1): 4-27. https://doi.org/10.1109/72.80202.
[4] Slotine, J. J. E., Li, W. (1991 Applied Nonlinear Control. Prentice Hall.
[5] Forgione, M., Piga, D. (2021). Continuous-time system identification with neural networks: Model structures and fitting criteria, European Journal of Control, 59: 69-81. https://doi.org/10.1016/j.ejcon.2021.01.008.
[6] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning, MIT Press.
[7] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory, Neural Comput., 9(8): 1735-1780. https://doi.org/ 10.1162/neco.1997.9.8.1735.
[8] Gonzalez, J., Yu, W. (2018). Non-linear system modeling using LSTM Neural Network, IFAC PapersOnline, 51(13): 485-489. https://doi.org/10.1016/j.ifacol.2018.07.326.

[9] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljacic, M., Hou, T. Y., Tegmark, M. (2024). KAN: Kolmogorov–Arnold Networks, arXiv.org. https://doi.org/10.48550/arXiv.2404.19756.
[10] Lerro, A., Gili, P., Fravolini, M. L., Napolitano, M. (2021). Experimental Analysis of Neural Approaches for Synthetic Angle-of-Attack Estimation, International Journal of Aerospace Engineering, https://doi.org/ 10.1155/2021/9982722.
[11] Sueker, M., Daghighi, A., Akhbardeh, A., Mackinnon, N. , Bearman, G., Baek, I. (2023). A Novel Machine-Learning Framework Based on a Hierarchy of Dispute Models for the Identification of Fish Species Using Multi-Mode Spectroscopy, Sensors, https://doi.org/ 10.3390/s23229062.
[12] Ogunmolu, O. P., Gu, X., Jiang, S. B., Gans, N. (2016). Nonlinear Systems Identification Using Deep Dynamic Neural Networks, arXiv.org. https://doi.org/ 10.48550/arXiv.1610.01439.
[13] Yousefian, R., Kamalasadan, S. (2017). A Review of Neural Network Based Machine Learning Approaches for Rotor Angle Stability Control, arXiv.org. https://doi.org/10.48550/arXiv.1701.01214.
[14] Linhares, L. L. S., Fontes, A. I. R., Martins, A., Araújo, F. M. U, Silveira, L. (2015). Fuzzy Wavelet Neural Network Using a Correntropy Criterion for Nonlinear System Identification, Mathematical Problems in Engineering, "Hindawi Publishing Corporation, https://doi.org/ 10.1155/2015/678965.
[15] Salem, M., Khelfi, M. F. (2012). Online RBF and Fuzzy Based Sliding Mode Control of Robot Manipulator, In. 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 896-901, https://doi.org/ 10.1109/SETIT.2012.6482033.
[16] Zhang, M., Zhang; J., Hou, A., Xia; A., Tuo, W. (2022). Dynamic System Modeling of a Hybrid Neural Network with Phase Space Reconstruction and a Stability Identification Strategy, Machines, https://doi.org/ 10.3390/machines10020122.
[17] Gong, H., Zhen, Z. (2012). A Neuro-Augmented Observer for Robust Fault Detection in Nonlinear Systems, Mathematical Problems in Engineering, "Hindawi Publishing Corporation, https://doi.org/ 10.1155/2012/789230.
[18] Atif, S.M., Khan, S., Naseem, I. et al. (2023). Multi-Kernel Fusion for RBF Neural Networks. Neural Process Lett55:1045–1069. https://doi.org/10.1007/s11063-022-10925-3.
[19] Yang, Q., Ye, Z., Li, X., Wei, D., Chen, S., Li, Z. (2021). Prediction of Flight Status of Logistics UAVs Based on an Information Entropy Radial Basis Function Neural Network, Sensors, https://doi.org/ 10.3390/s21113651.
[20] Bonassi, F., Oliveira da Silva, F. C., Scattolini, R. (2021). Nonlinear MPC for Offset-Free Tracking of Systems Learned by GRU Neural Networks, IFAC-PapersOnLine, 54(14): 54–59.
[21] Kemeth, F. P., Bertalan, T. S., Evangelou, N., Cui, T., Malani, S., Kevrekidis, I. (2021). Initializing LSTM Internal States via Manifold Learning, Chaos, 31(9), https://doi.org/10.1063/5.0055371.
[22] Plaster, B., Kumar, G. (2019). Data-Driven Predictive Modeling of Neuronal Dynamics using Long Short-Term Memory, Algorithms, 2(10), https://doi.org/ 10.3390/a12100203.
[23] Nascimento, S., Valdenegro-Toro, M. (2018). Modeling and Soft-fault Diagnosis of Underwater Thrusters with Recurrent Neural Networks, IFAC-PapersOnLine, 51(29): 80-85, https://doi.org/ 10.1016/j.ifacol.2018.09.473.
[24] Rostamijavanani, A., Li, S., Yang, Y. (2025). Data-driven Identification of Nonlinear Dynamical Systems with LSTM Autoencoders and Normalizing Flows, arXiv.org, https://doi.org/ 10.48550/arXiv.2503.03977.
[25] Kolmogorov, A. N. (1961). On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables, American Mathematical Society.
[26] Barasin, I., Bertalanic, B., Mohorcic, M., Fortuna, C. (2024). Exploring Kolmogorov-Arnold Networks for Interpretable Time Series Classification, arXiv.org, https://doi.org/ 10.48550/arXiv.2411.14904.
[27] Vaca-Rubio, C. J., Blanco, L., Pereira, R., Caus, M. (2024). Kolmogorov-Arnold Networks (KANs) for Time Series Analysis, arXiv.org, https://doi.org/ 10.48550/arXiv.2405.08790.
[28] Kich, V.A., Bottega, J.A., Steinmetz, R., Grando, R. B., Yorozu, A., Ohya, A. (2024). Kolmogorov-Arnold Networks for Online Reinforcement Learning, arXiv.org, https://doi.org/ 10.48550/arXiv.2408.04841.
[29] Billings, S. A. (2013). Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, Wiley, https://doi.org/ 10.1002/9781118535561.

Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[30] Leontaritis, I. J., Billings, S. A. (1985). Input-output parametric models for non-linear systems part I: deterministic non-linear systems, International Journal of Control, 41(2): 303–328, https://doi.org/10.1080/00207178508961129.