Journal of Innovative Technology, Vol. 7, No. 1, pp. 67-83, March 2025
http://doi.org/10.29424/JIT.202503_7(1).0007

A Comprehensive Review of Heuristic Approaches in Modelling, Analysis, and Identification of Nonlinear Systems

Faisal Altaf [1],Ching-Lung Chang [2], Naveed Ishtiaq Chaudhary [3*], Taimoor Ali Khan [4]

ABSTRACT

Nonlinear systems, commonly found in scientific research and engineering applications, present significant challenges due to their intricate and complex behavior, investigating the properties of non-linear systems in different scenarios, spanning over different types of nonlinear continuous and discrete time Multi-Input Multi-Output as well Single Input Single Output control systems with the help of modern computational heuristics. The review seeks to elucidate the distinguishing characteristics of these systems as well as the role, impact, and significance of the stochastic optimization computing paradigm based on evolutionary and swarming heuristic intelligence. In addition, this text describes how randomness significantly impacts the dynamics of such deterministic and stochastic nonlinear systems. Mathematical modeling approaches, which are rooted in the methodological foundations of ordinary differential equations and input-output models from an innovation studies perspective, may offer a conceptual framework to integrate these complex dynamics of nonlinear systems. This study comprehensively reviews the utilization of computational intelligence techniques, including genetic algorithms, particle swarm optimization, firefly algorithm, ant-colony optimization, simulated-annealing, tabu search optimizer, differential evolution heuristics, artificial-bee colony optimization, and Cuckoo Search for parameter estimation of nonlinear systems based on Hammerstein structure.

Keywords: Nonlinear systems; Heuristic computing; Computational intelligence

1. INTRODUCTION

Parameter estimation and system identification have long piqued the curiosity of scientists working in both linear and nonlinear fields. The interest in system identification concepts in various sectors of science and technology can be explained by their practical applications [1-4]. The primary goal of the systems identification task is to estimate the equivalent model that accurately replicates the behavior of the system. A standard system identification strategy often utilizes a gradient descent algorithm to update the model parameters [5-8]. The objective is to minimize the mean-square error (M-SE) among the model and system responses. The most common adaptive algorithms are those that are derived from the least-mean square (LMS) algorithm [9-14]. Several enhancements have been documented either to expedite the convergence or boost the precision of modeling [15-20]. In the system identification cases mentioned earlier, the error surface is typically multimodal. Using of conventional gradient-descent approach may result in a less than optimal solution [21]. One way to overcome the limitations of conventional gradient-descent algorithms for system identification is to reframe the task as optimization problem. This problem can then be solved using a structured stochastic search strategy, like swarm-based and evolutionary-computing algorithms [22-25]. Several studies have explored the use of SI and evolutionary-based computer techniques to identify linear and non-linear systems [26-29].

This paper aims to comprehensively review the utilization of computational intelligence techniques, including genetic algorithms (GAs), particle swarm optimization (PSO), firefly algorithm (FFA), ant-colony optimization (A-CO), simulated-annealing (SA), tabu search optimizer (TSO), differential evolution heuristics (DEH), artificial-bee colony optimization (ABCO), and Cuckoo Search algorithm (CSA), for the purpose of identifying Hammerstein systems. Furthermore, aside from the numerous offline applications of SI and evolutionary-bases computing algorithms that are based on the principles of system identification, Other interconnected domains of systems engineering, such as active noise control (ANC) systems, have also recorded multiple online case studies, including controller design [30-32]. An ANC system using PSO has been introduced in reference [33, 34]. Additionally, nonlinear variants of the system have been documented in references, and a multichannel version has also been described in [35-39]. An online system identification approach, basing quantum-behaved PSO, has been utilized to control the quality of ser-

Manuscript received October 21, 2024; revised December 2, 2024; accepted January 7, 2025.

1 Ph.D. Scholar, Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology, Taiwan, R.O.C.

2 Professor, Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Taiwan, R.O.C.

3* Assistant Professor (corresponding author), Future Technology Research Center, National Yunlin University of Science and Technology, Taiwan, R.O.C.
(email: chaudni@yuntech.edu.tw)

4 Masters Student, International Graduate School of Artificial Intelligence, National Yunlin University of Science and Technology, Taiwan, R.O.C.

vice provided by a web server, wireless networks and multiple engineering problems [40-44]. This work aims to provide readers with an overview of popular swarm and evolutionary computing methods are employed for solving non-linear system identification problems. For instance, the algorithms are visualized as a single flow diagram with numerous parallel paths, as depicted in Figure 1. The diagram includes not only GAs, DE algorithm, PSO along with cuckoo search algorithm (CSA), which is a good suggested scheme for swarm intelligence. The diagram represents the treatment of agents as chromosomes in the case of GA, vectors in DE, particles in PSO, and nests of cuckoo birds in the case of CSA. As indicated by the picture, the beginning procedures in all the algorithms under consideration are comparable and mostly involve a random placement of agents throughout the search space. The primary distinction is in the nature-inspired sources of inspiration that drove the evolution of these algorithms. Moreover, the schematic representation of experimental setup for the identification of a nonlinear system using heuristic technique is expressed in Figure 2.

The remaining portion of the paper is structured in the following manner. In Section 2, we provide a comprehensive discussion of the many classifications of nonlinear systems. Section 3 is a concise evaluation of the difficulties encountered in identifying nonlinear systems and the shortcomings of conventional methods. The application of heuristic algorithms for nonlinear system identification using block ARX models has been evaluated in Section 4, and the final conclusions have been presented in Section 5.

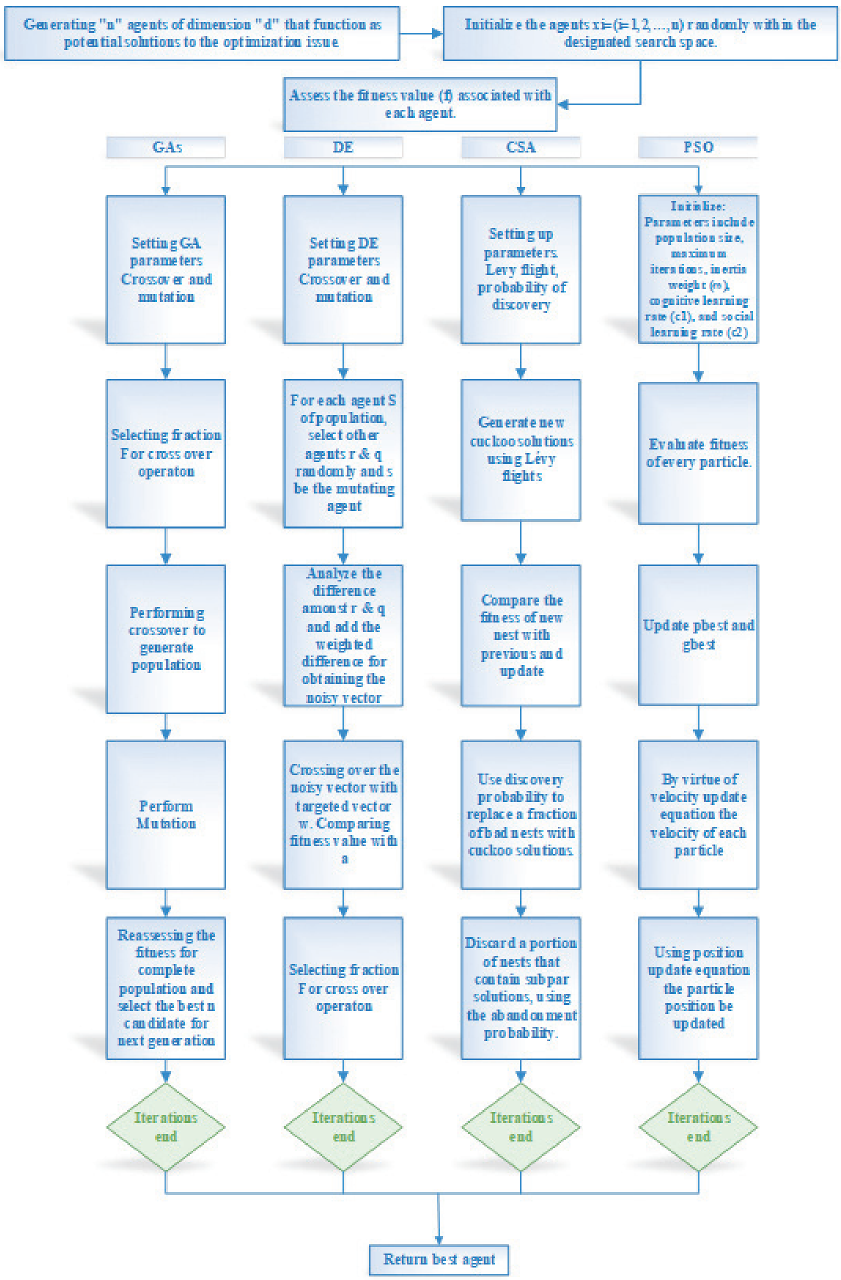

Fig. 1 Work Flow diagram of famous heuristics

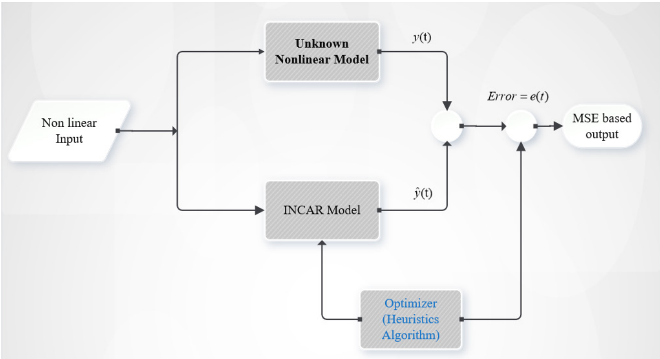

Fig. 2 Non-linear system identification using heuristics

2. CLASSIFICATION OF NONLINEAR SYSTEMS

Nonlinear systems can be classified into various types based on their mathematical properties, behavior, and characteristics [45]. Here are some common types of nonlinear systems, along with explanations for each:

2.1 Time-Invariant Nonlinear Systems

The dynamics of these systems do not change over time. The mathematical relationships governing the system remain constant. Time-invariant nonlinear systems are often encountered in many real-world applications, such as mechanical systems, electronic circuits, and chemical processes [46-51].

2.1.1 Property of Time Invariance:

Let x(t) be input while y(t) be the output. If the system is time-invariant, then for any constant time delay t_0 as shown in Equation 1:

$$y(t_i - t_0) = f(x(t_i - t_0)) \tag{1}$$

where f is the system's nonlinear function. This means delaying the input by t_0 simply delays the output by the same amount.

2.1.2 Mathematical Modeling

Nonlinear systems can be modeled using various techniques depending on the complexity. Here are two common approaches:

(i)Nonlinear Ordinary Differential Equations (ODEs): This method uses an equation system to describe the rate of change of the variables with in systems' state. The equations include nonlinear functions that represent the system's behavior as represented in Equation 2.

For example:

$$s^{\cdot}(t) = f\left(s(t),\, w(t)\right) \tag{2}$$

where s(t) is the state vector, w(t) is the input, f(.) is a nonlinear function representing the system dynamics.

(ii)Input-Output Model: This approach relates the system's output directly to the input without explicitly describing the internal states. The relationship in is expressed through a nonlinear function as shown in Equation 3.

$$q(t) = h\left(s(t),\, u(t)\right) \tag{3}$$

where q(t) is the output, h(.) is a nonlinear function that maps the state and input to the output.

2.1.3 Error Estimation:

Since real systems have imperfections, the actual output will deviate from the ideal output. This deviation is called the error. Here are two ways to estimate the error:

Norm-based Errors:

We can use norms (like L1 or L2 norm) to quantify difference in between the values desired output (y_d(t)) and the estimated output (y(t)) as expressed in Equation 4.

$$Error = \left\| y\_desired(t) - y\_estimated(t) \right\| \tag{4}$$

Frequency-domain Errors:

If the desired and actual outputs are in the form of signals, we can convert them to the frequency domain using tools like Fourier Transform. Determining the inaccuracy in the frequency domain involves comparing the desired-frequency and estimated-frequency spectra.

A time-invariant nonlinear system's block diagram usually has two primary blocks:

Non-linear Block (NLB): The NLB stands for the nonlinear function (f(.) or h(.)) of the system. It generates the output of the system (y(t)) from the inputs (u(t)) and (maybe) the state (x(t)).

Delay Block (DB): If the system adds a continuous time delay, a delay block can be added after the input to compensate for the delay in the output.

The block diagram will differ based on the selected modeling approach (state-space or input-output) and the existence of temporal delays.

Many branches of engineering, including circuit analysis, signal processing, and control systems, rely on a firm grasp of time-invariant nonlinear systems. Engineers are able to successfully build and analyze such systems by making use of mathematical models and error estimation techniques.

2.2 Time-Varying Nonlinear Systems:

These systems exhibit dynamics that change over time. The mathematical relationships governing the system evolve or are influenced by external factors or disturbances. Time-varying nonlinear systems are common in dynamic environments where system parameters, inputs, or operating conditions vary with time, such as communication networks, biological systems, and adaptive control systems [52-56].

Imagine a shock absorber in a car. It behaves differently depending on the speed and weight of the car (which are time-varying factors).

2.2.1 Mathematical Modeling

Modeling time-varying nonlinear systems is more complex than time-invariant ones. Here's a breakdown of two common approaches:

(i)Time-Varying Ordinary Differential Equations (ODEs):

This method is similar to time-invariant ODEs, but the nonlinear functions (f(.)) now explicitly depend on time (t) as represented in Equation 5:

$$\dot{s}(t) = f\left(s(t),\, w(t),\, t\right) \quad (5)$$

Here, the system's dynamics change with time, affecting how the state variables evolve.

(ii)Parameter-Varying Models:

This approach represents the system's nonlinearities using parameters that vary with time. These parameters are denoted by p(t) and included in the model equations as expressed in Equation 6:

$$q(t) = h\left(s(t),\, w(t),\, p(t)\right) \quad (6)$$

The key here is that p(t) captures the nature of the time-varying system.

2.2.2 Error Estimation

Error estimation remains vital for time-varying systems. The same approaches used for time-invariant systems (norm-based or frequency-domain) can be applied here. However, the interpretation might differ since the error can now change with time.

The block diagram for a time varying non-linear system is similar to the time-invariant case, with some key differences:

Time-Varying Nonlinear Block: The nonlinear block now explicitly shows the dependence on time (t), mathematically expressed as Equation 7:

$$f\left(s(t),\, w(t),\, t\right) or\, h\left(s(t),\, w(t),\, p(t)\right) \quad (7)$$

Time-Varying Elements: Depending on the specific system, additional blocks representing time-varying elements like time-dependent gains or filters might be included.

The complexity of the block diagram will depend on the chosen modeling approach and the specific time-varying characteristics of the system.

Understanding the time-varying nonlinear systems is important to analyze the systems whose behavior changes with time. This is crucial in fields like aerospace engineering, where flight dynamics are constantly changing, or in power systems, where load demands fluctuate.

2.3 Deterministic Nonlinear Systems

These systems respond based on their state variables and inputs without any randomness or uncertainty involved. Their behavior is described by deterministic mathematical equations often featuring non-linear functions. You will find such systems frequently in physics, engineering and mathematics, where their reactions are predictable and well defined [57-60].

Following are the preceding affects.

2.3.1 Time-Invariance vs. Time-Varying

System can either change over time or stay consistent:

- Time Varying Systems: It means that the system's behavior or characteristics can change as time goes on. Various factors can cause this such as external influences, time dependent non-linearities etc.
- Time-Invariant: As stated earlier, the system's behavior remains constant over time. The correlation between input and output remains consistent.

Both systems can be deterministic subject to the output that can be predicted for a starting condition and given input.

2.3.2 Mathematical Modeling

The modeling methods used to determine non-linear systems are the same as those used for general non-linear systems, as mentioned earlier.

- Non-Linear Ordinary Differential Equations: This approach uses a set of equations with non-linear function to measure how the system's variables change over time.
- Input-Output Model: In this method, the system's output is directly linked to its input through a function which is non-linear in nature, without stating the internal states.

The selection of these methodologies is contingent upon the particular system and the intended analysis.

2.3.3 Error Estimation

Similar to general nonlinear systems, error estimation in deterministic nonlinear systems aims to quantify the difference between the desired and actual outputs. The same techniques, such as norm-based errors or frequency-domain errors, can be applied.

In such systems, errors are not random; instead, they are due to things like imperfections in the model, external disturbances, or noise.

The block diagram for a deterministic non-linear system follows the already described structure.

Non-linear Block: This block represents a system non-linear function. It takes the input u(t) and possibly state x(t) and produces the system's output y(t).

Delay Block (Optional): A delay block can be included after the input if the system introduces a constant time delay.

The key point here is that the block diagram reflects the deterministic nature of the system, where the relationship between blocks is fixed and predictable.

Deterministic nonlinear systems are widely encountered in various fields. From the simple pendulum in mechanics to complex economic models, understanding their behavior is crucial for analysis, prediction, and control.

2.4 Stochastic Nonlinear Systems

These systems exhibit random fluctuations or uncertainties in their behavior. The system's dynamics are described by stochastic differential equations or probabilistic models, where randomness or uncertainty arises from external disturbances, noise, or inherent variability. Stochastic nonlinear systems are encountered in diverse fields, including finance, ecology, and signal processing, where randomness plays a significant role in system behavior [61-66].

Here's a breakdown of stochastic nonlinear systems in relation to the previous concepts:

Key Characteristic: Randomness

Unlike deterministic systems, stochastic nonlinear systems exhibit variability in their output for the same input due to random fluctuations. This randomness can be caused by:

Internal Noise: Unpredictable internal processes within the system can introduce random variations in the output.

External Noise: External factors with probabilistic behavior, like measurement noise or environmental disturbances, can affect the system's output.

2.4.1 Mathematical Modeling

Modeling stochastic nonlinear systems requires incorporating the element of randomness. Here are two common approaches:

(i)Stochastic Differential Equations (SDEs):

This method extends regular ODEs by including a term representing random noise. This term can be modeled using white noise or other noise processes.

(ii)State-Space Models with Stochastic Inputs:

This approach represents the system dynamics through state equations, but the input is considered stochastic.

Both methods allow us to analyze the statistical properties of the system's output, such as mean, variance, or probability distribution.

2.4.2 Error Estimation

Error estimation in stochastic nonlinear systems becomes more complex due to the inherent randomness. Some of the approaches are:

Statistical-Error Measures: Statistical measures such as Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) can be employed to calculate the average discrepancy between the intended output and the actual probabilistic outcome.

Distribution Analysis: Through the examination of the probability distribution of the error, we can gain insight into the possibility of encountering particular error values.

The objective of error estimation is not only to attain a solitary error-free result but rather to understand the statistical traits of the inaccuracy.

The block diagram of a stochastic nonlinear system is based on the following combination.

Non-linear Block (NB): This block represents the nonlinear function of the system.

Noise/Disturbance Source: An internal or external noise source block is included.

Delay Block (DB): To intricate ongoing time delays the DB can be provided if necessary.

The block diagram illustrates the interaction between the deterministic nonlinearities and the stochastic factors that affect the output of the system.

Stochastic nonlinear systems are commonly found in a wide range of real-life situations. Comprehending their behavior is essential for activities like risk assessment, filtering, and control design that take into consideration the probabilistic nature of the outputs, from forecasting the weather with inherent complexities to financial markets with unpredictable oscillations.

2.5 Continuous Nonlinear Systems

The state variables of these systems are defined over continuous domains, and they undergo continuous evolution throughout time. Integral equations describe the dynamics of continuous nonlinear systems. Instances encompass mechanical systems regulated by Newton's laws of motion, electric circuits explicated by Kirchhoff's rules, and fluid dynamics equations [67-72].

Key Characteristics:

Continuous Changes: State and output variables, in contrast to discrete systems, undergo continuous change over time.

Non-linear Relationships: The relationship between input, state, and output is non-linear. This implies that a proportional alteration in the input may not lead to a proportional alteration in the output.

Pertinent illustrations are followed as:

Spring-mass system: The motion of a mass attached to a spring is governed by a continuous nonlinear differential equation. As the mass moves, its position and velocity (state variables) change continuously.

Chemical reaction kinetics: The rate of change of chemical species in a reaction can be modeled by a system of continuous nonlinear differential equations. The concentrations of the species change smoothly over time.

Control systems with continuous actuators: Control systems where the control signal can take on any value within a range are considered continuous. For example, a system controlling the temperature of a furnace through a continuously adjustable valve.

2.5.1 Mathematical Modeling

The primary method for modeling continuous nonlinear systems is through:

Nonlinear Ordinary Differential Equations (ODEs): This method describes the rate of change of the system's state variables using a system of equations with nonlinear functions. The derivatives of the state variables represent their continuous change over time, as expressed in Equation (2). In Equation 8, s(t) is the state vector, w(t) is the input, and f(.) is a nonlinear function representing the system dynamics.

2.5.2 Error Estimation

The goal here is to quantify the difference of the desired-continuous output (q_desired(t)) and the estimated-continuous output (q(t)). The techniques include:

Norm-based Errors (NBE): The total variation between the two signals over time can be computed using norms.

Frequency-domain Errors (FDE): By transforming the desired and estimated responses into the frequency domain, it is possible to examine the error based on their respective frequency spectrums.

The block diagram illustrates:

Nonlinear Block (NB): The block symbolizes the system's nonlinear function (f(.)), which takes the input (u(t)) and state (x(t)) as inputs and generates the system's output (y(t)).

Integrator Blocks (IB): Integrator blocks may be added to the system dynamics to incorporate integration processes within the differential equations, depending on the unique requirements.

The block diagram accurately represents the continuous nature of the signals and the continuous interaction between the input, state, and output.

Engineers can utilize their understanding of continuous non-linear systems to efficiently design, control, and optimize several systems in domains such as mechanics, control engineering, and chemical processes, where continuous processes are widespread.

2.6 Discrete Nonlinear Systems

These systems evolve in discrete steps or time intervals, and their state variables are defined at discrete points in time. The dynamics of discrete nonlinear systems are described by difference equations or recursion relations. Discrete nonlinear systems are well exploited by the research community in various field of network security, PID controller and many more including digital control systems, discrete-time signal processing systems, and iterated maps in chaos theory [73-78].

Here's a breakdown of their key characteristics:

Key Characteristics:

Discrete Changes: The system's state and output change only at specific intervals of time. These changes can be significant jumps from one value to another.

Nonlinear Relationships: The relationship between input, state, and output is not linear. A proportional change in the input might not result in a proportional change in the output at the next time step.

Some of the examples are as follows:

Bouncing ball: The height of a bouncing ball can be modeled as a discrete nonlinear system. The ball's height changes only at the moments it hits the ground or another object.

Population growth model: Some models for population growth track population size at specific intervals (e.g., yearly). These models can be nonlinear to account for factors like birth rates and death rates.

Digital control systems: Control systems that use digital signals with discrete values (e.g., on/off) are considered discrete.

2.6.1 Mathematical Modeling

There are two main approaches to modeling discrete nonlinear systems:

Nonlinear Difference Equations: These equations relate the state of the system at one time-step (m) to the previous state at time-step (m-1) and the input applied at the current time step. They involve nonlinear functions that capture the system's behavior as shown in Equation 9.

$$s(m+1) = f\big(s(m),\ w(m)\big) \tag{8}$$

where s(m) is the state at time-step m, w(m) is the input at time-step m, and f(.) is a nonlinear function.

Input-Output Models: Similar to continuous systems, these models relate the system's output directly to the input through a nonlinear function, without explicitly describing the internal states. However, the function operates on discrete-time values.

2.6.2 Error Estimation

Since the system operates in discrete time steps, error estimation focuses on the difference amongst the desired response at each time-step (q_desire(m)) and the estimated response (q_estimated(m)). The same approaches used for continuous systems can be applied here, but adapted for discrete time:

Norm-based Errors: We can calculate norms to quantify the overall difference between the desired and actual output sequences.

Frequency-domain Errors (applicable in some cases): If the desired and actual outputs can be represented as periodic sequences, they can be converted to the frequency domain using tools like the Discrete Fourier Transform (DFT). The error can then be analyzed in terms of the frequency spectrum.

The block diagram for a discrete nonlinear system typically consists of:

Nonlinear Block: This block represents the system's nonlinear function (f(.) or h(.)) that takes the input (u(k)) and possibly the state (x(k)) as input and produces the system's output (y(k)).

Unit Delay Block(s): These blocks delay the input and/or state by one time step to account for the discrete nature of the system.

The block diagram reflects the discrete jumps in the signals and the non-linear rapport amongst input, state, and the output at each time step.

Understanding discrete nonlinear systems is crucial in various fields that involve digital processing and control. From analyzing communication systems to designing control algorithms for robots, these systems play a vital role in modern technology.

2.7 Non-linear Single Input Single Output (SISO) Systems

These systems have a single input and a single output, where the input-output relationship is characterized by nonlinear dynamics. SISO nonlinear systems are common in many engineering applications, such as feedback control systems, servo systems, and nonlinear filters [79-82].

Here's a breakdown of key aspects of SISO Nonlinear Systems:

Imagine a system where you can adjust one knob (input) and observe the corresponding effect on a single gauge (output). This is the essence of a SISO system. However, the relationship between the knob (input) and the gauge (output) is not always proportional or linear. This nonlinearity adds complexity in analyzing and controlling the system's behavior.

Some of the examples are as follows:

Temperature control system: Here, the input could be the control signal sent to a heater (e.g., percentage of power) and the output the measured room temperature. The relationship between heater power and temperature might not be perfectly linear, especially at extreme settings.

Chemical reaction: The amount of a reactant introduced (input) can affect the rate of a chemical reaction (output). However, the reaction rate might not increase proportionally with the added reactant due to factors like saturation effects.

2.7.1 Mathematical Modeling

Modeling SISO nonlinear systems can be achieved through various methods depending on the complexity:

Nonlinear Ordinary Differential Equations (ODEs): This method describes the system's dynamics using a system of equations with nonlinear functions relating the rate of change of the system's state (internal variables) to the input and the output as shown in Equation 2. where s(t) is the state vector, w(t) is the input, and f(.) is a nonlinear function representing the system dynamics.

Input-Output Model: This approach focuses directly on the relationship between the input and output, bypassing the internal states. Here, a nonlinear function maps the input to the output expressed as Equation 11.

$$q(t) = h\big(w(t)\big) \tag{9}$$

where q(t) is the output and h(.) is a nonlinear function representing the input-output relationship.

2.7.2 Analysis and Control

Nonlinear systems present greater challenges in terms of analysis and control compared to linear systems. Here are a few prevalent methods:

Linearization: Under some circumstances, the system's behavior can be estimated as linear in the vicinity of a particular operational point. This enables the utilization of linear control approaches for streamlined analysis.

Nonlinear Control Techniques: To manage the non-linearities several strategies have been developed. The strategies include the use of feedback control for incorporating nonlinear components based on optimization.

SISO non-linear system has following blocks:

The non-linear block (NB) represents the system's nonlinear function (f(.) or h(.)) that takes the input (u(k)) and possibly the state (x(k)) as input and produces the system's output (y(k)).

The optional block displays the non-linear function of the system. It is denoted as f(.) or h(.). It takes the input (u(t)) and (optional) the state (x(t)) and produces the output (y(t)) of the system.

Single input path and a single output path block diagram has been illustrated above. Where the central block is a nonlinear transformation function that connects the input to the output.

Understanding single-input single-output (SISO) nonlinear systems is essential for optimizing different real-world systems that exhibit non-linear behavior. For effectual design and operation, non-linear behavior must be wisely considered in systems ranging from basic temperature control systems to sophisticated chemical reaction processes.

2.8 Non-linear Multi-Input Multi-Output (MIMO) Systems

These systems possess a multitude of inputs and outputs, wherein the interactions between inputs and outputs are regulated by non-linear relationships. MIMO non-linear systems are present in intricate engineering systems that consist of several interconnected subsystems, such as robotic manipulators, chemical process plants, and communication networks [83-88].

The breakdown of key features of MIMO Non-linear Systems are given as:

Consider a system that consists of several knobs for input and multiple gauges for output. Manipulating a single knob can influence not just the gauge it is directly connected to, but also other gauges to different extents. Furthermore, the correlation between each knob and each gauge is not consistently proportionate or linear. This intricacy results from the dynamics of the system being interrelated.

Here are a few examples:

Flight control system (FCA): An airplane's control system comprises many inputs, such as ailerons, elevators, and rudders, which influence its motion in different manners, including changes in altitude, roll, and yaw. These relationships exhibit nonlinearity, particularly when operating at high speeds or during maneuvers.

Chemical plant: Within a chemical plant, the manipulation of several valves that regulate the movement of different reactants can have an impact on multiple characteristics of the end products, such as temperature, pressure, and purity. The interplay between these fluxes can introduce non-linearities in the properties of the final output.

2.8.1 Mathematical Modelin

Modeling MIMO nonlinear systems can be achieved through various methods:

Nonlinear Ordinary Differential Equations (ODEs): This method uses a system of equations with nonlinear functions to describe the systems' dynamics. Here, the rate of change of each state variable depends not just on the current state but also on all the input variables as expressed in Equation 2. where s(t) is the state vector, w(t) is the input vector (containing all input values), and f(.) is a vector-valued nonlinear function representing the system dynamics.

Input-Output Model: This approach directly relates each output to all the inputs through nonlinear functions. Mathematically, given as Equation 13.

$$q(t) = h(w(t)) \tag{10}$$

The output vector q(t) contains all output values, while the vector-valued nonlinear function h(.) represents the input-output relationships.

2.8.2 Analysis and Control:

Researchers are actively working on methods to analyze and control MIMO nonlinear systems. Below are a few prevalent methodologies:

Linearization: Linearization refers to the process of approximating a nonlinear function with a linear function to simplify calculations or analysis.

Multivariable Control Techniques: These methods are tailored for MIMO systems. In this method the system has multiple inputs and outs. Illustratively it encompass state-space control and model predictive control.

Non-linear Control Techniques: Techniques such as integrating nonlinear aspects into feedback control or by using of optimization-based methods to directly address multiple-input multiple-output (MIMO) non-linearities.

The blocks used in MIMO non-linear systems are:

Nonlinear Block (NB): The function takes the input vector (w(t)) and, optionally, the state vector (s(t)) as input and generates the output vector (q(t)). The block comprises non-linear functions that transform inputs into outputs.

Optional Blocks: Optional blocks to accommodate certain dynamic attributes.

MIMO nonlinear systems, ranging from flight control systems to chemical process management, necessitate the utilization of sophisticated analysis and control approaches to guarantee maximum performance.

2.9 Chaotic Nonlinear Systems

These systems display a high sensitivity to beginning conditions, deterministic chaos, and complicated dynamic behavior that is characterized by irregular, non-repetitive, and unpredictable paths. Chaotic nonlinear systems are governed by nonlinear equations with simple deterministic rules but exhibit highly complex and unpredictable behavior over time. Chaotic nonlinear systems have been exploited in control system, signal processing and quantum system [89-93]. Examples include the Lorenz system, the double pendulum, and certain electronic circuits.

They are characterized by three key features:

Sensitive Dependence on Initial Conditions (Butterfly Effect): Even tiny changes in the initial state of a chaotic system can lead to drastically different outputs over time. This is often described metaphorically as the "butterfly effect," where a butterfly flapping its wings in one place can eventually cause a tornado in another.

Aperiodic Long-Term Behavior: Unlike periodic systems (outputs repeat after a fixed time) or convergent systems (outputs settle to a specific value), chaotic systems never settle into a predictable pattern. Their outputs appear random, but they are still determined by the initial conditions and the system's rules.

Boundedness: Chaotic systems do not demonstrate exponential growth or decay. In spite of their apparent instability, their outputs remain within a finite range.

The key features of the system are illustrated in examples as given below:

The Double-Pendulum: This exemplary demonstration comprises of a pair of pendulums that are interconnected. Slight alterations in the initial beginning angles can result in significantly divergent swinging patterns as time progresses.

Population-Dynamics: Predator-prey interaction models can display chaotic dynamics, where even slight variations in starting population sizes can result in unpredictable and erratic fluctuations over time.

Weather Systems: The intricate interplay among temperature, pressure, humidity, and wind in the atmosphere gives rise to a turbulent system. Although the fundamental rules of physics are clearly established, accurately forecasting long-term weather patterns becomes progressively difficult because of the high sensitivity to initial conditions.

2.9.1 Mathematical Modeling

Chaotic systems can be represented using the same methodologies as other nonlinear systems, such as:

Nonlinear Ordinary Differential Equations (ODEs): These equations describe the rate of change of the system's state variables, but the functions involved may be highly complex and sensitive to initial conditions.

2.9.2 Analysis and Prediction

Understanding and forecasting the actions of chaotic systems is difficult because of their extreme sensitivity and lack of periodic patterns. Here are several methods:

Statistical Analysis: This technique includes investigative data through statistical measures to get a logic of its overall features, such as average values and the likelihood of various outcomes.

Chaotic non-linear systems are found in both natural and man-made settings. Following are the few examples:

Climate modeling includes the studying the behavior of weather systems helps improve long-term climate forecasts and assess the impact of global warming.

Economics includes financial markets often display that pattern, by understanding this, helps investors in developing strategies for managing risk.

Secure Communication using chaotic systems can generate pseudo-random sequences for encryption to enhance communication security.

Bifurcation Diagrams illustrates shifts from a state of order to a state of disorder and pinpoint areas in the parameter space where chaotic behavior takes place.

Numerical Simulations involves computer-based simulation to explore the behavior of system followed by different trends and pattern.

3. CHALLENGES OF NON-LINEAR SYSTEM IDENTIFICATION AND LIMITATIONS OF TRADITIONAL METHODS

Mathematical modeling of a non-linear system is often more difficult to find as compared to a linear system owing to multiple factors like:

- Non-linear models generally need more data than linear models. This data must accurately reflect the system's various operating conditions and behaviors.
- A model that's too complex might fit the data too closely, while a simpler one might miss important non-linear aspects of the system.
- Non-linear systems can often be described by several different models, making it hard to pinpoint a single, definitive one.

Although a non-linear model can be successfully created but there remain certain limitations.

- The model might have some errors with respect to the real systema and that may lead to decrease the or effect the system's performance.
- The model might work well under certain conditions but could lose accuracy in different scenarios.
- While the model can predict behavior, it might not always explain the fundamental physical processes, which bounds our understanding of the system.
- Real-world data often contains noise. Non-linear methods are more affected by this noise than linear ones, which can impact the model's accuracy.
- Working with non-linear systems, especially complex models, can be computationally intensive.

Feature	Challenges	Limitations
Focus	Difficulty of the identification process	Aspects of the identified model
Examples	Non-uniqueness, data requirements	Model accuracy, generalizability, physical meaning
Impact	Difficulty in obtaining a reliable model	Limitations in using the model for prediction, control, or analysis

4. HEURISTIC ALGORITHMS FOR IDENTIFYING NONLINEAR SYSTEMS USING BLOCK ARX MODELS

This section elaborates how heuristics-based optimization techniques are exploited to estimate the parameters in ARX models. Generally, the standard ARX models outrightly performs well for linear system, Block ARX models are tailor-made to cope the difficulties of non-linear behavior. Estimating of parameters in these models can be complicated owing to the complex nature of non-linear functions. In this study we will explore various heuristics methods to deal with the challenges.

4.1 Limitations of Traditional Optimization Methods

Prior to make investigation about the heuristic algorithms, it is important to understand the constraints of traditional optimization methods. The least square methods are effective for the linear ARX models they struggle with Block structured ARX models having non-linear elements. In such scenario the objective function will become non-convex, causing traditional methods to get stuck in sub-optimal solutions due to multiple local minima.

4.2 Advantages of Heuristic Algorithms

Heuristics based algorithms are strong alternative for the estimation of parameters in Block ARX models. The heuristics are inspired by the natural processes and these algorithms use iterative techniques in order to explore the complex search space and are overwhelmed to the traditional methods. They are flexible and need minimal information about the problem. The heuristic algos are good at result solutions in solving complex problems which helps in getting better performance for block ARX models.

4.3 Common Heuristic Algorithms for Block ARX Model Parameter Estimation

In this section some of the commonly used heuristics algos used for solving the parameter estimation problem are elaborated.

4.3.1 Genetic Algorithm (GA)

GA simulates natural selection and genetic recombination to solve difficult optimization and search problems. A genetic algorithm evolves a population of candidate solutions, represented as chromosomes or individuals, through selection, crossover, and mutation. Genetic algorithms use selection pressure and genetic operators to enhance population solutions and converge on optimal or near-optimal solutions. Engineering, computer science, economics, and biology employ genetic algorithms to solve optimisation issues where standard methods are inefficient or unfeasible[94-100].

Pseudocode:

```
Function GAs(population, fitness_function, max_generations)
  For i in range(max_generations):
    Select parents from population based on fitness
    Perform crossover on parents to create offspring
    Introduce mutations to offspring with a low probability
    Evaluate fitness of offspring
    Combine offspring with parents to form new population
    Return best individual in final population
End Function
```

4.3.2 Particle-Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a computational method inspired by the way flocks of birds or schools of fish move together. In PSO, a group of potential solutions, called particles, move through the search space to find the best solution for a given problem. Each particle represents a possible solution and adjusts its position and speed based on its own past experiences and the experiences of nearby particles.

PSO relies on swarm intelligence, where particles interact and share information to find the best solution. Each particle has its position, speed, and best-known solution, and particles are influenced by the global best solution found by any particle in the swarm.

Particles' movement in the search space is guided by both individual and collective factors. Cognitive factors encourage particles to improve their own best solutions, while social factors push them to consider the best solution found by the group. Acceleration coefficients help balance the search between exploring new areas and exploiting known good solutions.

By using both local and global information, PSO effectively explores and exploits the search space, often finding optimal or nearly optimal solutions.

The SI-based PSO algorithm is widely used in various optimization issues, including function optimization, parameter tuning, machine learning, and engineering design [101-106]. The reason for its effectiveness lies in its directness, effectiveness, and capacity to handle intricate and high-dimensional optimization tasks that encompass nonlinearity.

Velocity Update:

Equation 11 determines the speed and direction of each particle in the next iteration.

$$\begin{aligned} v_i(t) &= \omega_i * v_i(t-1) + c_{cl}1 * rand() * \left(pbest(t-1) - x(t-1)\right) \ldots \\ &\quad + c_{sl}2 * rand() * \left(g_{global}best(t) - x(t-1)\right) \end{aligned} \tag{10}$$

where:

- $v_i(t)$- particle velocity at iteration t
- $v_i(t-1)$- velocity of particle at previous iteration (t-1)
- ω_i- inertia weight (controls momentum of particles)
- $c_{cl}1$- cognitive learning rate (importance of individual experience)
- $c_{sl}2$- social learning rate (importance of swarm knowledge)
- $rand()$- random number between 0 and 1
- $pbesr$ (t-1) - best position found by the particle itself up to iteration t-1
- $x(t-1)$- current particles' position at iteration t-1
- $g_{global}best$- best-position found by the entire swarm up to iterationt

Position Update:

Equation 12 determines particles' new position based on its current position and the updated velocity.

$$x_{new}(t) = x_{current}(t-1) + v_i(t) \tag{12}$$

where:

- $x_{new}(t)$- new position of the particle at iteration t
- $x_{current}$ (t-1)- current particles' position at iteration t-1
- $v_i(t)$- particle updated velocity at iteration t

Pseudocode:

```
Function PSO(population, fitness_function, max_iterations)
  Initialize particles' positions and velocities
  For t in range(max_iterations):
    Evaluate individual particle fitness
    Update pbest for each particle
    Update g_global best for the entire swarm
    Update v_i(t) of each particle
    Update x_new(t) of each particle
  Return best particle in final population
End Function
```

Applications: PSO finds applications in various scientific and engineering domains, including:

Control System Design: Optimizing controller parameters for complex systems.

Power System Optimization: Optimizing power flow and stability in nonlinear power systems.

Machine Learning: Hyperparameter tuning in machine learning algorithms with nonlinearities.

Image Processing: Image segmentation and feature extraction tasks involving nonlinearities.

Robotics: Path planning for robots in complex environments.

4.3.3 Firefly Algorithm (FA)

In 2008, Xin-She Yang introduced the Firefly Algorithm (FA), a metaheuristic optimisation method that takes its cues from the flashing actions of fireflies[107]. In FA, potential answers are depicted as fireflies navigating the search space, with the brightness of each fly indicating its fitness value. Fireflies are drawn to lights that are brighter than their surroundings, and the degree of this attraction decreases as the distance between the lights increases. Exploring the search space and convergent towards optimal or near-optimal solutions, fireflies migrate towards brighter individuals through repeated updates. The simplicity and efficacy of FA in handling complex and multimodal landscapes have led to its widespread application to optimisation challenges. Efficiently managing environments that are both complex and multimodal.

FA has been employed for parameter optimisation in various engineering design fields, such as structural optimisation, mechanical design, and electromagnetic optimisation, image processing, picture augmentation, feature extraction, pattern identification, machine learning and artificial intelligence, neural networks, economics and finance to optimise portfolios, control risks, forecast financial outcomes, bioinformatics and computational biology for tasks such as predicting protein structures, aligning sequences, and optimising biological networks [108-112]. In summary, the Firefly Algorithm's wide range of applications makes it a powerful tool for solving optimisation problems in several areas of study and industry. The weight update relation is given in Equation 13.

Weight Update Equation (Attraction):

$$\beta(r) = exp(-\lambda * r \hat{} 2) \tag{13}$$

where:
- *β(r)* - attractiveness at distance r
- *λ* - light absorption coefficient

4.3.4 Ant Colony Optimization (ACO)

Ant Colony Optimisation (ACO) mimics foraging behavior of how ants deposit pheromone trails while searching for food. Artificial ants explore the search space, leaving "trails" based on solution quality [113]. This guides the search towards promising regions.

ACO has been utilised in diverse domains to address intricate optimisation challenges, data transmission channels in routing and networking, reducing latency and congestion in computer networks, transportation, and telecommunications, vehicle routing problem (VRP) and travelling salesman problem (TSP),supply chain management, resource allocation, and production scheduling in manufacturing and operations, robotics, swarm coordination, and multi-robot systems, hence enabling efficient navigation and collaboration in intricate situations [114-120]. These applications showcase the adaptability and efficacy of ACO in addressing a wide range of optimisation difficulties across several sectors. The recursive mathematical expression is given as Equation 14.

Weight Update Equation (Pheromone Update):

$$\Delta\tau(ij) = (1 / \rho) * \Delta Q(ij) \tag{14}$$

where:
- Δτ(ij) - change in pheromone level on path (i,j)
- ΔQ(ij) - quality of solution found on path (i,j)
- ρ - pheromone evaporation rate

4.3.5 Simulated Annealing (SA)

Simulated Annealing (SA) is a stochastic optimisation approach that draws inspiration from the annealing process observed in metallurgy [121]. It is utilised to discover nearly optimal solutions to combinatorial optimisation issues by simulating the process of cooling a material to achieve a state of low energy. The optimisation problem is viewed as a symbolic energy landscape in SA, with the objective function standing in for the system's energy. In an iterative fashion, the method takes an initial solution and uses it to explore the solution space by making minor random adjustments. These changes are approved or rejected based on a probability distribution, helping the algorithm avoid local optima and better search space exploration. As the method advances, the chance of accepting inferior answers drops with time, similar to the cooling process in annealing. SA keeps on in this iterative manner until a stopping criterion is satisfied, which usually happens when a predefined iterations have been finished or a specific degree of convergence has been attained. Simulated Annealing is renowned for its capacity to tackle intricate and multifaceted optimisation terrains, rendering it a favoured option for a diverse array of optimisation issues in numerous domains, such as engineering, finance, and artificial intelligence [122-125]. The weight update is equation is given as Equation 15.

Weight Update Equation (Metropolis Criterion - Probability of Accepting Uphill Move):

$$P(\Delta E > 0) = exp(-\Delta E / T) \tag{14}$$

where:
- ΔE - difference in cost function between new and current solution
- T - temperature (cooling parameter)

4.3.6 Tabu Search (TS)

The Tabu Search (TS) algorithm is a meta-heuristic optimisation technique that effectively maps solution spaces by iteratively transitioning between different solutions[126]. The system retains a temporary memory, referred to as the tabu list, in order to avoid repeating solutions that have been previously visited and to direct the search towards potentially fruitful areas within the solution space. Tabu Search effectively navigates intricate optimisation landscapes and identifies nearly optimum solutions to combinatorial optimisation problems by skillfully balancing exploration and exploitation. Tabu Search (TS) has been utilised in diverse domains, encompassing logistics and supply chain management for the purpose of optimising transportation routes, production scheduling, and inventory management. Additionally, it has been applied in telecommunications, networking, network design optimisation, routing optimisation, resource allocation optimisation, finance, portfolio optimisation, asset allocation optimisation, risk management, and investment strategy optimisation[127-132]. These applications demonstrate the adaptability and efficacy of Tabu Search in addressing various optimisation problems in practical situations.

4.3.7 Differential Evolution (DE)

In order to efficiently explore and exploit the search space, Differential Evolution (DE) iteratively refines a population of possible solutions by combining and mutating individuals. This stochastic optimisation technique is inspired by natural selection[133]. Differential Evolution (DE) has been extensively utilised in diverse domains of engineering, encompassing optimisation of mechanical and structural systems, parameter tuning in control systems, optimisation of electromagnetic devices, machine learning, data mining, feature selection, hyperparameter optimisation, neural network training, finance, risk management, algorithmic trading strategies, image processing for image segmentation, object detection, and image reconstruction tasks [134-139]. DE is a versatile technique for solving many optimisation difficulties in real-world applications due to its versatility, simplicity, and efficacy.

4.3.8 Artificial Bee Colony (ABC) Optimization

Artificial Bee Colony (ABC) A metaheuristic method known as optimisation draws inspiration from the foraging behaviour seen in honeybee colonies [140]. To solve optimisation challenges, it simulates artificial bees exploring and exploiting food sources. Artificial bees search ABC by iteratively changing the positions of candidate solutions, which are food sources. The algorithm contains three phases: employed, onlooker bee phase, and scout bees. Artificial bees use local information to improve food supplies during the employed bee phase. In the observer bee phase, artificial bees communicate potential food source knowledge and choose new food sources to explore. For population diversity, artificial bees randomly explore for new food sources in the scout bee phase. ABC effectively finds high-quality solutions to optimisation problems in engineering design, scheduling, image processing, and financial portfolio optimization [141-146].

4.3.9 Cuckoo Search (CS)

The Cuckoo Search (CS) algorithm is a metaheuristic optimisation technique that draws inspiration from the reproductive behavior observed in cuckoo birds, particularly the brood parasitism method employed by certain species of cuckoos [147]. The CS algorithm employs nests as representations of candidate solutions, with the objective of repeatedly enhancing the positions of these nests within the search space in order to identify the ideal solution. In the process of optimisation, cuckoo birds deposit eggs within their nests, symbolizing candidate solutions. Each individual egg serves as a potential solution to the optimisation problem at hand. The evaluation of egg quality is conducted by the utilization of a fitness function, whereby cuckoos employ a probabilistic process to replace eggs within their nests with superior alternatives. Furthermore, cuckoos have the ability to engage in random nest investigation as a means of preserving population variability. The Cuckoo Search algorithm has been utilized in diverse optimisation domains, including engineering design, image processing, wireless sensor networks, and machine learning. Its efficacy in efficiently identifying optimal or near-optimal solutions has been well-documented [148-152].

Mathematical Equations:

- Lévy Flight: CS utilizes Lévy flights, a random walk pattern with heavier tails compared to standard Brownian motion. This allows for efficient exploration of the search space, particularly for long jumps.
- Discovery and Replacement: A probability determines if a cuckoo (solution) lays its egg (new solution) in a randomly chosen nest (existing solution). Another probability determines if a host bird discovers the foreign egg and abandons the nest.

Pseudocode:

```
Function CS(population, fitness_function, max_generations)
  Initialize cuckoo positions (candidate solutions)
  For i in range(max_generations):
    Evaluate fitness of each cuckoo
    Generate new cuckoo solutions using Lévy flights
    Replace a fraction of worse nests with new solutions based
on a discovery probability
    Abandon a fraction of nests with low-quality solutions
  Return best cuckoo in final population
```

```
End Function
```

4.4 Choosing the Right Heuristic Algorithm

The choice of the optimal heuristic algorithm for a given Block ARX model identification challenge is contingent upon various factors like model complexity, cost function landscape, domain knowledge, convergence speed and implementation difficulty. Algorithms exhibits different behavior with respect to the above mentioned complexities. For instance, GA and PSO performs effective for simple models. However, SA and FFA can handle with complex models. SA and TS perform well for functions with local minima. GA and PSO are efficient, balancing exploration and exploitation to converge faster with clear minima. The ABC algorithm performed well for domain knowledge.

Choosing the right heuristic algorithm involves understanding these factors and matching them to the specific needs of your Block ARX model. Experimenting with different algorithms may be necessary to find the best fit.

For up-to-date advancements in nonlinear system identification, including Hammerstein systems, the Web of Science is a valuable resource. It offers a comprehensive database of journals, conference proceedings, and papers, along with tools for literature reviews and keeping up with recent developments.

Data was collected using Web of Science categories dated April 21, 2024, at 1930HRS. According to the Web of Science total number of articles published in 250 fields of research 236,569 have been published from 2014 to 2024. The clustere

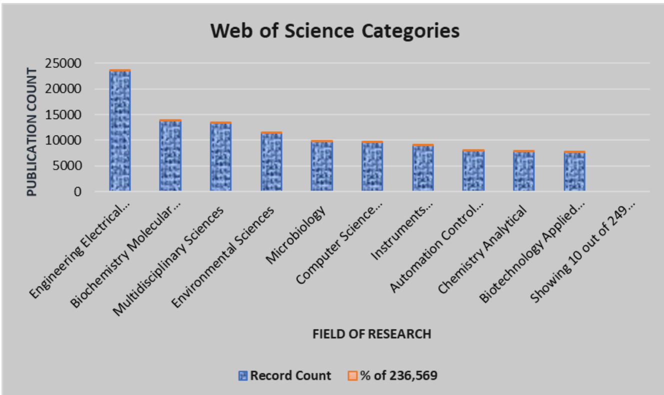

Fig. 3 Bar chart representation of web of science categories against publication count

The record shows that the out the maximum research in terms of system identification the maximum research has been carried out in the field of Electrical Engineering that is 9.9% of the total number of publications in 249 fields. Similarly, remaining fields of research have the significant contribution resulting in producing 236,569 publications. In order to further investigate the contribution of research community region wise it can be seen in Figure 4 that about 28.74% of total number of publications has been contributed from Peoples Republic of China, while the contribution of research community of USA is 23% and the trend shows declining trend as we move down. The trend can be witnessed as shown in Figure 4.

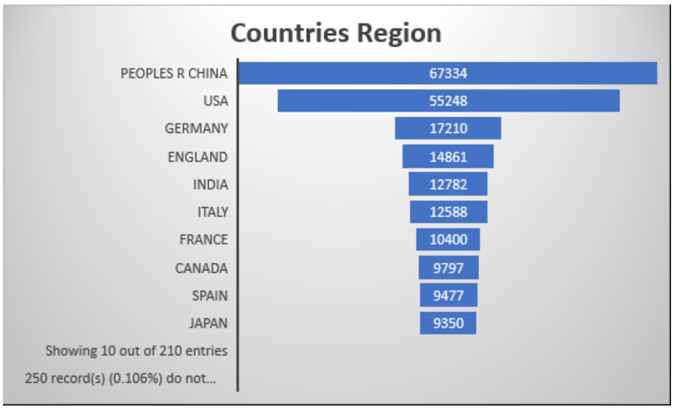

Fig. 4 Region wise statistics of system identification related publications

In order to analyze the research inclination in perspective of non-linear systems identification. The contribution of research community as of the information available on web of science Figure 5 shows that there are 12,245 and while the distribution accordingly in respect to the different field of research are shown in Figure 5 as Pie Chart. The data shows that 27% of 12,245 publications has been contributed by the research community in the field of electrical engineering while 17% in the field of Automation and control,14% by the Mechanical Engineering community while 11% in the field of Mechanics, Computer science and artificial intelligence, engineering multi-disciplinary, Instruments Instrumentation contributed 7% while, Mathematics and Civil Engineering has 6% and 5% contributions.

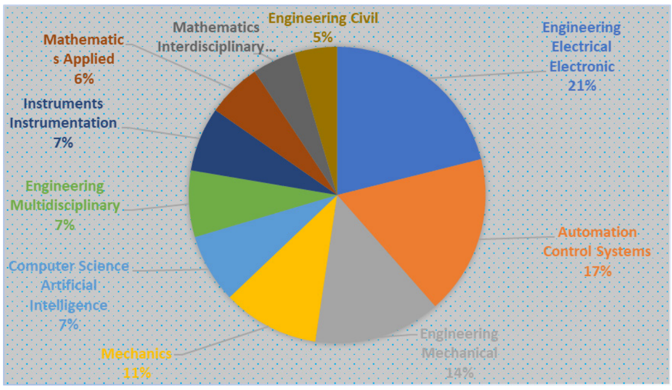

Fig. 5 Pie chart distribution representing web of science categories

Moreover, the further investigation in term of yearly publications to elaborate the trend of research community in nonlinear system identification. Figure 6 shows that yearly publication trend the particular field and it has been observed that starting from 2014 the number of publications were 792, while on going years shows an increasing trend of publication in the specific field in the preceding years till 2022 and reported publications were 1588. However, in 2023 the number of publications were 1535. The web of science record shows that since April 21, 2024 the publication are 338.

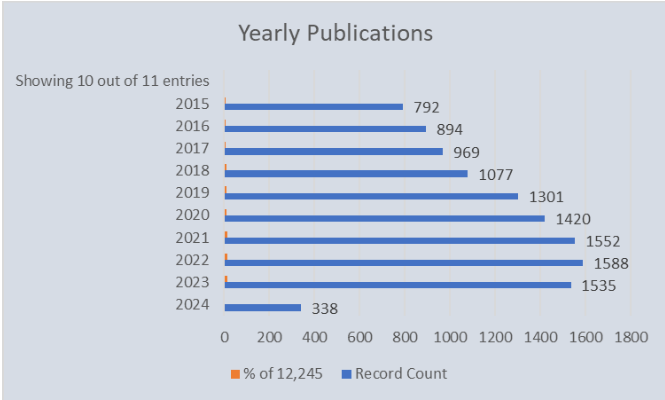

Fig. 6 Bar chart distribution representing yearly publications related to non-linear system identification

5. PROPOSED FUTURE PROSPECTIVE

Researchers and practitioners in the field are continuously addressing these challenges through the development of benchmark datasets for rigorous evaluation, the incorporation of hybrid approaches that combine heuristics with other methodology, and the utilisation of increasingly intricate heuristic methods. The aforementioned challenges may be mitigated via the progression of knowledge and technology; yet, it remains imperative to consider them while employing heuristics for parameter estimation in nonlinear Hammerstein systems.

Potential avenues for future research in heuristic-based parameter estimation of nonlinear Hammerstein models involve addressing existing challenges and using state-of-the-art technology. Here are potential future directions that heuristics in the field of system identification of non-linear systems may take:

- To develop hybrid models, it is necessary to integrate heuristics with advanced techniques such as machine learning algorithms or optimization approaches. The precision and effectiveness of estimating nonlinear Hammerstein models can be enhanced by integrating the benefits of multiple procedures.
- Develop adaptable heuristics capable of modifying system identification features. One important aspect to consider is the ability to accurately estimate the parameters of an unknown non-linear Hammerstein model.
- Make developing real-time heuristics a top priority. This is primarily significant for quickly resolving the system identification problem when resembling parameters of non-linear systems.
- Developing Clear Heuristics: the purpose is to create heuristics that deliver accurate and easily explainable results. It's vital to build reliable and meaningful models, particularly for important applications.

By using these approaches, researchers and practitioners can improve the application of heuristic techniques in identifying non-linear Hammerstein models. This will aid advance the reliability, flexibility, and accuracy of parameter estimates in complex systems, including areas like the human nervous system, brain signaling, etc.

6. CONCLUSIONS

This study presents a thorough review of non-linear systems and implications of system identification of non-linear systems. Moreover, the optimization based computational techniques like GAs, PSO, FFA, A-CO, SA, TSO, DEH, ABCO and CSA have been elaborated in term of their wide utilization in solving numerous scientific problems and their use in system identification of non-linear systems. This article might serve as convenient reference for researcher in the field of computational intelligence.

REFERENCES

[1] Abreu Archanjo Dutra, D., *Bayesian Networks for Variational System Identification.* arXiv e-prints, 2024: p. arXiv: 2404.10137.

[2] Chen, Y., et al., *Modified varying index coefficient autoregression model for representation of the nonstationary vibration from a planetary gearbox*. IEEE Transactions on Instrumentation and Measurement, 2023. **72**: p. 1-12.

[3] Guo, J., et al., *Identification of FIR systems with binary-valued observations against data tampering attacks*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2023.

[4] Ju, Y., B. Mu, and T. Chen, *On convergence of covariance matrix of empirical Bayes hyper-parameter estimator*. Control Theory and Technology, **2024**: p. 1-14.

[5] Ibnkahla, M., *Nonlinear system identification using neural networks trained with natural gradient descent*. EURASIP Journal on Advances in Signal Processing, 2003. **2003**: p. 1-9.

[6] Liu, J., et al., *A quasi fractional order gradient descent method with adaptive stepsize and its application in system identification*. Applied Mathematics and Computation, 2021. **393**: p. 125797.

[7] Ribeiro, A.H., et al., *Beyond occam's razor in system identification: Double-descent when modeling dynamics*. IFAC-PapersOnLine, 2021. **54**(7): p. 97-102.

[8] Wong, C.-C. and C.-C. Chen, *A hybrid clustering and gradient descent approach for fuzzy modeling*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1999. **29**(6): p. 686-693.

[9] Tobar, F.A., S.-Y. Kung, and D.P. Mandic, *Multikernel least mean square algorithm*. IEEE Transactions on Neural Networks and Learning Systems, 2013. **25**(2): p. 265-277.

[10] Al-Saggaf, U.M., et al., *The q-least mean squares algorithm. Signal Processing*, 2015. **111**: p. 50-60.

[11] Feng, D.-Z., Z. Bao, and L.-C. Jiao, *Total least mean squares algorithm*. IEEE Transactions on Signal Processing, 1998. **46**(8): p. 2122-2130.

[12] Chen, J., et al., *Nonnegative least-mean-square algorithm*. IEEE Transactions on Signal Processing, 2011. **59**(11): p. 5225-5235.

[13] Chen, B., et al., *Quantized kernel least mean square algorithm*. IEEE Transactions on Neural Networks and Learning Systems, 2011. **23**(1): p. 22-32.

[14] Diniz, P.S. and P.S. Diniz, *The least-mean-square (LMS) algorithm*. Adaptive Filtering: Algorithms and Practical Implementation, 2020: p. 61-102.

[15] Deng, H. and M. Doroslovacki, *Improving convergence of the PNLMS algorithm for sparse impulse response identification.* IEEE Signal Processing Letters, 2005. **1**2(3): p. 181-184.

[16] Li, X., et al., *An improved MPPT method for PV system with fast-converging speed and zero oscillation*. IEEE Transactions on Industry Applications, 2016. **5**2(6): p. 5051-5064.

[17] Younas, W., et al., *Improving Convergence Speed of Bat Algorithm Using Multiple Pulse Emissions along Multiple Directions.* Sensors, 2022. **2**2(23): p. 9513.

[18] Gomathi, K. and V. Saravanan, *Variable step size for improving convergence of FxLMS algorithm.* Procedia Technology, 2016. **25**: p. 420-426.

[19] Reiszadeh, M., H. Narimani, and M.S. Fazel, *Improving convergence properties of autonomous demand side management algorithms.* International Journal of Electrical Power & Energy Systems, 2023. **146**: p. 108764.

[20] Priya, V.V., et al. *Improving convergence speed of the neural network model using Meta heuristic algorithms for weight initialization. in 2021 International Conference on Computer Communication and Informatics (ICCCI).* 2021. IEEE.

[21] Cuevas, E., et al., *A comparison of evolutionary computation techniques for IIR model identification.* Journal of Applied Mathematics, 2014. **2014**.

[22] Dehghani, M., Š. Hubálovský, and P. Trojovský, *Northern goshawk optimization: a new swarm-based algorithm for solving optimization problems.* Ieee Access, 2021. **9**: p. 162059-162080.

[23] Mirjalili, S. and A. Lewis, *The whale optimization algorithm.* Advances in engineering software, 2016. **95**: p. 51-67.

[24] Neshat, M. and G. Sepidname, *A new hybrid optimization method inspired from swarm intelligence: Fuzzy adaptive swallow swarm optimization algorithm (FASSO).* Egyptian Informatics Journal, 2015. **16**(3): p. 339-350.

[25] Ali, M. A., Chaudhary, N. I., Khan, T. A., Mao, W. L., Lin, C. C., & Raja, M. A. Z. (2024). "Design of key term separated identification model for fractional input nonlinear output error systems: Auxiliary model based Runge Kutta optimization algorith." *Chaos, Solitons & Fractals,* **189**, 115696.

[26] Malik, N.A., et al., *Astrophysics-based transit search optimization heuristics for parameter estimation of multi-frequency sinusoidal signals.* Modern Physics Letters B, 2024: p. 2450342.

[27] Khan, T.A., et al., *Design of Runge-Kutta optimization for fractional input nonlinear autoregressive exogenous system identification with key-term separation.* Chaos, Solitons & Fractals, 2024. **182**: p. 114723.

[28] Mehmood, K., et al., *Parameter estimation of nonlinear systems: dwarf mongoose optimization algorithm with key term separation principle.* Journal of Ambient Intelligence and Humanized Computing, 2023: p. 1-11.

[29] Malik, M.F., et al., *Swarming intelligence heuristics for fractional nonlinear autoregressive exogenous noise systems.* Chaos, Solitons & Fractals, 2023. **167**: p. 113085.

[30] Khan, W.U., et al., *A novel application of fireworks heuristic paradigms for reliable treatment of nonlinear active noise control.* Applied Acoustics, 2019. **146**: p. 246-260.

[31] Elliott, S.J. and P.A. Nelson, *Active noise control.* IEEE signal

processing magazine, 1993. **10**(4): p. 12-35.

[32] Kuo, S.M., S. Mitra, and W.-S. Gan, *Active noise control system for headphone applications.* IEEE Transactions on Control Systems Technology, 2006. **14**(2): p. 331-335.

[33] Kuo, S.M., X. Kong, and W.S. Gan, *Applications of adaptive feedback active noise control system.* IEEE transactions on control systems technology, 2003. **11**(2): p. 216-220.

[34] Raja, M.A.Z., et al., *Design of hybrid nature-inspired heuristics with application to active noise control systems.* Neural Computing and Applications, 2019. **31:** p. 2563-2591.

[35] Yang, F., J. Guo, and J. Yang, *Stochastic analysis of the filtered-x LMS algorithm for active noise control.* IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020. **28**: p. 2252-2266.

[36] Shi, D., et al., *Feedforward selective fixed-filter active noise control: Algorithm and implementation.* IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020. **28**: p. 1479-1492.

[37] Lu, G., R. Chen, and H. Liu, *Active Noise Control Scheme for Smart Beds Based on a Wide and Narrow Band Hybrid Control Algorithm.* IEEE Access, 2023.

[38] Cheng, Y., R. Zhang, and S. Chen, *Wavelet packet transform applied to active noise control system for mixed noise in nonlinear environment.* Digital Signal Processing, 2023. **133**: p. 103860.

[39] Zhang, X., et al., *Performance optimization of the active noise control algorithm based on logarithmic function and its application in secondary channel online identification ANC system.* Measurement and Control, 2023. **56**(3-4): p. 751-762.

[40] Rugveth, V.S. and K. Khatter, *Sensitivity analysis on Gaussian quantum-behaved particle swarm optimization control parameters.* Soft Computing, 2023. **27**(13): p. 8759-8774.

[41] Velasquez, W., F. Jijon-Veliz, and M.S. Alvarez-Alvarado, *Optimal wireless sensor networks allocation for wooded areas using quantum-behaved swarm optimization algorithms.* IEEE Access, 2023. **11**: p. 14375-14384.

[42] Baş, E., *Improved particle swarm optimization on based quantum behaved framework for big data optimization.* Neural Processing Letters, 2023. **55**(3): p. 2551-2586.

[43] Elsedimy, E., S.M. AboHashish, and F. Algarni, *New cardiovascular disease prediction approach using support vector machine and quantum-behaved particle swarm optimization.* Multimedia Tools and Applications, 2024. **83**(8): p. 23901-23928.

[44] Tang, M., et al., *Mathematical modeling of resource allocation for cognitive radio sensor health monitoring system using coevolutionary quantum-behaved particle swarm optimization.* Expert Systems with Applications, 2023. **228**: p. 120388.

[45] Chen, H.-W., *Modeling and identification of parallel nonlinear systems: Structural classification and parameter estimation methods.* Proceedings of the IEEE, 1995. **83**(1): p. 39-66.

[46] Grussler, C. and R. Sepulchre, *Variation diminishing linear time-invariant systems.* Automatica, 2022. **136:** p. 109985.

[47] Mojahed, A., et al., *Exceeding the classical time-bandwidth product in nonlinear time-invariant systems.* Nonlinear Dynamics, 2022. **108**(4): p. 3969-3984.

[48] Ahmad, M. and R. Mohd-Mokhtar, *A Survey on Model-based Fault Detection Techniques for Linear Time-Invariant Systems with Numerical Analysis.* Pertanika Journal of Science & Technology, 2022. **30**(1).

[49] Dewayalage, I., et al., *Measurement Based Linear and Nonlinear Time Invariant Representations for High Voltage Inductive Transformer Frequency Response up to 10 kHz.* IEEE Access, 2024.

[50] Ye, H., et al., *Adaptive control with global exponential stability for parameter-varying nonlinear systems under unknown control gains.* IEEE Transactions on Cybernetics, 2023.

[51] Tomei, P. and R. Marino, *An enhanced feedback adaptive observer for nonlinear systems with lack of persistency of excitation.* IEEE Transactions on Automatic Control, 2022.

[52] Ye, H. and Y. Song, *Prescribed-time control for time-varying nonlinear systems: A temporal scaling based robust adaptive approach.* Systems & Control Letters, 2023. **181**: p. 105602.

[53] Zhang, Z., et al., *ADP-Based Prescribed-Time Control for Nonlinear Time-Varying Delay Systems With Uncertain Parameters.* IEEE Transactions on Automation Science and Engineering, 2024.

[54] Huang, S., et al., *Command filter-based adaptive fuzzy self-triggered control for MIMO nonlinear systems with time-varying full-state constraints.* International Journal of Fuzzy Systems, 2023. **25**(8): p. 3144-3161.

[55] Pinsky, M.A., *Optimal dichotomy of temporal scales and boundedness/stability of time-varying multidimensional nonlinear systems.* Mathematics of Control, Signals, and Systems, 2023. **35**(1): p. 125-158.

[56] Lenka, B.K. and S.N. Bora, *Nonnegativity, Convergence and Bounds of Non-homogeneous Linear Time-Varying Real-Order Systems with Application to Electrical Circuit System.* Circuits, Systems, and Signal Processing, 2023. **42**(9): p. 5207-5232.

[57] Ni, P., et al., *Non-stationary response of nonlinear systems with singular parameter matrices subject to combined deterministic and stochastic excitation.* Mechanical Systems and Signal Processing, 2023. **188**: p. 110009.

[58] Boyacioglu, B. and K.A. Morgansen. *Bioinspired observability analysis tools for deterministic systems with memory in flight applications. in AIAA Scitech 2021 Forum.* 2021.

[59] Li, Y., et al., *Adaptive optimized backstepping control-based RL algorithm for stochastic nonlinear systems with state constraints and its application.* IEEE Transactions on Cybernetics, 2021. ***52***(10): p. 10542-10555.

[60] Wang, W., et al., *Adaptive backstepping based consensus tracking of uncertain nonlinear systems with event-triggered communication.* Automatica, 2021. **133**: p. 109841.

[61] Li, Y., et al., *Observer-based adaptive optimized control for stochastic nonlinear systems with input and state constraints.* IEEE transactions on neural networks and learning systems, 2021. **33**(12): p. 7791-7805.

[62] Ren, X.-X. and G.-H. Yang, *Modified Kalman Filtering for Stochastic Nonlinear Systems Under Non-Gaussian–Lévy Noise and Cyber Attacks.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022. **53**(2): p. 1222-1232.

[63] Yuan, Y. and J. Zhao, *Fast finite-time stability and its application in adaptive control of high-order stochastic nonlinear systems.* Processes, 2022. **10**(9): p. 1676.

[64] Yuan, Y., et al., *Practically Fast Finite-Time Stability in the Mean Square of Stochastic Nonlinear Systems: Application to One-Link Manipulator.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2023.

[65] Zhang, Q. and Y. Zhou, *Recent advances in non-Gaussian stochastic systems control theory and its applications.* International Journal of Network Dynamics and Intelligence, 2022: p. 111-119.

[66] Liang, H., et al., *Cooperative fault-tolerant control for networks of stochastic nonlinear systems with nondifferential saturation nonlinearity.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020. **52**(3): p. 1362-1372.

[67] Luo, A.C.J., *Periodic Flows to Chaos Based on Discrete Implicit Mappings of Continuous Nonlinear Systems.* Int. J. Bifurc. Chaos, 2015. **25**: p. 1550044:1-1550044:62.

[68] Bian, T. and Z.-P. Jiang, *Reinforcement Learning and Adaptive Optimal Control for Continuous-Time Nonlinear Systems: A Value Iteration Approach.* IEEE Transactions on Neural Networks and Learning Systems, 2021. **33**: p. 2781-2790.

[69] Yang, Y., et al., *Robust Actor–Critic Learning for Continuous-Time Nonlinear Systems With Unmodeled Dynamics.* IEEE Transactions on Fuzzy Systems, 2021. **30**: p. 2101-2112.

[70] Xue, S., B. Luo, and D. Liu, *Event-Triggered Adaptive Dynamic Programming for Zero-Sum Game of Partially Unknown Continuous-Time Nonlinear Systems.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020. **50**: p. 3189-3199.

[71] Lam, H.K. and W.-K. Ling, *Sampled-data fuzzy controller for continuous nonlinear systems.* Iet Control Theory and Applications, 2008. **2**: p. 32-39.

[72] Lv, Y., et al., *Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics.* International Journal of Control, 2016. **89**: p. 112 - 99.

[73] Grigoras, C. and V. Grigoras, *Synchronizing Analog - Discrete Nonlinear Systems.* Bulletin of the Polytechnic Institute of Iași. Electrical Engineering, Power Engineering, Electronics Section, 2022. **68**: p. 47 - 60.

[74] Erbe, L., B. Jia, and Q. Zhang, *HOMOCLINIC SOLUTIONS OF DISCRETE NONLINEAR SYSTEMS VIA VARIATIONAL METHOD.* Journal of Applied Analysis & Computation, 2019.

[75] Guo, S.H., et al., *Fault Detection and Reconstruction for Discrete Nonlinear Systems via Takagi-Sugeno Fuzzy Models.* International Journal of Control, Automation and Systems, 2018. **16**: p. 2676 - 2687.

[76] Wang, X., et al., *Neural-network-based control for discrete-time nonlinear systems with denial-of-service attack: The adaptive event-triggered case.* International Journal of Robust and Nonlinear Control, 2021. **32**: p. 2760 - 2779.

[77] Mu, C., K. Liao, and K. Wang, *Event-triggered design for discrete-time nonlinear systems with control constraints.* Nonlinear Dynamics, 2021. **103**: p. 2645 - 2657.

[78] Verriest, E.I. *Pseudo Balancing for Discrete Nonlinear Systems.*

[79] Larguech, S., S. Aloui, and A. Chaari. *Sliding mode control for single input single output non linear system.* 2012.

[80] Al-khazraji, A. and K.M. Aljebory. *Robust Adaptive Type-2 Fuzzy Sliding Mode Control for Non-Linear uncertain SISO systems.* 2014.

[81] Yan, H.-s. and A.-M. Kang, *Asymptotic tracking and dynamic regulation of SISO non-linear system based on discrete multi-dimensional Taylor network.* Iet Control Theory and Applications, 2017. **11**: p. 1619-1626.

[82] Huang, Z.H., et al., *Frequency-domain L 2-stability conditions for switched linear and nonlinear SISO systems.* International Journal of Systems Science, 2014. **45**: p. 682 - 701.

[83] Shanin, N., L. Cottatellucci, and R. Schober, *Optimal Transmit Strategy for Multi-User MIMO WPT Systems With Non-Linear Energy Harvesters.* IEEE Transactions on Communications, 2021. **70**: p. 1726-1741.

[84] Feys, T., X. Mestre, and F. Rottenberg, *Self-Supervised Learning of Linear Precoders Under Non-Linear PA Distortion for Energy-Efficient Massive MIMO Systems.* ICC 2023 - IEEE International Conference on Communications, 2022: p. 6367-6372.

[85] Li, H., *Adaptive control of non-affine MIMO systems with input non-linearity and unmodelled dynamics.* The Journal of Engineering, 2019.

[86] Sachan, K. and R. Padhi, *Safety-constrained robust adaptive control for a class of MIMO non-linear systems.* IET Control Theory & Applications, 2020.

[87] Chernyshov, K.R., *A Non-Linear MIMO System Identification Approach Based on the Multiple Maximal Correlation Technique.* 2021 International Russian Automation Conference (RusAutoCon), 2021: p. 950-955.

[88] Palmieri, A., et al., *Tuning and Feasibility Analysis of Classical First-Order MIMO Non-Linear Sliding Mode Control Design for Industrial Applications.* Machines, 2019.

[89] Xia, Y., et al., *Further results on fuzzy sampled-data stabilization of chaotic nonlinear systems. Appl.* Math. Comput., 2020. **379**: p. 125225.

[90] Velmurugan, G. and R. Rakkiyappan, *Hybrid Projective Synchronization of Fractional-Order Chaotic Complex Nonlinear Systems With Time Delays.* Journal of Computational and Nonlinear Dynamics, 2016. **11**: p. 031016.

[91] Mahmoud, G.M., E.E. Mahmoud, and A.A. Arafa, *Passive control of n-dimensional chaotic complex nonlinear systems.* Journal of Vibration and Control, 2013. **19**: p. 1061 - 1071.

[92] Mahmoud, G.M. and E.E. Mahmoud, *Complex modified projective synchronization of two chaotic complex nonlinear systems.* Nonlinear Dynamics, 2013. **73**: p. 2231-2240.

[93] Rabah, K., S. Ladaci, and M. Lashab, *Bifurcation-based fractional-order PIλDμ controller design approach for nonlinear chaotic systems.* Frontiers of Information Technology & Electronic Engineering, 2018. **19**: p. 180-191.

[94] Ahmad, S.G., et al., *A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems.* J. Parallel Distributed Comput., 2016. **87**: p. 80-90.

[95] Messias, V.R., et al., *Combining time series prediction models using genetic algorithm to autoscaling Web applications host-*

ed in the cloud infrastructure. Neural Computing and Applications, 2015. **27**: p. 2383 - 2406.

[96] Bhoskar, T., et al., *Genetic Algorithm and its Applications to Mechanical Engineering: A Review.* Materials Today: Proceedings, 2015. **2**: p. 2624-2630.

[97] Hamdia, K.M., X. Zhuang, and T. Rabczuk, *An efficient optimization approach for designing machine learning models based on genetic algorithm.* Neural Computing and Applications, 2020. **33**: p. 1923 - 1933.

[98] Pramanik, S., R. Singh, and R. Ghosh, *Application of bi-orthogonal wavelet transform and genetic algorithm in image steganography.* Multimedia Tools and Applications, 2020. **79:** p. 17463 - 17482.

[99] Chen, Z., et al., *Online battery state of health estimation based on Genetic Algorithm for electric and hybrid vehicle applications.* Journal of Power Sources, 2013. **240**: p. 184-192.

[100] Jafar-Zanjani, S., S. Inampudi, and H. Mosallaei, *Adaptive Genetic Algorithm for Optical Metasurfaces Design.* Scientific Reports, 2018. **8**.

[101] Tiwari, S. and A. Kumar, *Advances and bibliographic analysis of particle swarm optimization applications in electrical power system: concepts and variants.* Evolutionary Intelligence, 2021. **16**: p. 23-47.

[102] Pozna, C., et al., *Hybrid Particle Filter–Particle Swarm Optimization Algorithm and Application to Fuzzy Controlled Servo Systems.* IEEE Transactions on Fuzzy Systems, 2022. **30**: p. 4286-4297.

[103] Pawan, Y.V.R.N., et al., *Particle swarm optimization performance improvement using deep learning techniques.* Multimedia Tools and Applications, 2022. **81**: p. 27949 - 27968.

[104] Li, H., et al., *An Overall Distribution Particle Swarm Optimization MPPT Algorithm for Photovoltaic System Under Partial Shading.* IEEE Transactions on Industrial Electronics, 2019. **66**: p. 265-275.

[105] Wang, F., H. Zhang, and A. Zhou, *A particle swarm optimization algorithm for mixed-variable optimization problems.* Swarm Evol. Comput., 2021. **60**: p. 100808.

[106] Chen, H., et al., *Particle Swarm Optimization Algorithm with Mutation Operator for Particle Filter Noise Reduction in Mechanical Fault Diagnosis.* Int. J. Pattern Recognit. Artif. Intell., 2020. **34**: p. 2058012:1-2058012:18.

[107] Zare, M., et al., *A Global Best-guided Firefly Algorithm for Engineering Problems.* Journal of Bionic Engineering, 2023: p. 1-30.

[108] Bačanin, N., et al., *A novel firefly algorithm approach for efficient feature selection with COVID-19 dataset.* Microprocessors and Microsystems, 2023. **98**: p. 104778 - 104778.

[109] Jovanovic, D., et al., *Tuning Machine Learning Models Using a Group Search Firefly Algorithm for Credit Card Fraud Detection.* Mathematics, 2022.

[110] Zivkovic, M., et al., *Novel hybrid firefly algorithm: an application to enhance XGBoost tuning for intrusion detection classification.* PeerJ Computer Science, 2022. **8**.

[111] Ewees, A.A., et al., *Improved Slime Mould Algorithm based on Firefly Algorithm for feature selection: A case study on QSAR model.* Engineering with Computers, 2021. **38**: p. 2407 - 2421.

[112] Bačanin, N., et al., *Optimized convolutional neural network by firefly algorithm for magnetic resonance image classification of glioma brain tumor grade.* Journal of Real-Time Image Processing, 2021. **18**: p. 1085 - 1098.

[113] López-Ibáñez, M., *Ant Colony Optimization.* Handbook of Machine Learning, 2001.

[114] Jia, Y.-H., Y. Mei, and M. Zhang, *A Bilevel Ant Colony Optimization Algorithm for Capacitated Electric Vehicle Routing Problem.* IEEE Transactions on Cybernetics, 2021. **52**: p. 10855-10868.

[115] Kishor, A. and C. Chakraborty, *Task Offloading in Fog Computing for Using Smart Ant Colony Optimization.* Wireless Personal Communications, 2021. **127**: p. 1683 - 1704.

[116] Ajeil, F.H., et al., *Grid-Based Mobile Robot Path Planning Using Aging-Based Ant Colony Optimization Algorithm in Static and Dynamic Environments.* Sensors (Basel, Switzerland), 2020. **20**.

[117] Miao, C., et al., *Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm.* Comput. Ind. Eng., 2021. **156**: p. 107230.

[118] Babaee Tirkolaee, E., et al., *A hybrid augmented ant colony optimization for the multi-trip capacitated arc routing problem under fuzzy demands for urban solid waste management.* Waste Management & Research, 2020. **38**: p. 156 - 172.

[119] Zhao, D., et al., *Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy.* Knowl. Based Syst., 2020. **216**: p. 106510.

[120] Guleria, K. and A.K. Verma, *Meta-heuristic Ant Colony Optimization Based Unequal Clustering for Wireless Sensor Network.* Wireless Personal Communications, 2019. **105**: p. 891 - 911.

[121] Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi, *Optimization by Simulated Annealing.* Science, 1983. **220**: p. 671 - 680.

[122] Lee, S. and S.B. Kim, *Parallel Simulated Annealing with a Greedy Algorithm for Bayesian Network Structure Learning.* IEEE Transactions on Knowledge and Data Engineering, 2020. **32**: p. 1157-1166.

[123] Dupanloup, I., S.W. Schneider, and L. Excoffier, *A simulated annealing approach to define the genetic structure of populations.* Molecular Ecology, 2002. **11**.

[124] Lyden, S. and M.E. Haque, *A Simulated Annealing Global Maximum Power Point Tracking Approach for PV Modules Under Partial Shading Conditions.* IEEE Transactions on Power Electronics, 2016. **31**: p. 4171-4181.

[125] Cruz-Chávez, M.A., M.G. Martínez-Rangel, and M.H. Cruz-Rosales, *Accelerated simulated annealing algorithm applied to the flexible job shop scheduling problem.* Int. Trans. Oper. Res., 2017. **24**: p. 1119-1137.

[126] Glover, F.W. and M. Laguna. *Tabu Search.* 1997.

[127] Goeke, D., *Granular tabu search for the pickup and delivery problem with time windows and electric vehicles.* Eur. J. Oper. Res., 2019. **278**: p. 821-836.

[128] Žulj, I., S. Kramer, and M. Schneider, *A hybrid of adaptive large neighborhood search and tabu search for the or-*

der-batching problem. Eur. J. Oper. Res., 2018. **264**: p. 653-664.

[129] Tellez, N., et al., *A Tabu Search Method for Load Balancing in Fog Computing.* International journal of artificial intelligence, 2018. **16**: p. 106-135.

[130] Diabat, A.H., T. Abdallah, and T.H. Le, *A hybrid tabu search based heuristic for the periodic distribution inventory problem with perishable goods.* Annals of Operations Research, 2016. **242**: p. 373-398.

[131] Gao, X., et al., *Turbo-Like Beamforming Based on Tabu Search Algorithm for Millimeter-Wave Massive MIMO Systems.* IEEE Transactions on Vehicular Technology, 2015. **65**: p. 5731-5737.

[132] Moridi, E. and H. Barati, *RMRPTS: a reliable multi-level routing protocol with tabu search in VANET.* Telecommunication Systems, 2016. **65**: p. 127 - 137.

[133] Price, K.V., R. Storn, and J. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization.* 2014.

134] Singh, D., et al., *Classification of COVID-19 patients from chest CT images using multi-objective differential evolution–based convolutional neural networks.* European Journal of Clinical Microbiology & Infectious Diseases, 2020. **39**: p. 1379 - 1389.

[135] Biswas, S., et al., *Improving Differential Evolution through Bayesian Hyperparameter Optimization.* 2021 IEEE Congress on Evolutionary Computation (CEC), 2021: p. 832-840.

[136] Gao, S., et al., *A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models.* Energy Conversion and Management, 2021. **230**: p. 113784.

[137] Liu, W., et al., *Coordinated Charging Scheduling of Electric Vehicles: A Mixed-Variable Differential Evolution Approach.* IEEE Transactions on Intelligent Transportation Systems, 2020. **21**: p. 5094-5109.

[138] Li, S., et al., *An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models.* Energy Conversion and Management, 2020.

[139] Yu, X., C. Li, and J. Zhou, *A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios.* Knowl. Based Syst., 2020. **204**: p. 106209.

[140] Cuevas, E. and A. Rodríguez. *Artificial Bee Colony (ABC) Algorithm.* 2020.

[141] Karaboğa, D., B. Akay, and N. Karaboga, *A survey on the studies employing machine learning (ML) for enhancing artificial bee colony (ABC) optimization algorithm.* Cogent Engineering, 2020. *7*.

[142] Savsani, P.V., R.L. Jhala, and V.J. Savsani, *Optimized trajectory planning of a robotic arm using teaching learning based optimization (TLBO) and artificial bee colony (ABC) optimization techniques.* 2013 IEEE International Systems Conference (SysCon), 2013: p. 381-386.

[143] N, P., et al., *AI Based Tagging the Optimal Power Point of a Solar Photovoltaic Using Artificial Bee Colony (ABC) Optimization Algorithm.* 2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2023. **10**: p. 1428-1432.

[144] Cui, Y., *Optimizing decision trees for English Teaching Quality Evaluation (ETQE) using Artificial Bee Colony (ABC) optimization.* Heliyon, 2023. **9**.

[145] Singh, T. and M.K. Sandhu, *An Approach in the Software Testing Environment using Artificial Bee Colony (ABC) Optimization.* International Journal of Computer Applications, 2012. **58**: p. 5-7.

[146] Rostami, O. and M. Kaveh, *Optimal feature selection for SAR image classification using biogeography-based optimization (BBO), artificial bee colony (ABC) and support vector machine (SVM): a combined approach of optimization and machine learning.* Computational Geosciences, 2021: p. 1-20.

[147] Yang, X.-S. and S. Deb, *Cuckoo Search via Lévy flights.* 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009: p. 210-214.

[148] Jawad, K., et al., *Novel Cuckoo Search-Based Metaheuristic Approach for Deep Learning Prediction of Depression.* Applied Sciences, 2023.

[149] *Task Scheduling in Cloud Computing Based on The Cuckoo Search Algorithm.* Iraqi Journal of Computer, Communication, Control and System Engineering, 2022.

[150] Liu, C., et al., *Solving the Multi-Objective Problem of IoT Service Placement in Fog Computing Using Cuckoo Search Algorithm.* Neural Processing Letters, 2022. **54**: p. 1823 - 1854.

[151] Imran, M., et al., *Intrusion detection in networks using cuckoo search optimization.* Soft Computing, 2022. **26**: p. 10651 - 10663.

[152] Eltamaly, A.M., *An Improved Cuckoo Search Algorithm for Maximum Power Point Tracking of Photovoltaic Systems under Partial Shading Conditions.* Energies, 2021.