



www.editada.org

Transfer Function Identification with Few-Shots Learning and a Genetic Algorithm Using Proposed Signal-Signature

Martín Montes Rivera¹, Marving Aguilar-Justo², Misael Perez Hernández¹

¹ Research and Postgraduate Studies Department, Universidad Politécnica de Aguascalientes (UPA), Aguascalientes 20342, Mexico

² Engineering Faculty, Universidad Veracruzana (UV), Ixtaczoquitlán Veracruz, 94452, Mexico

E-mails: martin.montes@upa.edu.mx

Abstract. Mathematical models help simulate system dynamics and identifying parameters impacts prediction and control. Various techniques exist for parameter estimation. The DC motor is commonly used for position and speed control due to its ease of use and precision, especially in low-power applications. However, accurately parameterizing with low error can be challenging. Few-shot learning, which involves identifying relevant parameters from a small amount of data, has gained popularity and is particularly useful when working with limited datasets. This work presents a new technique for identifying system transfer functions using few-shot learning. It assigns a unique signature value to each system and has been successfully tested on 1500 randomly generated systems. The approach reduced the search space significantly, enabling successful identification of all systems using a genetic algorithm. R-square values ranged from 0.99 to 1.0, with only 5% of samples falling out of range.

Keywords: System identification, DC Motor, Few-Shots Learning, Genetic Algorithms.

Article Info

Received May 1, 2024

Accepted June 1, 2024

1 Introduction

Mathematical models are used to replicate the dynamics of systems, which contain parameters representing physical constants like mass, inductance, capacitance, and viscous friction. These parameters are usually easy to measure, but they can change during system operation, which can significantly affect the prediction or control of system dynamics. Over the years, several parameter estimation techniques have been developed, including least square, output error, filtering, filter error, artificial neural networks, genetic algorithms, and their derivations. A deep and comprehensive analysis of these techniques is in (Raol et al., 2004) with a focus on their general application rather than specific use cases.

System identification is a critical area of research that involves creating models that accurately reflect the behavior of real-world systems, especially first and second order systems. In industrial settings, precise control systems are essential for web transport systems, and recent studies such as the one in (Giannoccaro & Sakamoto, 2022) have developed a direct identification method that is tailored to address this need. This highlights the importance of evolving methodologies in system identification, which play a crucial role in improving the efficiency and control of complex systems.

The DC motor is a popular choice for position and speed control applications due to its precision and ease of use, especially in low power applications. However, controlling the DC motor requires a mathematical model, which is essentially a linear, second-order system (Ishmeen & Jaswinder, 2018; Rahmatullah et al., 2023; Suman & Giri, 2016).

To express the mathematical model of a system, one can use different forms such as a differential equation, state space equation, or transfer function using Laplace transform. The transfer function comprises poles, zeros, and gain, which constitute the system parameters. The process of determining the parameter values is called identification or estimation, and one can perform it either online or offline (Ishmeen & Jaswinder, 2018; Poovizhi et al., 2017; Semenov et al., 2018).

According to Luengo in (Luengo et al., 2020), more than 300 articles on parameter estimation research have been summarized using the Monte Carlo method. This method is considered effective for real-world applications involving statistical signal processing.

Timofeev in (Timofeev et al., 2018) compared three methods for identifying parameters in linear regression models using computational simulation: ordinary least squares, weighted least squares, and Williamson algorithm.

Odhano's in (Rojas et al., 2018) deals with identifying parameters of oceanographic monitoring vehicles. This is crucial for intelligent control of the system. The process involves using empirical methods to estimate added mass and damping parameters, while rigid body parameters are estimated through computational modeling. To ensure accuracy, experimental tests are also conducted for parameter estimation.

In Gyuk et al. in (Gyuk et al., 2018), researchers utilized artificial intelligence to identify nine parameters of a mathematical model for the metabolism process of individuals who receive insulin due to type 2 diabetes. The approach combines Genetic Algorithms and brute force, resulting in better mean square error and Clarke's Error Grid Analysis compared to other studies on this topic. Estimating motor parameters is a well-researched topic with numerous articles emphasizing the use of automatic control and artificial intelligence techniques.

The paper Obeidat in (Obeidat et al., 2013), presents a method for parameter estimation of a PMDC motor using quantized output observations from sensors. The method is validated through simulation and experimentation, achieving a TSE of less than 0.001 in some tests.

Lian et al. developed a method in (Lian et al., 2019) to determine the inertia of a synchronous permanent magnet motor. They found a sampling period of 25 ms and gains of $k_p=0.008$ and $k_i=0.8$ to be optimal through experimentation. The direct calculation method had a 0.7% percentage error, while the proportional integral regulation method had zero overshoot, a settle time of 13.1 ms, and a steady state error of 0.

Artificial intelligence (AI) tools have gained popularity for parameter identification of electrical motors. This technology has proven its effectiveness in optimizing motor performance and enhancing overall efficiency. With its ability to accurately determine key parameters, AI is a valuable asset in the field of motor engineering.

Rubai and Kotaru in (Rubai & Kotaru, 2000) proposed a method to identify a DC motor using a Multilayered Feedforward Artificial Neural Network with dynamic back propagation. The method accounted for nonlinearities, showed low tracking error, and had shorter convergence time than the literature-reported DBP learning algorithm.

Machine learning, an AI technique, has proven highly effective in various fields, including rotating electrical machines, due to its ability to analyze complex data and identify patterns.

In (Cosmin & Sorin, 2023), Cosmin and Sorin conducted a study where they used reinforcement learning (RL) to control a DC motor with a two-quadrant converter. They trained the model offline to generate optimal control for speed regulation and achieved remarkable results. They recorded a rise time of 28 ms in 110 training episodes, with no long-term error. Furthermore, the tracking was admirable for frequencies below 20 Hz when a sine input was used.

Han et al. in (Han et al., 2022) conducted a study on fault detection in a Brushless DC motor using XGBoost, neural network, and convolutional neural network to detect demagnetization failure by measuring mechanical radial vibration. The three machine learning methods showed satisfactory results in predicting the demagnetization values, with the convolutional neural network achieving the lowest error rates of 0.3 RMSE.

Integral Reinforcement Learning was used in (Bujgoi & Sendrescu, 2022) Bujgoi and Sendrescu to control DC motors in a method that can handle continuous domain systems. A critic neural network is used to evaluate performance. The study includes simulation

and experimental tests using National Instruments hardware and software, showing that the controller can effectively regulate at varying levels.

Few-shot learning is an attractive subcategory of machine learning for identifying relevant parameters from a small amount of data. Object detection and image recognition are popular research fields that utilize this technique.

Li and Liu in (Li & Liu, 2022) developed a few-shot learning method using the Topic Snowball Model to extract structured knowledge from text and create a knowledge graph. Their study found that the Topic Snowball model performed better than the Neural Snowball when few initial seeds were used, although precision was slightly lower.

In a study by Ma et.al. in (Ma et al., 2023), Few-Shot Learning was explored as a fault diagnosis method. The study presented a classification of fault diagnosis methods, which included analytical model, qualitative knowledge, and data-driven approaches. The latter was further divided into statistical analysis, signal processing, and machine learning, where Few-Shot Learning was highlighted.

In this work, we propose a new technique for identifying the transfer function of systems using a few-shot learning approach. Our method uses a unique signature value assigned to each system. To test our approach, we applied a signal with varying length and frequency to 1500 randomly generated systems including 500 1st order systems, 500 2nd order systems and 500 motor systems. For all the experiments we collected six numerical values using exact time sampling. By reducing the signal components and simplifying the search space, we were able to significantly reduce the data size. As a result, we successfully identified all the transfer functions of the randomly generated systems using a genetic algorithm with R-square in ranges 0.99-1.0 with maximum 5% of samples out of the range.

2 Materials and Methods

2.1 First Order Systems and Transfer Function

First-order systems are characterized by having a single time constant and their behavior can be modeled using a first-order linear differential equation. This equation emerges by analyzing processes where the rate of change of a quantity is proportional to the difference between the current input and output (Chuk, 2012a). This model is prevalent in systems such as simple electrical circuits, thermal processes, and biological systems. Let's consider a simple thermal system as an example.

Suppose an object with heat capacity (C) that exchanges heat with its surroundings at a rate proportional to the temperature difference ($T(t) - T_{amb}$), where T_{amb} is the ambient temperature. Newton's law of cooling gives us (Márquez-Rubio et al., 2010): Here (k), it's a heat transfer coefficient.

Normalizing temperature as $y(t) = \frac{T(t) - T_{amb}}{T_{initial} - T_{amb}}$ and assuming an input $x(t)$ that affects temperature, reformulated in equation (1) like in (Márquez-Rubio et al., 2010):

$$\tau \frac{dy(t)}{dt} + y(t) = Kx(t) \tag{1}$$

where:

τ is the time constant of the system. It represents the time required for the system to reach approximately 63.2% of its final value in response to a step input. It reflects how quickly the system responds to changes in input.

$y(t)$ is the output of the system.

K is the gain of the system that relates $x(t)$ to its effect on $y(t)$, functioning as an amplification or attenuation factor.

$x(t)$ is the input of the system.

Time constant (τ) and gain (K) are critical parameters. The value of τ indicates how fast the system reaches its steady state. The transfer function, typically denoted as $G(s)$, is defined as the ratio of the Laplace transform of the output (response) of the system to the Laplace transform of the input (excitation), assuming all initial conditions are zero. It is expressed in equation (2) as in (Campos-Caba & Winckler, 2023):

$$G(s) = \frac{Y(s)}{X(s)} \tag{2}$$

where:

$Y(s)$ is the Laplace transform of the output,

$X(s)$ is the Laplace transform of the input,

s is a complex variable representing frequency ($s = \sigma + j\omega$ in engineering terms, where j is the imaginary unit).

The general transfer function for a first-order system is in equation (3).

$$G(s) = \frac{K}{\tau s + 1} \tag{3}$$

2.2 Second Order Systems and Transfer Function

Second-order systems are described with a second-degree linear differential equation. This equation is typical in systems where not only the output and its rate of change are important, but also the acceleration of that output (second derivative). Common examples include mass-spring-damper systems, RLC circuits, and certain control systems (Tourón et al., 2023).

We consider a system with a mass (m), a shock absorber with a damping coefficient (b), and a spring with constant (k). Newton's second law as in equation (4).

$$m \frac{d^2x(t)}{dt^2} + b \frac{dx(t)}{dt} + kx(t) = F(t) \tag{4}$$

Where $x(t)$ is the displacement and $F(t)$ is an applied external force. This equation is characteristic of systems where output, its rate of change, and its acceleration are relevant.

General Formula: Normalizing and assuming $F(t) = Kx(t)$ (where $x(t)$ is a normalized input and (K) its effect on strength), we get equation (5) as in (Molina, n.d.):

$$\frac{d^2y(t)}{dt^2} + 2\zeta\omega_n \frac{dy(t)}{dt} + \omega_n^2 y(t) = \omega_n^2 Kx(t) \tag{5}$$

where:

$\omega_n = \sqrt{\frac{k}{m}}$ It is the natural undamped frequency of the system. It is the intrinsic frequency of oscillation of the system in the absence of damping, indicating the natural tendency to oscillate (Peñasco-Goel et al., 2023).

$\zeta = \frac{b}{2\sqrt{mk}}$ It's the damping factor. It controls the nature of the system's response, determining whether it is under-damped (oscillatory), critically damped (fast without oscillating), or overdamped (slow and non-oscillatory) (Peñasco-Goel et al., 2023).

K is the gain of the system. Similar to the first-order case, it defines the magnitude of the system's response to a given input (Peñasco-Goel et al., 2023).

Step response: The response depends on the value of ζ .

- For $\zeta < 1$ (under-damped), the response oscillates with a frequency of $\omega_d = \omega_n \sqrt{1 - \zeta^2}$.
- For $\zeta = 1$ (critically damped), the response does not oscillate, but quickly approaches its final value.
- For $\zeta > 1$ (over-damped), the response also does not oscillate and is closer to its final value (Peñasco-Goel et al., 2023).

Key Parameters: Natural frequency (ω_n), damping factor (ζ), and gain (K) are crucial in determining system response.

The general transfer function for a second-order system is in equation (6) (Alvarez Brotons, 2004):

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{6}$$

In both 1st and 2nd order systems, the formulation of the differential equation is based on fundamental physical principles. The time constant (τ) in first-order systems and the natural frequency (ω_n) along with the damping factor (ζ) in second-order systems

are crucial parameters that describe how the system responds to different stimuli. These parameters are often the focus of analysis in system design and response optimization, are usually determined experimentally or calculated from the physical properties of the system (Olivar-Castellanos et al., 2023).

The solution of these differential equations provides the system response function, which describes how the output of the system evolves over time in response to specific inputs.

Understanding these systems and their parameters is essential in a variety of practical applications. For example, in process control, optimizing the response of a first-order system may involve adjusting the time constant to improve the speed of response without causing instability. In vibration control systems (second order), adjusting the damping factor and natural frequency is key to minimizing unwanted oscillations (Chuk, 2012b).

2.3 DC Motor model and Transfer Function

A DC motor is a device that converts direct current electrical energy into mechanical energy in the form of rotation. The rotation occurs due to the interaction between the magnetic field of the stator and the current flowing in the rotor windings. In control systems, a mathematical representation that relates the input and output of a system, known as a transfer function, is often used to describe and analyze the behavior of dynamic systems. The DC motor model can be synthesized in the electrical (equation (7)), mechanical (equation (8)) and electrotechnical force (equation (9)) equations as shown in the Fig. 1 (Poovizhi et al., 2017; Semenov et al., 2018).

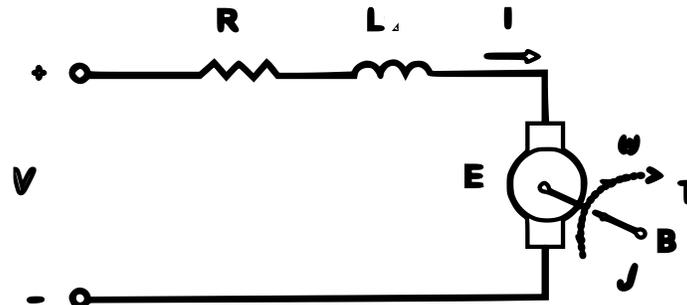


Fig. 1. Model of DC motor

$$V(t) = R \cdot i(t) + L \frac{di(t)}{dt} + e(t) \tag{7}$$

$$J \frac{d\omega(t)}{dt} = -b\omega(t) + k \cdot i(t) \tag{8}$$

$$e(t) = k \cdot \omega(t) \tag{9}$$

For obtaining the transfer function we start by combining electrical and electromechanical force equations as in equation (10).

$$V(t) = R \cdot i(t) + L \frac{di(t)}{dt} + k \cdot \omega(t) \tag{10}$$

Then we use mechanical and electrical equations fixed for current and relate them as in equation (11).

$$J \frac{d\omega(t)}{dt} + b\omega(t) = k \cdot \frac{V(t) - k \cdot \omega(t)}{R + L \frac{d}{dt}} \tag{11}$$

Applying Laplace transform and rearranging we get equation (12) with the transfer function of speed to voltage.

$$G(s) = \frac{\Omega(s)}{V(s)} = \frac{k}{((Js + b) * (Ls + R) + K^2)} \tag{12}$$

Similarly, we get equation (13) with the transfer function of position to voltage adding an extra integrator.

$$G(s) = \frac{\Theta(s)}{V(s)} = \frac{k}{((Js + b) * (Ls + R) + K^2)s} \quad (13)$$

2.4 Signature

Identification with classical control methods uses test signals like the step response and measures parameters associated with the TFs, as described in 2.1, 2.2, and 2.3. However, identifying those parameters demands using signals with several samples.

Since supervised algorithms like the GA imply working with populations with several chromosomes across generations evaluating the cost function, training could result in a slow and inefficient learning process.

The signature method proposed in this work changes those signals with several samples by six numerical components. Nevertheless, to still have representative information for identifying a transfer function. From the input to the system response, the six components must be unique.

Therefore, the signature s is a function that maps the signal containing the n_s samples of the system's output $y(t)$ after receiving and input signal $x(t)$, obtaining only six representative components of those changes, i.e., $s(x(t), y(t)): R^{n_s} \mapsto R^6$.

In order to get signature components that are representative of the system's behavior, the $x(t)$ signal must produce changes in length and frequency, allowing an optimization algorithm to perceive the system's constants, as other identification methods in classical control described in 2.1, 2.2, and 2.3.

Consequently, we use the $x(t)$ defined in equation (14) with three stages containing variations in length, faster, and slower frequencies, depending on the settling time with 2% of the variation ($ts_{2\%}$). The first one is a step signal maintained at 0 volts input during $\frac{ts_{2\%}}{2}$, then increases from 0 to 12 volts in $\frac{ts_{2\%}}{10}$ ms and remains at 12 volts until reaching $ts_{2\%}$. The second one varies length from 0 to 12 volts progressively with $\frac{ts_{2\%}}{t}$ and frequency with the time-squared t^2 from 0 to $1,000 \cdot ts_{2\%}^2$ Hz. The third one varies in length from 0 to 12 volts with $\frac{ts_{2\%}}{t}$ and frequency with t^2 from 0 to $0.5 \cdot ts_{2\%}^2$ Hz.

$$x(t) = \begin{cases} 0, & 0 \leq t \leq \frac{ts_{2\%}}{2} \\ 12 \cdot \min\left(\frac{t - \frac{ts_{2\%}}{2}}{\frac{ts_{2\%}}{10}}\right), & \frac{ts_{2\%}}{2} \leq t \leq ts_{2\%} \\ 12 \cdot \frac{ts_{2\%}}{t} \cdot \sin(1000t^2), & ts_{2\%} \leq t \leq 2ts_{2\%} \\ 12 \cdot \frac{ts_{2\%}}{t} \cdot \sin(0.5t^2), & 2ts_{2\%} \leq t \leq 3ts_{2\%} \end{cases} \quad (14)$$

Additionally, the $x(t)$ signal can generate less data for simulation when reducing the number of samples, allowing it to require fewer training points in a supervised learning algorithm. Moreover, this approach reduces the number of training points used in few-shots learning.

The process with few-shots learning used in this work with signature components starts by supplying $x(t)$ signal to the 1st order system, second order system or motor system.

The Fig. 2 shows the results of applying $x(t)$ signal with 5,000 samples generated to a motor with $J = 3.2284E - 06$, $b = 3.5077E - 06$, $K = 0.0274$, $R = 4.0000$, $L = 2.7500E - 06$ and $ts_{2\%} = 0.08001600320064013$. Similarly, the Fig. 3 shows $x(t)$ for the same motor but with six samples and Fig. 4 shows the six samples stem, showing directly the components of the signature.

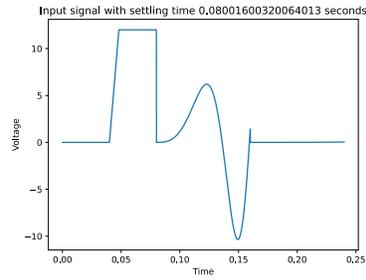


Fig. 2. Example of input signal $x(t)$ for signature generation with 5,000 samples.

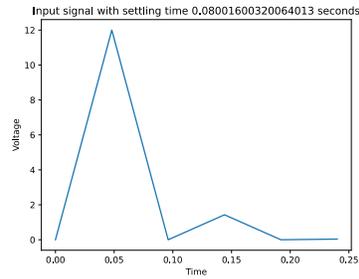


Fig. 3. Example of input signal $x(t)$ for signature generation with six samples.

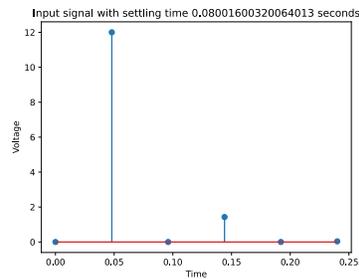


Fig. 4. Stem of example input signal $x(t)$ for signature generation with six components.

$x(t)$ produces different results with different samples, but even with fewer points, it is still representative, reducing the required data for training as occurs in the few-shots approach in **Fig. 5**.

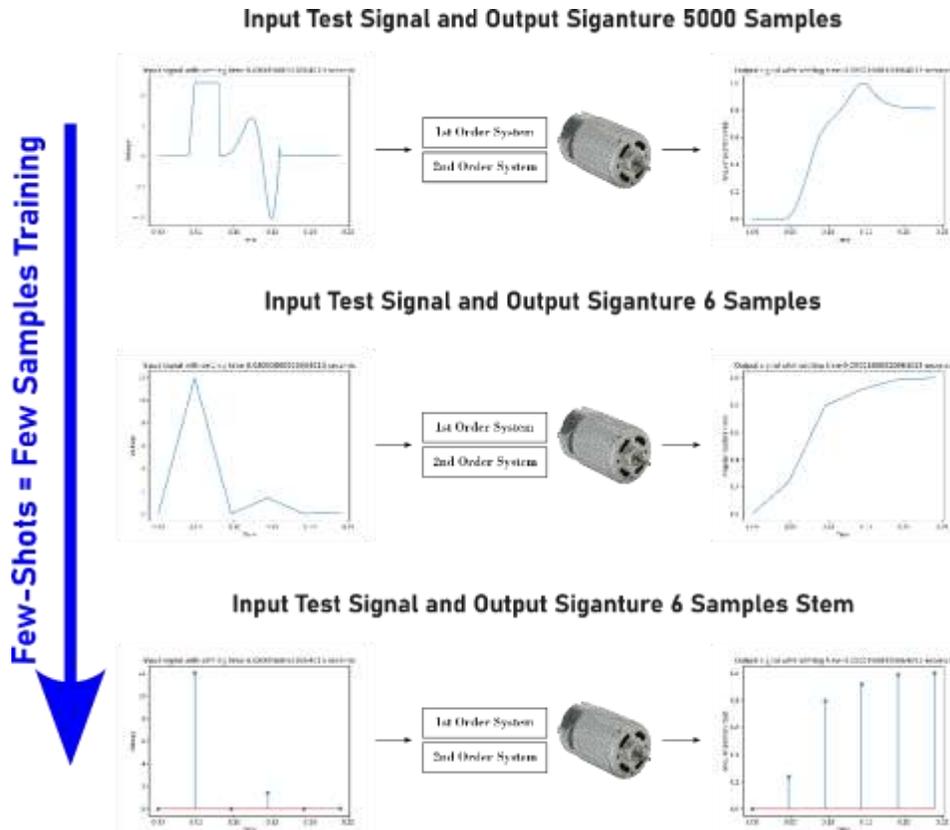


Fig. 5. Signature system identification reducing samples for few-shots learning.

In the case of the motor system described before, the output $y(t)$ is also the angular position of the motor $\theta(t)$. The results of applying $x(t)$ with 15,000 samples are in Fig. 6.

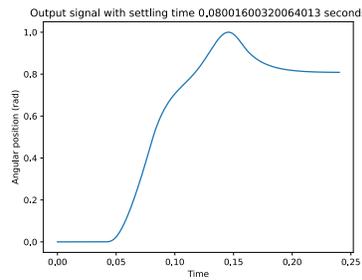


Fig. 6. Output $y(t) = \theta(t)$ after simulation with $x(t)$ using 15,000 samples.

Similarly, the signature response $\theta(t)$ decreases its samples or components if $x(t)$ decreases too. Fig. 7 shows the output signature obtained with six samples in $x(t)$, and Fig. 8 shows the stem of the same output with precise signature components used as training points in our proposal. However, as fewer components in $\theta(t)$, less representative is the signal. Therefore, the user must find a good value of samples for training and converge to a suitable candidate solution.

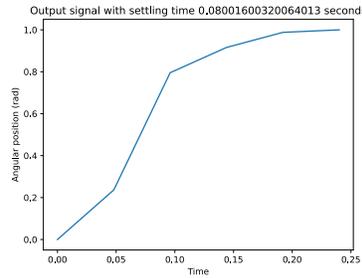


Fig. 7. Output $y(t) = \theta(t)$ after simulation with $x(t)$ using 6 samples.

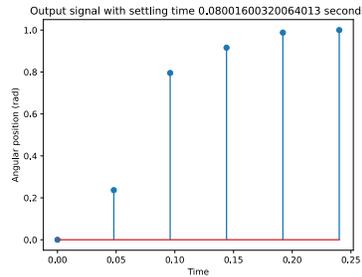


Fig. 8. Stem of $y(t) = \theta(t)$ after simulation with $x(t)$ using 6 samples.

2.4 Genetic Algorithm

Understanding these systems and their parameters is essential in a variety of practical applications. For example, in process control, optimizing the response of a first-order system may involve adjusting the time constant to improve the speed of response without causing instability. In vibration control systems (second order), adjusting the damping factor and natural frequency is key to minimizing unwanted oscillations (Chuk, 2012b).

The genetic algorithm is a search technique based on Charles Darwin's theory of natural selection. It selects the fittest candidates for reproduction to generate the next generations (Khatri et al., 2023).

In the natural selection process, the fittest candidates from a population are selected to produce better offspring. The offspring inherit the characteristics of their parents and, if the parents have better fitness, their offspring will have a better chance of surviving (Khatri et al., 2023). This process is repeated to create a generation with the fittest candidates. This concept can be applied to a hunt problem by selecting the best results from a set. The genetic algorithm involves five phases (Aziz et al., 2023; Nilanjan Dey, 2023).

1. Original population
2. Fitness function
3. Selection
4. Crossover
5. Mutation original Population

In the application to solve a problem with genetic algorithms it may vary depending on the problem but in general the following steps are followed (Jebari & Madiafi, 2013; Nilanjan Dey, 2023):

1. **Problem Definition:** To begin with a genetic algorithm, you must first identify the problem you want to solve, set a clear objective, and specify the variables to optimize or discover.
2. **Gene Encoding:** Define the variables or features that will make up each individual in the population to represent candidate solutions.
3. **Generate an Initial Population:** Generate a group of possible solutions to start your genetic algorithm, either randomly or using an appropriate initialization method.
4. **Fitness Function:** Create a fitness function to assess candidate solutions' quality for problem optimization. Assign a numerical value to each individual that reflects its quality relative to the objective.
5. **Selection:** Select individuals for reproduction based on their fitness, with higher fitness individuals being more likely to be chosen. Use different methods for selection, such as roulette wheel or tournament selection.

6. Crossover: Create new offspring by combining selected candidate solutions in pairs.
7. Mutation: This allows for the introduction of diversity into the population and prevents premature convergence.
8. Replacement: Determine how individuals from the previous generation will be replaced by the new offspring.
9. Termination Criterion: Define a criterion that signals when the algorithm should halt. This can be a maximum number of generations, achieving a target fitness value, or any other problem-specific criterion.
10. Iteration: Iterate through steps 5 to 9 for several generations until the termination criterion is met.
11. Result: Upon meeting the termination criterion, you will obtain a solution that approximates the optimal solution according to your fitness function. This solution represents the best solution found by the genetic algorithm (Jebari & Madiafi, 2013; Nilanjan Dey, 2023).

Key operators in a Genetic Algorithm include selection, crossover, and mutation. These operators are responsible for generating new diversity and driving the evolution of the population towards optimal solutions (Alzanin et al., 2023).

Selection: Selection methods, such as roulette wheel, tournament, or rank-based selection, determine which individuals are chosen for reproduction based on their fitness (Altarabichi et al., 2023).

Crossover: Crossover combines genetic information from two or more parents to create offspring. Common types of crossovers include single-point, multi-point, and uniform crossover (Altarabichi et al., 2023).

Mutation: Mutation introduces random changes into a chromosome's genes to explore new solutions in the search space (Altarabichi et al., 2023).

In this work we use a genetic algorithm with tournament selection, single-point crossover and uniform mutations all those mechanisms are described below:

Selection Method

Tournament selection (**Algorithm 1**) involves randomly choosing a small number of individuals from the population (P) with limit in the population size (P_s), referred to as the tournament size (t_s), and then selecting the best individual from that group for reproduction. This process is repeated multiple times to form the new generation as shown in the pseudocode below (Alzanin et al., 2023; Jebari & Madiafi, 2013; Nilanjan Dey, 2023).

1	function Tournament (P, t_s)
2	for 0 to P_s
3	Set best to 0
4	for 0 to t_s
5	Get current random element from population
6	If current element's fitness > best fitness
7	Current = best
8	End if
9	End for
10	Add best to mating pool.
11	End for

Algorithm 1 Tournament Selection

Advantages: Tournament selection maintains good genetic diversity, allowing even less fit individuals a chance of being selected. It can adapt well to multimodal problems with multiple optimal solutions (Alzanin et al., 2023; Jebari & Madiafi, 2013; Nilanjan Dey, 2023).

Disadvantages: The choice of tournament size can significantly impact algorithm performance, and determining the optimal size can be challenging. It does not guarantee that the fittest individuals are selected on every occasion (Alzanin et al., 2023; Jebari & Madiafi, 2013; Nilanjan Dey, 2023).

Crossover

The primary goal of crossover is to generate offspring that inherit favorable traits from both parents, potentially leading to the convergence towards optimal or improved solutions within the search space (Kieszek et al., 2023). The crossover mechanism with one single crossover point work with the below elements:

- Selection of Crossover Points: In the case of single-point or multi-point crossovers, random crossover points are selected on the parents' chromosomes.
- Offspring Creation: Segments of the parents' chromosomes before and after the crossover points are exchanged to form the chromosomes of the offspring.
- Number of Offspring: Depending on the genetic algorithm's design, crossover can generate one or more offspring for each pair of parents.

In this method, one single random crossover point is chosen on the chromosomes of the parents, and the segments before and after that point are exchanged to create offspring. The **Algorithm 2** shows how it works (Ali & Saeed, 2023).

```

1  function SinglePointCrossover(parent1, parent2):
2      length = length(parent1)
3      crossover_point = random_integer(1, length - 1)
4      child1 = []
5      child2 = []
6      for i in range(length):
7          if i < crossover_point:
8              child1.append(parent1[i])
9              child2.append(parent2[i])
10         else:
11             child1.append(parent2[i])
12             child2.append(parent1[i])
13     return child1, child2
14 end SinglePointCrossover

```

Algorithm 2 Single-Point Crossover

Mutation

Mutation is a fundamental operator in genetic algorithms that introduces randomness into the population by individually altering genes within an individual. Mutation plays a critical role in maintaining genetic diversity within the population and helps prevent premature convergence towards a suboptimal solution (Zeng et al., 2022). The mutation process implies working with the below elements:

Selection of the Individual. First, one or more individuals from the population are selected to undergo mutation. Selection can be random or follow specific criteria.

Selection of Genes. Next, one or more genes are chosen from the selected individual to be mutated. Gene selection is often random, with a mutation rate determining the probability of mutating each gene (Greenstein et al., 2023).

Gene Modification. The selected genes are then modified according to a mutation process. The nature of this modification depends on the problem and can vary widely. Some common examples of mutation include (Gen & Lin, 2023):

- Bit Mutation. - In binary problems, such as combinatorial optimization problems, a bit is flipped, changing 0 to 1 or vice versa.
- Value Mutation. In numerical problems, gene values are altered by adding or subtracting a small random value.
- Permutation Mutation: In problems involving permutations (e.g., the traveling salesman problem), two elements in a sequence may be swapped.
- Insertion or Deletion Mutation: In problems with sequences (e.g., genetic sequence design), an element can be inserted or removed from the sequence (Gen & Lin, 2023).

Creation of the New Individual:

The mutated individual is created by taking the original genes of the individual, with the mutated genes replacing the previously selected genes. This results in a new individual that is now part of the population (Gen & Lin, 2023).

Mutation Rate:

The mutation rate (P_m) is a critical parameter that controls the probability of an individual gene being mutated. A low mutation rate can maintain genetic stability within the population, while a high rate can increase exploration of the search space for solutions. The choice of the mutation rate depends on the problem and should be experimentally tuned to strike an appropriate balance between exploration and exploitation (Gen & Lin, 2023). The **Algorithm 3** describe the process of mutation.

```

1  function Mutation (individual,  $p_m$ ):
2  mutated_individual = copy(individual)
3  for i in range(length(mutated_individual)):
4  random_value = random_uniform(0, 1)
5  if random_value <  $p_m$ :
6  mutated_gene = mutate_gene(mutated_individual[i])
7  mutated_individual[i] = mutated_gene
8  return mutated_individual
9  end Mutation
    
```

Algorithm 3 Mutation

3 Results and Discussion

Our experiment includes obtaining the metrics and analyze 1500 randomly generated transfer functions with 500 1st order systems, 500 2nd order systems, and 500 motor systems according to the structures defined in sections 2.1, 2.2, and 2.3.

3.1 Results 1st Order Systems

We test the system identification with the 1st order systems by randomly generating a 500 transfer functions with the few-shots approach described in 2.4. The first ten randomly generated transfer functions with the structure defined in 2.1 and their corresponding six signature components ($c_1, c_2, c_3, c_4, c_5,$ and c_6) are in **Table 1**.

Table 1. First ten randomly generated 1st order transfer functions with six components.

ID	Randomly Generated Transfer Function	c_1	c_2	c_3	c_4	c_5	c_6
1	$\frac{0.548813503927325}{0.715189366372419s + 1.0}$	0.00E+00	9.50E-01	5.38E-01	1.00E+00	5.43E-01	-6.12E-01
	$\frac{0.417022004702574}{0.720324493442158s + 1.0}$	0.00E+00	1.00E+00	5.67E-01	7.48E-01	3.98E-01	-7.00E-01
2	$\frac{0.435994902142004}{0.0259262318278913s + 1.0}$	0.00E+00	1.00E+00	5.64E-01	-7.78E-01	-4.64E-01	-3.85E-02
	$\frac{0.550797902574576}{0.708147822618105s + 1.0}$	0.00E+00	1.00E+00	5.67E-01	9.92E-01	5.37E-01	-5.92E-01
3	$\frac{0.967029839013677}{0.547232249175722s + 1.0}$	0.00E+00	1.00E+00	5.67E-01	-8.83E-01	-5.26E-01	7.00E-01
	$\frac{0.221993171089739}{0.870732306177376s + 1.0}$	0.00E+00	1.00E+00	5.67E-01	-8.96E-01	-5.34E-01	-5.14E-01
4	$\frac{0.892860151436002}{0.331979805301177s + 1.0}$	0.00E+00	1.00E+00	5.67E-01	-5.93E-01	-3.61E-01	7.13E-01
	$\frac{0.0763082893739572}{0.779918792240115s + 1.0}$	0.00E+00	9.54E-01	5.40E-01	-8.43E-01	-5.02E-01	-1.00E+00
5	$\frac{0.873429402791816}{0.968540662820932s + 1.0}$	0.00E+00	1.00E+00	5.67E-01	7.90E-01	4.22E-01	8.22E-01
	$\frac{0.0103741538857}{0.501874592148739s + 1.0}$	0.00E+00	1.00E+00	5.67E-01	2.45E-01	1.13E-01	9.47E-01

After that, we use the GA to optimize the 500 randomly generated TFs, obtaining the TFs and signature components in **Table 2**. These results use the input parameters experimentally selected for the GA ($P_s = 2000, t_s = 100, n_g = 2, n_a = 16, p_m = 0.04, dec = 65,535, g = 15,000, c_d = 1.00 + E - 05$) and the cost function is the Mean Square Error (MSE) between the coefficients of the desired signature ($c_1, c_2, c_3, c_4, c_5,$ and c_6) and those obtained ($c_{1o}, c_{2o}, c_{3o}, c_{4o}, c_{5o},$ and c_{6o}). We use $n_g = 2,$ because the two genes optimized are the gain K and the τ time constant used in first order systems as described in 2.1.

Table 2. Results first ten TFs and signature coefficients obtained for 1st order systems with GA.

ID	Transfer Function Obtained with GA	c_{1o}	c_{2o}	c_{3o}	c_{4o}	c_{5o}	c_{6o}
1	$\frac{0.548378728923476}{0.718822003509575s + 1.0}$	0.00E+00	9.47E-01	5.39E-01	9.98E-01	5.44E-01	-6.09E-01
	$\frac{0.416983291371023}{0.714564736400397s + 1.0}$	0.00E+00	1.00E+00	5.64E-01	7.49E-01	3.96E-01	-7.03E-01
2	$\frac{0.435919737544823}{0.0257419699397269s + 1.0}$	0.00E+00	1.00E+00	5.62E-01	-7.82E-01	-4.62E-01	-3.76E-02
	$\frac{0.548378728923476}{0.706370641641871s + 1.0}$	0.00E+00	9.97E-01	5.63E-01	9.89E-01	5.34E-01	-5.90E-01
3	$\frac{0.969314106965743}{0.545601586938277s + 1.0}$	0.00E+00	1.00E+00	5.67E-01	-8.87E-01	-5.26E-01	7.04E-01
	$\frac{0.222751201647974}{0.87888914320592s + 1.0}$	0.00E+00	9.99E-01	5.71E-01	-8.94E-01	-5.38E-01	-5.15E-01

7	$\frac{0.887373159380484}{0.329655909056229s + 1.0}$	0.00E+00	9.97E-01	5.61E-01	-5.92E-01	-3.58E-01	7.11E-01
8	$\frac{0.0766765850308995}{0.78619058518349s + 1.0}$	0.00E+00	9.55E-01	5.45E-01	-8.42E-01	-5.06E-01	-1.00E+00
9	$\frac{0.872083619439994}{0.957808804455634s + 1.0}$	0.00E+00	1.00E+00	5.63E-01	7.91E-01	4.19E-01	8.23E-01
10	$\frac{0.0107423514152743}{0.531258106355383s + 1.0}$	0.00E+00	1.01E+00	6.03E-01	2.58E-01	1.22E-01	9.57E-01

Table 3 shows the cost value for the best and worst chromosomes in the population, the number of generations required to reach the desired cost $c_d = 1.00 + E - 05$, the average time per generation, and the MSE, MAE, MAPE, and R-square metrics for the obtained six-component signature in 1st order systems. Additionally, we show in **Fig. 9** the convergence diagram comparison in best fitness for the first ten 1st order systems.

Table 3. Metrics of the first ten solutions with six component signatures in 1st order systems.

id	best	worst	MSE	MAE	MAPE	R2	Generations	Average time per generation (seconds)
1	4.08E-06	1.20E+00	4.08E-06	1.70E-03	2.31E-03	1.00E+00	1.00E+00	3.73E+00
2	6.21E-06	9.70E-04	6.21E-06	2.14E-03	3.23E-03	1.00E+00	5.22E+02	2.48E-02
3	5.32E-06	3.49E-01	5.32E-06	1.92E-03	6.42E-03	1.00E+00	4.20E+01	1.80E-01
4	7.60E-06	7.87E-01	7.60E-06	2.44E-03	3.46E-03	1.00E+00	1.40E+01	3.88E-01
5	6.58E-06	4.48E-01	6.58E-06	1.96E-03	2.41E-03	1.00E+00	1.60E+01	3.33E-01
6	7.78E-06	4.92E+00	7.78E-06	2.22E-03	3.62E-03	1.00E+00	8.60E+01	7.42E-02
7	8.52E-06	3.48E-01	8.52E-06	2.23E-03	4.04E-03	1.00E+00	3.10E+01	1.80E-01
8	7.94E-06	6.78E+01	7.94E-06	2.14E-03	3.55E-03	1.00E+00	8.50E+01	7.54E-02
9	7.38E-06	4.42E-01	7.38E-06	2.25E-03	3.62E-03	1.00E+00	1.10E+01	4.78E-01
10	4.08E-06	2.87E-04	2.87E-04	1.26E-02	3.56E-02	9.98E-01	1.50E+04	1.89E-02

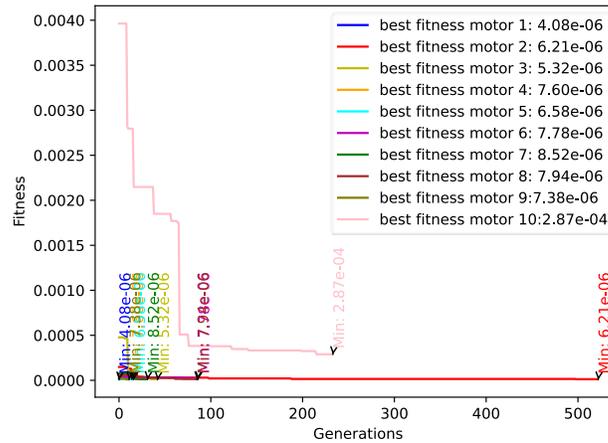


Fig. 9. Convergence diagram comparison in best fitness for the first 10 1st order systems.

Similarly, to identify the effect using the few-shots approach for training, we re-evaluate the candidate solutions using a signature with 15,000 components, i.e., without reducing the data for measuring. The MSE, MAE, MAPE, and R-square in this evaluation are in **Table 4**, showing the error increments changing from six to 15,000 components. However, the almost all the first ten trained TFs per 1st order system remain below 1% MAPE in the 15,000 samples evaluation. Despite that the ID 9 experiment presents 1.69% of MAPE it is an outlier as we show in the binomial test.

Table 4. Metrics of the first ten 1st order solutions with 15,000 component signatures.

ID	MSE	MAE	MAPE	R2
1	1.28E-06	7.25E-04	1.33E-02	1.00E+00
2	2.26E-06	9.22E-04	2.53E-02	1.00E+00
3	3.86E-07	3.15E-04	6.45E-03	1.00E+00
4	5.96E-06	1.18E-03	6.91E-03	1.00E+00
5	2.72E-06	9.29E-04	6.74E-03	1.00E+00
6	4.70E-06	1.20E-03	4.90E-02	1.00E+00
7	1.34E-05	1.77E-03	2.02E-02	1.00E+00
8	6.55E-06	1.28E-03	7.67E-01	1.00E+00
9	4.58E-06	1.28E-03	1.69E+00	1.00E+00
10	1.28E-06	7.25E-04	1.33E-02	1.00E+00

Furthermore, we obtain the step response in the desired transfer functions, which is a widely used criterion for system identification in classical control and compare them with the response of the obtained transfer functions. The metrics of this comparison are provided in **Table 5**.

Table 5. Metrics of the first ten 1st order systems step responses comparison.

ID	MSE	MAE	MAPE	R2
1	6.31E-07	7.34E-04	1.85E-03	1.00E+00
2	3.33E-07	3.98E-04	1.60E-03	1.00E+00
3	2.66E-07	3.54E-04	1.38E-03	1.00E+00
4	4.16E-06	1.93E-03	3.87E-03	1.00E+00
5	5.74E-06	2.36E-03	2.99E-03	1.00E+00
6	3.01E-07	4.94E-04	2.62E-03	1.00E+00
7	1.91E-05	4.00E-03	4.71E-03	9.99E-01
8	7.84E-08	2.50E-04	3.50E-03	1.00E+00
9	1.99E-06	1.23E-03	2.13E-03	1.00E+00
10	7.73E-08	2.47E-04	2.54E-02	9.85E-01

Fig. 10 presents a boxplot that helps us to understand the variability of the metrics obtained from the 500 1st order systems. The analysis reveals that the R-square metrics, namely R2r, R2so, and R2s, show an average of 1.0 for the system response, regression with a signature of 6, and regression with 15,000 points.

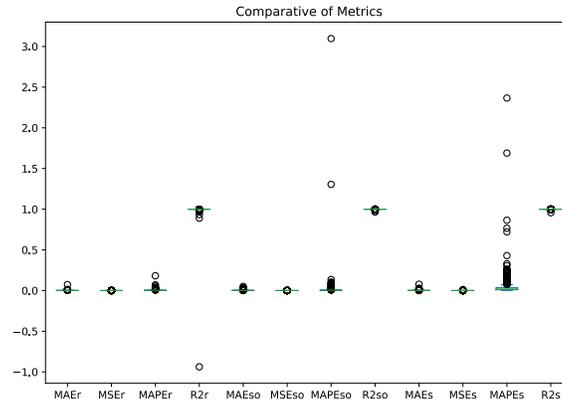


Fig. 10. Boxplot comparison of the different metrics with the system response, the signature with 6 and with 15,000 points, for the 500 1st order systems.

Fig. 11, Fig. 12 and Fig. 13 show the dispersion curves for the system response (R2r), regression with signature of six points (R2s0), and with 15,000 points (R2s), respectively. These figures provide insights into the averages of the R-square metrics for the 500 1st order systems. The curves' means are around 1.0, with upper and lower boundaries of 0.92 and 1.075 in the worst case.

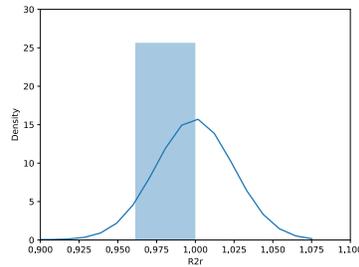


Fig. 11. Dispersion curve of R-square in the system response for the 500 1st order systems.

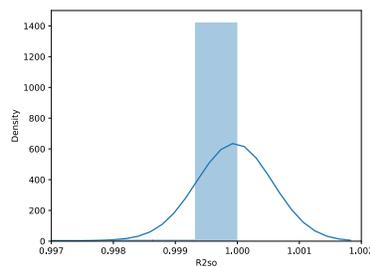


Fig. 12. Dispersion curve of R-square in signature with 6 points for the 500 1st order systems.

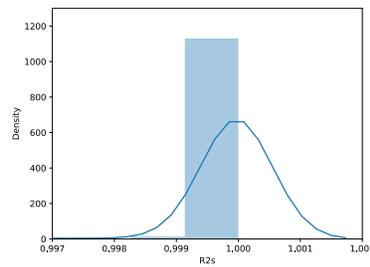


Fig. 13. Dispersion curve of R-square in signature with 15,000 points for the 500 1st order systems.

Moreover, we conducted an analysis to determine the boundaries of variation and the quality of regression obtained from GA transfer function using the six components signature. To evaluate the R-square metrics, we used a binomial test with an acceptable range of variation between 0.99 and 1.0 for the 500 randomly generated 1st order systems. We accepted up to a 5% probability of having a metric outside this range. The results for the system response (R2s) **Table 6**, regression with 6 components signature in **Table 7**, and 15,000 components signature in **Table 8**.

Our analysis showed that in all the regressions, we successfully maintained outliers in the range of 0.99-1.0 below 5%.

Table 6. Binomial test for R2 in the response of the system in the 500 1st order systems.

Data	Value
Range	0.99-1.0
Values within the range	490
Values outside the range	10
Null hypothesis p	0.05
Expected value	25
Observed p value	10
p-value result	9.308629941783915e-04
F-statistic	0.02

Table 7. Binomial test for R2 in the 6-points signature regression in the 500 1st order systems.

Data	Value
Range	0.99-1.0
Values within the range	498
Values outside the range	2
Null hypothesis p	0.05
Expected value	25
Observed p value	2
p-value result	6.544572734615963e-09
F-statistic	0.004

Table 8. Results of binomial test for R2 with 15,000 points signature regression in the 500 1st order systems

Data	Value
Range	0.99-1.0
Values within the range	499
Values outside the range	1
Null hypothesis p	0.05
Expected value	25
Observed p value	1
p-value result	4.132695012586317e-10
F-statistic	0.002

3.2 Results 2nd Order Systems

We also test the system identification with the 2nd order systems by randomly generating a 500 transfer functions with the few-shots approach described in 2.4. The first ten randomly generated transfer functions with the structure defined in 2.2 and their corresponding six signature components ($c_1, c_2, c_3, c_4, c_5,$ and c_6) are in **Table 9**.

Table 9. First ten randomly generated 2nd order transfer functions with six components.

ID	Randomly Generated Transfer Function	c_1	c_2	c_3	c_4	c_5	c_6
1	$\frac{56.5612383213192}{1.0s^2 + 20.3598943275128s + 103.060944959563}$	0.00E+00	1.00E+00	1.00E+00	9.89E-01	9.04E-01	2.47E-01
2	$\frac{43.4145792251058}{1.0s^2 + 8.16492993608866s + 104.1062071918}$	0.00E+00	1.00E+00	1.40E-01	8.74E-01	1.15E-01	3.57E-01
3	$\frac{4.63148266119851}{1.0s^2 + 6.19039825846692s + 10.6227908593528}$	0.00E+00	8.93E-01	1.00E+00	7.82E-01	7.72E-01	9.34E-01
4	$\frac{55.9810075619527}{1.0s^2 + 13.9306821634323s + 101.636203224963}$	0.00E+00	1.00E+00	7.51E-01	7.48E-01	6.03E-01	1.34E-01
5	$\frac{69.4136420588061}{1.0s^2 + 23.2596491540277s + 71.7802484043353}$	0.00E+00	1.00E+00	7.11E-01	3.27E-01	1.94E-01	6.09E-01
6	$\frac{30.4266977664568}{1.0s^2 + 14.2061143186503s + 137.06141327274}$	0.00E+00	1.00E+00	6.83E-01	9.18E-01	6.90E-01	4.91E-02
7	$\frac{35.6606982254408}{1.0s^2 + 15.4358428683643s + 39.9398474308514}$	0.00E+00	9.38E-01	7.25E-01	1.00E+00	7.29E-01	7.53E-01
8	$\frac{8.89926040864939}{1.0s^2 + 18.1082776924122s + 116.622459783335}$	0.00E+00	6.62E-01	9.92E-01	-5.33E-01	-1.00E+00	-8.37E-02
9	$\frac{140.551858875296}{1.0s^2 + 32.2004976658335s + 160.919541323017}$	0.00E+00	1.00E+00	7.48E-01	-2.14E-01	-2.03E-01	2.10E-01
10	$\frac{0.667061063449609}{1.0s^2 + 14.3659568817935s + 64.3002861533706}$	0.00E+00	7.80E-01	1.00E+00	-6.68E-01	-9.94E-01	2.36E-02

After that, we use the GA to optimize the 500 randomly generated TFs, obtaining the TFs and signature components in **Table 10**. These results use the input parameters experimentally selected for the GA ($P_s = 2000, t_s = 100, n_g = 3, n_a = 16, p_m = 0.04, dec = 65,535, g = 15,000, c_d = 1.00 + E - 05$) and the cost function is the Mean Square Error (MSE) between the coefficients of the desired signature ($c_1, c_2, c_3, c_4, c_5,$ and c_6) and those obtained ($c_{1o}, c_{2o}, c_{3o}, c_{4o}, c_{5o},$ and c_{6o}). We use $n_g = 3$, because the three genes optimized are the gain K and the natural undamped frequency ω_n and the damping factor ζ used in 2nd order systems as described in 2.2.

Table 10. Results first ten TFs and signature coefficients obtained for 2nd order systems with GA.

ID	Transfer Function Obtained with GA	c_{1o}	c_{2o}	c_{3o}	c_{4o}	c_{5o}	c_{6o}
1	$\frac{56.9977427416421}{1.0s^2 + 20.6784751146128s + 104.132783322838}$	0.00E+00	9.96E-01	9.95E-01	9.88E-01	9.01E-01	2.49E-01
2	$\frac{45.8741517270323}{1.0s^2 + 8.43630160775223s + 110.252403309046}$	0.00E+00	1.00E+00	1.34E-01	8.76E-01	1.12E-01	3.57E-01
3	$\frac{4.5559846626064}{1.0s^2 + 6.09950296067459s + 10.460217729257}$	0.00E+00	8.89E-01	1.00E+00	7.78E-01	7.73E-01	9.30E-01
4	$\frac{53.7881002030657}{1.0s^2 + 12.8750994713425s + 98.0911383239066}$	0.00E+00	1.01E+00	7.48E-01	7.39E-01	6.04E-01	1.22E-01
5	$\frac{66.9455967859891}{1.0s^2 + 22.170740966962s + 69.1040777055474}$	0.00E+00	1.01E+00	7.12E-01	3.26E-01	1.93E-01	6.11E-01
6	$\frac{30.7974285371641}{1.0s^2 + 14.6102966708585s + 138.525015729791}$	0.00E+00	9.95E-01	6.86E-01	9.19E-01	6.92E-01	5.43E-02
7	$\frac{37.9416727734761}{1.0s^2 + 16.6054268064669s + 42.4043713156955}$	0.00E+00	9.36E-01	7.24E-01	1.00E+00	7.29E-01	7.57E-01
8	$\frac{9.32431058210019}{1.0s^2 + 19.05830618256s + 120.3127966131}$	0.00E+00	6.72E-01	9.92E-01	-5.31E-01	-9.94E-01	-9.43E-02
9	$\frac{133.999885017814}{1.0s^2 + 30.7838627490388s + 153.140388962096}$	0.00E+00	9.97E-01	7.55E-01	-2.13E-01	-2.05E-01	2.09E-01
10	$\frac{0.411569967456178}{1.0s^2 + 8.77451553114811s + 45.5612125291227}$	0.00E+00	9.96E-01	9.95E-01	9.88E-01	9.01E-01	2.49E-01

Table 11 shows the cost value for the best and worst chromosomes in the population, the number of generations required to reach the desired cost $c_d = 1.00 + E - 05$, the average time per generation, and the MSE, MAE, MAPE, and R-square metrics for the obtained six-component signature in 2nd order systems. Additionally, we show in **Fig. 14** the convergence diagram comparison in best fitness for the first ten 2nd order systems.

Table 11. Metrics of the first ten solutions with six component signatures in 2nd order systems.

id	best	worst	MSE	MAE	MAPE	R2	Generations	Average time per generation (seconds)
1	9.98E-06	4.28E-04	9.98E-06	2.62E-03	3.64E-03	1.00E+00	5.86E+02	2.90E-02
2	9.36E-06	2.25E-05	9.36E-06	2.30E-03	1.26E-02	1.00E+00	1.99E+03	1.72E-02
3	8.84E-06	1.01E-05	8.84E-06	2.63E-03	3.00E-03	1.00E+00	3.68E+03	1.83E-02
4	5.91E-05	5.91E-05	5.91E-05	5.89E-03	2.00E-02	1.00E+00	1.50E+04	1.70E-02
5	7.09E-06	1.90E-01	7.09E-06	1.87E-03	2.92E-03	1.00E+00	1.16E+02	6.36E-02
6	9.35E-06	1.52E-05	9.35E-06	2.41E-03	1.93E-02	1.00E+00	8.94E+02	1.97E-02
7	5.54E-06	1.58E-01	5.54E-06	1.74E-03	2.05E-03	1.00E+00	2.35E+02	3.78E-02
8	4.04E-05	4.04E-05	4.04E-05	4.77E-03	2.51E-02	1.00E+00	1.50E+04	1.71E-02
9	9.48E-06	1.14E-02	9.48E-06	2.13E-03	4.57E-03	1.00E+00	4.61E+02	5.16E-02
10	9.98E-06	4.28E-04	4.30E-03	4.98E-02	6.49E-01	9.92E-01	1.50E+04	1.82E-02

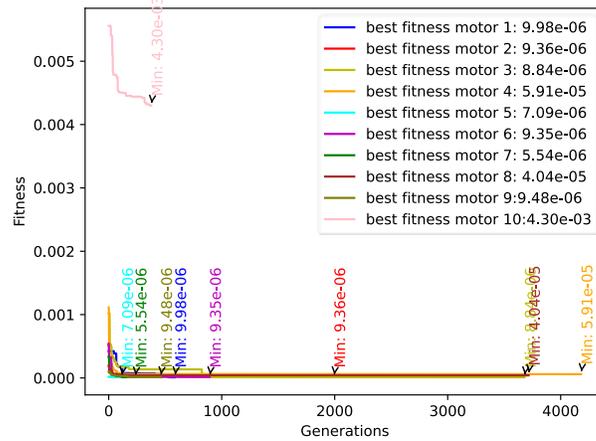


Fig. 14. Convergence diagram comparison in best fitness for the first 10 2nd order systems.

Similarly, to identify the effect using the few-shots approach for training, we re-evaluate the candidate solutions using a signature with 15,000 components, i.e., without reducing the data for measuring. The MSE, MAE, MAPE, and R-square in this evaluation are in **Table 12**, showing the error increments changing from six to 15,000 components. However, almost all the ten trained TFs per 2nd order system remain below 1% MAPE in the 15,000 samples evaluation. Despite that the ID 11 experiment presents 11.15% of MAPE it is an outlier as we show in the binomial test.

Table 12. Metrics of the first ten 2nd order solutions with 15,000 component signatures.

ID	MSE	MAE	MAPE	R2
1	9.37E-06	1.92E-03	8.15E-02	1.00E+00
2	3.72E-04	8.52E-03	6.09E-01	9.98E-01
3	1.30E-06	7.60E-04	1.14E-01	1.00E+00
4	1.50E-04	6.53E-03	4.99E-01	1.00E+00
5	9.15E-06	1.92E-03	7.70E-02	1.00E+00
6	2.70E-05	2.60E-03	9.89E-02	1.00E+00

7	1.49E-05	2.32E-03	1.19E-01	1.00E+00
8	1.27E-04	6.00E-03	8.36E-01	1.00E+00
9	1.27E-05	1.95E-03	3.86E-02	1.00E+00
10	1.08E-02	5.90E-02	1.15E+01	9.77E-01

Furthermore, we obtain the step response in the desired transfer functions, which is a widely used criterion for system identification in classical control and compare them with the response of the obtained transfer functions. The metrics of this comparison are provided in **Table 13**.

Table 13. Metrics of the first ten 2nd order systems step responses comparison.

ID	MSE	MAE	MAPE	R2
1	3.45E-06	1.71E-03	3.73E-03	1.00E+00
2	2.00E-05	3.22E-03	1.08E-02	9.98E-01
3	2.73E-07	4.13E-04	2.41E-03	1.00E+00
4	2.02E-05	3.69E-03	8.77E-03	9.99E-01
5	1.14E-05	3.02E-03	4.57E-03	1.00E+00
6	5.34E-07	5.84E-04	3.19E-03	1.00E+00
7	5.17E-06	1.91E-03	4.88E-03	1.00E+00
8	4.89E-07	6.25E-04	1.33E-02	9.99E-01
9	3.73E-06	1.76E-03	4.37E-03	1.00E+00
10	3.45E-06	1.71E-03	3.73E-03	1.00E+00

Fig. 15 presents a boxplot that helps us to understand the variability of the metrics obtained from the 500 2nd order systems. The analysis reveals that the R-square metrics, namely R2r, R2so, and R2s, show an average of 1.0 for the system response, regression with a signature of 6, and regression with 15,000 points, respectively.

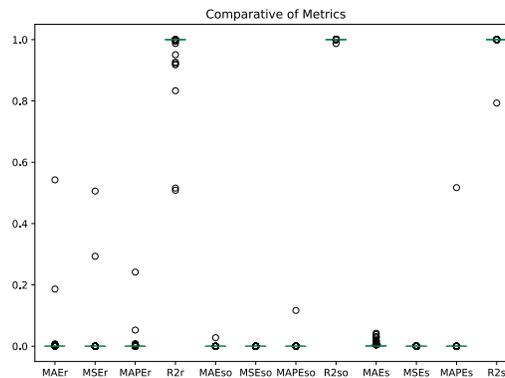


Fig. 15. Boxplot comparison of the different metrics with the system response, the signature with 6 and with 15,000 points, for the 500 2nd order systems.

Fig. 16, **Fig. 17** and **Fig. 18** show the dispersion curves for the system response (R2r), regression with signature of six points (R2so), and with 15,000 points (R2s), respectively. These figures provide insights into the averages of the R-square metrics for the 500 2nd order systems. The curves' means are around 1.0, with upper and lower boundaries of 0.98 and 1.020 in the worst case.

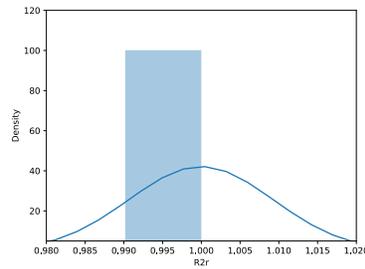


Fig. 16. Dispersion curve of R-square in the system response for the 500 2nd order systems.

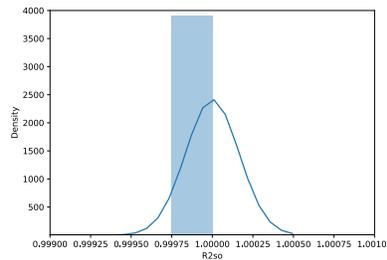


Fig. 17. Dispersion curve of R-square in signature with 6 points for the 500 2nd order systems.

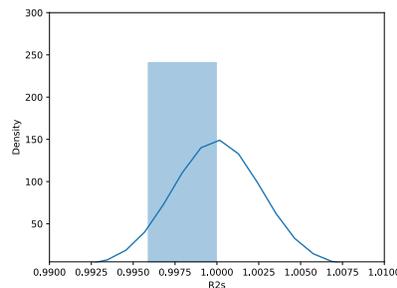


Fig. 18. Dispersion curve of R-square in signature with 15,000 points for the 500 2nd order systems.

Moreover, we conducted an analysis to determine the boundaries of variation and the quality of regression obtained from GA transfer function using the six components signature. To evaluate the R-square metrics, we used a binomial test with an acceptable range of variation between 0.99 and 1.0 for the 500 randomly generated 2nd order systems. We accepted up to a 5% probability of having a metric outside this range. The results for the system response (R2s) **Table 14**, regression with 6 components signature in **Table 15**, and 15,000 components signature in **Table 16**.

Our analysis showed that in all the regressions, we successfully maintained outliers in the range of 0.99-1.0 below 5%.

Table 14. Binomial test for R2 in the response of the system in the 500 2nd order systems.

Data	Value
Range	0.99-1.0
Values within the range	492
Values outside the range	8
Null hypothesis p	0.05
Expected value	25
Observed p value	8
p-value result	1.2196053082001523 e-04

F-statistic	0.016
-------------	-------

Table 15. Binomial test for R2 in the 6-points signature regression in the 500 1st order systems.

Data	Value
Range	0.99-1.0
Values within the range	499
Values outside the range	1
Null hypothesis p	0.05
Expected value	25
Observed p value	1
p-value result	4.132695012586317e-10
F-statistic	0.002

Table 16. Results of binomial test for R2 with 15,000 points signature regression in the 500 2nd order systems

Data	Value
Range	0.99-1.0
Values within the range	499
Values outside the range	1
Null hypothesis p	0.05
Expected value	25
Observed p value	1
p-value result	4.132695012586317e-10
F-statistic	0.002

3.2 Results Motor Identification

The first stage in the proposed method starts by generating a 500 DC motor dataset for validating the transfer function identification achieved with the few-shots signature approach described in 2.4. The first ten randomly generated transfer functions and their corresponding six signature components ($c_1, c_2, c_3, c_4, c_5,$ and c_6) are in **Table 17**.

Table 17. First ten randomly generated transfer functions with the components of their signature.

ID	Randomly Generated Transfer Function	c_1	c_2	c_3	c_4	c_5	c_6
1	$\frac{0.391796194446569}{9.8234408137143 \cdot 10^{-10}s^2 + 0.00126344210671416s + 0.15515071791453}}$	0.00E+00	1.35E-01	4.50E-01	6.34E-01	9.51E-01	1.00E+00
2	$\frac{7.43436312741763 \cdot 10^{-5}}{2.58571841214713 \cdot 10^{-10}s^2 + 0.000532685639229194s + 0.00092011570628111}}$	0.00E+00	1.13E-01	4.05E-01	5.81E-01	8.76E-01	1.00E+00
3	$\frac{0.357280610621161}{7.74350474154462 \cdot 10^{-10}s^2 + 0.000801898049318912s + 0.127697119213499}}$	0.00E+00	1.49E-01	5.01E-01	6.76E-01	9.55E-01	1.00E+00
4	$\frac{0.189088080293414}{2.07799573291231 \cdot 10^{-9}s^2 + 0.00118876038975219s + 0.0372826597621439}}$	0.00E+00	2.83E-01	1.00E+00	8.84E-01	2.30E-01	9.62E-02
5	$\frac{0.63224483397175}{2.85071140511024 \cdot 10^{-9}s^2 + 0.00292052608327723s + 0.401386224878578}}$	0.00E+00	1.38E-01	4.86E-01	6.65E-01	9.44E-01	1.00E+00
6	$\frac{0.134367450970627}{4.58091182819256 \cdot 10^{-10}s^2 + 0.000861586393993046s + 0.0214340381014662}}$	0.00E+00	1.66E-01	5.89E-01	7.55E-01	9.56E-01	1.00E+00
7	$\frac{0.533798929987591}{4.06116969101815 \cdot 10^{-10}s^2 + 0.000157293755739376s + 0.284999781955137}}$	0.00E+00	3.79E-01	9.02E-01	9.47E-01	9.99E-01	1.00E+00
8	$\frac{0.284966000436581}{3.15306285748907 \cdot 10^{-10}s^2 + 0.00023250220978434s + 0.0835895526754455}}$	0.00E+00	2.31E-01	6.82E-01	8.09E-01	9.85E-01	1.00E+00
9	$\frac{0.564976451139048}{8.58823205088309 \cdot 10^{-10}s^2 + 0.00195898544909002s + 0.321370696582876}}$	0.00E+00	1.52E-01	5.04E-01	6.78E-01	9.57E-01	1.00E+00
10	$\frac{0.322252640537195}{6.22884258808504 \cdot 10^{-12}s^2 + 5.8661566754782 \cdot 10^{-6}s + 0.10413053916331}}$	0.00E+00	4.46E-01	9.09E-01	9.54E-01	1.00E+00	1.00E+00

After that, we use the GA to optimize the 500 randomly generated TFs, obtaining the TFs and signature components in **Table 18**. These results use the same input parameters experimentally selected for the GA ($P_s = 1000, t_s = 2, n_g = 5, n_a = 16, p_m = 0.08, dec = 1,000, g = 10,000, c_d = 1.00 + E - 05$) and the cost function is the MAPE between the coefficients of the desired signature ($c_1, c_2, c_3, c_4, c_5,$ and c_6) and those obtained ($c_{10}, c_{20}, c_{30}, c_{40}, c_{50},$ and c_{60}).

Table 18. Results of the first ten TFs and signature coefficients obtained with GA.

ID	Transfer Function Obtained with GA	C_{10}	C_{20}	C_{30}	C_{40}	C_{50}	C_{60}
1	$\frac{0.39465}{2.26899992 \cdot 10^{-10}s^3 + 0.00127277669844898s^2 + 0.156278426208s}$	0.00E+00	1.35E-01	4.50E-01	6.34E-01	9.51E-01	1.00E+00
2	$\frac{0.00034}{1.63502955 \cdot 10^{-9}s^3 + 0.00243601297434437s^2 + 0.004208071077s}$	0.00E+00	1.13E-01	4.05E-01	5.81E-01	8.76E-01	1.00E+00
3	$\frac{0.35287}{1.065423513 \cdot 10^{-9}s^3 + 0.000792032702404111s^2 + 0.126119553609s}$	0.00E+00	1.49E-01	5.01E-01	6.76E-01	9.55E-01	1.00E+00
4	$\frac{0.19668}{3.8234262 \cdot 10^{-10}s^3 + 0.00123651982205871s^2 + 0.038780234136s}$	0.00E+00	2.83E-01	1.00E+00	8.84E-01	2.30E-01	9.62E-02
5	$\frac{0.63093}{1.486766944 \cdot 10^{-9}s^3 + 0.00291444768139098s^2 + 0.400553157376s}$	0.00E+00	1.38E-01	4.86E-01	6.65E-01	9.44E-01	1.00E+00
6	$\frac{0.15551}{1.90768675 \cdot 10^{-10}s^3 + 0.000997181219258281s^2 + 0.024806743912s}$	0.00E+00	1.66E-01	5.89E-01	7.55E-01	9.56E-01	1.00E+00
7	$\frac{0.53201}{4.0680684 \cdot 10^{-11}s^3 + 0.000156837315841788s^2 + 0.284044123978s}$	0.00E+00	3.79E-01	9.02E-01	9.47E-01	9.99E-01	1.00E+00
8	$\frac{0.29155}{3.61943768 \cdot 10^{-10}s^3 + 0.000238615417425186s^2 + 0.085521838031s}$	0.00E+00	2.31E-01	6.82E-01	8.09E-01	9.85E-01	1.00E+00
9	$\frac{0.56325}{1.14732653 \cdot 10^{-10}s^3 + 0.00195307263427662s^2 + 0.32038746885s}$	0.00E+00	1.52E-01	5.04E-01	6.78E-01	9.57E-01	1.00E+00
10	$\frac{0.32313}{1.367769 \cdot 10^{-11}s^3 + 5.866448428706 \cdot 10^{-6}s^2 + 0.104414467493s}$	0.00E+00	4.46E-01	9.09E-01	9.54E-01	1.00E+00	1.00E+00

Table 19 shows the cost value for the best and worst chromosomes in the population, the number of generations required to reach the desired cost $c_d = 1.00 + E - 05$, the average time per generation, and the MSE, MAE, MAPE, and R-square metrics for the obtained six-component signature. Additionally, we show in **Fig. 19** the convergence diagram comparison in best fitness for the first ten motors.

Table 19. Metrics of the first ten candidate solutions with six component signatures.

id	best	worst	MSE	MAE	MAPE	R2	Generations	Average time per generation (seconds)
1	8.84E-06	2.31E-03	3.74E-11	4.33E-06	8.84E-06	1.00E+00	3.62E+03	2.64E-02
2	8.23E-06	6.72E-04	1.57E-11	2.98E-06	8.23E-06	1.00E+00	7.49E+03	2.45E-02
3	9.93E-06	5.33E-04	2.71E-11	3.93E-06	9.93E-06	1.00E+00	5.23E+03	2.52E-02
4	9.62E-06	6.68E-04	1.70E-10	7.79E-06	9.62E-06	1.00E+00	5.52E+03	2.49E-02
5	8.16E-06	1.36E-03	1.50E-11	3.17E-06	8.16E-06	1.00E+00	3.37E+03	2.62E-02
6	6.14E-06	4.18E-04	1.65E-11	3.54E-06	6.14E-06	1.00E+00	8.24E+03	2.58E-02
7	8.90E-06	4.50E-04	2.39E-10	7.61E-06	8.90E-06	1.00E+00	6.53E+03	2.50E-02
8	8.86E-06	1.44E-04	4.22E-11	5.36E-06	8.86E-06	1.00E+00	5.15E+03	2.31E-02
9	3.12E-06	1.31E-03	5.43E-12	1.77E-06	3.12E-06	1.00E+00	3.39E+03	2.50E-02
10	8.91E-06	4.52E-04	6.47E-11	4.99E-06	8.91E-06	1.00E+00	4.67E+03	2.55E-02

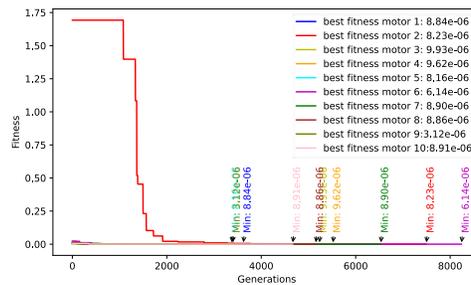


Fig. 19. Convergence diagram comparison in best fitness for the first 10 motors

Similarly, to identify the effect using the few-shots approach for training, we re-evaluate the candidate solutions using a signature with 15,000 components, i.e., without reducing the data for measuring. The MSE, MAE, MAPE, and R-square in this evaluation are in **Table 20**, showing the error increments changing from six to 15,000 components. However, the trained TFs per DC motor remain below 1% MAPE in the 15,000 samples evaluation.

Table 20. Metrics of the first ten candidate solutions with 15,000 component signatures.

ID	MSE	MAE	MAPE	R2
1	1.50E-06	1.07E-03	1.26E-05	1.00E+00
2	2.49E-10	1.32E-05	1.10E-05	1.00E+00
3	1.19E-06	9.57E-04	7.62E-06	1.00E+00
4	9.72E-07	8.63E-04	1.48E-05	1.00E+00
5	2.50E-07	4.38E-04	3.56E-06	1.00E+00
6	9.41E-09	8.53E-05	3.20E-06	1.00E+00
7	1.18E-06	9.50E-04	1.60E-06	1.00E+00
8	7.09E-06	2.33E-03	9.57E-06	1.00E+00
9	1.96E-07	3.87E-04	3.09E-06	1.00E+00
10	5.58E-06	2.07E-03	3.37E-06	1.00E+00

Furthermore, we convert the transfer functions acquired by the GA that express the relationship between position and voltage to those that relate speed and voltage by eliminating an s integrator in the transfer functions. This conversion enables us to model and contrast the step response, which is a widely used criterion for system identification in classical control. The outcomes of this conversion are provided in **Table 21**.

Table 21. Metrics of the first ten candidate solutions among the step responses of the speed in the DC motors.

ID	MSE	MAE	MAPE	R2
1	2.30E-09	4.03E-05	5.98E-05	1.00E+00
2	8.13E-13	8.16E-07	1.45E-05	1.00E+00
3	1.50E-09	2.94E-05	5.21E-05	1.00E+00
4	9.71E-09	9.55E-05	5.30E-05	1.00E+00
5	3.75E-10	1.36E-05	5.86E-05	1.00E+00
6	1.71E-09	3.79E-05	1.28E-05	1.00E+00
7	4.32E-07	1.74E-04	7.95E-04	1.00E+00
8	2.79E-09	4.44E-05	3.65E-05	1.00E+00
9	3.66E-10	1.23E-05	4.43E-05	1.00E+00
10	3.44E-06	1.72E-04	2.05E-04	1.00E+00

Fig. 20 presents a boxplot that helps us to understand the variability of the metrics obtained from the 500 motors. The analysis reveals that the R-square metrics, namely R2r, R2so, and R2s, show an average of 1.0 for the system response, regression with a signature of 6, and regression with 15,000 points.

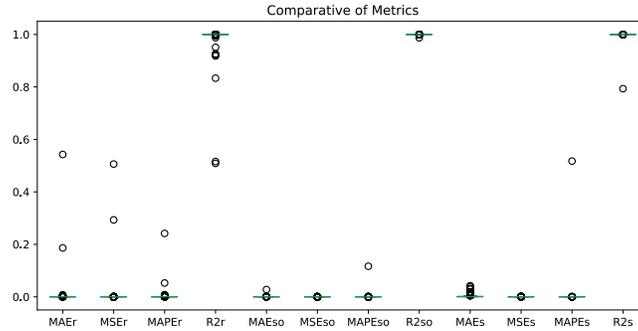


Fig. 20. Boxplot comparison of the different metrics with the system response, the signature with 6 and with 15,000 points, for the 500 motors.

Fig. 21, Fig. 22 and **Fig. 23** show the dispersion curves for the system response (R2r), regression with signature of six points (R2so), and with 15,000 points (R2s), respectively. These figures provide insights into the averages of the R-square metrics for the 500 motors. The curves' means are around 1.0, with upper and lower boundaries of 0.99 and 1.02.

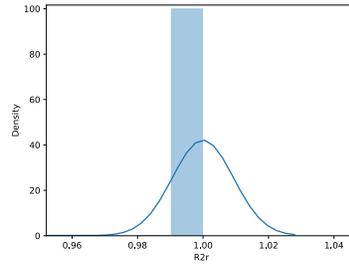


Fig. 21. Dispersion curve of R-square in the system response for the 500 motors.

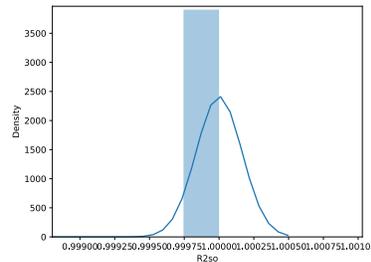


Fig. 22. Dispersion curve of R-square in signature with 6 points for the 500 motors.

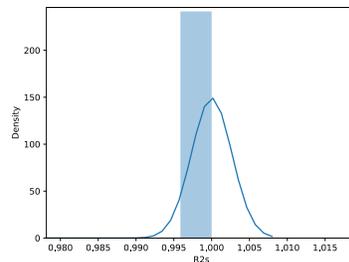


Fig. 23. Dispersion curve of R-square in signature with 15,000 points for the 500 motors.

Moreover, we conducted an analysis to determine the boundaries of variation and the quality of regression obtained from GA transfer function using the six components signature. To evaluate the R-square metrics, we used a binomial test with an acceptable range of variation between 0.99 and 1.0 for the 500 randomly generated motors. We accepted up to a 5% probability of having a metric outside this range. The results for the system response (R2s) are in **Table 22**, regression with 6 components signature in **Table 23**, and 15,000 components signature in **Table 24**.

Our analysis showed that in all the regressions, we successfully maintained outliers in the range of 0.99-1.0 below 5%.

Table 22. Results of binomial test for R2 in the response of the system

Data	Value
Range	0.99-1.0
Values within the range	492
Values outside the range	8
Null hypothesis p	0.05
Expected value	25
Observed p value	8
p-value result	0.00012196053082001523
F-statistic	0.016

Table 23. Results of binomial test for R2 in the signature regression with 6 points

Data	Value
Range	0.99-1.0
Values within the range	492
Values outside the range	8
Null hypothesis p	0.05
Expected value	25
Observed p value	8
p-value result	4.132695012586317e-10
F-statistic	0.002

Table 24. Results of binomial test for R2 in the signature regression with 15,000 points

Data	Value
Range	0.99-1.0
Values within the range	492
Values outside the range	1
Null hypothesis p	0.05
Expected value	25
Observed p value	1
p-value result	4.132695012586317e-10
F-statistic	0.002

4 Conclusions

In this work, we present a new method for identifying transfer function systems using a few-shot learning technique. Our approach involves assigning a distinctive signature value to each system, which we generate by exposing the system to an input signal of varying length and frequency, and recording six numerical values through precise time sampling. By reducing the signal components to just six parameters, our method simplifies and reduces the search space, making it much easier to identify transfer function systems.

In order to verify the effectiveness of our method, we conducted extensive testing on a total of 1500 transfer functions, randomly generated from three different categories: 500 1st order systems, 500 2nd order systems, and 500 motor systems. The optimization process involved minimizing the Mean Squared Error (MSE) for 1st and 2nd order systems, and the Mean Absolute Percentage

Error (MAPE) for motor systems. The objective was to compare the generated transfer function against the one optimized using the Genetic Algorithm (GA).

We performed an analysis on transfer functions, comparing signature and step response metrics with 6 and 15,000 components. Our approach utilized the GA method and yielded highly significant results, with R-square values ranging from 0.99 to 1.0. We evaluated the accuracy of our approach using a binomial test, which produced metrics within the expected range with a 5% probability.

Our research has demonstrated that the application of the signature method to optimize transfer functions yields results of high quality, even when the number of components increases from six to 15,000. Furthermore, the optimization processing time for each generation averaged at 2.51E-02 seconds. Our approach significantly improves existing methods and facilitates the identification process by simplifying the procedure and reducing the required time.

The outcomes of our study demonstrate that the methodology we employed holds substantial promise for future applications in system identification, control, and automation. Our findings suggest that this approach can be effectively harnessed to achieve superior precision, efficiency, and safety in control systems. These promising results highlight the significance of further research and development in this field to fully exploit its potential and investigate its broader implications.

4.1 Future work

Within the scope of this study, we have exclusively identified transfer functions of 1st, 2nd order systems and DC motors using genetic algorithms. Nevertheless, we believe that the process signature method, in conjunction with few-shot approaches, has the potential to be extended to other models or transfer functions of any kind. We recommend repeating the process and assessing the results with different systems to evaluate its effectiveness, and test this approach using other different optimizing algorithms.

References

- Ali, W., & Saeed, F. (2023). Hybrid Filter and Genetic Algorithm-Based Feature Selection for Improving Cancer Classification in High-Dimensional Microarray Data. *Processes* 2023, Vol. 11, Page 562, 11(2), 562. <https://doi.org/10.3390/PR11020562>
- Altarabichi, M. G., Pashami, S., Nowaczyk, S., & Mashhadi, P. S. (2023). Fast Genetic Algorithm For Feature Selection - A Qualitative Approximation Approach. *GECCO 2023 Companion - Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion*, 11–12. <https://doi.org/10.1145/3583133.3595823>
- Alvarez Brotons, X. (2004). *Control predictivo de canales de riego utilizando modelos de predicción de tipo Muskingum (primer orden) y de tipo Hayami (segundo orden)*. <https://upcommons.upc.edu/handle/2099.1/3330>
- Alzanin, S. M., Gumaiei, A., Haque, M. A., & Muaad, A. Y. (2023). An Optimized Arabic Multilabel Text Classification Approach Using Genetic Algorithm and Ensemble Learning. *Applied Sciences* 2023, Vol. 13, Page 10264, 13(18), 10264. <https://doi.org/10.3390/APP131810264>
- Aziz, R. M., Mahto, R., Goel, K., Das, A., Kumar, P., & Saxena, A. (2023). Modified Genetic Algorithm with Deep Learning for Fraud Transactions of Ethereum Smart Contract. *Applied Sciences* 2023, Vol. 13, Page 697, 13(2), 697. <https://doi.org/10.3390/APP13020697>
- Bujgoi, G., & Sendrescu, D. (2022). DC Motor Control based on Integral Reinforcement Learning. *2022 23rd International Carpathian Control Conference, ICCO 2022*, 282–286. <https://doi.org/10.1109/ICCC54292.2022.9805935>
- Campos-Caba, R., & Winckler, P. (2023). Evaluación de caudal de sobrepaso en una defensa costera mediante un modelo basado en las ecuaciones de Navier-Stokes promediadas por Reynolds en el volumen (VARANS). *Obras y Proyectos*, 33, 6–14. <https://doi.org/10.21703/0718-281320233301>
- Chuk, D. (2012a). *Los sistemas de primer orden y los Controladores PID*. <http://dea.unsj.edu.ar/control2/ControladoresPID.pdf>
- Chuk, D. (2012b). *Los sistemas de primer orden y los Controladores PID*.
- Cosmin, B., & Sorin, T. (2023). Reinforcement Learning for a Continuous DC Motor Controller. *15th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2023 - Proceedings*. <https://doi.org/10.1109/ECAI58194.2023.10193912>
- Gen, M., & Lin, L. (2023). Genetic Algorithms and Their Applications. *Springer Handbooks*, 635–674. https://doi.org/10.1007/978-1-4471-7503-2_33/COVER
- Giannoccaro, N. I., & Sakamoto, T. (2022). Implementation and Validation of a Simple Direct Identification Method for an Experimental Multi-Span Web Transport System. *Systems* 2022, Vol. 10, Page 17, 10(1), 17. <https://doi.org/10.3390/SYSTEMS10010017>
- Greenstein, B. L., Elsey, D. C., & Hutchison, G. R. (2023). *Best Practices for Using Genetic Algorithms in Molecular Discovery*. <https://doi.org/10.26434/CHEMRXIV-2023-67HFC>
- Gyuk, P., Vassanyi, I., & Kosa, I. (2018). Blood glucose level prediction with improved parameter identification methods. *IEEE 30th Jubilee Neumann Colloquium, NC 2017, 2018-January*, 85–88. <https://doi.org/10.1109/NC.2017.8263257>
- Han, L., Fu, X., Tian, S., & Shen, Z. (2022). Demagnetization Fault Detection Research of Brushless DC Motor Based on Machine Learning. *2022 5th World Conference on Mechanical Engineering and Intelligent Manufacturing, WCMEIM 2022*, 634–638. <https://doi.org/10.1109/WCMEIM56910.2022.10021516>

- Jebari, K., & Madiafi, M. (2013). Selection Methods for Genetic Algorithms. *Int. J. Emerg. Sci*, 3(4), 333–344. <https://www.researchgate.net/publication/259461147>
- Khatri, K. C. A., Shah, K. B., Logeshwaran, J., & Shrestha, A. (2023). GENETIC ALGORITHM BASED TECHNO-ECONOMIC OPTIMIZATION OF AN ISOLATED HYBRID ENERGY SYSTEM. *ONLINE) ICTACT JOURNAL ON MICROELECTRONICS*, 4. <https://doi.org/10.21917/ijme.2023.0249>
- Kieszek, R., Kachel, S., & Kozakiewicz, A. (2023). Modification of Genetic Algorithm Based on Extinction Events and Migration. *Applied Sciences 2023*, Vol. 13, Page 5584, 13(9), 5584. <https://doi.org/10.3390/APP13095584>
- Li, H. G., & Liu, B. (2022). Knowledge Extraction: A Few-shot Relation Learning Approach. *Proceedings - 2022 International Conference on Machine Learning and Knowledge Engineering, MLKE 2022*, 261–265. <https://doi.org/10.1109/MLKE55170.2022.00057>
- Lian, C., Xiao, F., Gao, S., & Liu, J. (2019). Load Torque and Moment of Inertia Identification for Permanent Magnet Synchronous Motor Drives Based on Sliding Mode Observer. *IEEE Transactions on Power Electronics*, 34(6), 5675–5683. <https://doi.org/10.1109/TPEL.2018.2870078>
- Luengo, D., Martino, L., Bugallo, M., Elvira, V., & Särkkä, S. (2020). A survey of Monte Carlo methods for parameter estimation. *EURASIP Journal on Advances in Signal Processing 2020 2020:1*, 2020(1), 1–62. <https://doi.org/10.1186/S13634-020-00675-6>
- Ma, L., Shi, F., Wu, Z., & Peng, K. (2023). A Survey of Few-shot Learning-based Compound Fault Diagnosis Methods for Industrial Processes. *Proceedings - 2023 IEEE 6th International Conference on Industrial Cyber-Physical Systems, ICPS 2023*. <https://doi.org/10.1109/ICPS58381.2023.10128105>
- Márquez-Rubio, J. F., del-Muro-Cuéllar, B., Velasco-Villa, M., & Álvarez-Ramírez, J. (2010). Control basado en un esquema observador para sistemas de primer orden con retardo. *Revista Mexicana de Ingeniería Química*, 9(1), 43–52. http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-27382010000100006&lng=es&nrm=iso&tlng=es
- Molina, S. (n.d.). *La investigación de segundo orden en ciencias sociales y su potencial predictivo: el caso del proyecto de Identidad y tolerancia*.
- Nilanjan Dey (Ed.). (2023). *Applied Genetic Algorithm and Its Variants: Case Studies and New Developments* (Springer). Springer Nature.
- Obeidat, M. A., Wang, L. Y., & Lin, F. (2013). Real-time parameter estimation of PMDC motors using quantized sensors. *IEEE Transactions on Vehicular Technology*, 62(7), 2977–2986. <https://doi.org/10.1109/TVT.2013.2251431>
- Olivar-Castellanos, G. S., Vela-Valdés, L. G., Aguayo- Alquicira, J., Olivar-Castellanos, G. S., Vela-Valdés, L. G., & Aguayo- Alquicira, J. (2023). Control por ganancias programadas aplicado a un convertidor DC-DC para la regulación de la intensidad luminosa de una lámpara LED. *Ingenius. Revista de Ciencia y Tecnología*, 2023(29), 66–78. <https://doi.org/10.17163/INGS.N29.2023.06>
- Peñasco-Goel, S. R., Traslosheros-Michel, A., & Ramírez-Villa, G. (2023). Simulador de vuelo y comportamiento dinámico para modelos de aeronaves. *Pádi Boletín Científico de Ciencias Básicas e Ingenierías Del ICBI*, 11(Especial4), 189–203. <https://doi.org/10.29057/ICBI.V11IESPECIAL4.11392>
- Poovizhi, M., Senthil Kumaran, M., Ragul, P., Irene Priyadarshini, L., & Logambal, R. (2017). Investigation of mathematical modelling of brushless dc motor(BLDC) drives by using MATLAB-SIMULINK. *International Conference on Power and Embedded Drive Control, ICPEDC 2017*, 178–183. <https://doi.org/10.1109/ICPEDC.2017.8081083>
- Rahmatullah, R., Ak, A., & serteller, N. F. O. (2023). SMC Controller Design for DC Motor Speed Control Applications and Performance Comparison with FLC, PID and PI Controllers. *Lecture Notes in Networks and Systems*, 579, 607–617. https://doi.org/10.1007/978-981-19-7663-6_57/COVER
- Raol, J. R., Girija, G., & Singh, J. (2004). Modelling and Parameter Estimation of Dynamic Systems. *Modelling and Parameter Estimation of Dynamic Systems*. <https://doi.org/10.1049/PBCE065E>
- Rojas, J., Eichhorn, M., Baatar, G., Matz, S., & Glotzbach, T. (2018). Parameter identification and optimization of an oceanographic monitoring remotely operated vehicle. *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans, OCEANS - Kobe 2018*. <https://doi.org/10.1109/OCEANSKOB.2018.8559202>
- Rubaai, A., & Kotaru, R. (2000). Online identification and control of a dc motor using learning adaptation of neural networks. *IEEE Transactions on Industry Applications*, 36(3), 935–942. <https://doi.org/10.1109/28.845075>
- Semenov, A. S., Khubieva, V. M., & Kharitonov, Y. S. (2018). Mathematical modeling of static and dynamic modes DC motors in software package MATLAB. *2018 International Russian Automation Conference, RusAutoCon 2018*. <https://doi.org/10.1109/RUSAUTOCON.2018.8501666>
- Ishmeen, K., & Jaswinder, S. (2018). Position Control of DC Motor by using PID, FLC, ANN Controller Techniques and the Comparison of Performances. *International Journal of Engineering Research & Technology*, 4(15). <https://doi.org/10.17577/IJERTCONV4IS15018>
- Suman, S. K., & Giri, V. K. (2016). Speed control of DC motor using optimization techniques based PID Controller. *Proceedings of 2nd IEEE International Conference on Engineering and Technology, ICETECH 2016*, 581–587. <https://doi.org/10.1109/ICETECH.2016.7569318>
- Timofeev, V. S., Shchekoldin, V. Y., & Timofeeva, A. Y. (2018). Identification Methods for Linear Regression Based on Data of Household Budget Surveys. *2018 14th International Scientific-Technical Conference on Actual Problems of Electronic Instrument Engineering, APEIE 2018 - Proceedings*, 311–314. <https://doi.org/10.1109/APEIE.2018.8545726>
- Tourón, M., Navarro-Asencio, E., & Tourón, J. (2023). Validez de constructo de la Escala de Detección de alumnos con Altas Capacidades para Padres, (GRS 2), en España. *Revista de Educación*, 1(402), 53–80. <https://doi.org/10.4438/1988-592X-RE-2023-402-595>
- Zeng, D., Yan, T., Zeng, Z., Liu, H., & Guan, P. (2022). A Hyperparameter Adaptive Genetic Algorithm Based on DQN. <https://doi.org/10.1142/S0218126623500627>