Iranian Journal of Fuzzy Systems

Volume 22, Number 2, (2025), pp. 147-170

Original paper

University of Sistan and Baluchestan

An incremental learning-based fuzzy control scheme for a class of uncertain Euler-Lagrange systems

A. A. Haghrah [1], S. Ghaemi [2] and M. A. Badamchizadeh [3]

[1,2,3] *Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran*

haghrah@tabrizu.ac.ir, ghaemi@tabrizu.ac.ir, mbadamchi@tabrizu.ac.ir

Abstract

Euler-Lagrange systems describe a wide range of mechanical and robotic systems. Uncertainties and dynamic parameter changes pose significant challenges to control the Euler-Lagrange systems using traditional methods. In such cases, intelligent methods can cover the limits of the classical techniques. In this research, we intend to present a fuzzy logic-based controller trained by the proposed incremental learning algorithm to face the challenges of Euler-Lagrange systems. Incremental learning aims to accumulate experiences over time to train the models. We have performed several simulations to test the capabilities of the proposed method and compared the results with well-known machine learning-based methods using various criteria. Considering the integral of absolute error, the results show that the proposed method has improved by 40.89%, 38.32%, 34.12%, and 34.79% compared to the best other method in nominal system scenario and three other scenarios considering three different levels of uncertainty. The overshoot of the system response achieved by the proposed control scheme is approximately $44-48\%$ less than the best other method in four scenarios. Also, we have studied the system response to disturbance and noise.

Keywords: Fuzzy systems, artificial intelligence, incremental learning, Euler-Lagrange systems.

1 Introduction

Fuzzy logic is fundamentally based on the concept of computing with words. Despite facing numerous challenges, fuzzy logic has gained significant acceptance in both academic and industrial communities due to its ability to handle imprecise information and model complex systems using linguistic rules. One of its most significant features is the capacity to approximate any mathematical function through a limited number of fuzzy if-then rules, which allows the transformation of expert qualitative insights into quantitative models. This capability not only enhances the interpretability of the resulting mathematical models by expressing them in verbal terms but also supports a gradual improvement in accuracy with the addition of more rules. Moreover, expert knowledge can be systematically incorporated into these models and further refined using field data, establishing fuzzy models as a unique machine learning technique despite their relatively higher computational complexity compared to other methods [42]. It is noteworthy that fuzzy logic has been successfully integrated into various machine learning paradigms, including supervised, unsupervised, semi-supervised, and reinforcement learning, to extend its applicability and performance [19].

It is worth mentioning that approximation theory plays a significant role in the development and analysis of intelligent systems, particularly in quantifying errors and convergence properties of computational methods. For instance, advancements in the study of q-Baskakov-Durrmeyer type operators have introduced novel inequalities for error estimation and numerical analysis in multidimensional domains [25]. Similarly, tensor product quantum beta-type operators have expanded the scope of error approximations using generalized Boolean sum operators and mixed modulus of continuity [27]. Additionally, the development of Kantorovich-type rational operators has demonstrated improved approximation rates, offering practical tools for systems requiring precise convergence properties [26]. Finally, fuzzy

Corresponding Author: S. Ghaemi

Received: May 2024; Revised: March 2025; Accepted: April 2025.

https://doi.org/10.22111/ijfs.2025.48879.8616

Meyer-König and Zeller operators have contributed to the field by exploring fuzzy Korovkin-type approximation results and related inequalities [28]. While the direct applications of these operators differ from the control strategies presented in this study, the underlying principles of error estimation and convergence analysis remain conceptually relevant to the evaluation and refinement of intelligent controllers, such as the fuzzy logic-based approach proposed in this paper.

Fuzzy logic-based methods have been applied across an extensive range of fields [13, 21, 30, 31]. For example, in the realm of medical diagnosis, integrating fuzzy logic with deep learning has been shown to address the uncertainties present in neuroimaging data and expert annotations. Tanveer et al. [37] provide a comprehensive study on fuzzy deep learning (FDL) for improving both the accuracy and interpretability of automated Alzheimer's diagnosis. Their work details how fuzzy logic enhances image preprocessing, segmentation, and classification processes and effectively manages uncertainty by fusing multimodal data including genomics, proteomics, and metabolomics within a fuzzy framework.

In computer vision, the adaptability of fuzzy logic is evident in its application to face mask recognition. Balovsyak et al. [3] combine the classical Viola-Jones detection method with fuzzy logic, using Haar cascades to detect facial features followed by a Mamdani fuzzy inference system to ascertain mask usage. This approach, which evaluates multiple detection outcomes via fuzzy logic, demonstrates significant improvements in recognition accuracy, especially under variable conditions where detection confidence may fluctuate.

Industrial automation has similarly benefited from fuzzy logic. Santhosh et al. [32] integrated fuzzy logic with machine learning and the Taguchi method to optimize the painting process of an IRB1410 industrial robot. Their detailed analysis involved comparing various painting trajectories, linear, circular, and zigzag, by evaluating surface smoothness and paint quality using Scanning Electron Microscope (SEM) images. The study employed Taguchi's L9 orthogonal design and ANOVA to isolate key factors affecting surface roughness, and machine learning algorithms were used to automate flaw detection with an accuracy of 94%. Fuzzy logic in this context was instrumental in refining the decision-making process and adapting to the variable demands of the industrial process.

In risk and uncertainty management, fuzzy logic has been pivotal. Siami et al. [34] introduced an Autonomous Fuzzy Decision Support System (AFDSS) that combines fuzzy logic with artificial intelligence and unsupervised learning to evaluate risks in big data environments. Their system uses fuzzy clustering to identify patterns from large, unlabeled datasets and then applies a fuzzy inference system to assess the risk associated with these clusters. This detailed approach demonstrates how fuzzy logic can extract and evaluate critical risk indicators in complex, data-driven scenarios.

Wind power forecasting is another field where fuzzy logic has shown promise. Shi et al. [33] developed a fuzzy time series model tailored for wind power prediction, which minimizes the need for extensive data preprocessing and offers clear interpretability for each prediction step. The model incorporates non-stationary sets to better capture long-term variations and introduces an improved artificial hummingbird algorithm (SLG-AHA) to enhance both prediction accuracy and stability. Their work provides detailed insights into how fuzzy logic can be adapted for renewable energy forecasting, balancing the trade-off between model complexity and interpretability.

Traditional learning paradigms, including deep learning, are not without their drawbacks. Notable challenges include time-intensive training processes and the phenomenon of catastrophic forgetting, where previously learned information is lost as new data is acquired [1]. Incremental learning has emerged as a promising strategy to overcome these issues by enabling models to gradually enhance their knowledge over successive iterations. This approach involves explicitly storing acquired experiences within a limited capacity and emphasizes transforming raw observations into actionable experiences. However, incremental learning also faces significant challenges: changing data distributions over time, balancing the integration of new and old knowledge, dynamically adjusting model complexity, and maintaining a compact yet informative memory of past data. While incremental learning has been thoroughly explored in image processing applications [49], its application to dynamic control problems remains relatively underdeveloped.

Within the scope of fuzzy models, early work on incremental learning dates back to the development of Takagi-Sugeno fuzzy systems, where researchers laid the foundation for data-driven online learning approaches. For instance, Lughofer et al. [17] proposed a data-driven online learning method that adapts both the linear parameters in rule consequents and the premise parameters in membership functions through a modified vector quantization technique. This approach enabled automatic rule evolution and cluster generation based on incoming data, with evaluations performed on high-dimensional datasets from engine test benches and rolling mills, as well as on nonlinear system identification tasks.

Building on this foundation, Yang et al. [45] introduced the Statistical Evolving Fuzzy Inference System (SEFIS), which is designed to update system parameters continuously and adjust its structure in the presence of non-Gaussian noise. SEFIS employs an adaptive maximum correntropy extended Kalman filter (AMCEKF) to improve parameter estimation by incorporating statistical uncertainty measures, alongside dynamic rule evolution to add or remove rules as needed. This detailed methodology illustrates how structural adaptation, combined with statistical modeling, can yield faster learning and improved accuracy compared to traditional evolving fuzzy systems.

Another advancement include PANFIS (Parsimonious Network based on Fuzzy Inference System) [29] and evolving

Takagi-Sugeno-Kang (TSK) models for time series forecasting [2]. PANFIS, for example, starts with an empty rule set and incrementally constructs and refines fuzzy rules based on the degree of nonlinearity in the system, merging similar fuzzy sets to improve interpretability while keeping the model compact. The evolving TSK model introduced by Alves et al. [2] adapts its structure by dynamically adjusting its rule base while maintaining a pre-defined maximum number of rules. These detailed works illustrate the potential of incremental learning to maintain a balance between accuracy and interpretability, even in nonstationary environments.

Fuzzy logic has also been widely applied in fuzzy control of Euler-Lagrange systems, which serve as models for many mechanical systems and robotic arms [20]. The dynamics of these systems are typically modeled using the Euler-Lagrange method, providing a robust framework for controller design [40]. However, addressing uncertainties within these systems, particularly when precise modeling is unattainable, remains a significant challenge. Recent studies have tackled various aspects of this challenge: Zhang et al. [47] proposed a finite-time event-triggered containment control strategy that utilizes a settling-time performance function and adaptive neural networks with saturated terms to manage unknown control coefficients; Yang et al. [44] developed a distributed optimal consensus control strategy using filters and observers to ensure bounded control signals and global consensus; and Xiao et al. [41] introduced a fixed-time formation-containment control strategy that employs distributed estimators and adaptive integral sliding mode disturbance observers.

Additional studies have further refined control methodologies for Euler-Lagrange systems. He et al. [9] proposed a dynamic output feedback control strategy that requires only position and velocity tracking errors, thereby eliminating the need for full state feedback or explicit system parameter knowledge. Sun et al. [36] employed nonsingular fast terminal sliding mode control in conjunction with control barrier functions to ensure both finite-time convergence and constraint satisfaction. Leal et al. [14] developed a fault-tolerant control scheme using a generalized proportional integral observer, which efficiently estimates disturbances and faults using only the inertia matrix. Liu et al. [16] addressed the nonlinear challenges in Euler-Lagrange systems through an inverse optimal formation control method that ensures semi-global uniform ultimate boundedness while optimizing performance via a cost function. Su et al. [35] presented a distributed formation optimal tracking protocol that allows each agent to track an optimal reference trajectory without exchanging virtual states, and Ober et al. [24] integrated variational principles with backward error analysis for system identification from discrete position data.

Machine learning-based approaches have also been applied to address uncertainties in Euler-Lagrange systems. Chen et al. [4] combined reinforcement learning with an echo state network to provide fault-tolerant adaptive tracking control, while Yao et al. [46] developed a reinforcement learning-based direct signal tracking control strategy that integrates data-driven learning with backstepping control for enhanced adaptation and stability. Nguyen et al. [23] proposed an event-triggered robust optimal control method for mobile Euler-Lagrange systems affected by actuator saturation and dead zones, ensuring closed-loop stability and avoiding Zeno behavior. Gao et al. [6] addressed singularity issues by dynamically adjusting coefficients in the potential energy function through a fuzzy system, and He et al. [10] utilized Gaussian processes to construct a nonparametric uncertainty model that supports adaptive robust control with low gains.

Despite these significant advances, the application of incremental learning specifically to fuzzy controllers for Euler-Lagrange systems remains relatively underexplored. Preliminary studies in this area include a random forest-based incremental learning algorithm for robot control by Gijsberts et al. [7] and the Infinite Mixture of Linear Experts (IMLE) algorithm for multi-valued function approximation in sensor mapping by Jamone et al. [12]. Yang et al. [43] further demonstrated the benefit of incorporating an incremental learning mechanism into neural-network estimators for enhancing the control performance of biped robots. However, a comprehensive framework that integrates incremental learning with fuzzy logic for dynamic control tasks in Euler-Lagrange systems is still needed.

In this paper, we propose an innovative incremental learning algorithm tailored specifically for fuzzy models applied to control tasks in Euler-Lagrange systems. Our approach introduces an adaptive, experience-driven fuzzy model that continuously evolves its rule base in response to real-time performance measures such as error, error derivative, and error integral. The proposed algorithm periodically transforms observations into stored experiences within a limited-size queue, which are then used to update rule outputs dynamically. This methodology allows the fuzzy model to refine its control strategy over time without the need for complete retraining. By combining the interpretability of fuzzy systems with the adaptability of incremental learning, our work provides a detailed and novel solution to the challenges of controlling nonlinear and uncertain Euler-Lagrange systems.

In addition to its theoretical contributions, the proposed incremental learning-based fuzzy model has significant practical implications. Real-world dynamic systems-such as autonomous robotics, industrial automation, and renewable energy management-demand controllers that can adapt in real time to uncertainties and evolving environmental conditions. By continuously updating its rule base without the need for complete retraining, our approach offers a robust solution to the challenges of sensor noise, changing data distributions, and unforeseen disturbances.

The remainder of this paper is organized as follows. In Section 2, we examine the dynamic model of Euler-Lagrange systems. Section 3 introduces the proposed incremental learning-based control scheme. Section 4 presents simulation results and their analysis, and Section 5 concludes the paper.

2 System dynamics

The Euler-Lagrange model serves as a cornerstone for deriving dynamic equations governing mechanical systems, offering insights applicable across various domains. Typically, a system with n degrees of freedom (DoF) under the Euler-Lagrange framework is described by the following equations:

$$M(q)\,\ddot{q} + H(q,\ \dot{q})\,\dot{q} + d(q) = \tau, \tag{1}$$

in which, $q \in \mathbb{R}^n$ is the vector of system variables, $M(q) \in \mathbb{R}^{n\times n}$ is the symmetric and positive definite inertia matrix, $H(q,\ \dot{q}) \in \mathbb{R}^{n\times n}$ denotes the centrifugal Coriolis turques, $d(q) \in \mathbb{R}^n$ represents the vector of gravitational torque, $\tau \in \mathbb{R}^n$ demonstrates the vector of control input.

In practice, the nominal model of Euler-Lagrange systems (1) is typically established via standard test procedures and measurements. However, the exact model of Euler-Lagrange systems (1) remains elusive due to factors such as wear, friction, and modeling simplifications. A viable approach to address these challenges is to decompose the model (1) into a nominal component and an uncertain component [40].

$$M(q) = M_o(q) + \Delta M(q), \tag{2}$$
$$H(q,\ \dot{q}) = H_o(q,\ \dot{q}) + \Delta H(q,\ \dot{q}), \tag{3}$$
$$d(q) = d_o(q) + \Delta d(q). \tag{4}$$

In which, M_o indicates nominal inertia matrix, H_o denotes the nominal centrifugal Coriolis turques, and d_o represents the nominal vector of gravitational torque. Note that the terms containing the Δ prefix demonstrate the uncertain components of the system dynamics.

Now, by assuming the state variables as $x_1 = q$, $x_2 = \dot{q}$, and $\mathbf{x} = \left[q^T,\ \dot{q}^T,\ \ddot{q}^T\right]^T$, let's put the system equations in the standard state-space form. Using equation (1), $\dot{x}_2$ would be as following:

$$\begin{aligned}\dot{x}_2 = {} & M_o^{-1}(x_1)\,\tau \\ & - M_o^{-1}(x_1)\left[H_o(x_1,\ x_2)\,x_2 + d_o(x_1)\right] \\ & - M_o^{-1}(x_1)\left[\Delta M(x_1)\,\dot{x}_2 + \Delta H(x_1,\ x_2)\,x_2 + \Delta d(x_1)\right].\end{aligned} \tag{5}$$

So by assuming $g(\mathbf{x})$ and $f(\mathbf{x})$ as:

$$\begin{aligned} g(\mathbf{x}) &= \left[g_1(\mathbf{x}),\ \cdots,\ g_n(\mathbf{x})\right]^T \\ &= -M_o^{-1}(x_1)\left(H_o(x_1,\ x_2)\,x_2 + d_o(x_1)\right), \end{aligned} \tag{6}$$

$$\begin{aligned} f(\mathbf{x}) &= \left[f_1(\mathbf{x}),\ \cdots,\ f_n(\mathbf{x})\right]^T \\ &= -M_o^{-1}(x_1)\left(\Delta M(x_1)\,\dot{x}_2 + \Delta H(x_1, x_2)\,x_2 + \Delta d(x_1)\right), \end{aligned} \tag{7}$$

we will have:

$$\begin{aligned}\dot{x}_1 &= x_2, \\ \dot{x}_2 &= M_o^{-1}(x_1)\,\tau + g(\mathbf{x}) + f(\mathbf{x}).\end{aligned} \tag{8}$$

According to their respective definitions, $g(\mathbf{x})$ represents a known dynamic function, whereas $f(\mathbf{x})$ embodies an unknown entity, encompassing unstructured dynamics and external disturbances. Referring to [40], it is assumed that the inertia matrix M is symmetric and positive definite. As can be seen, we have also considered the additive uncertainties in addition to multiplicative uncertainties, which makes the control problem harder.

3 Incremental learning

Classical machine learning paradigms operate under the assumption that all training data is available at once, allowing the algorithm to learn a single, comprehensive model. The main problem of such paradigms is catastrophic forgetting.

Catastrophic forgetting, also known as catastrophic interference, refers to the phenomenon that an artificial neural network forgets previously learned information when it is trained on new information [18]. Such "forgetting" can be significant and abrupt, leading to a drastic drop in performance on previously learned tasks. Continual learning, in contrast, presents a scenario where data arrives sequentially, often with changes in the underlying data distribution over time. This scheme demands adaptation from the learning algorithm, requiring integrating new information continuously while preserving previously acquired knowledge. Thus, one of the principal and open problems in the machine learning context is enabling the models to incrementally learn from evolving data [5].

We can categorize the incremental learning into three scenarios: Task–Incremental Learning (TIL), Domain–Incremental Learning (DIL), and Class–Incremental Learning (CIL) [38]. TIL addresses problems where the model sequentially learns to solve distinct tasks. DIL focuses on training a model to solve a single problem in different contexts. Finally, CIL aims to discriminate between incrementally observed classes. Considering these definitions, we can deduce that the control problem of dynamic systems is a CIL because we deal with a fixed (albeit uncertain) system, so any changes and learning should occur based on the different initial conditions of the system. In the following sections, we will introduce the proposed fuzzy incremental learning control scheme and explore some theoretical aspects of its application to a class of uncertain Euler-Lagrange systems.

3.1 Preliminaries

In this subsection, we introduce the preliminaries of the proposed control scheme: Formulation of the TSK fuzzy systems, formulation of the Mamdani fuzzy systems, and Human Evolutionary Optimization Algorithm.

3.1.1 TSK fuzzy systems

The rule base of a TSK system is defined as a number of IF-THEN rules as follows:

$$\begin{aligned} R^l : IF\ \chi_1\ is\ F_{1,l}\ AND\ \cdots\ AND\ \chi_p\ is\ F_{p,l}, \\ THEN\ \nu_1 = \nu^*_{1,l}\ AND\ \cdots\ AND\ \nu_q = \nu^*_{q,l}. \end{aligned} \tag{9}$$

In which χ_i represents the i^{th} input of the system, $F_{i,l}$ is the fuzzy set describing the i^{th} input in the l^{th} rule, ν_j is the j^{th} output of the system, the value of the j^{th} output in the l^{th} rule is $\nu^*_{j,l}$. Finally, p and q are the number of inputs and outputs of the fuzzy TSK subsystem, respectively. Considering the product operator as the t-norm of the fuzzy system, the j^{th} output of a TSK system consisting of L rules can be calculated as follows:

$$\nu_j = \frac{\sum_{l=1}^{L} \nu^*_{j,l} \prod_{i=1}^{p} \mu_{F_{i,l}}(\chi_i)}{\sum_{l=1}^{L} \prod_{i=1}^{p} \mu_{F_{i,l}}(\chi_i)}, \quad j = 1,\ \cdots,\ q. \tag{10}$$

3.1.2 Mamdani fuzzy systems

The rule base of a Mamdani fuzzy system can be illustrated as the following set of rules [22]:

$$R^l : IF\ \chi_1\ is\ G_{1,l}\ AND\ \cdots\ AND\ \chi_p\ is\ G_{p,l}\ THEN\ \nu\ is\ B_l. \tag{11}$$

In which χ_i represents the i^{th} input of the system, $G_{i,l}$ is the fuzzy set describing the i^{th} input in the l^{th} rule, ν is the output of the system, B_l is the fuzzy set describing the output of the l^{th} rule, and p is the number of system inputs. Considering the product as the t-norm, maximum as the s-norm, and singleton fuzzification, the fuzzy output of a Mamdani system consisting of L rules can be calculated as follows:

$$\mu_{B'}(\nu) = \max_{l=1,\ \cdots,\ L} \left[\prod_{i=1}^{p} \mu_{G_{i,l}}(\chi_i)\, \mu_{B_l}(\nu) \right]. \tag{12}$$

These choices are commonly used in fuzzy logic systems due to their mathematical simplicity and intuitive interpretation. The product t-norm is selected as it provides a smooth and multiplicative combination of fuzzy membership values, which helps in achieving a finer degree of intersection between fuzzy sets. Also, the maximum s-norm is employed because it effectively represents the union of fuzzy sets by selecting the highest degree of membership. These norms are standard practices in fuzzy systems and are particularly suitable for the proposed method's requirements.

Now, to achieve the crisp output of the system, we must defuzzify the output set, $B^{'}$. In this paper, the center of gravity method has been chosen for the defuzzification of the final output fuzzy set, which can be calculated using the formula below:

$$\nu^* = \frac{\int_V \nu \ \mu_{B'}(\nu)\, d\nu}{\int_V \mu_{B'}(\nu)\, d\nu}. \tag{13}$$

In which ν^* is the crisp output of the system, and V is the universe of discourse of the output.

3.1.3 Human evolutionary optimization algorithm

The Human Evolutionary Optimization (HEO) Algorithm is a recent metaheuristic algorithm inspired by evolution in human societies. In this algorithm, the initial population experiences two stages. The first stage is human exploration, and the second is human development. In the human development stage, we divide the population into Leaders, Explorers, and Losers, and each sub-population evolves with a different rule. The reported results for this algorithm suggest a significant capability of being used in various scientific and engineering applications. The pseudo-code below describes the HEO algorithm [15]:

Logistic chaotic mapping initialization
Compute initial fitness values
for $i = 1$ *to* M **do** ▷ Starting the search
  $bestFitness \leftarrow fitness(1)$
  $worstFitness \leftarrow fitness(N)$
  $r \leftarrow random()$
  for $j = 1$ *to* N **do**
    if $i <= M/4$ **then** ▷ Human exploration stage
      $X_j^* \leftarrow X_{gb} \times \left(1 - \frac{i}{M}\right) + (mean\,(X_j) - X_{gb}) \times \lfloor \frac{random()}{F_{jump}} \rfloor \times F_{jump} + b \times Levy\,(P)$
    else ▷ Human development stage
      for $j = 1$ *to* N_L **do** ▷ Leaders update
        if $r < A$ **then**
          $X_j^* = \omega X_j \times e^{\frac{-i \times normal()}{M \times random()}}$
        else
          $X_j^* = \omega X_j + normal()$
        end if
      end for
      for $j = N_L$ *to* $N_L + N_E$ **do** ▷ Explorers update
        $X_j^* \leftarrow normal() \times e^{\frac{X_N - X_j}{j^2}}$
      end for
      for $j = N_L + N_E$ *to* $N_L + N_E + N_F$ **do** ▷ Followers update
        $X_j^* \leftarrow X_j + \omega \times normal() \times (X_1 - X_j)$
      end for
      for $j = N_L + N_E + N_F$ *to* N **do** ▷ Losers update
        $X_j^* \leftarrow X_{gb} + (X_{gb} - X_j) \times normal()$
      end for
    end if
    Update and boundary control
  end for
  Store best solution
end for

The main parameters of this algorithm are population size, number of leaders, number of explorers, number of losers, A, and number of iterations. In this paper, we use the values 50, 20, 20, 10, 0.6, and 100, respectively. Notice that *Levy* is a random number generator with Levy probability distribution, and the input of the *Levy* function is the size of the randomly generated vector. Also, *random* and *normal* functions are random generators with uniform and normal distributions, respectively. More details about the HEO algorithms are available in the original study [15].

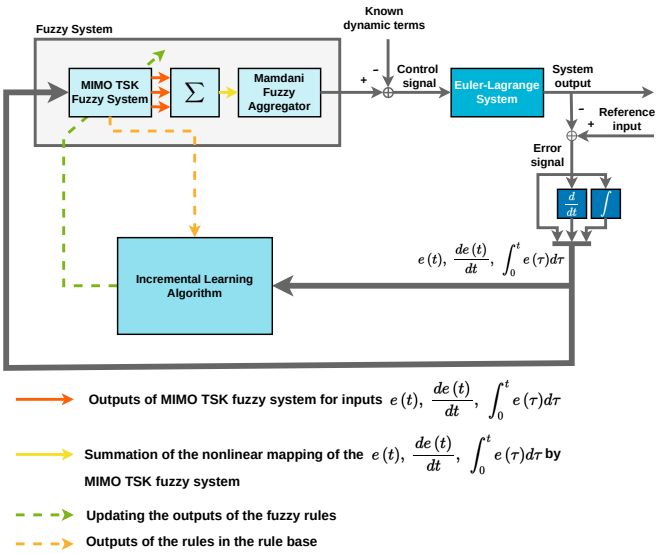


Figure 1: Block diagram of the proposed incremental fuzzy learning scheme, illustrating its structural components and learning process.

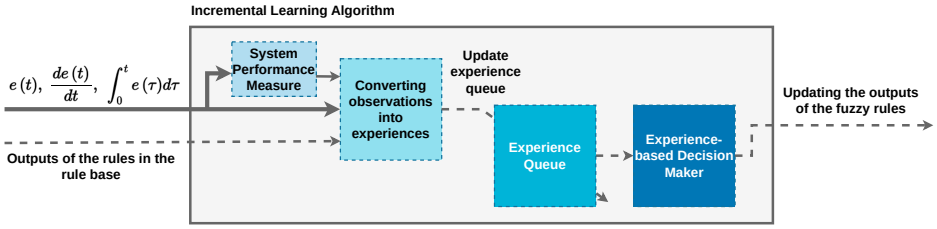


Figure 2: Block diagram of the proposed incremental algorithm, outlining its key components and operational flow.

3.2 Proposed fuzzy incremental learning control scheme

Figure 1 represents an overall view of the proposed fuzzy model and the incremental learning scheme corresponding with it. Also, Figure 2 demonstrates the structure of the proposed incremental learning algorithm for converting observations to experiences. In these block-diagrams, the dashed line shows procedures that occur at discrete time instances. The fuzzy model consists of two fuzzy subsystems: a Multi-Input Multi-Output (MIMO) Takagi-Sugeno-Kang (TSK) fuzzy system and a Single-Input Single-Output (SISO) Mamdani fuzzy system. The inputs of the MIMO TSK fuzzy subsystem are the components of the error vector, and its outputs are applied to the aggregator SISO Mamdani subsystem after passing through the summation block. The main use of the Mamdani subsystem is to ensure the control signal lies within a specific bound.

Our selection of the Takagi-Sugeno-Kang (TSK) fuzzy model is primarily driven by its ability to capture nonlinear dynamics in a mathematically concise and interpretable manner. The TSK framework is particularly well-suited for dynamic control tasks in Euler-Lagrange systems, where a precise quantitative approximation of the system behavior is essential. In contrast, the Mamdani fuzzy subsystem was chosen for its strength in handling qualitative, linguistic rule-based reasoning, which facilitates robust uncertainty modeling in scenarios where crisp boundaries are difficult to define. Together, these subsystems provide a complementary balance-leveraging the quantitative strengths of TSK and the qualitative robustness of Mamdani-to address the complex challenges of our control problem.

In the proposed scheme, we adjust the outputs of the fuzzy rules in the rule base of the TSK fuzzy subsystem using a specific algorithm, both periodically and event-based. We assume the adjustment period of the TSK fuzzy subsystem as T_u. Also, the event that triggers an adjustment process is a change in the rule with the highest firing strength in the rule base. Before starting to describe the rule-adjustment process, we should define some terms. An *experience* is a set of outputs of the TSK fuzzy rules. A *successful experience* is an experience in a T_u period in which the cost function is reduced. The *cost function* is a function of input variables and time, which has an inverse relation with the

performance of the system. That is, reducing the cost improves the performance. The *active rule* is the rule with the highest firing strength. We will introduce the exact relation of the cost function in the following. Each rule-adjustment process has three steps:

1. Updating the experiences list by storing the outputs of the rules in the rule base and the change of cost function till the last update process.
2. Determining the output of the active rule based on previous successful experiences by performing a weighted averaging.
3. Updating the output of the active rule based on the 2nd step and the HEO-based algorithm.

The mentioned cost function can be a time-invariant function as the norm, or a time-variant function, which can force a specific convergence rate in the training process. Let's assume the experience FIFO queue as follows:

$$\mathcal{E} = [\cdots,\ \mathcal{E}_i,\ \cdots]\,. \tag{14}$$

In which, we define the i^{th} experience as:

$$\mathcal{E}_i = (\mathcal{Y}_i,\ \mathcal{C}_i)\,. \tag{15}$$

In this definition,

$$\mathcal{Y}_i = \{\mathcal{Y}_{i,l,j},\ l = 1, \cdots, L_{TSK},\ j = 1, \cdots, q\}\,, \tag{16}$$

is the set of outputs, L_{TSK} is the number of rules in the fuzzy TSK subsystem rule base, and $\mathcal{C}_i$ is the cost function, which we evaluate it as below:

$$\mathcal{C}_i = sat\left(\gamma\left(t_{i-1}\left\|e_{i-1}\right\| - t_i\left\|e_i\right\|\right), \sigma\right). \tag{17}$$

In this formula, γ is a constant. *sat* is the saturation function defined as:

$$sat(\beta,\ \sigma) = \begin{cases} \sigma & \beta > \sigma, \\ -\sigma & \beta < -\sigma, \\ \beta & Otherwise. \end{cases}$$

with the input β and the saturation level σ. Furthermore, t_i is the storing time of the i^{th} experience. In Figure 2, the "Converting observations into experiences" block is responsible for the first step described above. After adding the experience to the list, the experience-based decision maker summates the achieved experiences in the experience list and proposes a set of output values for the rules of the TSK fuzzy subsystem rule base. The updated outputs are achieved using a weighted average as below:

$$\bar{\mathcal{Y}}_{l,j} = \frac{\sum_{i \in \mathcal{I}_{\mathcal{E}_s}} \mathcal{C}_i \mathcal{Y}_{i,l,j}}{\sum_{i \in \mathcal{I}_{\mathcal{E}_s}} \mathcal{C}_i}. \tag{18}$$

In which, $\bar{\mathcal{Y}}$ is the set of outputs of the TSK fuzzy subsystem achieved by previous experiences, and $\mathcal{E}_s$ is the set of successful experiences. If $\mathcal{C}_i$ is positive, the experience is considered successful, and if negative or zero, it is considered unsuccessful. $\mathcal{I}_{\mathcal{E}_s}$ represents the indices set of successful experiences. The mentioned procedure, described as the second step of the updating rules process, is done in the "Experience-based Decision Maker" block.

Note that the system error is large at the begining of the simulation and gradually converges over time. As mentioned, based on the equation (18), the updated output is obtained using an averaging procedure with $\mathcal{C}_i$s as weights. Having large weights for experiences occurring while the error is high is inevitable, and this affects the contribution of more successful experiences obtained when the error is lower. Therefore, using only the error difference as the cost function does not yield proper results. To mitigate this issue, we introduced a time coefficient to the error value. Further more, the coefficients γ and σ are used for a similar purpose: to prevent the weights from growing beyond a specific value, thereby avoiding premature convergence of the rule outputs.

After introducing Equation (18), we now delve deeper into periodic and event-based rule updates. These two types of rule adjustments are designed to prevent experiences from leaving unwanted impact when applied in Equation (18). We will illustrate the significance of these rule update strategies using two examples.

1. First, consider a scenario where the rule with the highest firing strength has not been updated for a considerable period. Based on the defined fuzzy sets, this indicates that the system states are either changing at a very low rate or have converged to a specific value. Regardless of which case applies, this results in lost learning time without acquiring new experience. If this process continues, it delays the acquisition of new experience, slows the convergence of fuzzy rule outputs, and prolongs the training process. To address this issue, periodic rule updates are introduced at fixed intervals.

2. Next, suppose the rule outputs are updated periodically, but due to improper initialization or other issues, the system error fails to converge. In this case, waiting until the next scheduled update can cause errors to rise unexpectedly. If the system starts from a high error value at an advanced time step and then experiences a successful learning instance, this would assign an undesirably large weight in Equation (18) to an experience that should not have such a high $\mathcal{C}_i$. This can also lead to delayed convergence of the rule outputs. To prevent this issue, event-based rule updates are introduced. By triggering updates when necessary, system errors are prevented from diverging unexpectedly.

The rationale behind these two types of rule updates has been described above. However, further research can be conducted to improve the convergence speed of this learning method.

The last step for updating the rule base of the TSK fuzzy subsystem is to deviate the outputs of the rules from the values achieved by the experiences using the HEO-based method, which ensures the possibility of generating better $\bar{\mathcal{Y}}_{l,j}$ over the training process. In other words, pure use of the results achieved by the experiences yields an unmatured convergence, ensembling a situation in which we are not looking for new and different conditions and corresponding new experiences. Utilizing the HEO-based method, we can experience new conditions and improve the system using successful experiences. The final output of the active rule of the TSK fuzzy subsystem is calculated as follows:

$$\hat{\mathcal{Y}}_{l,j} = \left(1 - \frac{k}{\Gamma}\right)\left(\bar{\mathcal{Y}}_{l,j} - \hat{\mathcal{Y}}^b_{l,j}\right) Levy\,() + \left(1 - \frac{k}{\Gamma}\right)\hat{\mathcal{Y}}^b_{l,j} + \left(\hat{\mathcal{Y}}^m_{l,j} - \hat{\mathcal{Y}}^b_{l,j}\right)\left\lfloor \frac{random()}{F_{jump}} \right\rfloor F_{jump}. \tag{19}$$

In which, k is the rule-adjustment process index, Γ is the maximum number of rule-adjustments, $\hat{\mathcal{Y}}^b_{l,j}$ indicates the best achieved $\hat{\mathcal{Y}}_{l,j}$, and $\hat{\mathcal{Y}}^m_{l,j}$ shows the mean value of achieved $\hat{\mathcal{Y}}_{l,j}$s. $Levy\,()$ and $random()$ indicate randomly generated numbers considering Levy and uniform probability distributions. F_{jump} is an algorithm parameter defined as one-thousandth of the length of the definition interval of the $\hat{\mathcal{Y}}_{l,j}$.

For further clarification, we summarize the "Converting Observations into Experiences" block as Algorithm (1). Additionally, a more structured, step-by-step explanation of the "Experience-Based Decision Maker" block is provided in Algorithm (2). The final outputs of the fuzzy TSK subsystem used in the training process are denoted as $\hat{\mathcal{Y}}_{l,j}$, and the output of the fuzzy TSK subsystem is computed using Equation (23).

Algorithm 1 Convert Observations into Experiences

1: **Let** $\mathcal{C}_i \leftarrow$ *Cost function using equation* (17);
2: **Let** $\mathcal{Y}_i \leftarrow$ *Outputs of the fuzzy TSK subsystem rules*;
3: **PUSH** $\mathcal{E}_i = (\mathcal{Y}_i,\ \mathcal{C}_i)$ *into experience FIFO buffer with fixed size* $(\mathcal{E})$;
4: *Return* $\mathcal{E}$;

Algorithm 2 Experience-based Decision Maker

1: **Let** $\bar{\mathcal{Y}_{l,j}} \leftarrow$ *Weighted average of the* $\mathcal{Y}_{i,l,j}$ *using equation* (18);
2: **Let** $\hat{\mathcal{Y}}_{l,j} \leftarrow HEO-$ *based update of the outputs for new experiences using equation* (19);
3: *Return* $\hat{\mathcal{Y}}_{l,j}$

As can be seen in Figure 1, the input of the TSK fuzzy subsystem is the error vector defined below:

$$\mathbf{e}_{\theta_i} = \left[e_{\theta_i}, \dot{e}_{\theta_i}, \int^t e_{\theta_i}(\tau)\, d\tau\right], \tag{20}$$

And the input of the Mamdani fuzzy subsystem is the summation of the outputs of the TSK fuzzy subsystem. We assume three fuzzy sets, Negative, Zero, and Positive, for describing each input variable of the TSK fuzzy subsystem. Figure 3 represents the input sets that are used similarly for all inputs of the TSK fuzzy subsystem. All possible rules using these sets for all inputs are used for building up the rule base of the system. The Mamdani fuzzy subsystem maps the result to a control signal inside the operation range of the actuator.

Considering the structure of the fuzzy controller and the mathematical formulation of its components, the final

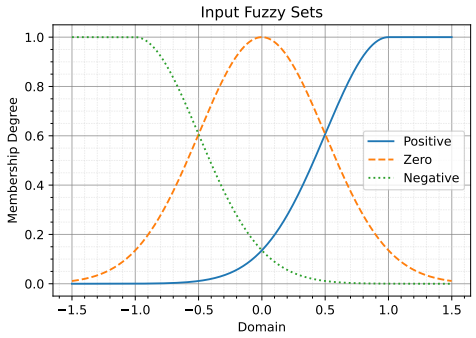


Figure 3: Fuzzy sets representing the input variables of the TSK fuzzy subsystem, illustrating their linguistic categories and membership functions.

input-output relation of the controller model can be summarized as:

$$y^* = \frac{\int_V y\ \mu_{B'}(y)\, dy}{\int_V \mu_{B'}(y)\, dy}, \tag{21}$$

$$\mu_{B'}(y) = \max_{l=1,\cdots,L_{Mamdani}} \left[\mu_{G_l} \left(\sum_{j=1}^{q} y_j \right) \mu_{B_l}(y) \right], \tag{22}$$

$$y_j = \frac{\sum_{l=1}^{L_{TSK}} \hat{\mathcal{Y}}_{l,j} \prod_{i=1}^{p} \mu_{F_{i,l}}(e_i)}{\sum_{l=1}^{L_{TSK}} \prod_{i=1}^{p} \mu_{F_{i,l}}(e_i)},\ j = 1, \cdots, q. \tag{23}$$

In which, e_i is the i^{th} component of the error vector, and $\hat{\mathcal{Y}}_{l,j}$ is the j^{th} output of the l^{th} rule during the learning process. The membership functions used in this formula have been defined above. As we know, fuzzy systems are universal approximators, and the defined fuzzy controller also has this feature, meaning that, considering the inputs of the TSK subsystem as error, error derivative, and error integral, the control scheme provided in this article is a general nonlinear PID controller. From this point of view, we can initialize the controller parameters in such a way that the system is stable. Since the successful experiences are only those in which the system performance is enhanced, considering the definition of cost function, the system will maintain stability. Improving system performance, in this case, is guaranteed by successful experiences, achieved by equation (19).

Note that, in our proposed fuzzy system, we continuously monitor performance measures throughout the simulation. When an experiment is deemed unsuccessful, indicated by a decrease in the performance measure, we update the output of the fuzzy rule with the highest firing strength. This targeted adjustment is designed to correct the observed deficiency. If the updated rule subsequently results in improved performance, the experiment is classified as successful, and the updated output is stored for use in future intervals. This adaptive, experience-driven updating process directly contributes to the overall performance gains observed in our simulations, as it enables the system to learn from its shortcomings and reinforce effective control strategies.

Regarding parameter selection, system parameters and initializations are determined through an empirical tuning. They are adjusted based on cross-validation experiments across diverse operating conditions to ensure stable and efficient adaptation during incremental learning. This systematic approach to parameter tuning not only optimizes the control performance but also enhances the transparency and reproducibility of our methodology.

3.3 Stability analysis

Assuming the general formulation of the Euler-Lagrange systems as:

$$M(q)\,\ddot{q} + H(q,\ \dot{q})\,\dot{q} + d(q) = \tau(t) + \tau_d(t), \tag{24}$$

and defining the system error using the desired trajectory $q_d(t)$ as:

$$e = q - q_d, \tag{25}$$

$$\dot{e} = \dot{q} - \dot{q}_d, \tag{26}$$

$$\xi = \int_{\vartheta=0}^{t} e(\vartheta)\, d\vartheta, \tag{27}$$

we define the computed-torque controller with PID structure as following:

$$\tau(t) = M(q)(\ddot{q}_d - K_d\dot{e} - K_P e - K_I\xi) + H(q,\dot{q})\dot{q}_d + d(q), \tag{28}$$

where K_P, K_D, and K_I are positive definite gain matrices. By substituting this control law into the Euler-Lagrange system dynamics, the closed loop error dynamics become:

$$M(q)\ddot{e} + H(q,\dot{q})\dot{e} + K_P e + K_D\dot{e} + K_I\xi = \tau_d(t). \tag{29}$$

To study the stability of the closed-loop system error dynamics let's assume the Lyapunov candidate function as:

$$V(e,\dot{e},\xi) = \frac{1}{2}\dot{e}^T M(q)\dot{e} + \frac{1}{2}e^T K_P e + \frac{1}{2}\xi^T K_I \xi. \tag{30}$$

which is positive definite provided that $M(q)$, K_P, and K_I are positive definite. Taking the derivative of V with respect to time and following similar steps as before (including the cancellation due to the skew-symmetry property of $\dot{M}(q) - 2H(q,\dot{q})$, we obtain:

$$\dot{V} = -\dot{e}^T K_D\dot{e} + \dot{e}^T\tau_d(t). \tag{31}$$

In the absence of $\tau_d(t)$, $\dot{V} = -\dot{e}^T K_D\dot{e}$, which guarantees asymptotic stability when the linearized characteristic polynomial of the closed-loop system (in a scalar or decoupled diagonal case) is Hurwitz.

$$s^3 + K_D s^2 + K_P s + K_I = 0. \tag{32}$$

Which means that $K_D K_P > K_I$ and $K_P, K_I, K_D > 0$. The additional term $\dot{e}^T\tau_d(t)$ represents the effect of the disturbance. Suppose that the disturbance is bounded in norm:

$$\| \tau_d(t) \| \leq \delta, \quad \forall t \geq 0. \tag{33}$$

Then we have

$$\dot{e}^T\tau(t) \leq \| \dot{e} \| \tau(t) \leq \delta \| \dot{e} \|. \tag{34}$$

Thus, the derivative of V can be bounded as

$$\dot{V} \leq -\lambda_{min}\{K_D\} \| \dot{e} \|^2 + \delta \| \dot{e} \|, \tag{35}$$

where $\lambda_{min}\{K_D\}$ denotes the smallest eigenvalue of K_D. To make the dissipation term more transparent, one may complete the square. For any $\varrho > 0$, by Young's inequality we have

$$\delta \| \dot{e} \| \leq \frac{\delta^2}{2\varrho} + \frac{\varrho}{2} \| \dot{e} \|^2. \tag{36}$$

Choosing $\varrho = \lambda_{min}\{K_D\}$ yields

$$\delta \| \dot{e} \| \leq \frac{\delta^2}{2\lambda_{min}\{K_D\}} + \frac{\lambda_{min}\{K_D\}}{2} \| \dot{e} \|^2. \tag{37}$$

Substitute back into the expression for $\dot{V}$:

$$\begin{aligned} \dot{V} &\leq -\lambda_{min}\{K_D\} \| \dot{e} \|^2 + \frac{\lambda_{min}\{K_D\}}{2} \| \dot{e} \|^2 + \frac{\delta^2}{2\lambda_{min}\{K_D\}} \\ &= -\frac{\lambda_{min}\{K_D\}}{2} \| \dot{e} \|^2 + \frac{\delta^2}{\lambda_{min}\{K_D\}}. \end{aligned} \tag{38}$$

This inequality shows that when $\| \dot{e} \|$ is large, the negative term dominates; however, when $\| \dot{e} \|$ is small, the disturbance term prevents $\dot{V}$ from being strictly negative. In other words, the system is input-to-state stable (ISS): the errors will converge to a bounded set whose size depends on δ.

The candidate function V is still positive definite, and its derivative satisfies

$$\dot{V} \leq -\frac{\lambda_{min}\{K_D\}}{2} \| \dot{e} \|^2 + \frac{\delta^2}{\lambda_{min}\{K_D\}}, \tag{39}$$

which implies that $V(t)$ is bounded for all t.

By standard ISS arguments, the tracking error e, its derivative $\dot{e}$, and the integral error ξ are ultimately bounded by a function of the disturbance magnitude δ. More specifically, there exists a residual set around the origin (i.e., e, $\dot{e}$, ξ converge to a neighborhood of zero) whose size can be made small if δ is small and if the gain K_D is sufficiently large.

As before ($\tau_d(t) = 0$), the gains must satisfy:

- K_P, K_I, and K_D are positive definite.
- In the scalar (or decoupled) case, the condition $K_D K_P > K_I$ (or its matrix analogue) remains important.

These conditions, combined with the choice of K_D to ensure sufficient damping, guarantee that the disturbance's effect is attenuated.

Now, starting from a set of PID parameters that satisfy these conditions, we will demonstrate that the system remains stable while the rule outputs evolve over time. Based on equation (23), while the system is under test, the output of the TSK fuzzy subsystem is given by:

$$y_j = \sum_{l=1}^{L} \omega^l \bar{\mathcal{Y}}_{l,j},\ j = 1, \cdots, q. \tag{40}$$

In which,

$$\omega^l = \frac{\prod_{i=1}^{p} \mu_{F_{i,l}}(e_i)}{\sum_{k=1}^{L} \prod_{i=1}^{p} \mu_{F_{i,k}}(e_i)}. \tag{41}$$

Additionally, e_i represents the i^{th} element of the introduced error vector. By temporarily disregarding the Mamdani subsystem and considering the stability analysis provided for the PID controller along with the inputs of the TSK fuzzy subsystem, the stability condition of the closed-loop system is given by:

$$y_1, y_2, y_3 > 0, \tag{42}$$

$$y_1 y_2 < y_3. \tag{43}$$

Which can be written as:

$$\left(\sum_{l=1}^{L} \omega^l \bar{\mathcal{Y}}_{l,1}\right)\left(\sum_{l=1}^{L} \omega^l \bar{\mathcal{Y}}_{l,2}\right) < \sum_{l=1}^{L} \omega^l \bar{\mathcal{Y}}_{l,3}, \tag{44}$$

$$\longrightarrow \quad \sum_{l=1}^{L}\sum_{k=1}^{L} \omega^l \omega^k \bar{\mathcal{Y}}_{l,1} \bar{\mathcal{Y}}_{k,2} < \sum_{l=1}^{L} \omega^l \bar{\mathcal{Y}}_{l,3}. \tag{45}$$

A sufficient condition is that each pair satisfies:

$$\bar{\mathcal{Y}}_{l,1} \bar{\mathcal{Y}}_{k,2} < \sqrt{\bar{\mathcal{Y}}_{l,3} \bar{\mathcal{Y}}_{k,3}}, \quad l, k = 1, \cdots, L. \tag{46}$$

This ensures that:

$$\sum_{l=1}^{L}\sum_{k=1}^{L} \omega^l \omega^k \bar{\mathcal{Y}}_{l,1} \bar{\mathcal{Y}}_{k,2} < \sum_{l=1}^{L}\sum_{k=1}^{L} \omega^l \omega^k \sqrt{\bar{\mathcal{Y}}_{l,3} \bar{\mathcal{Y}}_{k,3}}. \tag{47}$$

By applying Jensen's inequality to the convex function $f(x) = x^2$, we obtain $y_1 y_2 < y_3$. As mentioned earlier, the Mamdani fuzzy subsystem acts as a saturation function, preventing the control signal from growing beyond the capacity of the actuators. In the stability analysis, this manifests as a condition that renders stability local. Up to this point, we have demonstrated that the system remains stable by initializing the outputs of the TSK fuzzy subsystem using stable PID controller coefficients.

Now, we show that, starting from a stable controller, the closed-loop system remains stable at the end of the training process. Since only successful experiences are used, and based on the time term in the cost function formula (equation (17)), an experience is considered successful only when the error converges at a rate higher than linear, with a specific active rule, until a new rule update is triggered. If the update is triggered by a change in the active rule, it indicates that one of the error states have passed through zero. Consequently, the update process searches for a better output for the active rule. Using the previous stability analysis and the output initialization method, we ensure convergence for all active rules. As a result, the system error remains bounded during the training process, and by optimizing the parameter set with the HEO algorithm, this bound will converge to zero or reach exactly zero when asymptotically stable outputs for the fuzzy TSK subsystem are obtained for every rule.

3.4 Scalability and computational complexity

Scalability is a critical consideration in our proposed control scheme, particularly when extending its application to systems with higher degrees of freedom or more complex dynamics. The incremental learning framework is designed to adapt its rule base continuously, allowing the controller to adjust to new information without the need for complete

retraining. As the system scales, the computational complexity associated with rule evolution and updating can increase; however, our approach mitigates this by employing strategies such as dynamic rule pruning. This techniques help to manage the number of active rules, ensuring that the computational load grows sub-linearly relative to the system's complexity.

Furthermore, a trade-off exists between the accuracy of the control performance and the efficiency of computation. To balance this, learning rates and weighting factors are carefully tuned, allowing for effective incremental updates while minimizing computational overhead. Preliminary analyses suggest that, even as the system size increases, our method maintains a robust performance without incurring prohibitive computational costs. In addition, future enhancements such as parallel processing and hardware acceleration are envisioned to further improve scalability in large-scale applications.

4 Simulation results

In this section, we will evaluate the proposed fuzzy incremental learning-based control scheme in contrast with reinforcement learning models. For this purpose, we will use a robotic manipulator dynamic model. This model is widely used in studies concerning the Euler-Lagrange systems and is a well-known benchmark system. The comparison criteria for different models, in addition to convergence characteristics, are the minimum number of required runs in the simulation horizon for achieving a certain convergence performance, settling time of joint angles' error, and overshoot of joint angles' norm. We assume the simulation horizon as the interval $[0,\ 5]$ seconds for simulating nominal and uncertainty scenarios.

According to equation (1), the dynamics of the robotic manipulator is given by following definitions:

$$M\left(q\right)=\left[\begin{array}{cc} M_{11} & M_{12} \\ M_{21} & M_{22} \end{array}\right], \tag{48}$$

where the elements of matrix $M\left(q\right)$ are achieved using the formulas:

$$M_{11}=I_{Z_1}+I_{Z_2}+m_1\frac{1}{4}l_1^2+m_2\left(l_1^2+\frac{1}{4}l_2^2+l_1l_2cos\left(q_2\right)\right), \tag{49}$$

$$M_{12}=M_{21}=I_{Z_2}+m_2\left(\frac{1}{4}l_2^2+\frac{1}{2}l_1l_2cos\left(q_2\right)\right), \tag{50}$$

$$M_{22}=I_{Z_2}+\frac{1}{4}m_2l_2^2. \tag{51}$$

In the above relations, I_{Z_i}, m_i, and l_i indicate the inertial momentum, mass, and length of the i^{th} link, respectively. Also,

$$H\left(q,\ \dot{q}\right)=\left[\begin{array}{cc} -c\dot{q}_2+c_1^* & -c\left(\dot{q}_1+\dot{q}_2\right) \\ c\dot{q}_1 & c_2^* \end{array}\right], \tag{52}$$

in which, c_i^* is the friction coefficient of the i^{th} link, and

$$c=\frac{1}{2}m_2l_1l_2sin\left(q_2\right). \tag{53}$$

The coefficient c is often referred to as a centrifugal and Coriolis coupling coefficient or simply as a dynamic coupling coefficient. It typically arises from the terms that model the interaction between the velocities and/or accelerations of the links and quantifies how the motion of one joint affects the dynamic forces on another joint through Coriolis and centrifugal effects. c can also be broadly referred to as a dynamic coupling coefficient because it appears in the off-diagonal elements of the Coriolis matrix and represents coupling between joint motions. Finally,

$$d\left(q\right)=\left[\begin{array}{c} \frac{1}{2}m_1gl_1cos\left(q_1\right)+m_2g\left(l_1cos\left(q_1\right)+\frac{1}{2}l_2cos\left(q_1+q_2\right)\right) \\ \frac{1}{2}m_2gl_2cos\left(q_1+q_2\right) \end{array}\right]. \tag{54}$$

The actuator for each joint in this robot is a DC motor with a peak torque of $8.5\ N\cdot m$, and Table 1 represents the joint parameters [40]. All the system parameters are assumed to be uncertain with errors of the order 10% [48], 25%, and 50% in three scenarios. Therefore, using the equation (7) introduced in section 2, the system uncertainties can be calculated.

Table 1: Parameters of the system model, detailing the key variables and coefficients used in the analysis.

Description	**Value**	**Unit**
Mass of link-i: m_i	0.506	kg
Length of link-i: l_i	0.305	m
Inertial moment of link-i: I_{Z_i}	0.027	$kg \cdot m^2$
Friction coefficient of link-i: c_i^*	0.01625	$kg \cdot m^2/s$
Gravity acceeration: g	9.81	m/s^2

Note that, in simulations, we initialized the robot from $q(0) = [0.8,\ 0.6]^T$ and $\dot{q}(0) = [0.4,\ 0.3]^T$. We trained the proposed control scheme over 50 executions with $T_u = 0.1$, meaning a minimum number of 2500 epochs. Also, we trained the MLP, RBFN, and TSK controllers using the HEO algorithm with population size of 50 in 100 iterations, meaning 5000 complete executions and also 5000 epochs. Based on the theoretical analysis of system stability presented in Section 3.3, the HEO algorithm improves the convergence characteristics of the closed-loop system without affecting its stability. However, it can influence the quality of the system's response. In this study, the selected parameters are those recommended by the authors of the article that introduced the HEO algorithm [15]. Future research could investigate the use of alternative optimization algorithms with different parameter settings for a comparative analysis.

Since the robot under study has two joints, we must utilize two controllers for the system. As we mentioned, each controller has three inputs, and each input is represented using three fuzzy sets. So, the full rule base of the TSK subsystem consists of 27 rules. Also, since the TSK subsystem has three outputs, each rule similarly needs three output values. It means that, L_1 and p in equation (16) are 27 and 3, respectively. We have initialized these $27 \times 3 = 81$ output values using simple and not optimized PID controller coefficients as follows:

$$\bar{\mathcal{Y}}^{init} = \left[\bar{\mathcal{Y}}_{l,j}^{init}\right]_{L_1 \times p} = \begin{bmatrix} 2 & 0.5 & 0.5 \\ 2 & 0.5 & 0.5 \\ & \cdots & \\ 2 & 0.5 & 0.5 \end{bmatrix}. \tag{55}$$

In this matrix, the columns in the l^{th} row represent the Proportional, Integral, and Derivative coefficients of the PID controller, which serve as the output of the l^{th} rule. When starting the training process or when there is still no successful experience, we should use the initial coefficients $\bar{\mathcal{Y}}_{l,j}^{init}$ in Equation (19).

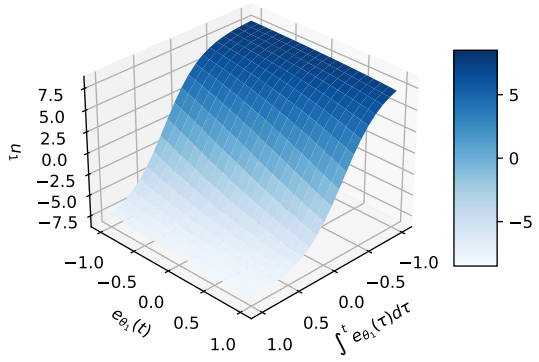
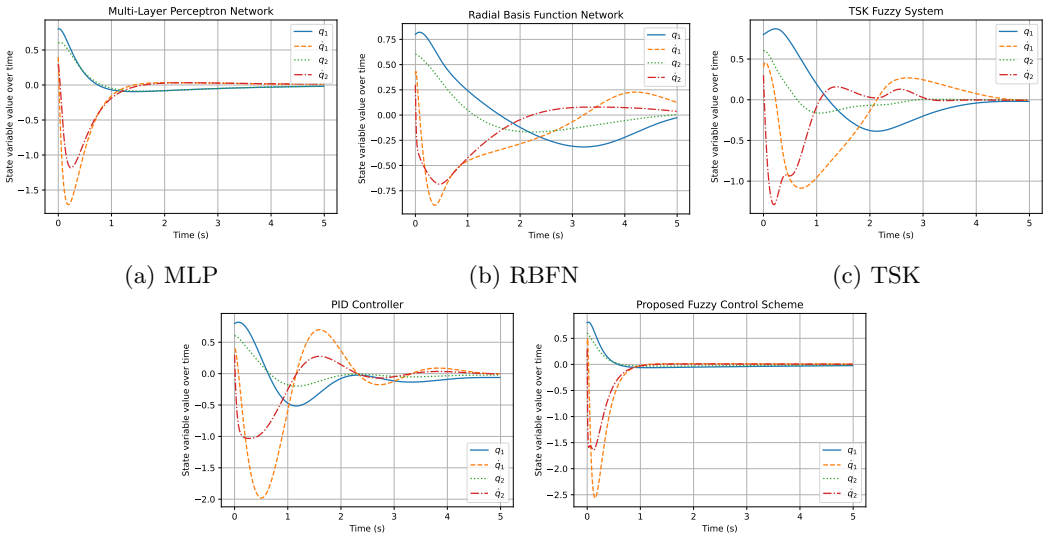


Figure 4: 3D illustration of the control surface for the first joint's controller, showing how the control law responds to variations in input variables of the controller while $\dot{e}_{\theta_1} = 0$.

Along with the reinforcement-learning-based MLP, RBFN, and TSK controllers, we have mentioned the results achieved by the PID controller to show that the proposed scheme can achieve significant results despite starting from a poor initialization. Of course, we can select the PID controller parameters to achieve better results, but it may suffer from a lack of adaptability. The MLP model, used as the controller, has three layers with 3, 9, and 1 neurons, respectively. The RBFN model used in simulations has a 3-dimensional feature vector and 12 kernels. The TSK

method has 3 inputs and 12 rules with all fuzzy sets as Gaussian. The t-norm and s-norm of the TSK system have been selected as product and maximum, respectively. These assumptions have been made to have approximately the same total parameter number for all four learning-based models. We have demonstrated the achieved control surfaces for the first joint controller after the learning process in Figure 4. In the first scenario, considering the nominal system, Figure 5 shows the system state variables achieved by five different control methods. Figure 6 depicts the convergence diagram of the joint angles error. Also, Figure 7 demonstrates the norm of the control input vector over the simulation horizon. Figures 8, 9, and 10 represents the system response achieved by different methods considering the 10%, 25%, and 50% uncertainty scenarios, respectively. Due to the poor results in uncertainty scenarios by the RBFN and TSK controllers, which are reflected in the comparison table, we avoided illustrating their results to save space. Table 2 compares the performance measures achieved by different methods for the control task of nominal system and three uncertainty scenarios.

To evaluate the effectiveness of our proposed control scheme, we consider a comprehensive set of performance metrics that capture both transient and steady-state behaviors under nominal and uncertain conditions. First, the settling time-defined as the time required for the system response to remain within a 4% error band of the desired value-is measured for both the nominal and uncertain systems. A shorter settling time indicates a faster response, which is critical for applications requiring rapid adaptation. Next, the Integral of Absolute Error (IAE) is computed over the entire response period for both the nominal and uncertain systems. The IAE provides a cumulative measure of tracking accuracy, reflecting the controller's overall precision in minimizing deviations from the target trajectory. Finally, overshoot is evaluated for both system conditions, offering insight into the stability and robustness of the control scheme; excessive overshoot can be indicative of an unstable or overly aggressive response. Together, these metrics-settling time, IAE, and overshoot-provide a detailed and balanced assessment of the control performance. Minimizing settling time and IAE, along with controlling overshoot, aligns with our study's goals of achieving rapid, accurate, and stable performance in both nominal and uncertain environments. As can be seen, in all cases, the proposed incremental learning-based control scheme has achieved significantly better results. The proposed method shows an improvement of 40.89% compared with the best other method in nominal system scenario. The major improvement can also be tracked in uncertainty scenarios with 38.32%, 34.12%, and 34.79% less IAEs. In the case of settling time, while the RBFN, TSK, and PID controllers have not settled down in the tough most simulation scenario, the proposed method achieved the $4.6054s$ compared with $4.8303s$ for the MLP method. In all other scenarios, the proposed method overcomes the results achieved by the MLP method. Finally, in four scenarios, the proposed control scheme reduces the system response overshoot by approximately $44-48\%$ compared to the best-performing alternative method. The commendable outcomes of the proposed control method are contingent upon the structure of the controller, which effectively fosters and facilitates incremental learning.



(a) MLP (b) RBFN (c) TSK

(d) PID (e) Proposed control scheme

Figure 5: System response obtained using the proposed control scheme, PID controller, and three reinforcement-based control schemes, highlighting their performance in trajectory tracking and stability.

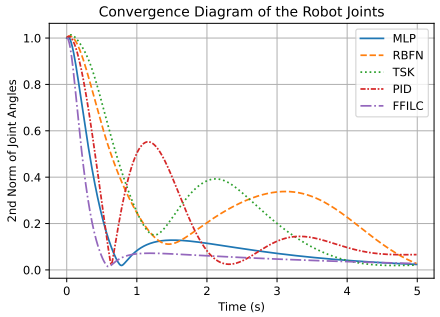


Figure 6: Comparison of joint angle error convergence achieved by different methods, illustrating the rate and stability of error reduction over time.

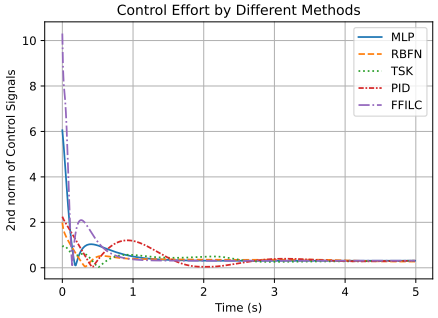
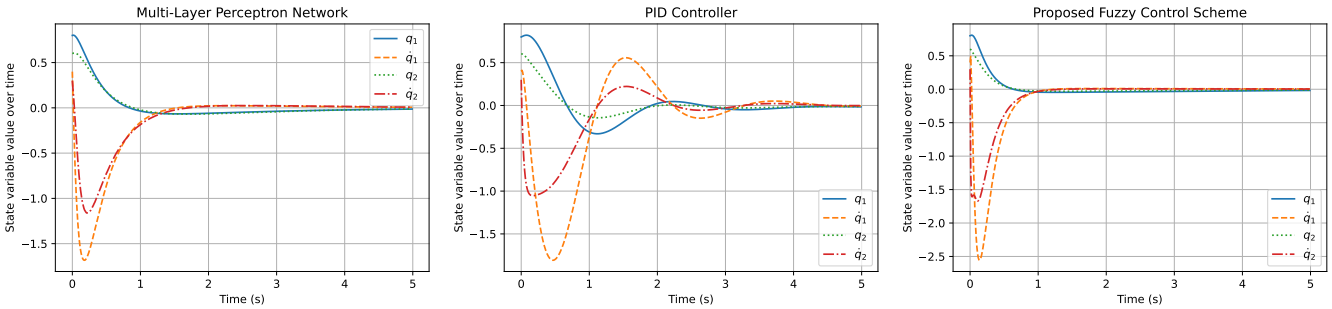
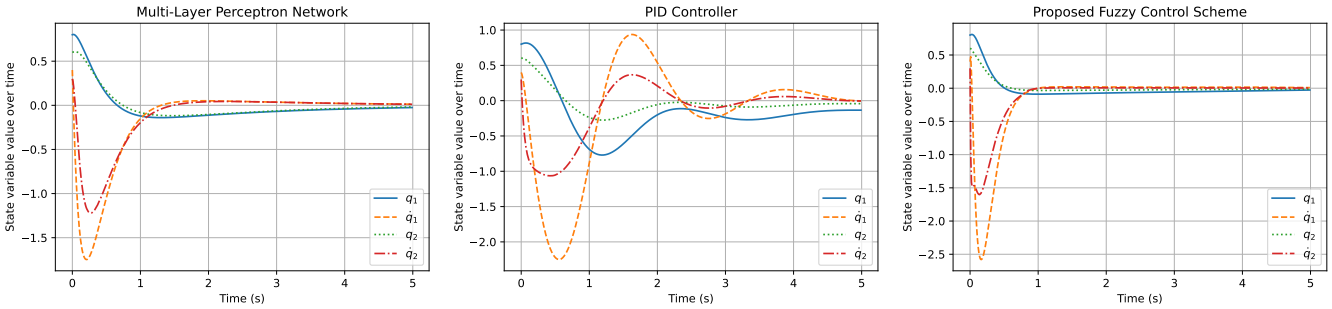


Figure 7: Comparison of the control signal norm achieved by different methods, highlighting the efficiency and smoothness of control actions.



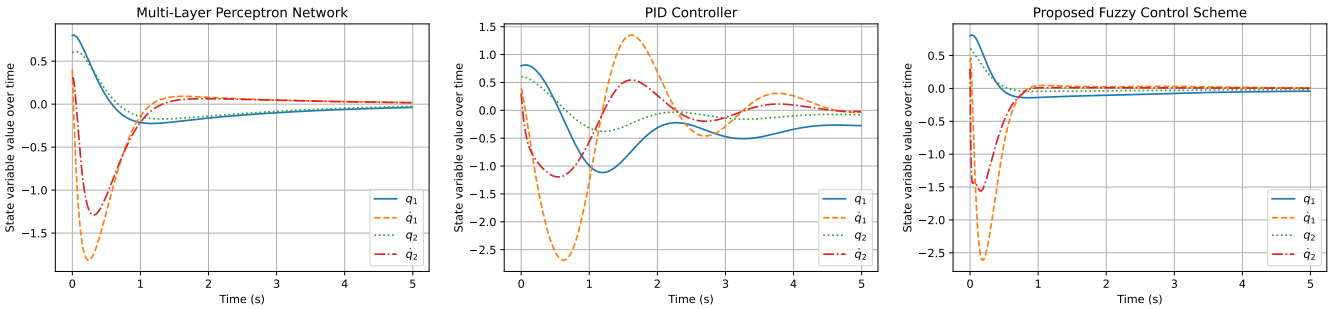
(a) MLP (b) PID (c) Proposed control scheme

Figure 8: System response obtained using different methods under uncertainty scenario 1, highlighting their robustness and adaptability to uncertainties.



(a) MLP (b) PID (c) Proposed control scheme

Figure 9: System response obtained using different methods under uncertainty scenario 2, highlighting their robustness and adaptability to uncertainties.



(a) MLP (b) PID (c) Proposed control scheme

Figure 10: System response obtained using different methods under uncertainty scenario 3, highlighting their robustness and adaptability to uncertainties.

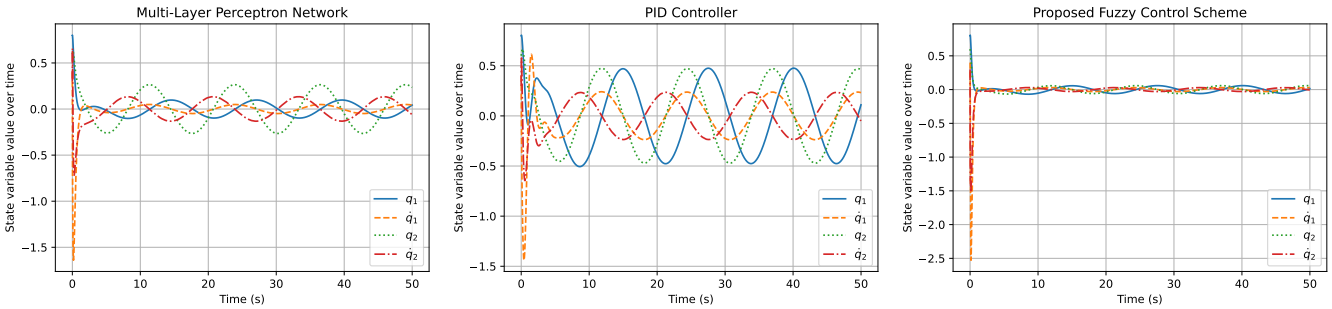
Table 2: Comparison of the performance metrics of different control methods, highlighting their effectiveness and efficiency.

	MLP	**RBFN**	**TSK**	**PID**	**Proposed**
Settling time (4%) of the nominal system	3.6340s	–	–	–	**2.6893s**
Settling time (4%) of the uncertain system (Scenario 1)	4.0480s	–	–	–	**3.5810s**
Settling time (4%) of the uncertain system (Scenario 2)	4.5567s	–	–	–	**4.2715s**
Settling time (4%) of the uncertain system (Scenario 3)	4.8303s	–	–	–	**4.6054s**
IAE of the nominal system	0.5915	1.4349	1.1761	0.7684	**0.3496**
IAE of the uncertain system (Scenario 1)	0.6612	1.5004	1.3926	1.1328	**0.4078**
IAE of the uncertain system (Scenario 2)	0.7772	1.8328	1.7641	1.7252	**0.5120**
IAE of the uncertain system (Scenario 3)	0.9902	2.7643	–	2.5746	**0.6457**
Overshoot of the nominal system	9.57%	28.12%	28.61%	36.09%	**5.52%**
Overshoot of the uncertain system (Scenario 1)	12.84%	33.81%	39.25%	55.25%	**7.19%**
Overshoot of the uncertain system (Scenario 2)	18.16%	47.86%	55.79%	81.81%	**9.59%**
Overshoot of the uncertain system (Scenario 3)	28.13%	85.07%	–%	117.80%	**14.90%**

To evaluate the disturbance rejection performance of the proposed controller, we assume the disturbance vector applied to the system as an additive term to the control signal as below [40]:

$$\tau_d = \left[\ sin(0.5t), \ cos(0.5t) \ \right]^T. \tag{56}$$

In this case, the simulation horizon has been considered as $[0, 50]$ seconds. According to the results, the RBFN and TSK controllers were not able to properly reject the disturbance. The IAE for these controllers were 20.5683 and 23.6115, respectively. Figures 11(a), (b), and (c) demonstrate the system response achieved by MLP, PID, and the proposed control methods, respectively. Compared with other controllers, the proposed control scheme has a better disturbance rejection performance without any additional disturbance observer or predictor. In this case, the IAE achieved by the MLP method is 9.7823, while the proposed controller yields an IAE equal to 3.1105. Also, the MLP method achieves an error resonating about the constant value 0.1811, while the proposed control scheme experiences an error of about 0.0588 in the presence of disturbance signals.



(a) MLP (b) PID (c) Proposed control scheme

Figure 11: System response obtained using different methods in the presence of disturbance signals, illustrating their disturbance rejection capabilities.

For going more deep in noise rejection property of the controllers, we have plotted the frequency response of the closed loop system considering different controllers in Figure 12. As can be seen, while in low frequencies the proposed controller has just a slight superiority over other methods, in high frequencies its supremacy is more obvious. Note that, these figures have been plotted after the training phase and achieving the final controller gains. Furthermore, for having an insight into noise rejection capability of the proposed method compared with other four methods, we have repeated the simulation for the nominal system in the presence of input noise with $500Hz$ frequency. Figure 13 demonstrates the system response achieved by the proposed control scheme, PID controller, and three reinforcement-based control schemes in this case. Also, Figures 14 and 15 illustrate the joint angles error and control signal norm achieved by different methods, respectively. Similarly, the proposed controller was able to handle the noise better compared with the other methods.

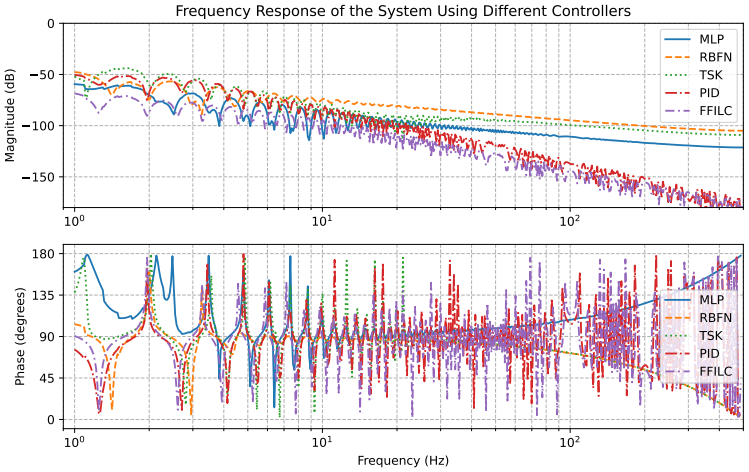
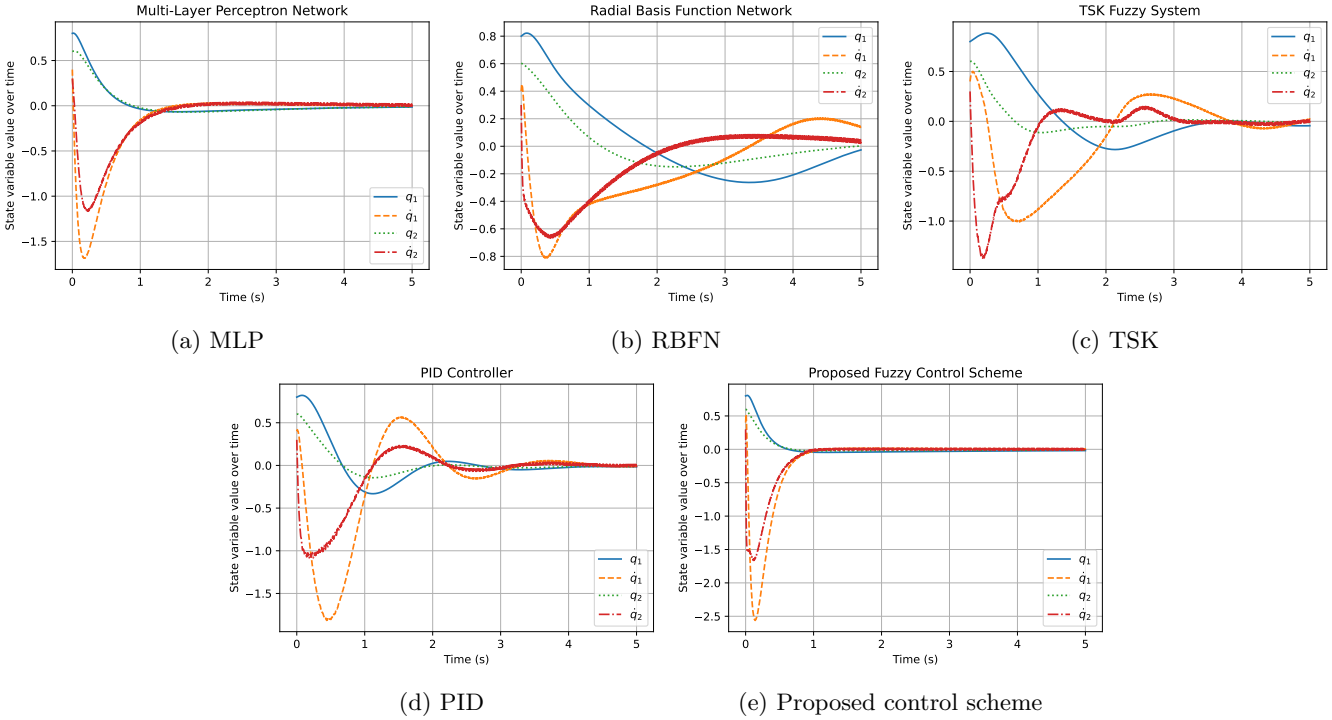


Figure 12: Frequency response of the system for different controllers, highlighting their impact on stability and dynamic performance.



(a) MLP (b) RBFN (c) TSK

(d) PID (e) Proposed control scheme

Figure 13: System response obtained using the proposed control scheme, PID controller, and three reinforcement-based control schemes under high-frequency stimulation.

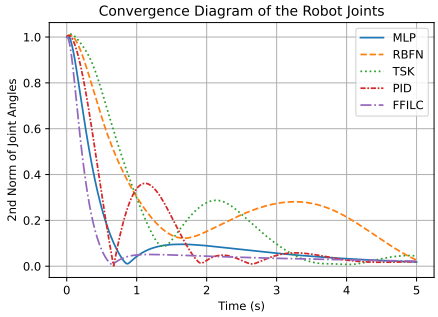


Figure 14: Joint angle error convergence achieved by different control methods under high-frequency system stimulation, highlighting their tracking accuracy and robustness.

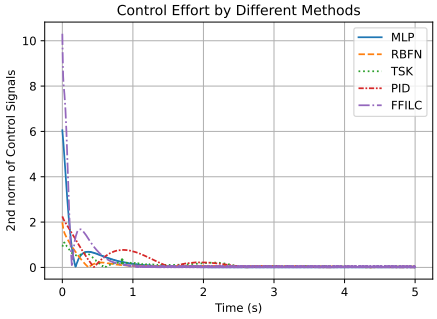
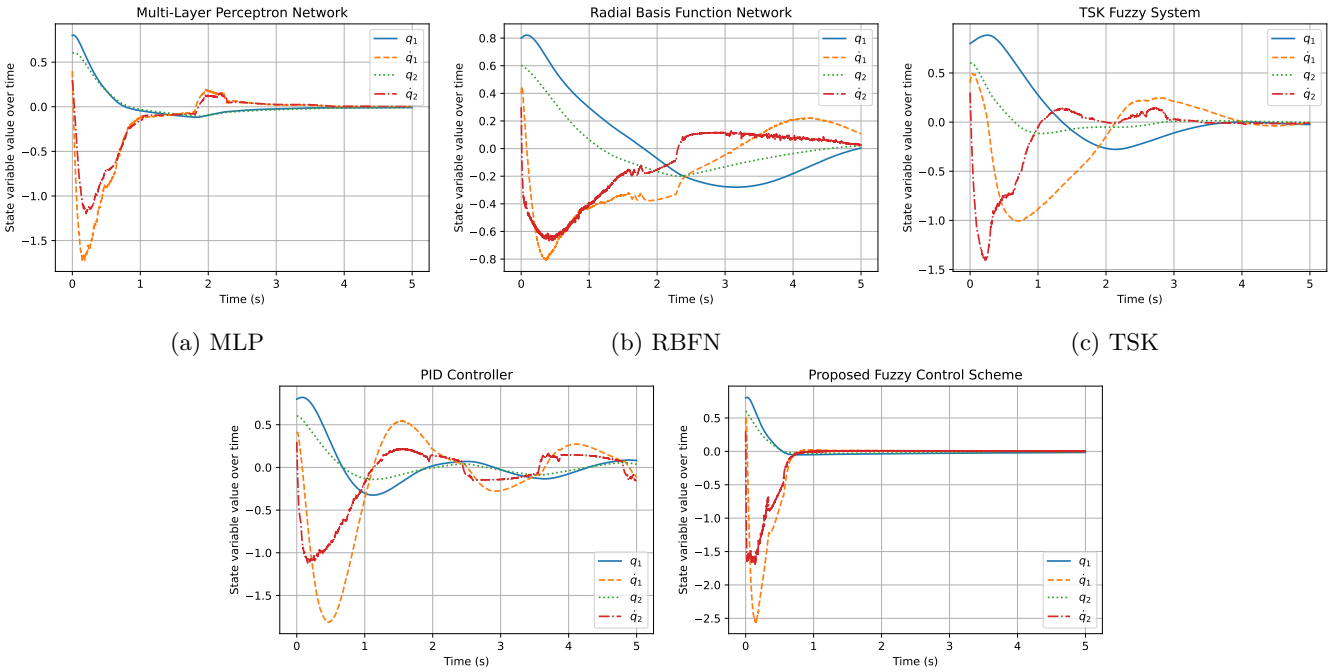


Figure 15: Control signal norm obtained using different methods under high-frequency system stimulation, highlighting their efficiency and control effort.

Another important issue in control systems is noise in the state feedback signals. To evaluate the robustness of the controllers against high-frequency state feedback noise, we introduced a randomly generated noise signal into the state feedback. Figure 16 illustrates the system response obtained using the proposed control scheme, the PID controller, and three reinforcement-based control schemes under this condition. Additionally, Figures 17 and 18 depict the joint angle errors and control signal norms achieved by different methods, respectively. The proposed controller demonstrated superior noise-handling capability compared to the other methods. The Integral of Absolute Error (IAE) achieved by the proposed controller in this scenario is 0.4066, whereas the best-performing alternative method (MLP) achieved an IAE of 0.5933.



(a) MLP (b) RBFN (c) TSK

(d) PID (e) Proposed control scheme

Figure 16: System response obtained using the proposed control scheme, PID controller, and three reinforcement-based control schemes under high-frequency state feedback noise.

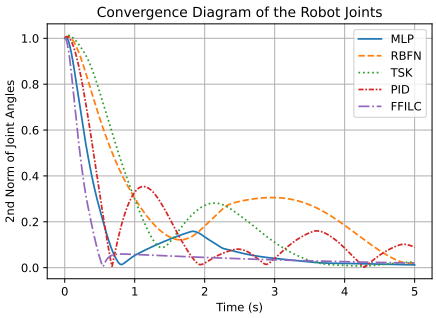


Figure 17: Joint angle error convergence achieved by different methods under high-frequency state feedback noise, highlighting tracking accuracy and robustness.

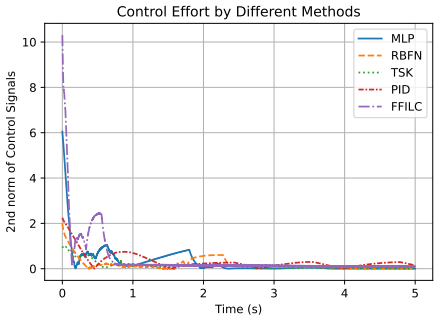


Figure 18: Control signal norm obtained using different methods under high-frequency state feedback noise, highlighting efficiency and control effort.

While we have trained the MLP, RBFN, and TSK algorithms over 5000 executions and 5000 epochs, the proposed incremental learning scheme only needs 50 execution and more than 2500 epochs to overcome other methods. Considering that each execution is a simulation of 5 seconds, the total simulation time for reinforcement learning-based methods is $2.5e4$ seconds of simulation, while for the proposed method, it is just $2.5e2$ seconds. Figure 19 represents the normalized cost function and its moving average over training process. Due to randomized initial condition of system during the training process, the cost function demonstrates a noisy fluctuating pattern. However, the moving average of cost function is decreasing over different runs, confirming the efficiency of the mechanism for assigning performance measure, storing experiences, and updating the outputs of the active fuzzy if-then rule as presented in Section 3. Finally, it should be noted that we have done the simulations using Numpy [8] and SciPy [39] Python libraries and the data representation by Matplotlib library [11].

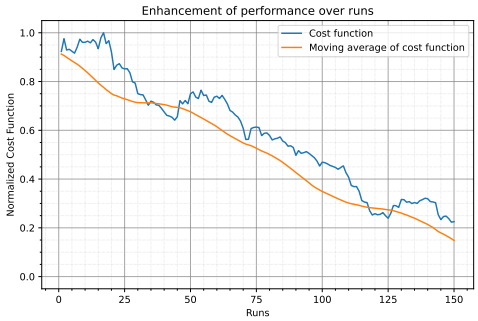


Figure 19: Normalized cost function and its moving average over the first 150 runs during the training process of the proposed incremental learning scheme, which continues for a total of 200 runs.

5 Conclusions

In this paper, we proposed an incremental learning-based fuzzy control scheme for uncertain Euler-Lagrange systems and initialized the system parameters using simple PID controller coefficients. The incremental learning algorithm evolved initial parameters to meet the expectations. The proposed model can transform the system performance into experiences and use the experiences to make decisions about the controller parameters. We have done the simulations on a well-known and extensively used Euler-Lagrange benchmark system, and the results confirmed the capabilities of the proposed scheme. The results prove that, with much shorter training, the proposed control scheme can achieve better performance. Compared with other methods, the IAE, settling time, overshoot, and disturbance rejection performance of the proposed fuzzy incremental learning-based controller is better. In future work, we will also involve the unsuccessful experiences in the decision-making process and add more heuristics to the algorithm, updating the outputs of TSK fuzzy subsystem rules.

Acknowledgement

The authors would like to express their sincere appreciation to the referees for their valuable and insightful suggestions that have significantly improved the quality of this paper.

Conflict of interest

The authors affirm that they have no conflicts of interest, whether financial or otherwise, that could have affected the conduct of this research or the reporting of its results.

Data and Code Availability

The simulation code and all data supporting the findings of this study are available in the following GitHub repository: https://github.com/Haghrah/Incremental-Fuzzy-Control-of-Euler-Lagrange-Systems. This includes the initial version of the simulation code used for the analyses presented in the article, as well as the weights and coefficients obtained during the learning process (both reinforcement and incremental).

References

[1] Z. Alammar, L. Alzubaidi, J. Zhang, Y. Li, A. Gupta, Y. Gu, *Generalisable deep learning framework to overcome catastrophic forgetting*, Intelligent Systems with Applications, **23** (2024), 200415. https://doi.org/10.1016/j.iswa.2024.200415

[2] K. S. T. R. Alves, C. D. de Jesus, E. P. de Aguiar, *A new takagi–sugeno–kang model to time series forecasting*, Engineering Applications of Artificial Intelligence, **133** (2024), 108155. https://doi.org/10.1016/j.engappai.2024.108155

[3] S. Balovsyak, O. Derevyanchuk, V. Kovalchuk, H. Kravchenko, M. Kozhokar, *Face mask recognition by the viola-jones method using fuzzy logic*, International Journal of Image, Graphics and Signal Processing (IJIGSP), **16** (2024), 39–51. https://doi.org/10.5815/ijigsp.2024.03.04

[4] Q. Chen, Y. Jin, Y. Song, *Fault-tolerant adaptive tracking control of euler-lagrange systems–an echo state network approach driven by reinforcement learning*, Neurocomputing, **484** (2022), 109–116. https://doi.org/10.1016/j.neucom.2021.10.083

[5] Z. Chen, B. Liu, *Lifelong machine learning*, **1**, Springer, (2018). https://doi.org/10.1007/978-3-031-01581-6

[6] H. Gao, W. Bi, X. Wu, Z. Li, Z. Kan, Y. Kang, *Adaptive fuzzy-region-based control of euler–lagrange systems with kinematically singular configurations*, IEEE Transactions on Fuzzy Systems, **29** (2020), 2169–2179. https://doi.org/10.1109/TFUZZ.2020.2994991

[7] A. Gijsberts, G. Metta, *Incremental learning of robot dynamics using random features*, 2011 IEEE International Conference on Robotics and Automation, IEEE (2011), 951–956. https://doi.org/10.1109/ICRA.2011.5980191

[8] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, *Array programming with NumPy*, Nature, **585** (2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[9] X. He, M. Lu, *Output feedback control of uncertain euler–lagrange systems by internal model*, Automatica, **156** (2023), 111189. https://doi.org/10.1016/j.automatica.2023.111189

[10] Y. He, Y. Zhao, *Adaptive robust control of uncertain euler–lagrange systems using gaussian processes*, IEEE Transactions on Neural Networks and Learning Systems, **35** (2022), 7949–7962. https://doi.org/10.1109/TNNLS.2022.3222405

[11] J. D. Hunter, *Matplotlib: A 2d graphics environment*, Computing in Science and Engineering, **9** (2007), 90–95. https://doi.org/10.1109/MCSE.2007.55

[12] L. Jamone, B. Damas, J. Santos-Victor, *Incremental learning of context-dependent dynamic internal models for robot control*, 2014 IEEE International Symposium on Intelligent Control (ISIC), IEEE (2014), 1336–1341. https://doi.org/10.1109/ISIC.2014.6967617

[13] L. T. Le, L. K. Ngo, *Application of combining type 2 fuzzy cmac network and jordan neural network in nonlinear system control*, Iranian Journal of Fuzzy Systems, **22** (2025), 25–39. https://doi.org/10.22111/ijfs.2025.50552.8930

[14] I. E. Leal-Leal, E. Alcorta-Garcia, *An efficient fault tolerant control scheme for euler–lagrange systems*, International Journal of Applied Mathematics and Computer Science, **33** (2023), 219–228. https://doi.org/10.34768/amcs-2023-0017

[15] J. Lian, G. Hui, *Human evolutionary optimization algorithm*, Expert Systems with Applications, **241** (2024), 122638. https://doi.org/10.1016/j.eswa.2023.122638

[16] Y. Liu, Y. Li, *Application of inverse optimal formation control for euler-lagrange systems*, IEEE Transactions on Intelligent Transportation Systems, **24** (2023), 5655–5662. https://doi.org/10.1109/TITS.2023.3240135

[17] E. D. Lughofer, *Flexfis: A robust incremental learning approach for evolving takagi–sugeno fuzzy models*, IEEE Transactions on Fuzzy Systems, **16** (2008), 1393–1410. https://doi.org/10.1109/TFUZZ.2008.925908

[18] M. McCloskey, N. J. Cohen, *Catastrophic interference in connectionist networks: The sequential learning problem*, Psychology of Learning and Motivation, Elsevier, **24** (1989), 109–165. https://doi.org/10.1016/S0079-7421(08)60536-8

[19] O. Mehrabi, A. Fakharian, M. Siahi, A. Ramezani, *Fuzzy radial basis function least square policy iteration: A novel critic-only reinforcement learning framework*, Iranian Journal of Fuzzy Systems, **22** (2025), 59–80. https://doi.org/10.22111/ijfs.2025.48345.8503

[20] M. Mirzaee, R. Kazemi, *Type-ii fuzzy inference system-based fractional terminal sliding mode control for zero-force exoskeleton robots*, Iranian Journal of Fuzzy Systems, **21** (2024), 147–171. https://doi.org/10.22111/ijfs.2025.49399.8718

[21] A. Mirzaei, A. Nazemi, *Parallel synchronization and a rbf neuro-fuzzy system to synchronization of chaotic systems*, Iranian Journal of Fuzzy Systems, **21** (2024), 101–126. https://doi.org/10.22111/ijfs.2025.48808.8609

[22] A. Naghsh, F. Sheikholeslam, M. Danesh, *Design of an adaptive fuzzy estimator for force/position tracking in robot manipulators*, Iranian Journal of Fuzzy Systems, **11** (2014), 75–89. https://doi.org/10.22111/ijfs.2014.1396

[23] T. L. Nguyen, H. T. Tran, T. T. Tran, C. T. Pham, *Reinforcement learning-based event-triggered robust optimal control for mobile euler-lagrange systems with dead-zone and saturation actuators*, International Journal of Mechanical Engineering and Robotics Research, **10** (2021), 116–127. https://doi.org/10.18178/ijmerr.10.3.116-127

[24] S. Ober-Blöbaum, C. Offen, *Variational learning of euler–lagrange dynamics from data*, Journal of Computational and Applied Mathematics, **421** (2023), 114780. https://doi.org/10.1016/j.cam.2022.114780

[25] E. Y. Özkan, *Inequalities and numerical results of approximation for bivariate q–baskakov–durrmeyer type operators including q–improper integral*, Journal of Mathematical Inequalities, **16** (2022), 499–512. https://doi.org/10.7153/jmi-2022-16-36

[26] E. Y. Özkan, *A new kantorovich-type rational operator and inequalities for its approximation*, Mathematics, **10** (2022), 1982. https://doi.org/10.3390/math10121982

[27] E. Y. Özkan, *Some inequalities and numerical results estimating error of approximation for tensor product kind bivariate quantum beta-type operators and pertaining to gbs variant*, Journal of Inequalities and Applications, **2022** (2022), 66. https://doi.org/10.1186/s13660-022-02798-w

[28] E. Y. Özkan, *Approximation process of the fuzzy meyer-könig and zeller operators*, Operators, Inequalities and Approximation: Theory and Applications, Springer (2024), ISBN 978-981-97-3237-1, 33–51. https://doi.org/10.1007/978-981-97-3238-8_2

[29] M. Pratama, S. G. Anavatti, P. P. Angelov, E. Lughofer, *Panfis: A novel incremental learning machine*, IEEE Transactions on Neural Networks and Learning Systems, **25** (2013), 55–68. https://doi.org/10.1109/TNNLS.2013.2271933

[30] M. S. Rani, R. Sai, *Qpmmcoa and bayesian fuzzy clustering: A novel approaches for optimizing queries in big data*, Iranian Journal of Fuzzy Systems, **22** (2025), 1–24. https://doi.org/10.22111/ijfs.2025.47412.8350

[31] L. Saihi, B. Berbaoui, *Hybrid neuro-fuzzy integral sliding mode control technique optimized upqc for power quality improvement in solar systems*, Iranian Journal of Fuzzy Systems, **22** (2025), 147–168. https://doi.org/10.22111/ijfs.2025.49224.8678

[32] R. Santhosh, D. J. Sut, M. Uma, P. Sethuramalingam, *Optimizing irb1410 industrial robot painting processes through taguchi method and fuzzy logic integration with machine learning*, International Journal of Intelligent Robotics and Applications, **8** (2024), 380–400. https://doi.org/10.1007/s41315-024-00325-2

[33] X. Shi, J. Wang, B. Zhang, *A fuzzy time series forecasting model with both accuracy and interpretability is used to forecast wind power*, Applied Energy, **353** (2024), 122015. https://doi.org/10.1016/j.apenergy.2023.122015

[34] M. Siami, M. Naderpour, F. Ramezani, J. Lu, *Risk assessment through big data: An autonomous fuzzy decision support system*, IEEE Transactions on Intelligent Transportation Systems, **25** (2024), 9016–9027. https://doi.org/10.1109/TITS.2024.3392959

[35] P. Su, J. Yu, Y. Hua, Q. Li, X. Dong, Z. Ren, *Distributed time-varying formation optimal tracking for uncertain euler–lagrange systems with time-varying cost functions*, Aerospace Science and Technology, **132** (2023), 108019. https://doi.org/10.1016/j.ast.2022.108019

[36] G. Sun, Q. Zeng, *A finite-time sliding mode control approach for constrained euler–lagrange system*, Mathematics, **11** (2023), 2788. https://doi.org/10.3390/math11122788

[37] M. Tanveer, M. Sajid, M. Akhtar, A. Quadir, T. Goel, A. Aimen, S. Mitra, Y. D. Zhang, C. T. Lin, J. Del Ser, *Fuzzy deep learning for the diagnosis of alzheimer's disease: Approaches and challenges*, IEEE Transactions on Fuzzy Systems, **32** (2024), 5477–5492. https://doi.org/10.1109/TFUZZ.2024.3409412

[38] G. M. van de Ven, T. Tuytelaars, A. S. Tolias, *Three types of incremental learning*, Nature Machine Intelligence, **4** (2022), 1185–1197. https://doi.org/10.1038/s42256-022-00568-3

[39] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, *SciPy 1.0: Fundamental algorithms for scientific computing in Python*, Nature Methods, **17** (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[40] Y. Wu, X. Yang, H. Yan, M. Chadli, Y. Wang, *Adaptive fuzzy event-triggered sliding mode control for uncertain euler-lagrange systems with performance specifications*, IEEE Transactions on Fuzzy Systems, **31** (2022), 1566–1579. https://doi.org/10.1109/TFUZZ.2022.3205777

[41] W. Xiao, H. Ma, L. Zhou, H. Li, *Adaptive fuzzy fixed-time formation-containment control for euler-lagrange systems*, IEEE Transactions on Fuzzy Systems, **31** (2023), 3700–3709. https://doi.org/10.1109/TFUZZ.2023.3267425

[42] X. Xie, F. Yang, L. Wan, J. Xia, K. Shi, *Enhanced local stabilization of constrained n-ts fuzzy systems with lighter computational burden*, IEEE Transactions on Fuzzy Systems, **31** (2022), 1064–1070. https://doi.org/10.1109/TFUZZ.2022.3187182

[43] L. Yang, G. Lai, Y. Chen, Z. Guo, *Online control for biped robot with incremental learning mechanism*, Applied Sciences, **11** (2021), 8599. https://doi.org/10.3390/app11188599

[44] Y. Yang, R. Li, J. Huang, X. Su, *Distributed optimal output feedback consensus control for nonlinear euler-lagrange systems under input saturation*, Journal of the Franklin Institute, **360** (2023), 5857–5877. https://doi.org/10.1016/j.jfranklin.2023.03.042

[45] Z. X. Yang, H. J. Rong, P. Angelov, Z. X. Yang, *Statistically evolving fuzzy inference system for non-gaussian noises*, IEEE Transactions on Fuzzy Systems, **30** (2021), 2649–2664. https://doi.org/10.1109/TFUZZ.2021.3090898

[46] Z. Yao, J. Yao, *Toward reliable designs of data-driven reinforcement learning tracking control for euler–lagrange systems*, Neural Networks, **153** (2022), 564–575. https://doi.org/10.1016/j.neunet.2022.05.017

[47] F. Zhang, Y. Y. Chen, Y. Zhang, *Finite-time event-triggered containment control of multiple euler–lagrange systems with unknown control coefficients*, Journal of the Franklin Institute, **360** (2023), 777–791. https://doi.org/10.1016/j.jfranklin.2022.11.043

[48] Y. Zhang, C. Hua, K. Li, *Disturbance observer-based fixed-time prescribed performance tracking control for robotic manipulator*, International Journal of Systems Science, **50** (2019), 2437–2448. https://doi.org/10.1080/00207721.2019.1622818

[49] D. W. Zhou, Q. W. Wang, Z. H. Qi, H. J. Ye, D. C. Zhan, Z. Liu, *Class-incremental learning: A survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **46** (2024), 9851–9873. https://doi.org/10.1109/TPAMI.2024.3429383