ISA Transactions 178 (2026) 333–347

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatra

Research article

Mitigating exposure bias for reliable ANN-based control: The enhanced scheduled sampling framework

Pau Comas*, Antoni Morell, Jose Lopez Vicario, Ramon Vilanova

IDEAS Research Group, Universitat Autónoma de Barcelona (UAB) Campus UAB, 08193 Bellaterra, Barcelona, Spain

HIGHLIGHTS

- The Enhanced Scheduled Sampling (ESS) framework is presented to address exposure bias when creating ANN-based feedback controllers.
- The Scheduled Sampling curriculum learning technique is applied to control, with stability and convergence enhancements in an online, closed-loop setting.
- Validation on FOPDT and a non-linear CSTR system with second-order dynamics shows the framework's versatility.
- ESS-trained controllers achieve superior performance and robustness compared to conventional Batch Learning.
- The open-source implementation of ESS is shared to encourage further research.

ARTICLE INFO

Keywords:
Artificial neural networks
Industrial control
Scheduled sampling
Exposure bias
PID controller

ABSTRACT

The design of Artificial Neural Networks (ANNs) as controllers for industrial processes offers significant potential but is often hindered by the challenge of exposure bias. Classic training methods, such as Batch Learning (BL), can suffer from it when trained controllers are deployed in closed-loop systems. This paper introduces the Enhanced Scheduled Sampling (ESS) framework, a novel and comprehensive training methodology designed to mitigate these limitations and produce reliable ANN-based controllers. Based on the Scheduled Sampling curriculum learning strategy, the ESS framework integrates an online simulation environment with key enhancements such as Weighted ANN Actuations to blend controller outputs, a sequence-based control error term and a controlled dynamic learning rate adjustment strategy. The framework is validated against the baseline learning approach on a conventional PID control scenario acting on First-Order Plus Dead-Time processes as well as on a non-linear Continuous Stirred-Tank Reactor exhibiting second-order dynamics. The PID benchmark provides an industrially relevant reference, enabling the proposed ANN-based controller to be compared with the established standard for process-level feedback control. The results demonstrate that controllers trained with ESS consistently outperform their BL counterparts in both control performance and robustness to parametric variations and training initialisation. By effectively mitigating exposure bias, the ESS framework represents a significant step towards developing dependable, high-performance ANN controllers ready for real-world industrial deployment or to advance research in transfer learning for control. The open-source implementation is shared to encourage further research.

1. Introduction

Industrial Control Systems are integral to modern manufacturing and production, encompassing critical control mechanisms designed to automate and manage industrial operations, thereby ensuring the efficiency, safety, and reliability of process industry plants across diverse sectors. Within this landscape, the Proportional-Integral-Derivative (PID) controller has long been a cornerstone due to its simplicity, robustness, and broad applicability. It is successfully employed in numerous applications, from temperature and pressure regulation to advanced motion control in robotics and speed control in motor drives [1]. Well-established tuning methodologies have contributed to its widespread adoption. This has led to feedback control solutions even where a precise process model is unavailable [2–4]. However, the reliance of PID controllers on accurate tuning and often on a good identification

* Corresponding author.
Email addresses: pau.comas@uab.cat (P. Comas), antoni.morell@uab.cat (A. Morell), jose.vicario@uab.cat (J. Lopez Vicario), ramon.vilanova@uab.cat (R. Vilanova).

https://doi.org/10.1016/j.isatra.2026.09.013
Received 29 September 2025; Received in revised form 8 September 2026; Accepted 8 September 2026
Available online 9 September 2026
0019-0578/© 2026 The Author(s). Published by Elsevier Ltd on behalf of International Society of Automation. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

of process dynamics can present challenges, particularly in environments characterised by significant complexity or rapid changes in the conditions.

In the preceding years, the application of Artificial Neural Networks (ANNs) within control theory has gained considerable research interest. Putting ANNs to operate alongside, supplement, or even replace conventional controllers such as PIDs has opened new paths for enhancing control system efficiency, adaptability, and overall performance [5–8]. One of the main concerning scenarios causing the surge of ANN-based solutions is one where there is a lack of substantial expertise and a deep understanding of the underlying process dynamics. Consequently, there is a growing conviction that ANNs, when combined with appropriate data-driven training techniques, could simplify this complexity [9]. These networks can be trained to learn desired control behaviours, leveraging the inherent flexibility and adaptive capabilities of neural network architectures. On the other hand, data-based training carries the risk of not generalising well to unseen situations not covered by the training. Concerns about fully autonomous ANN controllers have motivated extensive work on hybrid uses that support conventional loops, for instance, in the intelligent tuning of PID parameters [2], or in the development of "soft sensors" to estimate critical process variables [7]. Therefore, the direct modelling and deployment of ANNs as standalone replacements for PID controllers have received comparatively less, though increasing, attention [10,11].

The development of direct ANN-based controllers could bring several potential advantages. Such controllers can be computationally lightweight, making them suitable for deployment on resource-constrained hardware. In addition, the adaptive nature of ANNs opens the door to controllers that can learn from ongoing operational data and adjust their behaviour in response to changing process conditions, unexpected disturbances, unmodelled dynamics, etc. Robust, autonomous ANN-based replicas of PID controllers would provide a foundation for applying Transfer Learning (TL) in the industrial control domain. TL involves using the knowledge gained from one task to improve performance or reduce learning time on a related but different task [12], which could dramatically reduce engineering effort and time required to commission new control systems or reconfigure existing ones in diverse industrial plants or processes. It is frequent to encounter plants with numerous control loops that could take advantage of simplified controller transferability. Related to that, the use of transferability metrics [13,14] researches the ability to compare candidate networks for adapting or fine-tuning to new tasks. Said metrics could allow for choosing between existing ANN-based controllers for new control scenarios. The ability to reliably create stable and accurate ANN-based controller models is thus a critical prerequisite for advancing TL investigations in control engineering.

The practical implementation of direct ANN-based controllers faces substantial challenges. A primary obstacle is the phenomenon known as "exposure bias". This term, as defined by Ranzato et al. [15] in the context of sequence generation, refers to a

> "behavioural mismatch between maximum likelihood-based training and generation procedure. During maximum likelihood-based training, the next token distribution is conditioned on ground truth data whereas, during generation, it has to rely on the model's own previously generated tokens".

In that sense, autoregressive models are often trained to predict the next element of a sequence based on a history of correct, ground-truth elements. At inference, however, these models must generate subsequent elements based on the history of *their own* previous predictions [16].

This discrepancy is particularly critical in closed-loop control systems, where complexity arises. When training an ANN-based controller, it might learn to output a control signal $u(t)$ based on past process states $y(t-1), y(t-2), \ldots$ and ideal past control actions $u_{\text{ideal}}(t-1), u_{\text{ideal}}(t-2), \ldots$ that are provided during the training phase. This mirrors the 'teacher-forcing' paradigm, where the model is guided by correct historical data. This training methodology can be referred as Batch Learning (BL) or offline training, which comprises parallel processing and backpropagation on fixed datasets with epoch-based progress. Upon deployment in an autonomous control loop, the controller's own output $u(t)$ directly influences the future process state $y(t+1)$. The controller must then determine the next control action $u(t+1)$ based on the current state $y(t)$ and its own past action $u(t)$, without reference to any external 'ideal' or 'ground-truth' actions from the immediate past. If an emitted control action $u(t)$ is to some degree suboptimal, the subsequent process state $y(t+1)$ will deviate from the ideal trajectory, encountering new situations in inference and difficulties during generalisation. This deviation then forms the input for the next control decision, creating a recursive loop where initial errors can compound. This potential where the model's errors influence future inputs and lead to further errors, is the core concern stemming from exposure bias in control applications, potentially leading to performance degradation or even instability of the control system [9].

To address the challenge of exposure bias, the Scheduled Sampling (SS) technique offers a compelling strategy. Originally proposed for a text-to-sequence challenge [17], SS aims to mitigate the discrepancy between training and inference by gradually transitioning the model from a 'teacher-forcing' regime to conditioning on its own previously generated outputs. The exposure bias in sequence generation mirrors the challenges in ANN-based control loops, and similar deep learning applications, where the ANN controller model must handle its own errors during autonomous operation [18–20]. Previous work on the challenge of adapting SS for ANN-based PID control [9] has proposed the creation of a specific framework, tested on systems that can be described in terms of First-Order Plus Dead Time (FOPDT) models. It also outlines the need to further evaluate the framework and to deal with potential challenges when applied to more complex systems.

This paper presents the complete and comprehensive development, methodological underpinnings, and validity analysis of the Enhanced Scheduled Sampling (ESS) framework, which is the corresponding evolved training framework. This work presents a novel, consolidated and efficient training methodology specifically tailored for control applications, enabling neural networks to perform reliably in closed-loop settings. Its architecture, operational pipeline, and design rationale are detailed, reflecting the evolutionary decision-making process that culminated in this current refined form. The ESS implementation resources are available at [21] to foster discussion and collaborative research.

The remainder of this paper is organised as follows. Section 2 provides background on the Scheduled Sampling technique, while Section 3 describes the methodology. Section 4 details the proposed Enhanced Scheduled Sampling (ESS) framework. Section 5 presents and analyses the results. Finally, Section 6 discusses the findings, future work, and concludes the paper.

2. Background: scheduled sampling

Deploying ANNs as direct controllers hinges on addressing the discrepancies between their training environments and their operational realities during inference (exposure bias), as in many autoregressive tasks. Scheduled Sampling (SS), a curriculum learning strategy proposed by Bengio et al. [17], emerged as a technique to tackle this very issue, primarily in the domain of sequence generation tasks such as machine translation and language modelling.

2.1. The scheduled sampling mechanism

In a standard 'teacher-forcing' setup in Recurrent Neural Networks (RNNs) or input-to-sequence scenarios, to predict the t-th element of a sequence, y_t, the model is always provided with the ground-truth $(t-1)$-th element, y_{t-1}, from the training data. During autonomous generation, the model must rely on its own previously generated element, $\hat{y}_{t-1}$. A single error in this chain can lead the model into unobserved parts of the state space, potentially causing a cascade of subsequent errors and

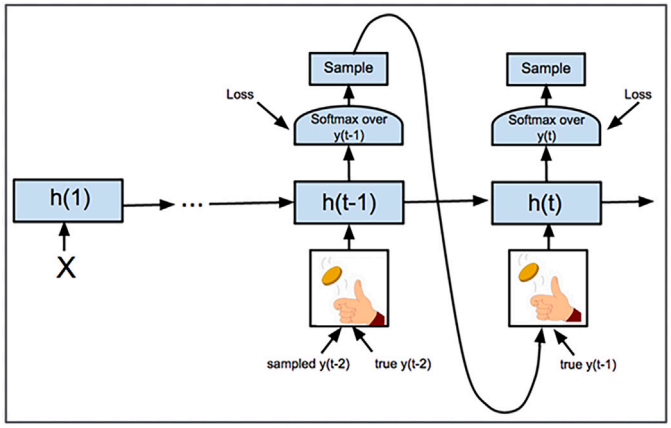


Fig. 1. Illustration of the scheduled sampling approach [17].

a significant drop in output quality [15]. In discrete state-based models such as Hidden Markov Models, where dynamic programming can be used, beam search to generate several candidate sequences can help. However, it is harder to use efficiently for continuous state-based models such as RNNs.

Scheduled Sampling introduces a simple yet effective modification to this training procedure for text generation (Fig. 1). Instead of deterministically feeding y_{t-1}, at each generation step t, a stochastic decision is made: the model receives either the ground-truth word y_{t-1} or its own previous prediction $\hat{y}_{t-1}$ as input. This choice is governed by a biased 'coin flip' probability, ϵ_t, which is scheduled to change as training progresses (where t typically denotes the training iteration or epoch). Initially, ϵ_t is high (e.g., 1), ensuring full 'teacher-forcing'. As training advances, ϵ_t is gradually decreased – this can follow various decay functions (e.g., linear, exponential, inverse sigmoid) – thus increasing the frequency with which the model is exposed to its own outputs and enabling exploration of other input subspaces. The central aim, as articulated in [9], is to "encourage the model to learn to cope with errors, leading to improved generalisation performance during inference". This creates a curriculum where the model gradually adapts its behaviour to the scenarios it will face when deployed.

2.2. Context, strengths, and limitations

The original SS formulation demonstrated significant improvements on challenging sequence generation tasks, such as image captioning (MSCOCO dataset [22]) and machine translation [17]. Despite these empirical successes, SS sparked critical discussion regarding its theoretical properties. Huszár [23] raised questions about the consistency of the learning objective, arguing that SS yields a biased estimator, in that even as the number of examples and the capacity go to infinity, the procedure may not converge to the correct model. Still, experiments with SS clearly showed some improvements in terms of the robustness of the generated sequences. More recently, Korakakis et al. [24] investigated how SS interacts with catastrophic forgetting in continual learning settings, finding that while SS can ameliorate exposure bias, it could sometimes worsen performance when prefixes at inference were correct. That led to the proposal of elastic weight consolidation as a form of regularisation.

In current deep learning, SS is less prominent in large-scale, data-intensive settings, but remains useful in specific domains. It has been adapted to Transformer models [25] and applied to semantics-assisted video captioning with recurrent architectures [26], supporting its relevance for lightweight models and moderate-data regimes.

One notable alternative is Professor Forcing (PF) by Lamb et al. [27], which incorporates an adversarial training component. PF could offer advantages in learning global sequence properties, whilst SS provides a more straightforward and computationally lighter mechanism. Another alternative is Reinforcement Learning (RL) [28], which can achieve powerful long-term solutions by directly optimising a task-level objective, but often requires substantially more state-space coverage in training, while leaving stability-wise concerns unanswered in many cases. The acknowledged limitations and alternative methods serve to refine the SS application and motivate domain-specific enhancements, such as those developed in the framework presented in this paper.

2.3. Scheduled sampling applied to control systems

Adapting the SS mechanism from the text-generation context to the feedback control systems domain requires a shift from a static, offline setting to a dynamic, online simulation environment. This involves integrating an implementation of a baseline controller (e.g., a PID) alongside the ANN in the same environment, with both interacting with the process model [14], as illustrated in Fig. 2.

The training process is built around the dynamic generation of data. At each time step, the stochastic 'coin flip' (governed by the ϵ_t probability) decides the source of the control action fed to the process. At the beginning, the probability is set to favour the baseline PID controller. In this phase, the PID not only provides the control signal that drives the process but also generates the ground-truth target labels for the ANN's learning objective, which is done throughout the whole training process.

As training progresses, ϵ_t probability is gradually adjusted according to the decay schedule. This increases the likelihood that the ANN's output controls the process in the coin flip. When the ANN's action is selected, it acts as the controller for generating the next process state and, therefore, the input vector that the ANN will receive. The baseline PID continues to run in parallel, providing the ideal target action for the loss

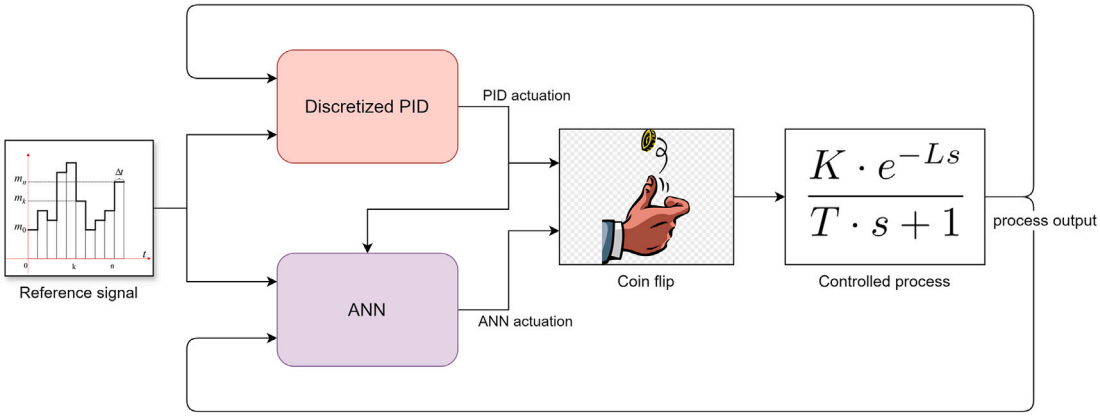


Fig. 2. Conceptual diagram of the SS methodology for an FOPDT control loop [9].

calculation, regardless of which controller's action was actually applied to the process.

Consequently, the ANN is progressively exposed to the consequences of its own actions, learning to adapt and respond to a wider range of control scenarios than would be possible with a static, historical dataset. The duration of every training epoch is determined by the length of every simulation run, which follows the predefined trajectory of setpoint signals.

3. Methodology: control processes and controller design

Here, we establish the foundational elements for developing the Enhanced Scheduled Sampling (ESS) framework. We detail the benchmark process models, baseline controllers, and ANN architectures that were systematically evaluated throughout our research journey, from initial investigations with simpler first-order linear systems to the more challenging non-linear, second-order application. Finally, we demonstrate the limitations of conventional training for ANNs in control, illustrating the core problem that motivated the ESS innovations.

3.1. Benchmark process models and baseline PID control

3.1.1. First-order plus dead time (FOPDT) systems

First-Order Plus Dead Time (FOPDT) processes represent a fundamental class of dynamic systems widely encountered in industrial control applications. The transfer function characterising the FOPDT family is as follows:

$$P(s) = \frac{K \cdot e^{-Ls}}{T \cdot s + 1} \tag{1}$$

where K represents the process gain, T the time constant, and L the dead time. This simple yet effective model serves as an approximation for many real-world industrial processes, from temperature control in heat exchangers to level control in storage tanks [29].

For our approach, a set of 10 different FOPDT processes is considered, each characterised by varying normalised dead time $\tau = L/T$, spanning the range $\tau \in [0.11, 2.0]$ [9]. The specific processes are generated by systematically varying the dead time L from 0.13 to 2.40 min while maintaining constant gain $K = 1.2$ and time constant $T = 1.2$ minutes. The systematic variation of the normalised dead time $\tau = L/T$ allows the evaluation of a wide range of dynamic process behaviours encountered in industry. Normalised dead time τ is widely recognised as the primary factor influencing control difficulty and closed-loop performance in FOPDT systems [30]. This approach ensures that the presented results are generalisable beyond a specific operating point and applicable across diverse process dynamics.

Tuning of baseline PID controllers for FOPDT processes is done using Alfaro and Vilanova's robust guidelines [4], employing standard continuous-time 2DOF PID controllers with an M_S robustness level of 2 to balance setpoint tracking, disturbance rejection, and robustness to process variations.

3.1.2. Second-order non-linear system

The Continuous Stirred-Tank Reactor (CSTR) system represents a benchmark with industrial relevance, particularly for testing control strategies on non-linear, higher-order processes with inverse response characteristics [14]. The specific CSTR configuration investigated in this work is based on the Van de Vusse reaction scheme detailed in [31]. The overall control system is shown in Fig. 3. The reactor dynamics are governed by the following non-linear differential equations for the concentrations of components A and B:

$$\frac{dC_A(t)}{dt} = -k_1 C_A(t) - k_3 C_A^2(t) + \frac{1}{V}(C_{Ai} - C_A(t))Q(t) \tag{2}$$

$$\frac{dC_B(t)}{dt} = k_1 C_A(t) - k_2 C_B(t) - \frac{1}{V} C_B(t) Q(t) \tag{3}$$

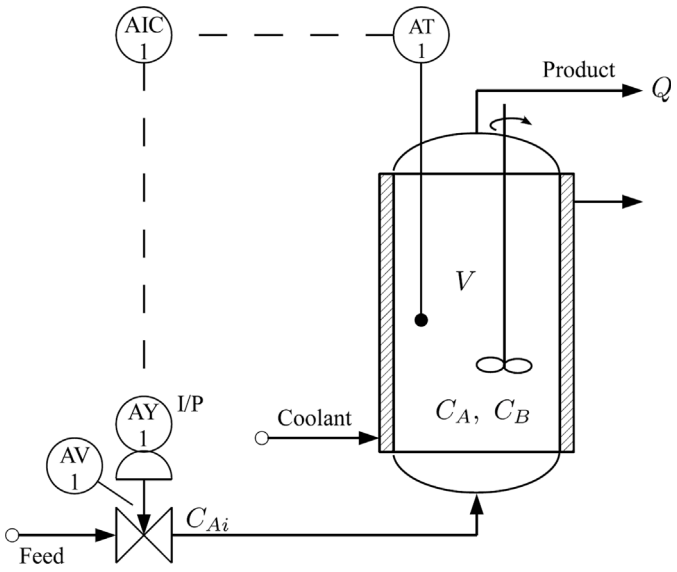


Fig. 3. Continuous stirred-tank reactor control system [31].

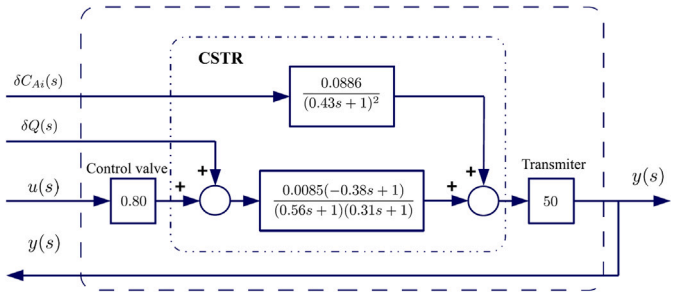


Fig. 4. CSTR linear model [31].

where C_A and C_B are the concentrations of components A and B, $k_1 = 5/6$ min^{-1}, $k_2 = 5/3$ min^{-1}, and $k_3 = 1/6$ min^{-1} are the reaction rate constants, $V = 70$ L is the reactor volume, $C_{Ai} = 10$ gmol/L is the inlet concentration of component A, and Q is the volumetric flow rate. The control objective focuses on regulating the concentration of product B ($C_B(t)$) through manipulation of $Q(t)$, while accounting for disturbances in both the inlet concentration C_{Ai} (primary disturbance) and the flow rate Q.

Linearising around the nominal operating point ($Q_0 = 40$ L/min, $C_{A0} = 3$ gmol/L, $C_{B0} = 1.117$ gmol/L) yields the following transfer functions:

$$P_u(s) = \frac{0.34(-0.38\ \mathrm{s} + 1)}{(0.56\ \mathrm{s} + 1)(0.31\ \mathrm{s} + 1)} \tag{4}$$

$$P_d(s) = \frac{4.43}{(0.43\ \mathrm{s} + 1)^2} \tag{5}$$

where $P_u(s)$, characterising the control path from flow rate Q to C_B, was identified via step response analysis specifically developed for inverse response systems [32]. Similarly, $P_d(s)$, representing the disturbance path from C_{Ai} to C_B, was derived using process reaction curve analysis [33].

Fig. 4 illustrates the overall linearised CSTR control loop, which incorporates the control valve gain $K_v = 0.80$ (L/min)/% and transmitter gain $K_t = 50\%/(\mathrm{gmol/L})$.

The baseline controller is a two-degree-of-freedom (2DOF) PID implemented as:

$$u(s) = K_p \left[\beta r(s) - y(s) + \frac{1}{T_i s}(r(s) - y(s)) - \frac{T_d s}{\gamma T_d s + 1} y(s) \right] \tag{6}$$

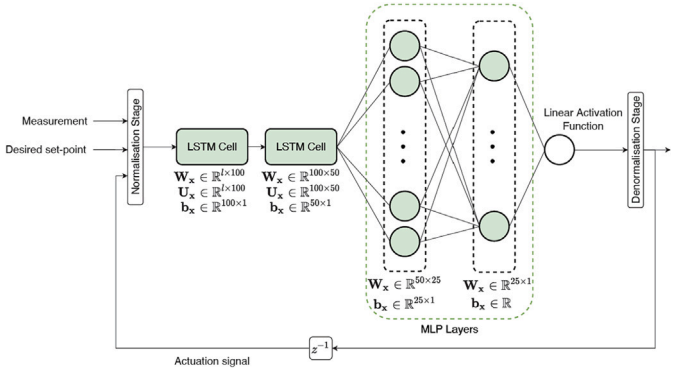


Fig. 5. LSTM_FF architecture in a three-input case.

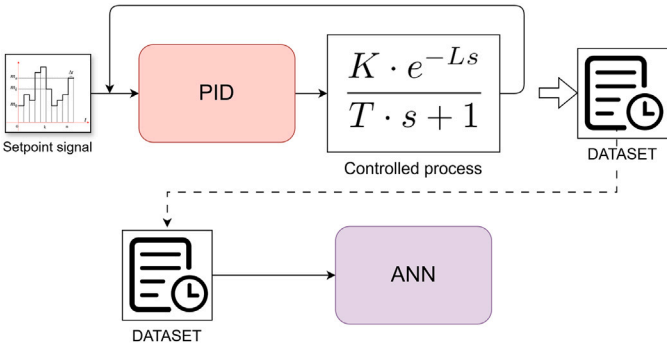


Fig. 6. Basic diagram of the batch learning (BL) training methodology, where the ANN is trained offline on a static dataset.

Parameters were tuned for $M_S = 1.6$ robustness (tighter than in previous setups) using the methodology in [31], giving $K_p = 2.372$, $T_i = 0.663$ min, $T_d = 0.162$ min, filter constant $\gamma = 0.1$, and setpoint weight $\beta = 0$. Overall, the system under control presents three key control challenges: non-minimum phase behaviour (inverse response) in $P_u(s)$, second-order disturbance dynamics with characteristics different from those of $P_u(s)$, and non-linear coupling between concentration states under large disturbances.

3.2. Artificial neural network (ANN) controller design

3.2.1. Architectures explored

Two main neural network architectures were evaluated for ANN-based PID control: the LSTM_FF network and Multi-Layer Perceptrons (MLPs).

The LSTM_FF architecture (see Fig. 5) consists of two LSTM blocks followed by two MLP layers. This hybrid design takes advantage of the ability of the LSTM to capture temporal dependencies and the non-linear mapping power of the MLP while remaining compact. LSTM cells maintain memory of past control states and process responses, which is important for emulating integral and derivative PID actions. The LSTM_FF architecture has shown good results as an ANN-based controller for PID emulation tasks, for instance, in wastewater treatment plant control loops [8].

As non-recurrent alternatives, we explored MLPs with 4 and 8 hidden layers (MLP4, MLP8), along with their Leaky ReLU variants (LeakyMLP4, LeakyMLP8) employing leaky ReLU activations instead of standard ReLU. The MLP4 architecture uses hidden layers of 100, 50, and 25 neurons, while MLP8 comprises hidden layers of 512, 256, 128, 64, 40, 30, and 20 neurons. Both architectures output a single control action value, emulating the scalar PID output. While non-recurrent networks might initially seem less suitable for control tasks with temporal dependencies, we found that with the right input design, they can be competitive with recurrent models.

Choosing between LSTM-based and MLP-based controllers depends on performance, with computational simplicity and fast inference achieved. Importantly, more complex architectures could be more sensitive to training instability, especially when the training framework is applied to challenging control systems.

3.2.2. Input vector formulation

Designing the input vector for ANN-based PID controllers requires balancing the provision of sufficient information with avoiding excessive complexity that could hinder training and generalisation. In the experiments, for simplicity and consistency, the same input vector formulation is used across both recurrent and non-recurrent architectures, with recurrent networks receiving sequences of these input vectors to capture temporal dependencies.

The candidate input variables considered throughout the project included:

- **Current Error Signal** ($e(t)$): The instantaneous difference between the setpoint and process output, from which proportional-like control actions can be derived.
- **Cumulative Error Window**: A sliding window of past error values with a specified window size (w_s), allowing the network to capture recent error trends and dynamics relevant to process time constants or delays.
- **Previous Control Action** ($u(t-1)$): The last used control output actuation, aiding in smooth control transitions.
- **Cumulative Integral Error** ($i_e(t)$): The integral of the error over time, explicitly providing information necessary to address steady-state offsets and emulate integral control action.
- **Setpoint Value** ($r(t)$): The current desired reference value, which, along with the process output, defines the error but can provide additional context.
- **Process Output** ($y(t)$): The current measured output of the process, which, along with the setpoint, defines the error signal.

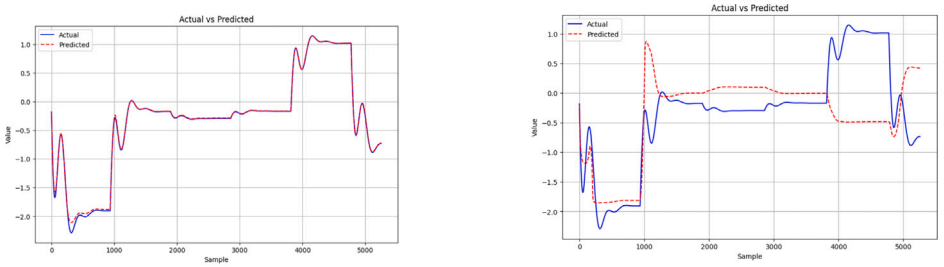
Theoretically, classical PID controllers receive the setpoint and process output to compute the error. In our ANN controllers, including the integral error input, has been found to help address steady-state error more effectively. While the setpoint and process output are highly correlated with the current error signal, adding the cumulative error window allows the network to focus on recent dynamics influenced by process characteristics such as dead time. The window size is selected to approximate the specific process settling time, ensuring the input encapsulates the complete transient response.

Our current input vector configuration comprises the current error, cumulative error window, previous control action, and cumulative integral error. However, alternative configurations have also obtained good results, particularly in simpler FOPDT systems [9].

3.3. Conventional training for ANNs: batch learning

The classical approach to training an ANN is through Batch Learning (BL), the offline methodology shown in Fig. 6. When applied to control, the ANN is trained on a static, pre-recorded dataset to minimise a loss function, typically the Mean Squared Error (MSE) for regression problems, between its output and the historical actions of a target PID controller. Since this process is computationally inexpensive, the dataset can be iterated over for many epochs to achieve a very low prediction error. While this approach is computationally efficient and can excel at 1-step-ahead prediction tasks, when complexity arises and the controllers show imperfections, it suffers in an autonomous, closed-loop setting upon deployment (exposure bias).

The consequence is illustrated in Fig. 7. The left plot shows an ANN controller, trained via BL, operating in a 1-step prediction mode where



(a) BL ANN performance in 1-step prediction mode.

(b) Same ANN in autonomous closed-loop operation.

Fig. 7. Illustration of the effect of exposure bias. ANN controller outputs (predicted) against baseline PID actions (actual). The ANN trained by BL appears to perform well when receiving known control states at each step (a), but its performance degrades and becomes unstable when it must rely on its own outputs in autonomous operation (b).

it receives the ground-truth history at each step. In this context, the ANN appears to be close to the target PID's actuation. However, the right plot reveals the same controller's performance when operating autonomously under the same conditions. It can be seen that its errors rapidly compound, leading to erratic and unstable control behaviour.

This failure demonstrates that minimising prediction error with BL does not guarantee generalisation and effective closed-loop control performance. This challenge motivates the development of the Enhanced Scheduled Sampling (ESS) framework, which is detailed in the following section.

4. The enhanced scheduled sampling (ESS) framework

ESS extends the SS approach previously applied to FOPDT processes [9] to address the training instability and convergence difficulties encountered in the CSTR application. ESS balances several key innovations, control-specific decisions, and configurable components to make it a stable and effective learning tool for ANN-based controllers on complex control tasks. The code for the framework implementation is shared with the community at [21]. We present the framework's architecture (Section 4.1), detail the development journey with the fixes (Section 4.2), and outline the three critical enhancements—Weighted ANN Actuations (WANNA), the chosen Dynamic Learning Rate strategy, and the Augmented Loss function that differentiates ESS from the standard literature (Section 4.3).

4.1. Core pipeline architecture

The ESS framework is built with these interconnected core elements that allow the SS basis to adapt to an online control simulation environment:

- **Process**: A discrete-time implementation of the plant model (e.g., FOPDT or CSTR), coded in Python using state-space representations with the *control* library [34]. This block receives the control action and any relevant disturbance trajectories.
- **Controller**: A discrete-time implementation of the baseline controller (e.g., 2DOF PID with trapezoidal integration). This serves as the ground-truth control action generator for ESS with the ANN.
- **ANN Training Block**: This represents the ANN-based controller itself, along with how we manage and process the data involved, backpropagation and learning losses.
- **Coin Flip Block**: The heart of the SS idea. At each time step, this block decides whether the process simulation receives the control action from the PID (teacher-forcing) or the ANN, based on the current SS decay probability or any criteria we impose.

Therefore, the pipeline operates in a closed loop, where the key is how we manage the ANN learning from dynamically generated data that reflect the consequences of its own actions, or online PID actuations, as opposed to static, offline BL. The proposed ESS framework diagram, when applied to the CSTR system, is depicted in Fig. 8.

4.2. Framework development

The journey from a basic SS adaptation to the final ESS framework was iterative. Specific challenges encountered during development prompted the methodological refinements and enhancements detailed below.

4.2.1. Simulation environment and training structure

The first challenge was adapting the concept of a training "epoch" to a continuous, online simulation in which the ground truth is dynamically generated. We decided to tackle this with the notion of artificial epochs, defined by a fixed number of simulation steps based on the historical data dimensions. This choice is important for structured training, validation progression, and managing events such as learning rate adjustments or the option of early stopping.

The entire framework was developed in Python for flexibility and speed, given that co-execution with MATLAB/Simulink was proven to be rigid and slow even in the simplest case. The ESS framework is a full Python implementation with exported models and results are subsequently validated in Simulink to ensure fidelity. Scalers are also exported, and data standardisation is chosen over min-max scaling to handle inputs in our case, as it is less sensitive to outliers that can occur during unstable training phases, providing more robust performance.

4.2.2. Evolution of the loss function and validation strategy

The methods for validating the model and guiding its learning have significantly evolved. In initial formulations, model validation relied on 1-step-ahead prediction loss, as it was computationally simple and provided a good indicator of initial convergence. However, it was acknowledged that this metric loses value as the model becomes more autonomous; a model can be excellent at 1-step prediction but fail in a closed loop due to exposure bias. This led to a shift towards using pure simulation-based validation, where the ANN controls the process autonomously for predefined sequences of setpoints and, if applicable, disturbances or external effects. While this is the true measure of a controller's competence, it is often too unstable to be a reliable metric in the early or medium stages of ESS. The final framework thus allows for a hybrid approach, where a 1-step validation loss can guide early-stage learning, but final model selection is based on performance in pure simulation mode.

The loss function also underwent refinement. The baseline was a Mean Squared Error (MSE) between the ANN and PID outputs. To address observed steady-state errors, the loss was augmented with terms for mean integral or absolute errors. In posterior formulations, to address stability issues, a smoothing loss penalising rapid output changes

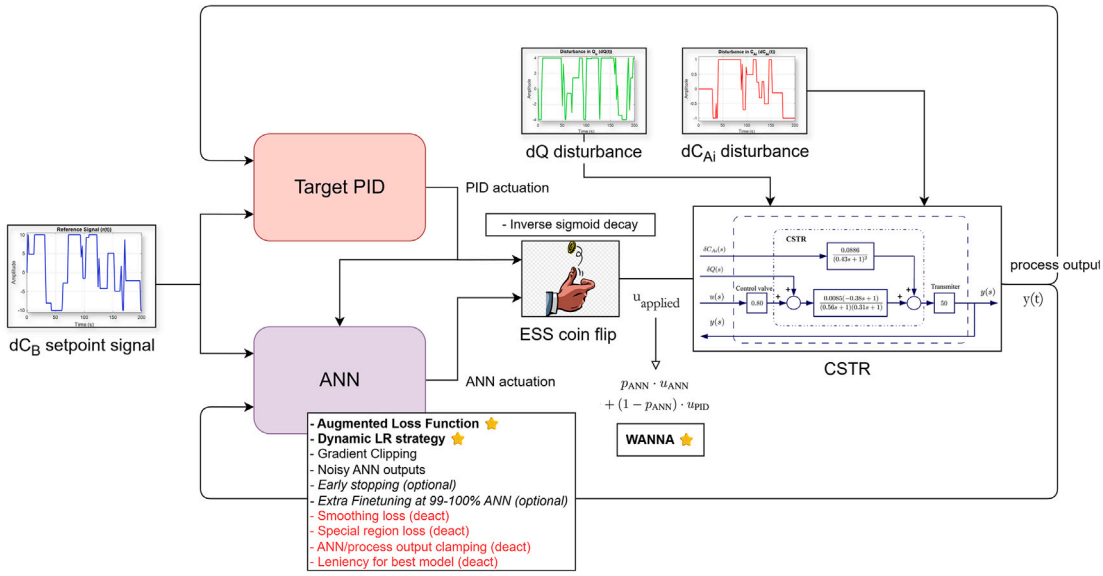


Fig. 8. Proposed strategy block diagram for ESS on CSTR, along with the list of activated, optional and deactivated (red) enhancements within the framework. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

was tested. While these offered marginal benefits, the most effective solution was to incorporate a sequence-based control error term alongside the MSE loss, which directly penalises deviations from the regulatory objective over a specified time window. While loss function engineering is a well-known practice, this specific adaptation was critical for this control application: it allowed us to mitigate those steady-state errors in late training phases and align the optimisation objective with closed-loop performance. Finally, the explored idea of a special loss penalty for unstable training regions was ultimately deactivated following final ESS enhancements.

4.2.3. Mechanisms for stability and convergence

The application of basic SS adaptations to the CSTR system, with its significantly more complex dynamics, immediately highlighted critical methodological challenges. The core issue revolves around ensuring training stability and convergence throughout the SS training run. For the CSTR, increasing ANN autonomy under SS frequently led to instability (e.g., exploding gradients, divergent outputs), particularly if the ANN was underfitted or the training regimen too aggressive, risking ANN weight corruption. While the intra-epoch randomness in selecting between ANN and PID is crucial for generating dynamic training data and enhancing the ANN's resilience to its own errors, this stochasticity also introduced a significant risk: premature ANN actions could disrupt vital parts of the training, potentially undermining the entire learning session. While initial measures – gradient clipping, output or ANN clamping, and a smoothing loss – enabled the ANN to reach the later SS phases (90–95% of control actuations generated by the ANN), a critical limitation emerged: *the ANN often remained over-reliant on PID corrections in the final 95–100% SS phase*, indicating it was not yet in a good position to operate independently.

To address the issue, the learning rate (LR) strategy became a critical point in our final stages of development. For simple FOPDT processes, a fixed or simple decay LR is functional. However, for the CSTR, the convergence and stability challenges demanded a more careful approach. Various strategies were tested, including progressively reducing the LR on training loss plateaus. A key difficulty was deciding whether to base the LR on training loss or validation loss, as the simulation-based validation loss is unreliable at the start. The strategy that proved most effective, and is a core component of ESS, is a dynamic, stage-based LR adjustment. The LR starts higher in the early/middle stages of SS and is adjusted based on training loss, then in the late and optional fine-tuning phases, it is guided by the simulation-based validation loss, all within controlled boundaries. Instead of a single metric, we drive early-stage learning with 1-step prediction loss (on training data) and switch to closed-loop simulation loss (on validation data) only when the network becomes capable of autonomous operation. This strategy is highly customisable within the framework.

In parallel with dynamic LR adjustments, another crucial step for ensuring stability and effective convergence was the introduction of Weighted ANN Actuations (WANNA). Instead of a binary switch between the PID and ANN outputs during the SS schedule, this mechanism blends them. When the "coin flip" selects the ANN, the applied control action is a weighted sum of the ANN's and the PID's outputs, with the ANN's weight increasing throughout training. This provides a much smoother transition as the ANN takes on more control, mitigating the impact of premature or inaccurate actions that could otherwise destabilise the entire training run. The effectiveness of WANNA allowed for the deactivation of other "brute-force" stability measures like clamping ANN and process outputs, which masked underlying issues. As a standard safeguard, gradient clipping remains a default feature to prevent exploding gradients.

4.2.4. Training regime and model selection

The configuration of the training schedule itself provides further flexibility. The choice of the decay function for the SS schedule directs the training stages. A linear decay is effective for simpler processes, but for the CSTR, an inverse sigmoid decay proved to be more suitable. This schedule naturally allocates more training time to the critical early (0–10%) and late (95–100%) phases of the ANN actuation, allowing more time for initial stabilisation and final refinement. Besides, we found that additional performance could be gained by adding a fine-tuning phase after the regular training completes. In other words, it is difficult to infer the ideal training length, so this phase involves continuing training with the ANN operating at 99% or 100% autonomy, often with small LR and a small amount of noise added to the ANN outputs to encourage further exploration and robustness. Choosing 99% allows for a small degree of stochasticity to persist in an already low-effect fine-tuning phase. Finally, the strategy for selecting the best model evolved. An early approach entailed leniency, favouring models that successfully reached later, more autonomous stages of the SS schedule, even if their 1-step prediction loss was slightly worse. This was later replaced by a stricter policy that prioritises performance in full simulation mode as the

Table 1
Control performance metrics over the FOPDT processes.

Controller	Metric	τ (L/T)									
		0.11	0.31	0.52	0.74	0.95	1.16	1.37	1.58	1.79	2.00
PID	IAE	5.55	14.72	23.24	30.86	37.74	44.14	50.58	56.97	63.18	69.27
	ISE	1.59	5.47	9.90	14.18	18.22	22.10	26.01	30.09	34.41	38.82
Scheduled Sampling (SS)	IAE	**8.02**	**17.18**	**25.16**	**32.63**	**40.57**	**46.66**	**53.31**	**59.41**	**65.94**	**72.82**
	ISE	**2.01**	**6.51**	**11.27**	**15.96**	**20.59**	**24.95**	**29.46**	**33.44**	**36.77**	**43.19**
	Avg Best Epoch	29	30	32	30	28	30	29	31	27	30
	IAE% Improv. vs BL	6.64	33.67	25.91	30.40	25.42	21.51	17.18	12.19	22.55	22.28
Batch Learning ANN (BL)	IAE	8.59	25.90	33.96	46.88	54.40	59.45	64.37	67.66	85.14	93.70
	ISE	2.26	7.98	13.11	19.44	24.22	28.49	34.58	36.13	45.49	51.63
	Avg Best Epoch	91	83	75	77	69	80	73	84	69	78

primary criterion for model competency, assuming that the model has reached the late stage of the training regimen.

4.3. Definition of key ESS enhancements

In summary, the effectiveness of the ESS framework in addressing the challenges of training ANN-based controllers for complex systems comes from three main modifications, which work together to deliver the necessary stability and performance:

Weighted ANN actuation (WANNA). Instead of the binary coin flip used in original SS, WANNA ensures training stability by blending the ANN's output (u_{ANN}) with the PID's (u_{PID}) using a scheduled weight (p_{ANN}) that evolves with the training phase. By default, $p_{\text{ANN}}(e) = e/N_{\text{epochs}}$, where e is the current epoch and N_{epochs} is the total training duration. This continuous mixing mechanism mitigates the impact of premature or erratic ANN actions that can destabilise training, preserving valid gradient information by avoiding the use of hard clamping and its value loss, ensuring that the applied control action, u_{applied}, becomes a smooth transition rather than a binary switch:

$$u_{\text{applied}} = p_{\text{ANN}} \cdot u_{\text{ANN}} + (1 - p_{\text{ANN}}) \cdot u_{\text{PID}} \tag{7}$$

Dynamic learning rate (LR) strategy. A multi-stage optimisation strategy that ensures robust convergence by switching the guiding metric based on the training phase, which is a crucial step for navigating complex optimisation landscapes. The strategy is twofold:

- *Early/Middle Stages:* The LR is higher and exclusively guided by training loss (or 1-step prediction variations) to allow for faster initial learning, avoiding the instability of closed-loop metrics in undeveloped models.
- *Late/Fine-tuning Stages:* The LR is reduced and switches to track the simulation-based validation loss. This ensures that the final convergence is driven by closed-loop performance.

While optimal ranges may vary, default stability bounds for the learning rate are generally on the order of 10^{-3} for the initial phase, 10^{-4} for intermediate and late phases, and 10^{-5} in case of fine-tuning.

Augmented loss function. A composite objective function improves control performance fidelity by complementing the standard Mean Squared Error ($\mathcal{L}_{\text{MSE}}$) between ANN and PID outputs with a weighted (λ) sequence-based control error term ($\mathcal{L}_{\text{seq_loss}}$). This introduces a more balanced learning signal related to actual control performance, beyond simple output mimicry, which helps to address late training steady-state offsets:

$$\mathcal{L}_{\text{ESS}} = \mathcal{L}_{\text{MSE}} + \lambda \mathcal{L}_{\text{seq_loss}} \tag{8}$$

5. Results

This section evaluates the proposed ESS framework. The analysis is structured into two main parts. First, we assess the core control performance, beginning with the foundational results on FOPDT systems to establish the SS method's efficacy, followed by a detailed investigation of the more complex CSTR system. Second, we evaluate the robustness of the framework from two critical perspectives: Parametric Robustness, where we analyse performance degradation when faced with plant-model mismatch, and Training Robustness, which assesses the consistency of the framework's training outcome across multiple runs. In the analysis, performance metrics (IAE, ISE) act as the primary indicators of exposure bias mitigation. A controller that achieves high stability and low error in closed-loop operation has, by definition, successfully mitigated the exposure bias that causes offline-trained agents to diverge.

5.1. Performance: FOPDT

The goal is first to validate the basic premise: that the SS approach is a more effective methodology for training ANN controllers than standard Batch Learning (BL). For this purpose, each training method was executed for 100 epochs, with each epoch comprising 35,041 generated samples, obtained via MATLAB simulations, or real samples in the case of the historical dataset. Table 1 presents the average results, consistent with prior experiments [9], over three distinct iterations for the two main training scenarios: the conventional BL, referred to as the baseline scenario, and the SS approach with the LSTM_FF model.

The conventional BL framework operates in a fully offline manner, as shown in Fig. 6. In this setup, the ANN is trained using a static dataset of inputs (process errors, cumulative errors, etc.) and their corresponding target outputs (the simulation's PID control actions). The training is organised into fast, computationally inexpensive epochs, where the model's weights are updated based on the MSE loss function, which measures the discrepancy between the ANN's predictions and the PID's historical actions. This method does not involve a live simulation; the ANN's relation with the process dynamics is captured only by the data sequences in the static dataset.

The results in Table 1 highlight a significant performance difference between the two training methodologies. The BL models consistently show higher Integral Absolute Error (IAE) and Integral Squared Error (ISE) scores compared to the SS models, indicating less effective control. While each BL epoch is computationally faster than an SS epoch (which involves simulation), achieving a competent controller with BL requires a much larger number of epochs, as seen by the Average Best Epoch (ABE) values, which are typically between 70 and 90. The ABE for the SS models is consistently around 30. This implies that BL, despite its faster epochs, ultimately requires more training iterations and data processing to converge to a solution.

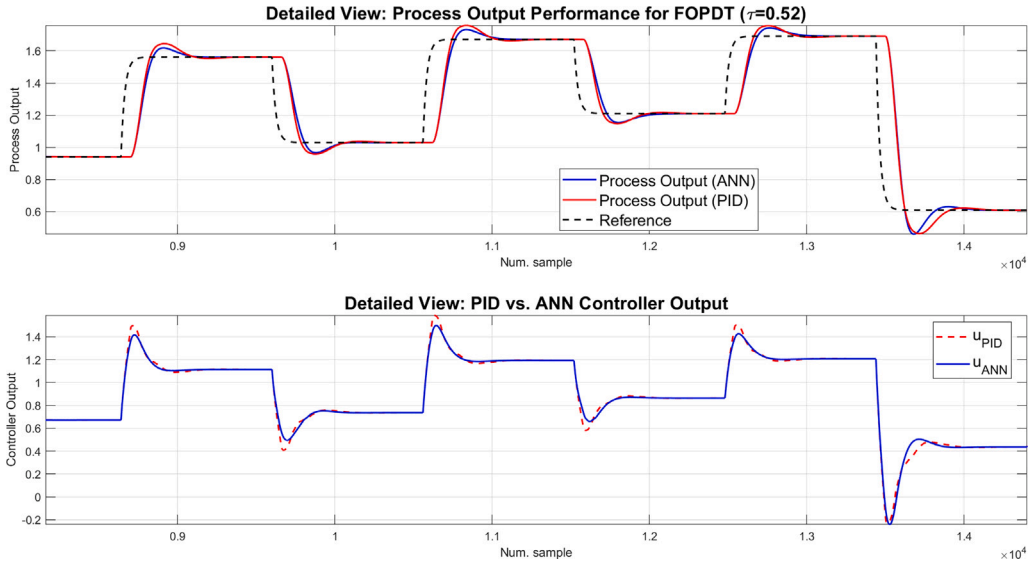


Fig. 9. SS ANN controller on $\tau = 0.52$ process loop results (samples 8500 to 15000).

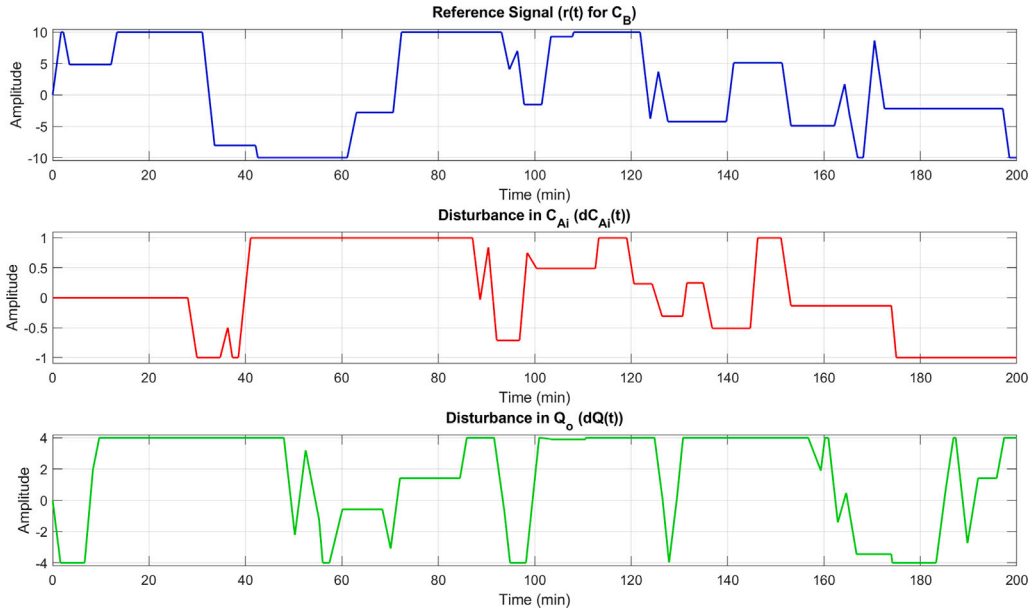


Fig. 10. Process excitation signals of the CSTR training set. The values represent deviations around the nominal operating point of the linearised system. The ranges for these excitation signals were chosen to reflect approximately ±10% variations around the nominal operating point of the CSTR, consistent with typical step change magnitudes used for control system analysis in [31] (e.G., $\delta C_B(t) \in [-10, 10]\%$, $\delta C_{A_i}(t) \in [-1, 1]\, g\ mol\ L^{-1}$, and $\delta Q(t) \in [-4, 4]\ L\ \mathrm{min}^{-1}$).

The SS framework demonstrates consistently superior performance. This is quantified in the IAE% of improvement against the BL over the ten processes, showing that the SS framework achieves substantial IAE improvements over BL, going from 6.64% to as high as 33.67%. In Fig. 9, a fraction of the results with controller `SS for` τ`=0.52` and the baseline PID is shown. Because the ABE is around 30, this suggests that for these simpler FOPDT systems, the full 100-epoch SS may not be strictly necessary, and an effective controller can be obtained at an earlier SS stage. These findings on FOPDT systems validate that the SS framework is an option to address the inefficiencies and performance limitations of conventional offline training for ANN-based controllers, showcasing convincing results in simple but frequent processes and providing a strong motivation for its application and evaluation on more challenging systems.

5.2. Performance: CSTR

The analysis now turns to the CSTR system with non-linear dynamics. The training and validation datasets were generated from a 200-minute simulation of the model under PID control, using a sample time of $dt = 0.01$ min. To ensure the data was sufficiently rich for robust training, persistently exciting signals (Fig. 10) consisting of pseudo-random ramps and holds were applied to the setpoint and disturbances, covering a wide range of frequencies and amplitudes around the nominal operating point [31,35]. The ANN controllers were subsequently trained in a Python environment using a linearised version of the CSTR model.

The performance of ANN-based controllers, trained via ESS and BL, was benchmarked against the baseline PID on the CSTR system. Table 2 summarises the key performance indicators for the three most representative models obtained from our latest tests for each methodology. The

Table 2
Comparative performance of ESS and BL controllers on the CSTR with multiple validation metrics.

Controller	ANN Model	IAE (Train)	IAE (Val)	Train Loss	Val Loss (Pred)	Val Loss (Sim)	Val Loss (Hist)
Baseline PID	—	320	245	—	—	—	—
ESS (Enhanced Scheduled Sampling)							
ESS (308e)	MLP8	358	272	2.0×10^{-4}	1.97×10^{-3}	9.32×10^{-2}	1.26×10^{-2}
ESS (308e)	LeakyMLP4	345	272	3.2×10^{-3}	8.70×10^{-4}	1.41×10^{-1}	7.40×10^{-3}
ESS + FineT	LSTM (ws = 500)	352	284	1.0×10^{-3}	8.90×10^{-3}	1.17×10^{-1}	3.00×10^{-2}
BL (Batch Learning/Offline Training)							
BL (1000e)	LeakyMLP4	383	286	1.7×10^{-3}	1.30×10^{-3}	1.58×10^{-1}	2.60×10^{-2}
BL (15e)	MLP8	399	343	3.0×10^{-3}	3.60×10^{-4}	2.74×10^{-1}	3.50×10^{-2}
BL (1000e)	LSTM (ws = 15)	1464	2994	7.71×10^{-1}	1.40×10^{-4}	4.04	5.17

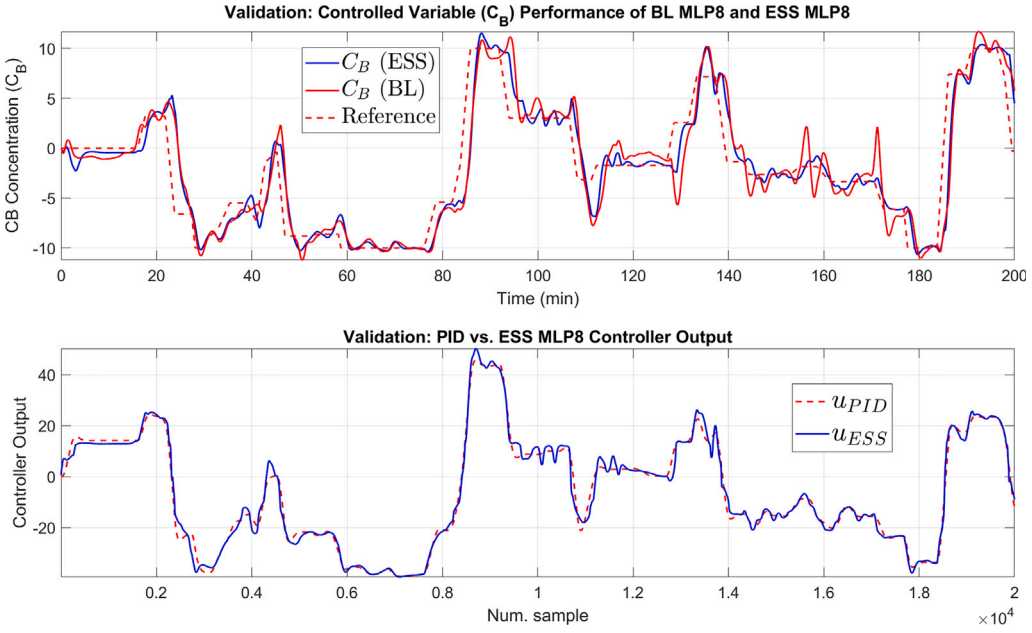


Fig. 11. ESS MLP8 controller results on the validation set.

BL epoch takes around 0.5–1 s, while the epoch for the ESS takes 40–60 s, due to the live simulations. BL models' epoch limit is set to 1000, but usually the best model is achieved and selected well before. The primary metric for control performance is IAE on the validation dataset. To assess the learning process, we report several validation losses: `Val Loss (Pred)` for 1-step-ahead prediction accuracy (mimicking the PID given a perfect history), `Val Loss (Sim)` for autonomous closed-loop simulation where the ANN is compared to a live PID, and `Val Loss (Hist)` where the ANN is compared to the historical PID actions from the original validation data.

The BL models show mixed but ultimately inferior results. While it is possible to achieve a reasonable IAE score of 286 with the `BL (1000e) LeakyMLP4` model through extensive offline training, this result is outperformed by the best ESS models and the benchmark. Furthermore, other BL models showcase the typical pitfalls of this approach: the `BL (15e) MLP8` demonstrates how stopping training early to prevent overfitting can result in a significantly underfitted model (IAE 343), while the `BL (1000e) LSTM` highlights the danger of exposure bias, where an excellent 1-step prediction score (1.40×10^{-4}) leads to catastrophic failure in autonomous control (IAE 2994).

In contrast, controllers trained with the ESS framework consistently demonstrated superior robust performance. The top-performing models, `ESS MLP8` and `ESS LeakyMLP4`, achieved validation IAE scores of 272, closer to the PID benchmark of 245. The control actuations of the `ESS MLP8` controller compared against the BL counterpart are shown in Fig. 11. The performance of ESS models shows a strong correlation between good control (low IAE) and low simulation-based validation losses (`Val Loss (Sim)` and `Val Loss (Hist)`). For instance, the `ESS MLP8` model's low IAE is supported by its low simulation loss (9.32×10^{-2}). This shows that the ESS framework successfully trains controllers that are not only capable of mimicry but are also robust when operating in an autonomous, closed-loop fashion. The ESS methodology effectively mitigates the exposure bias that harms the BL models.

For evaluating ANN-based controllers, closed-loop simulation losses (`Val Loss (Sim)` and `Val Loss (Hist)`) are the most reliable predictors of true control capability. Optimising for 1-step-ahead prediction loss (`Val Loss (Pred)`) is misleading and often detrimental to the final control performance. These results affirm that for a complex, non-minimum phase system like the CSTR, a dynamic training regimen such as ESS is essential for developing effective and reliable ANN-based controllers. In addition, to validate the individual contributions of the ESS components, we conducted an ablation study on the LeakyMLP4 model (Table 3). The results confirm the importance of the enhancements for this complex process: WANNA is critical for preventing training divergence during the middle phase, the sequence-based error term improves performance by reducing steady-state offsets, and the dynamic LR strategy ensures stable convergence to the optimal policy.

While the CSTR serves as a strong benchmark, to offer further validation of the applicability of the framework to high-order industrial systems, ESS's performance was also evaluated on a fifth-order

Table 3
Ablation study of ESS enhancements on CSTR control performance (LeakyMLP4).

Configuration	IAE	ISE	Status	Observation
Full ESS Framework	**345.00**	**272.00**	**Converged**	**Optimal performance**
w/o Sequence Loss (L_{seq})	369.15	291.04	Converged	Steady-state offset (+7% IAE)
w/o Dynamic LR	395.60	312.80	Unstable	Fragile convergence/ Oscillation risk
w/o WANNA (p_{ANN})	–	–	Failed	Divergence (Training crash ≈ 60%)

superheated steam temperature system described by a fifth-order model using two different robust PID tuning strategies. The details of this additional study are provided in Appendix A.

5.3. Parametric robustness

A Parametric Robustness (PR) analysis is performed to assess how controllers react to changes in plant conditions or to suboptimal plant identification. Focussing on the CSTR environment, the performance of each controller was evaluated across five distinct plant variations: 1) the nominal plant, 2) a −5% change in process gain, 3) a −5% change in process dynamics (faster plant), 4) a +10% gain with a −10% slowest time constant, and 5) a −10% gain with a +10% slowest time constant. They were chosen to systematically test the controllers against common forms of plant-model mismatch, including simple gain and dynamic changes and more complex, combined parameter drifts that simulate more aggressive and more sluggish plant behaviours.

The results, detailed in Table 4 (training set) and Table 5 (validation set), are evaluated using three key metrics. The Mean IAE across Variations indicates the average control performance under uncertainty. The Mean Degradation quantifies the percentage drop in performance from the nominal case, with lower values being preferable. Then, the Std. Dev. of IAE measures the consistency of that performance; a lower value implies that the controller's effectiveness is more stable across different plant conditions.

The top 2 controllers trained with the ESS framework demonstrate strong overall PR, though with important distinctions between models. The `ESS LeakyMLP4` emerges as the best-performing robust model on the validation set. It achieves a Mean IAE of 273.70, the lowest among all ANN models, combined with a low standard deviation (5.66) and low degradation (+0.39%), showing a good balance of nominal performance and resilience. It is important to note that the `ESS + FineT LSTM` model, while effective under nominal conditions, shows less PR under parametric changes. Its standard deviation is higher (7.64), and its performance on the training set degrades by nearly 3%. This suggests that the extended fine-tuning, while improving its nominal IAE, may have led to a degree of overspecialisation toward the nominal plant, slightly reducing its adaptability.

The BL methodology outputs controllers with a wider and more inconsistent range of PR characteristics. On the one hand, it can produce controllers that are tentatively robust, as demonstrated by the `BL (1000e) LeakyMLP4` model. This model achieves the lowest standard deviation of all ANNs (2.46), indicating very stable performance across plant variations. However, this stability is achieved at a suboptimal performance level; its mean validation IAE of 288.41 is higher than the best ESS controllers, making it robustly weaker. Furthermore, its mean IAE degradation is above +0.6% for both training and validation, which is higher than that of the top two ESS models. This suggests a lesser adaptability to parametric changes, possibly because the BL framework allows a simpler model like the LeakyMLP4 to slightly overfit to the nominal plant conditions. On the other hand, the BL framework also produces models such as the `BL (1000e) LSTM`, which highlight the methodology's unreliability. The `BL (1000e) LSTM` model's negative mean degradation is not a sign of PR but a consequence of its catastrophic nominal failure, where some plant variations coincidentally make the control problem slightly less unmanageable. Its instability is confirmed in the training set analysis (Table 4), where its IAE standard deviation exceeds 1000.

5.4. Training robustness

Finally, we assessed the Training Robustness (TR) of our framework. This is critical for evaluating a methodology's ability to produce

Table 4
Parametric robustness analysis: controller performance across plant variations (Training set).

Controller	ANN Model	Nominal IAE (Train)	Mean IAE across Variations (Train)	Mean Degradation (Train)	Std. Dev. of IAE (Train)
Baseline PID	—	320.79	320.86	+0.02%	5.79
ESS (Enhanced Scheduled Sampling)					
ESS (308e)	LeakyMLP4	345.42	346.76	+0.39%	1.62
ESS (308e)	MLP8	358.28	358.65	+0.10%	7.80
ESS + FineT	LSTM (ws = 500)	352.47	362.89	+2.96%	13.77
BL (Batch Learning/Offline Training)					
BL (1000e)	LeakyMLP4	383.56	385.94	+0.62%	13.55
BL (15e)	MLP8	399.84	400.20	+0.09%	5.68
BL (1000e)	LSTM (ws = 15)	1464.00	2225.80	+52.04%	1007.37

Table 5
Parametric robustness analysis: controller performance across plant variations (Validation set).

Controller	ANN Model	Nominal IAE (Val)	Mean IAE across Variations (Val)	Mean Degradation (Val)	Std. Dev. of IAE (Val)
Baseline PID	—	245.95	246.18	+0.09%	5.94
ESS (Enhanced Scheduled Sampling)					
ESS (308e)	LeakyMLP4	272.65	273.70	+0.39%	5.66
ESS (308e)	MLP8	272.77	273.23	+0.17%	5.11
ESS + FineT	LSTM (ws = 500)	284.14	285.08	+0.33%	7.64
BL (Batch Learning/Offline Training)					
BL (1000e)	LeakyMLP4	286.57	288.41	+0.64%	2.46
BL (15e)	MLP8	343.56	343.63	+0.02%	5.76
BL (1000e)	LSTM (ws = 15)	2994.00	2987.80	−0.21%	232.51

Table 6
Training robustness analysis: FOPDT processes across different initial seeds.

Controller	Metric	τ (L/T)									
		0.11	0.31	0.52	0.74	0.95	1.16	1.37	1.58	1.79	2.00
PID	IAE	5.55	14.72	23.24	30.86	37.74	44.14	50.58	56.97	63.18	69.27
Scheduled Sampling (SS)	IAE	8.02	17.18	25.16	32.63	40.57	46.66	53.31	59.41	65.94	72.82
	AMAD% IAE	**1.97**	**1.54**	**0.53**	**2.44**	**0.44**	**1.44**	**1.82**	**1.17**	**1.54**	**0.79**
	AMAD% ISE	**2.32**	**0.85**	**0.39**	**0.13**	**0.85**	**0.80**	**1.73**	1.30	**1.55**	**0.58**
Batch Learning ANN (BL)	IAE	8.59	25.90	33.96	46.88	54.40	59.45	64.37	67.66	85.14	93.70
	AMAD% IAE	1.99	14.16	2.00	9.06	3.53	5.87	1.85	4.95	2.50	5.35
	AMAD% ISE	5.01	9.58	0.85	8.41	1.99	1.92	2.99	**0.41**	3.50	7.50

Table 7
Training robustness analysis: CSTR across different initial seeds.

Controller	ANN Model	Mean IAE (Train)	AMAD% (Train)	Mean IAE (Val)	AMAD% (Val)
ESS (Enhanced Scheduled Sampling)					
ESS (308e)	LeakyMLP4	361.67	3.07%	287.67	3.63%
ESS (308e)	MLP8	367.33	1.69%	286.33	3.34%
ESS + FineT	LSTM (ws = 500)	375.33	4.14%	292.33	1.98%
BL (Batch Learning/Offline Training)					
BL (1000e)	LeakyMLP4	389.88	3.80%	302.86	3.27%
BL (15e)	MLP8	403.34	1.42%	341.12	2.37%
BL (1000e)	LSTM (ws = 15)	621.33	16.17%	535.33	13.91%

reliable controllers, irrespective of the inherent randomness in the training process (e.g., weight initialisation). To quantify this, we use the Average Mean Absolute Deviation (AMAD%), which measures the average percentage of performance variation from the mean across multiple runs with different random seeds. A lower AMAD% signifies greater consistency and reliability.

5.4.1. FOPDT systems

For the FOPDT processes, the TR results are presented in Table 6. They reveal a clear and consistent advantage for the SS methodology. The SS models exhibit remarkably low AMAD% values across all tested process time constants (τ), consistently staying below 2.5%. In contrast, the BL models, particularly for more challenging process dynamics (e.g., $\tau = 0.31$ and $\tau = 0.74$), show significantly higher variability, with AMAD% values reaching up to 14.16%. This demonstrates that for these simple systems, the SS framework provides much greater stability, reliably producing controllers of a consistent quality.

5.4.2. CSTR system

The analysis was extended to the more complex CSTR environment, with the three most representative models for both ESS and BL being trained with five seeds. The results are presented in Table 7. For the CSTR, the variability between the best BL and ESS models is more comparable, with most models showing an AMAD% between 2–4%. However, the ESS framework consistently achieves this stability at a significantly better performance level, with lower Mean IAE values across all architectures. The ESS has many random factors, also due to its coin flip behaviour and noise additions to ANN outputs, so achieving a comparable TR to offline training with enough epochs is a good indicator. For instance, the `ESS + FineT LSTM (ws=500)` model achieves the best validation consistency (AMAD% of 1.98%), likely due to its robust fine-tuning phase and the large input window size, which helps stabilise its predictions.

Again, the unreliable nature of the `BL (1000e) LSTM` model is proven, as it exhibits extremely high variability (over 13% AMAD% in validation). Overall, this analysis was essential to confirm that the training frameworks did not "explode" or produce wildly divergent outcomes dependent on random factors, with ESS proving to be the more dependable choice.

6. Conclusion and future directions

This work presented and evaluated the Enhanced Scheduled Sampling (ESS) framework, demonstrating its advancements in the training of ANN-based controllers for industrial processes. Our findings consistently show that ESS is a superior alternative to traditional Batch Learning (BL) methodologies for replicating controller-like behaviour, whether the lightweight ANN architecture incorporates memory (e.g., LSTMs) or not (e.g., MLPs). The incorporated enhancements, such as Weighted ANN Actuations (WANNA), the augmented loss function, and our dynamic learning rate strategy, jointly with customisable modules, have been crucial in achieving controllers that aim not only to be high-performing but also to be safe. This emphasis on both performance and robustness (across both parametric variations and training initialisation) is key to the idea of developing a methodology capable of delivering dependable control solutions. In summary, the proposed ESS framework offers a holistic stance on bridging the gap between neural network training and control-oriented deployment. ESS provides a practical methodology that addresses the multiple challenges of creating reliable ANN controllers in a way that more isolated approaches have not.

The empirical evidence from both FOPDT and CSTR systems validates ESS's ability to mitigate critical issues inherent in offline training, most notably the difficulties in achieving training success due to exposure bias. Despite the stochasticity of the ESS framework due to its probabilistic scheduling and noise additions, it demonstrates a level of training robustness comparable to the more deterministic BL approach. This is a strong indicator of the framework's potential, as it consistently produces competent controllers without a notable penalty for its dynamic, online nature. To encourage further research and facilitate broader adoption, the implementation of the ESS framework is open-access and free for the research community to build upon, customise, and test [21].

From a practical standpoint, the ESS hyperparameters are intentionally tunable rather than universal. Reference configurations and tested ranges used in this work are provided in the repository. For instance, for λ in Eq. (8), the reference implementation uses $\lambda = 0.1$; users can increase it when steady-state or long-horizon errors persist and reduce it when PID-output imitation or training convergence deteriorates, while aiming to keep the weighted sequence-loss contribution of a comparable

order to the MSE term. The default WANNA schedule and learning-rate orders reported in Section 4.3 provide additional starting points. A systematic sensitivity analysis and process-dependent automatic tuning of these parameters remain future work.

As introduced in Section 2.2, there are alternative paradigms applicable to ANN-based controller generation, such as Professor Forcing (PF) and Reinforcement Learning (RL). While the primary focus of this work has been the presentation and validation of ESS, it is relevant to contextualise the advantages regarding sample efficiency, stability, and computational demand when comparing ESS to other methods.

PF emerged as an alternative to the claimed drawbacks of SS, specifically regarding the learning objective as highlighted by Huszár [23]. It was argued that standard SS can lead to a biased estimator by training on a distribution (its own errors) that differs from the target. PF attempts to resolve this by forcing the free-running dynamics to match the teacher-forced dynamics via adversarial training. In the context of control, however, the "corrupted inputs" are entirely possible state disturbances that the controller must reject through its corrective policy, which is different from generative modelling applications. ESS addresses threats of control errors and divergences encountered in SS training by stabilising the learning trajectory via the WANNA mechanism and penalising control deviations through the sequence-based error term, which forces the network to make good long-term decisions.

PF relies on an adversarial training scheme, where an ANN discriminator network challenges the ANN controller to produce trajectories indistinguishable from those of the teacher. This adversarial nature introduces significant computational overhead and hyperparameterisation due to the concurrent training of two ANNs. Furthermore, adversarial dynamics are often prone to training instability, such as mode collapse (failing to learn the full distribution of the data) or oscillatory convergence, which can be detrimental in safety-critical control applications. ESS retains the inherent advantage of SS of maintaining a lightweight single controller network, and includes modifications with WANNA and dynamic LR for sample efficiency, and fits with limited trajectory data. ESS is designed to offer more predictable convergence than PF and a computationally lighter alternative for classic industrial scenarios with constrained resources.

Comparatively, RL represents the shift to self-learned policy optimisation, demanding extensive state-space exploration. While autonomous policy discovery facilitates the search for global loss maxima, the controllers obtained lack stability guarantees and remain highly sensitive to reward shaping. ESS leverages the baseline PID teacher, aiming for both a lightweight solution and a progressive learning journey. The incorporation of the sequence-based control error term allows ESS to bridge the gap between the limitations of standard imitation learning on nominal SS and the performance goals of closed-loop control. The framework aims to attain results with orders of magnitude fewer interactions than RL, making it an efficient route to ANN-based alternatives with known controller trajectories.

Consequently, there are strong architectural arguments for ESS as the framework for typical loops, but these relative operational boundaries remain to be better quantified. Upcoming research directions should include direct empirical benchmarks against PF and RL to strictly validate these expected advantages in terms of sample efficiency and stability within industrial control environments.

Looking ahead, several paths for the future emerge from this work. Future efforts will focus on fine-tuning ESS strategies to further close this gap. In addition, a further research path involves integrating reinforcement learning techniques into the methodology to enable the ANN-based controllers to autonomously develop control policies superior to traditional baselines. Crucially, the ESS framework's ability to produce robust and adaptable ANN-based controllers makes it an ideal foundation for advancing Transfer Learning (TL) for control research. That direction is very promising, with the existence of a framework able to create ANN controllers such as ESS. Ultimately, the goal is to leverage TL to drastically reduce the time and engineering effort required to commission and tune new controllers, particularly in large-scale plants with numerous, similar control loops. In that sense, the investigation can entail using ESS-trained controllers as source models for TL in complex control loops, such as the CSTR. This will involve comparing new transferability metrics and validating the framework's effectiveness in adapting controllers across diverse industrial environments, building upon prior work with FOPDT systems [14]. Finally, an alternative future research direction, with a different interpretation for the TL idea can also be explored. This is the use of the ESS approach to tune the parameters of a PID instead of generating an ANN-based controller. This may be useful to explore in which way a PID can replicate the behaviour of a more advanced controller by attempting to transfer the "control knowledge" of the advanced, more complex controller, and project it into the PID controller.

CRediT authorship contribution statement

Pau Comas: Writing – original draft, Software, Investigation, Conceptualization. **Antoni Morell:** Writing – review & editing, Supervision. **Jose Lopez Vicario:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Ramon Vilanova:** Writing – review & editing, Validation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has received support from the OPEN4CEC project (PCI2023-145975-2) and the SMART-LEM project (PCI2023-145977-2), funded by MCIU/AEI/10.13039/501100011033 and co-funded by the European Union.

Appendix A. Application to high-order dynamics: SHSTS of a 500 MW boiler

A.1. Process description and baselines

The Superheated Steam Temperature System (SHSTS) of a 500 MW boiler is characterised by high-order lag dynamics, represented by the fifth-order transfer function [36]:

$$P_B(s) = \frac{0.7732}{(19\ \mathrm{s} + 1)^5} \tag{A.1}$$

where time constants are in seconds.

We utilised two robust standard PID baselines designed for a robustness level of $M_S = 2.0$. The first baseline is the NORpid (Near-Optimal Robust PID) controller, tuned based on a 4th-order Repeated-Pole approximation [36], with parameters $K_p = 1.872$, $T_i = 58.134$s, and $T_d = 24.734$s. The second baseline is the eSIMC (Extended SIMC) controller, tuned based on a FOPDT approximation using the extended SIMC rule [37], with parameters $K_p = 1.452$, $T_i = 66.902$s, and $T_d = 12.428$s.

Training and validation datasets were generated using a randomised signal of step changes with variable step intervals, a sample time of $T_s =$

Table A.8
Performance comparison on 500 MW boiler process (IAE).

Target Strategy	Controller	Training IAE	Test IAE
NORpid ($M_S = 2.0$)	Baseline PID	510	539
	ESS-ANN	544	578
eSIMC ($M_S = 2.0$)	Baseline PID	638	668
	ESS-ANN	625	651

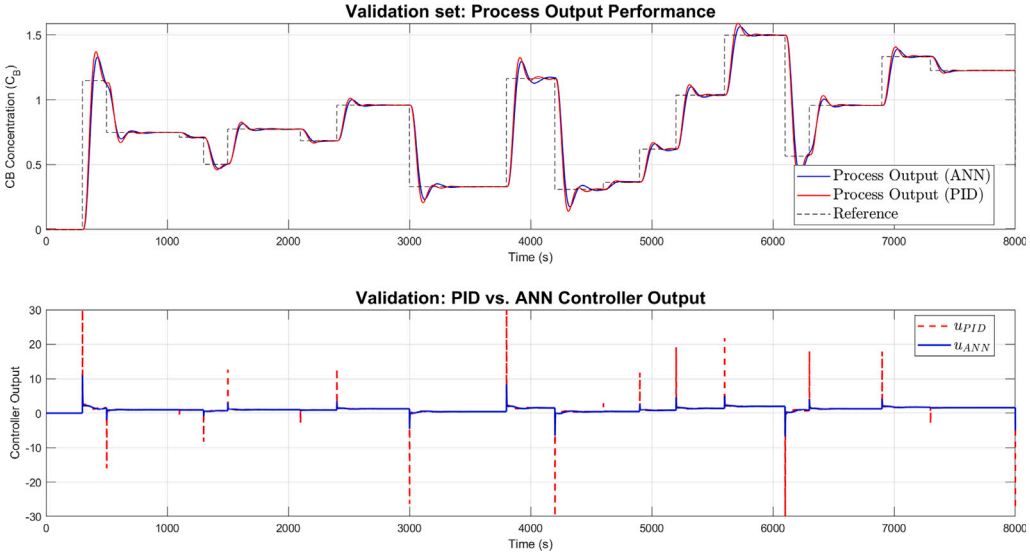


Fig. A.12. ESS-ANN control response tracking the NORpid baseline ($M_S = 2.0$) on the test set.

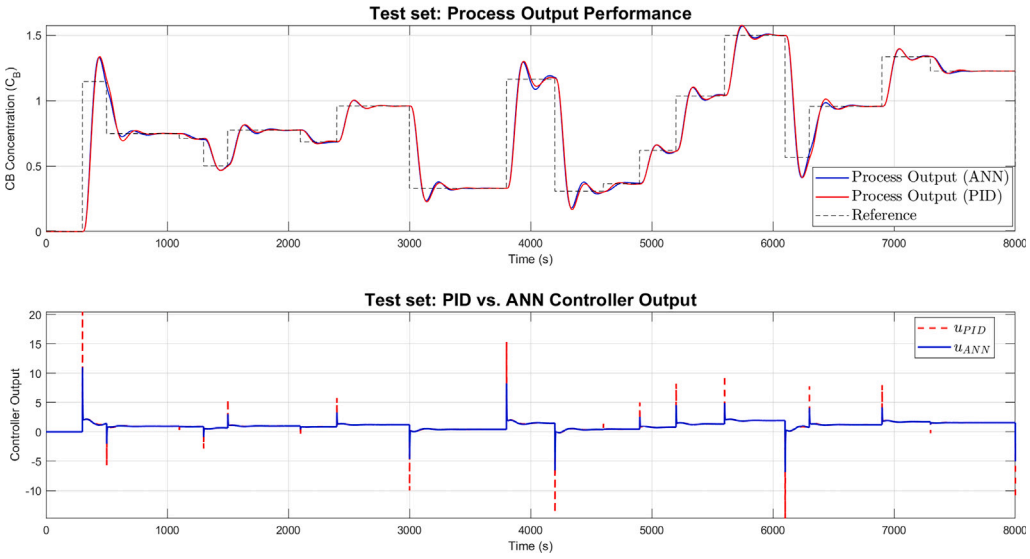


Fig. A.13. ESS-ANN control response tracking the eSIMC baseline ($M_S = 2.0$) on the test set.

1.0s, and a total simulation duration of $T_{sim} = 8000$s. The ANN controller uses the LeakyMLP4 architecture described in Section 3.2.1.

A.2. Performance results

Table A.8 compares the IAE of the baseline PID controllers against their respective ESS ANN-based controllers' training and test sets. Results show that the ESS framework effectively clones the behaviour of the eSIMC PID, with a little improvement in terms of error due to the sequence-based error term based on control performance. For the NORpid strategy, based on a higher-order process than the previous one, the ANN maintains performance close to the theoretical optimum with room for improvement, which is due to the ANN's tendency to produce smooth control outputs and regulate abrupt changes.

The time-domain responses for the test dataset are illustrated in Figs. A.12 and A.13, which show the process output (y_t) and control signal (u_t).

References

[1] Borase RP, Maghade DK, Sondkar SY, Pawar SN. A review of PID control, tuning methods and applications. Int J Dyn Control 2021;9(2):818–27. https://doi.org/10.1007/s40435-020-00665-4

[2] Sundaram VS, Jayabarathi T. Load frequency control using PID tuned ANN controller in power system. In: Proc 1st Int Conf electr energy syst (ICEES); 2011. p. 269–74. https://doi.org/10.1109/ICEES.2011.5725341

[3] Rodríguez-Abreo O, Rodríguez-Reséndiz J, Fuentes-Silva C, Hernández-Alvarado R, Torres Falcón MDCP. Self-tuning neural network PID with dynamic response control. IEEE Access 2021;9:65206–15. https://doi.org/10.1109/ACCESS.2021.3075452

[4] Alfaro VM, Vilanova R. Simple robust tuning of 2DoF PID controllers from a performance/robustness trade-off analysis. Asian J Control 2013;15(6):1700–13. https://doi.org/10.1002/asjc.653

[5] Kumar R, Srivastava S, Gupta JRP. Artificial neural network based PID controller for online control of dynamical systems. In: Proc IEEE 1st Int Conf power electron intell control energy syst (ICPEICES); 2016. p. 1–6. https://doi.org/10.1109/ICPEICES.2016.7853092

[6] Almachi JC, Vicente R, Bone E, Montenegro J, Cando E, Reina S. Implementation of a neural network for adaptive PID tuning in a high-temperature thermal system. Energies 2025;18(12):3113. https://doi.org/10.3390/en18123113

[7] Pisa I, Santín I, Lopez Vicario J, Morell A, Vilanova R. ANN-based soft sensor to predict effluent violations in wastewater treatment plants. Sensors 2019;19(6):1280. https://doi.org/10.3390/s19061280
[8] Pisa I, Santín I, Morell A, Lopez Vicario J, Vilanova R. LSTM-based wastewater treatment plants operation strategies for effluent quality improvement. IEEE Access 2019;7:159773–86. https://doi.org/10.1109/ACCESS.2019.2950852
[9] Comas P, Lopez Vicario J, Morell A, Vilanova R. Scheduled sampling training framework for ANN-based PID control. In: Proc IEEE 29th Int Conf emerg technol fact autom (ETFA); 2024. p. 1–4. https://doi.org/10.1109/ETFA61755.2024.10710958
[10] Efheij H, Albagul A. Comparison of PID and artificial neural network controller in online real-time industrial temperature process control. In: Proc IEEE 1st Int maghreb meet Conf sci tech autom control comput Eng (MI-STA); 2021. p. 110–5. https://doi.org/10.1109/MI-STA52233.2021.9464484
[11] Aamir M. On replacing PID controller with ANN controller for DC motor position control. Int J Res Stud Comput 2013;2(1):21–9. https://doi.org/10.5861/ijrsc.2013.236
[12] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, et al. A comprehensive survey on transfer learning. Proc IEEE 2021;109(1):43–76. https://doi.org/10.1109/JPROC.2020.3004555
[13] Jiang J, Shu Y, Wang J, Long M. Transferability in deep learning: a survey. arXiv preprint. arXiv:2201.05867. 2022.
[14] Comas P, Morell A, Vilanova R, Lopez Vicario J. Predicting transfer learning suitability in ANN-based control of FOPDT industrial processes. J Process Control 2025;152:103476. https://doi.org/10.1016/j.jprocont.2025.103476
[15] Ranzato M, Chopra S, Auli M, Zaremba W. Sequence level training with recurrent neural networks. arXiv preprint. arXiv:1511.06732. 2015.
[16] Zargar SA. Introduction to sequence learning models: RNN, LSTM. GRU [preprint ed. Raleigh (NC): North Carolina State University; 2021. https://doi.org/10.13140/RG.2.2.36370.99522
[17] Bengio S, Vinyals O, Jaitly N, Shazeer N. Scheduled sampling for sequence prediction with recurrent neural networks. arXiv preprint. arXiv:1506.03099. 2015.
[18] Peussa A, Damskägg E-P, Sherson T, Mimilakis SI, Juvela L, Gotsopoulos A, et al. Exposure bias and state matching in recurrent neural network virtual analog models. In: Proc 24th Int Conf digit audio eff. IEEE; 2021. p. 284–91. https://doi.org/10.23919/DAFx51585.2021.9768259 Vienna, Austria.
[19] Manzanas Lopez D, Musau P, Tran H, Johnson TT. Verification of closed-loop systems with neural network controllers. EPiC Ser Comput 2019;61:201–10. https://doi.org/10.29007/btv1
[20] Abiodun OI, Jantan A, Omolara AE, Dada KV, Mohamed NA, Arshad H. State-of-the-art in artificial neural network applications: a survey. Heliyon 2018;4(11):e00938. https://doi.org/10.1016/j.heliyon.2018.e00938
[21] Comas P. Code repository for: “Mitigating exposure bias for reliable ANN-based control: the enhanced scheduled sampling framework”. Github 2026. https://github.com/PauCoH/ESS
[22] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: common objects in context. In: Proc eur Conf comput vis (ECCV). Cham: Springer; 2014. p. 740–55. https://doi.org/10.1007/978-3-319-10602-1_48
[23] Huszár F. How (not) to train your generative model: scheduled sampling, likelihood, adversary? arXiv preprint. arXiv:1511.05101. 2015.
[24] Korakakis M, Vlachos A. Improving scheduled sampling with elastic weight consolidation for neural machine translation. Findings Assoc Comput Linguist: EMNLP 2022 2022;7247–58. https://doi.org/10.18653/v1/2022.findings-emnlp.536
[25] Mihaylova T, Martins AFT. Scheduled sampling for transformers. In: Proc 57th annu meet assoc comput linguist: stud RES workshop; 2019. p. 351–6. https://doi.org/10.18653/v1/P19-2049
[26] Chen H, Lin K, Maye A, Li J, Hu X. A semantics-assisted video captioning model trained with scheduled sampling. Front Robot AI 2020;7:475767. https://doi.org/10.3389/frobt.2020.475767
[27] Lamb AM, Goyal A, Zhang Y, Zhang S, Courville AC, Bengio Y. Professor forcing: a new algorithm for training recurrent networks. arXiv preprint. arXiv:1610.09038. 2016.
[28] Buşoniu L, De Bruin T, Tolić D, Kober J, Palunko I. Reinforcement learning for control: performance, stability, and deep approximators. Annu Rev Control 2018;46:8–28. https://doi.org/10.1016/j.arcontrol.2018.09.005
[29] Tridianto E, Ariwibowo TH, Almasa SK, Prasetya HEG. Cascaded PID temperature controller for FOPDT model of shell-and-tube heat exchanger based on MATLAB/Simulink. Proc Int Electron Symp Eng Technol Appl 2017;185–91. https://doi.org/10.1109/ELECSYM.2017.8240400
[30] Alfaro VM, Vilanova R. Model reference based robust tuning of five-parameter 2DoF PID controllers for first-order plus dead-time models. Proc Eur Control Conf (ECC) 2013;3931–6. https://doi.org/10.23919/ECC.2013.6669850
[31] Alfaro VM, Vilanova R. Model-reference robust tuning of PID controllers. vol. 10. Cham: Springer International Publishing; 2016. p. 101–5. https://doi.org/10.1007/978-3-319-28213-8.
[32] Balaguer P, Alfaro V, Arrieta O. Second order inverse response process identification from transient step response. ISA Trans 2011;50(2):231–8. https://doi.org/10.1016/j.isatra.2010.11.005
[33] Alfaro VM. Low-order models identification from the process reaction curve. Cienc Tecnol (Costa Rica) 2006;24(2):197–216. Spanish.
[34] Fuller S, Greiner B, Moore J, Murray R, van Paassen R, Yorke R. The Python control systems library (Python-control). In: Proc 60th IEEE Conf decis control (CDC); 2021. p. 4875–81. https://doi.org/10.1109/CDC45484.2021.9683368
[35] Brunton SL, Dawson STM, Rowley CW. State-space model identification and feedback control of unsteady aerodynamic forces. J Fluids Struct 2014;50:253–70. https://doi.org/10.1016/j.jfluidstructs.2014.06.026
[36] Alfaro VM, Vilanova R. Control of high-order processes: near-optimal robust tuning of PID regulators for repeated-pole plus dead-time models. Int J Control 2024;97(1):72–85. https://doi.org/10.1080/00207179.2021.1983642
[37] Grimholt C, Skogestad S. Optimal PID-control on first order plus time delay systems and verification of the SIMC rules. IFAC Proc 2013;46(32):265–70. https://doi.org/10.3182/20131218-3-IN-2045.00122