

Article

Control of Nonlinear Systems Using Fuzzy Techniques Based on Incremental State Models of the Variable Type Employing the “Extremum Seeking” Optimizer

Basil Mohammed Al-Hadithi ^{1,2}  and Gilberth André Loja Acuña ^{2,*}

¹ Intelligent Control Group, Universidad Politécnica de Madrid, Centre for Automation and Robotics UPM—CSIC, C/J. Gutierrez Abascal, 2, 28006 Madrid, Spain; basil.alhadithi@upm.es

² Department of Electrical, Electronics, Control Engineering and Applied Physics, School of Industrial Design and Engineering, Universidad Politécnica de Madrid, C/Ronda de Valencia, 3, 28012 Madrid, Spain

* Correspondence: andre.loja.acuna@alumnos.upm.es

Abstract

This work presents the design of a control algorithm based on an augmented incremental state-space model, emphasizing its compatibility with Takagi–Sugeno (T–S) fuzzy models for nonlinear systems. The methodology integrates key components such as incremental modeling, fuzzy system identification, discrete Linear Quadratic Regulator (LQR) design, and state observer implementation. To optimize controller performance, the Extremum Seeking Control (ESC) technique is employed for the automatic tuning of LQR gains, minimizing a predefined cost function. The control strategy is formulated within a generalized framework that evolves from conventional discrete fuzzy models to a higher-order incremental- N state-space representation. The simulation results on a nonlinear multi-variable thermal mixing tank system validate the effectiveness of the proposed approach under reference tracking and various disturbance scenarios, including ramp, parabolic, and higher-order polynomial signals. The main contribution of this work is that the proposed scheme achieves zero steady-state error for reference inputs and disturbances up to order $N-1$ by employing the incremental- N formulation. Furthermore, the system exhibits robustness against input and load disturbances, as well as measurement noise. Remarkably, the ESC algorithm maintains its effectiveness even when noise is present in the system output. Additionally, the proposed incremental- N model is applicable to fast dynamic systems, provided that the system dynamics are accurately identified and the model is discretized using a suitable sampling rate. This makes the approach particularly relevant for control applications in electrical systems, where handling high-order reference signals and disturbances is critical. The incremental formulation, thus, offers a practical and effective framework for achieving high-performance control in both slow and fast nonlinear multivariable processes.

Keywords: incremental model; Takagi–Sugeno model; Extremum Seeking Control



Received: 17 June 2025

Revised: 7 July 2025

Accepted: 8 July 2025

Published: 11 July 2025

Citation: Al-Hadithi, B.M.; Loja Acuña, G.A. Control of Nonlinear Systems Using Fuzzy Techniques Based on Incremental State Models of the Variable Type Employing the “Extremum Seeking” Optimizer. *Appl. Sci.* **2025**, *15*, 7791. <https://doi.org/10.3390/app15147791>

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fuzzy logic methods emerged in the mid-1960s with the foundational work presented in [1], where a fuzzy set is defined through a membership function that assigns each element a degree of membership ranging from zero to one. Subsequently, in 1973, Zadeh expanded on this concept by introducing fuzzy algorithms, which are structured sequences

of instructions formulated through IF-THEN rules [2]. In the same year, the first documented industrial application of fuzzy logic was reported in [3]. This study explored the development of a “linguistic” controller for a steam engine model, aiming to analyze human interaction with adaptive control strategies. Unexpectedly, the fuzzy control approach demonstrated superior performance, significantly exceeding expectations and fostering broader interest in the application and advancement of fuzzy logic systems

The work presented in [4] compares fuzzy control methods with other nonlinear control strategies. It points out that a major drawback of fuzzy control lies in its inability to clearly define a direct link between membership functions and the system’s states, an unresolved issue. The authors also suggest that nonlinear control approaches, especially those specifically adapted to the system’s structure, often demonstrate superior performance compared to fuzzy techniques. The study presented in [5] focuses on Takagi–Sugeno–Kang (TSK) fuzzy systems, which enable the representation of general nonlinear dynamics through a compact set of rules. This work also introduces a methodology for designing fuzzy controllers capable of ensuring closed-loop stability when using TSK models. A practical implementation of fuzzy control is illustrated in [6], where a fuzzy logic controller is applied to a magnetic levitation system of a metallic sphere. The main contribution lies in the comparative analysis between the fuzzy controller and both linear (PD and PID) and nonlinear alternatives. The results reveal a superior performance of the fuzzy controller, primarily due to its effectiveness in managing the inherent nonlinearities of the system.

The research outlined in [7] introduces the well-established Takagi–Sugeno (T–S) identification method for modeling systems based on input–output data. A notable limitation of this method is its inapplicability in scenarios where membership functions exhibit pairwise overlap. To address this shortcoming, the approaches proposed in [8,9] incorporate parameter weighting to refine the values obtained through standard T–S identification, thereby mitigating the constraints of the original formulation. Furthermore, a Linear Quadratic Regulator (LQR) is employed to evaluate the performance of the proposed estimation framework. Validation through two illustrative examples confirms the robustness, simplicity, and general applicability of the algorithm developed.

The study conducted in [10] illustrates the application of modern control theory, rooted in time-domain analysis and state-space modeling, to systems with multiple inputs and outputs, including those exhibiting nonlinear behavior. In this study, the authors address the challenges posed by complex airflow dynamics, noting that few models adequately capture the dynamic distribution of indoor temperature. To improve accuracy in temperature prediction and control, a state-space model is employed and experimentally validated, yielding realistic and reliable results. Similarly, the study in [11] proposes an enhanced aeroservoelastic modeling framework for gust load analysis using time-domain state-space representations. The approach effectively handles the computational complexity of gust calculations and proves advantageous for control system design. A subsystem realization methodology is introduced to construct the full aeroservoelastic model, with simulation results confirming its high predictive accuracy.

The contributions presented in [12,13] demonstrate further applications of the state-space modeling paradigm in control design. Di Ruscio [12] incorporates the model within a model predictive control framework, whereas [13] utilizes it for the regulation of a four-tank process. In the latter study, as is common in system identification tasks, the derivation of accurate models typically requires full knowledge of both input and output signals. However, when one or more inputs are partially unknown, the identification process becomes significantly more challenging or even infeasible. To address this limitation, the methodology proposed in [14] introduces an iterative estimation procedure capable of reconstructing the system state trajectory and inferring the unknown input signal based

solely on output measurements. This technique effectively mitigates one of the primary obstacles associated with conventional identification approaches.

The control strategy outlined in [15] leverages the incremental state-space model to achieve zero steady-state error in response to step-type reference signals. This formulation is further employed in [16] for the optimal multivariable control of a wind turbine system. The evaluation involves the application of Linear Quadratic Regulator (LQR) controllers to a nonlinear wind turbine model, with simulation results indicating favorable transient performance and complete steady-state error rejection, even in the presence of external disturbances, nonlinearities, and model inaccuracies. The work in [17] reinforces the suitability of the incremental modeling approach for addressing the intrinsic nonlinearity and uncertainty of the plant. Additional findings in [18] demonstrate that incremental control strategies can effectively resolve continuous-time nonlinear control problems without requiring a comprehensive global model of the system.

The foundational work in [19] demonstrates that the state vector of a linear system can be reconstructed by monitoring the system's inputs and outputs. This study introduces the concept of a state observer, which enables estimation of the system's internal states when direct measurement is impractical—a common scenario in real-world control applications. The relevance of this methodology is further substantiated in [20,21], where state observers are employed to estimate the system states and facilitate the design of optimal state-feedback controllers. Simulation analyses presented in both references validate the proposed approaches, highlighting their ability to ensure global stability and accurate state estimation.

The comparative study conducted in [22] analyzes the performance of three advanced state observers: high-gain, sliding mode, and extended state observers developed to address the inherent limitations of classical observer designs. The results indicate that the extended state observer outperforms the others in terms of robustness to dynamic uncertainties, sensor noise, and external disturbances. A similar evaluation is presented in [23], where the Luenberger estimator, Kalman filter, and sliding mode observer are assessed for estimating the speed of a DC motor. The findings underscore that observer performance is highly dependent on the specific characteristics of the system, with each method exhibiting distinct advantages under different operational conditions.

The Linear Quadratic Regulator (LQR) has been widely adopted in state-space control for both continuous- and discrete-time systems due to its inherent robustness and optimal performance characteristics. Wong [24] employs the LQR to demonstrate its ability to ensure system stability under predefined conditions, even in the presence of actuator faults and significant external disturbances. A complementary study is presented in [25], where the LQR is integrated into a fuzzy control architecture. In this case, the nonlinear system is linearized using the Lyapunov linearization technique and modeled through the Takagi–Sugeno (T–S) fuzzy framework. This representation enables the implementation of an LQR controller within each locally linearized region. The simulation results confirm that the combined approach yields quasi-optimal system performance, underscoring the effectiveness of embedding LQR within fuzzy modeling structures.

The investigation performed in [26] presents a comparative evaluation of the Linear Quadratic Regulator (LQR) and the Proportional-Derivative (PD) controller, applied to the attitude control of a quadrotor, a system characterized by nonlinear dynamics. MATLAB 2014 based simulations are employed to assess key performance indicators, including settling time and overshoot percentage. The results indicate that the LQR, despite being a linear control strategy, is capable of delivering satisfactory performance in controlling the nonlinear system, thus validating its applicability in such contexts.

The work presented in [27] introduces an enhancement to the Linear Quadratic Regulator (LQR) through the inclusion of an integral feedback component aimed at improving steady-state accuracy. This modification achieves a notable reduction in steady-state error—from over 40 cm to less than 1 cm—demonstrating the effectiveness of integral action as an extension to the LQR for applications demanding high precision. A comparable methodology is explored in [28], where an optimal control scheme based on the LQR is designed for the position and attitude regulation of a quadcopter UAV. In this case, integral compensation is selectively implemented for vertical position control. The simulation outcomes confirm that the proposed control structure successfully eliminates steady-state error in altitude regulation.

The methodology applied in [29] presents a comparative evaluation of a state feedback controller (SFC), designed using a linearized model and a Linear Quadratic Regulator (LQR), against a conventional PID controller. The findings indicate that, with appropriate parameter tuning, the SFC markedly improves the dynamic performance of the system. Specifically, it achieves a reduction in steady-state error and delivers an enhanced transient response, thereby offering a more robust control solution compared to the traditional PID approach.

Extremum Seeking Control (ESC), initially introduced in [30], constitutes an adaptive control strategy that enables a system to converge toward the optimum of a measured performance index without requiring a priori knowledge of the plant model. The method operates by iteratively minimizing a cost function through the adjustment of specific optimization parameters. A significant resurgence of interest in ESC was driven by the contribution in [31], where a formal proof of convergence was established for the standard ESC scheme utilizing periodic perturbations in the context of nonlinear systems.

The application of Extremum Seeking Control (ESC) for PID tuning is explored in [32], where ESC is utilized to minimize a cost function and, thereby, identify the optimal parameters of a PID controller. The comparative analysis reveals that ESC achieves superior performance—specifically, a reduced settling time—relative to traditional tuning techniques such as Ziegler–Nichols (ZN), Internal Model Control (IMC), and Iterative Feedback Tuning (IFT). A related contribution is presented in [33], which proposes a novel PI controller design method grounded in ESC principles. This approach is formulated to address real-time optimization tasks, thereby illustrating the adaptability of ESC for dynamic control environments.

As discussed in [34], enhancing process performance typically involves the application of real-time optimization (RTO) or model-based control techniques. Although these strategies can be effective, they often entail considerable computational overhead, and model-based controllers are notably susceptible to disturbances not accounted for in the model. In contrast, Extremum Seeking Control (ESC) offers a model-free alternative that circumvents these issues but is constrained by slow convergence rates. To address this limitation, the integration of automatic optimization control is proposed as a means to reduce computational complexity. The case study involving an ammonia reactor illustrates that the combined use of ESC and automatic optimization enables effective compensation for unmeasured disturbances and plant-model discrepancies, thus improving overall system robustness.

The methodology presented in this work aims to ensure zero steady-state error in response to various reference and disturbance signals, including step, ramp, and second-order polynomial (parabolic) inputs. The proposed control scheme combines the incremental state and incremental-N state modeling frameworks with the Takagi–Sugeno (T–S) fuzzy representation to address nonlinear system dynamics. To improve the controller tuning process, Extremum Seeking Control (ESC) is integrated into the design. The effectiveness

of the approach is validated through a case study involving a thermal tank system, selected due to its inherent nonlinear behavior and multivariable characteristics.

The structure of the paper is as follows: Sections 2–4 introduce the incremental state-space model, its fuzzy extension via Takagi–Sugeno rules, and a nonlinear parameter identification method. Sections 5 and 6 address discrete LQR control and state estimation. Sections 7 and 8 detail the construction of incremental-2 and generalized incremental-N models, including their fuzzy formulations. Section 9 presents the Extremum Seeking Control (ESC) algorithm for automatic LQR tuning. Section 10 applies the proposed fuzzy control strategy to a nonlinear thermal mixing tank, evaluating its performance and robustness across various reference signals and disturbance scenarios.

2. Incremental State Model

It is well established that discrete-time models are suitable for identifying continuous-time systems with multiple delays through the use of difference equations, as such models permit the integration of delays directly into the state formulation. In this context, the incremental state-space model offers distinct advantages over its non-incremental counterpart. Specifically, it simplifies the computation of the reference state, since a zero incremental state can be directly associated with a desired output, effectively eliminating the need for explicit reference state calculation. Additionally, the incremental formulation inherently introduces an integrative effect into the control law, thereby enabling zero steady-state error in response to constant references or disturbances [15].

We use a discrete-time multivariable system with p inputs, $u \in \mathbb{R}^p$, and q outputs, $y \in \mathbb{R}^q$, where $q \leq p$ [35]:

$$y_i(k+1) = a_{i0} + a_{i1}y_i(k) + \cdots + a_{in_i}y_i(k - n_i + 1) + \sum_{j=1}^p (b_{ij1}u_j(k) + b_{ij2}u_j(k-1) + \cdots + b_{ijn_i}u_j(k - n_i + 1)) \quad (1)$$

The system has been identified using the least squares method with affine terms; therefore, some of the parameters a_{in_i} , b_{ij1} , ..., b_{ipn_i} are non-zero. For each output, the difference equations can be transformed into a discrete-time state-space model with affine terms:

$$\begin{aligned} x_i(k) &\in \mathbb{R}^{n_i} \\ x_i(k+1) &= \begin{bmatrix} a_{i0} \cdot a_{i1} \\ a_{i0} \cdot a_{i2} \\ \vdots \\ a_{i0} \cdot a_{in_i} \end{bmatrix} + \begin{bmatrix} a_{i1} & 1 & \cdots & 0 \\ a_{i2} & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 \\ a_{in_i} & 0 & \cdots & 0 \end{bmatrix} x_i(k) \\ &+ \begin{bmatrix} b_{i11} & b_{i21} & \cdots & b_{ip1} \\ b_{i12} & b_{i22} & \vdots & b_{ip2} \\ \vdots & \vdots & \vdots & \vdots \\ b_{i1n_i} & b_{i2n_i} & \cdots & b_{ipn_i} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_p(k) \end{bmatrix} \\ y_i(k) &= a_{i0} + \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} x_i(k) \end{aligned} \quad (2)$$

In matrix notation,

$$\begin{aligned} x_i(k+1) &= a_{xi} + A_i x_i(k) + B_i u(k) \\ y_i(k) &= a_{i0} + C_i x_i(k) \end{aligned} \quad (3)$$

where $a_{xi} \in \mathbb{R}^{(n_i \times 1)}$, $A_i \in \mathbb{R}^{(n_i \times n_i)}$, $B_i \in \mathbb{R}^{(n_i \times p)}$ y $C_i \in \mathbb{R}^{(1 \times n_i)}$. By grouping the entire multivariable system, the following is obtained:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_q(k+1) \end{bmatrix} = \begin{bmatrix} a_{x1} \\ a_{x2} \\ \vdots \\ a_{xq} \end{bmatrix} + \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_q \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_q(k) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_q \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_p(k) \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_q(k) \end{bmatrix} = \begin{bmatrix} a_{10} \\ a_{20} \\ \vdots \\ a_{q0} \end{bmatrix} + \begin{bmatrix} C_1 & 0 & \cdots & 0 \\ 0 & C_2 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_q \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_q(k) \end{bmatrix}$$

This can be represented as follows:

$$\begin{aligned} x(k+1) &= a_x + Ax(k) + Bu(k) \\ y(k) &= a_y + Cx(k) \end{aligned} \quad (5)$$

$a_x \in \mathbb{R}^{(n \times 1)}$, $A \in \mathbb{R}^{(n \times n)}$, $B \in \mathbb{R}^{(n \times p)}$, $a_y \in \mathbb{R}^{(q \times 1)}$ y $C \in \mathbb{R}^{(q \times n)}$, where $n = n_1 + n_2 + \cdots + n_q$. Equation (5) will be referred to as the discrete-time state-space model, which has been obtained through input-output identification; therefore, it is controllable and observable. If Equation (5) is applied for a previous sample, the following is obtained:

$$\begin{aligned} x(k) &= a_x + Ax(k-1) + Bu(k-1) \\ y(k-1) &= a_y + Cx(k-1) \end{aligned} \quad (6)$$

Subtracting (6) from (5) yields

$$\begin{aligned} x(k+1) - x(k) &= A(x(k) - x(k-1)) + B(u(k) - u(k-1)) \\ y(k) - y(k-1) &= C(x(k) - x(k-1)) \end{aligned} \quad (7)$$

The incremental state Δx and the manipulated input Δy are defined as follows:

$$\begin{aligned} \Delta x(k) &= x(k) - x(k-1) \\ \Delta y(k) &= u(k) - u(k-1) \end{aligned} \quad (8)$$

Substituting (8) into (7) yields

$$\begin{aligned} \Delta x(k+1) &= A\Delta x(k) + B\Delta u(k) \\ y(k) &= y(k-1) + C\Delta x(k) \end{aligned} \quad (9)$$

In this formulation, p new states are introduced, achieving the following:

$$\begin{aligned} y(k+1) &= y(k) + C\Delta x(k+1) \\ &= y(k) + C(A\Delta x(k) + B\Delta u(k)) \end{aligned} \quad (10)$$

Thus, the new incremental state model can be expressed as follows:

$$\begin{bmatrix} y(k+1) \\ \Delta x(k+1) \end{bmatrix} = \begin{bmatrix} I & CA \\ 0 & A \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix} + \begin{bmatrix} CB \\ B \end{bmatrix} \Delta u(k) \quad (11)$$

$$y(k) = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix}$$

Thus, a new extended incremental state vector, $x_a(k) \in \mathbb{R}^{(q+n)}$, is defined:

$$x_a(k) = \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix} \quad (12)$$

The incremental state model in matrix notation is:

$$\begin{aligned} x_a(k+1) &= A_a x_a(k) + B_a \Delta u(k) \\ y(k) &= C_a x_a(k) \end{aligned} \quad (13)$$

where $A_a \in \mathbb{R}^{((q+n) \times (q+n))}$, $B_a \in \mathbb{R}^{((q+n) \times p)}$ y $C_a \in \mathbb{R}^{(q \times (q+n))}$.

3. Fuzzy Incremental State Model

Nonlinear systems can be modeled using the T-S fuzzy model, based on a set of measurable system variables $\{z_1(k), z_2(k), \dots, z_m(k)\}$. If $\{r_1, r_2, \dots, r_m\}$ are chosen as fuzzy sets for these variables, the system can be defined through IF-THEN rules as follows [15]:

$$\begin{aligned} S^{(i_1 \dots i_m)}: & \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\ y_i(k+1) &= \\ & a_{i0}^{(i_1 \dots i_m)} + a_{i1}^{(i_1 \dots i_m)} y_i(k) + a_{i2}^{(i_1 \dots i_m)} y_i(k-1) + \dots + a_{in_i}^{(i_1 \dots i_m)} y_i(k-n_i+1) \\ & + \sum_{j=1}^p (b_{ij1}^{(i_1 \dots i_m)} u_j(k) + b_{ij2}^{(i_1 \dots i_m)} u_j(k-1) + \dots + b_{ijn_i}^{(i_1 \dots i_m)} u_j(k-n_i+1)) \end{aligned} \quad (14)$$

where j is the index of each fuzzy variable, and i_m is the index of each fuzzy rule associated with that fuzzy variable. Starting from the result obtained in Equation (11), the fuzzy incremental state model can be derived:

$$\begin{aligned} S^{(i_1 \dots i_m)}: & \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\ \begin{bmatrix} y(k+1) \\ \Delta x(k+1) \end{bmatrix} &= \begin{bmatrix} I & CA^{(i_1 \dots i_m)} \\ 0 & A^{(i_1 \dots i_m)} \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix} + \begin{bmatrix} CB^{(i_1 \dots i_m)} \\ B^{(i_1 \dots i_m)} \end{bmatrix} \Delta u(k) \\ y(k) &= \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix} \end{aligned} \quad (15)$$

where $A^{(i_1 \dots i_m)} \in \mathbb{R}^{n \times n}$, $B^{(i_1 \dots i_m)} \in \mathbb{R}^{n \times p}$ y $C \in \mathbb{R}^{q \times n}$. Moreover, its matrix form can be expressed as follows:

$$\begin{aligned} S^{(i_1 \dots i_m)}: & \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\ x_a(k+1) &= A_a^{(i_1 \dots i_m)} x_a(k) + B_a^{(i_1 \dots i_m)} \Delta u(k) \\ y(k) &= C_a x_a(k) \end{aligned} \quad (16)$$

where $A_a^{(i_1 \dots i_m)} \in \mathbb{R}^{((q+n) \times (q+n))}$, $B_a \in \mathbb{R}^{((q+n) \times p)}$ y $C_a \in \mathbb{R}^{(q \times (q+n))}$.

4. Identification T-S

A noteworthy method for the identification of T-S fuzzy models is presented in [7], where the estimation of a nonlinear system's parameters is carried out through the minimization of a quadratic cost index. It is important to highlight that this method fails when the membership functions of the fuzzy rules are defined as triangular and pairwise overlapping. This failure arises because the T-S matrix is not of full rank and, consequently, is not invertible. Nevertheless, [9,36] propose a generalized T-S model identification approach

based on parameter weighting, which successfully overcomes this limitation. Starting from (14), the fuzzy estimation of each output can be represented as follows:

$$\hat{y}_i(k+1) = \sum_{i_1=1}^{r_1} \cdots \sum_{i_m=1}^{r_m} \beta^{(i_1 \dots i_m)}(z_{(i_1 \dots i_m)}(k)) \left[a_{i_0}^{(i_1 \dots i_m)} + a_{i_1}^{(i_1 \dots i_m)} y_i(k) + \dots + a_{i_{n_i}}^{(i_1 \dots i_m)} y_i(k - n_i + 1) + \sum_{j=1}^p \left(b_{ij1}^{(i_1 \dots i_m)} u_j(k) + \dots + b_{ijn_i}^{(i_1 \dots i_m)} u_j(k - n_i + 1) \right) \right] \quad (17)$$

where

$$\beta^{(i_1 \dots i_m)}(z_{(i_1 \dots i_m)}(k)) = \frac{\prod_{j=1}^m (\mu_{ji_j}(z_j(k)))}{\sum_{i_1=1}^{r_1} \cdots \sum_{i_m=1}^{r_m} \left(\prod_{j=1}^m (\mu_{ji_j}(z_j(k))) \right)} \quad (18)$$

with $\mu_{ji_j}(z_j)$ being the membership function corresponding to the fuzzy set $M_j^{i_j}$. In order to apply the parameter weighting method, an initial estimation of the linearized model is required:

$$P_0 = \begin{bmatrix} a_{i_0}^0 & \cdots & a_{i_{n_i}}^0 & b_{i_{11}}^0 & \cdots & b_{i_{1n_i}}^0 & \cdots & b_{ip1}^0 & \cdots & b_{ipn_i}^0 \end{bmatrix} \quad (19)$$

The initial estimation can be obtained by applying the classical least squares method to the system's dataset. This initial estimation (19) serves as the reference parameters for all T-S subsystems. Subsequently, the fuzzy system parameters can be determined by minimizing

$$\begin{aligned} J &= \sum_{k=1}^{n_s} (y(k) - \hat{y}(k))^2 + \gamma^2 \sum_{i_1=1}^{r_1} \cdots \sum_{i_m=1}^{r_m} \sum_{j=0}^n \left(p_j^0 - p_j^{(i_1 \dots i_m)} \right)^2 \\ &= \|Y - XP\|^2 + \gamma^2 \|p_0 - P\|^2 = \left\| \begin{bmatrix} Y \\ \gamma p_0 \end{bmatrix} - \begin{bmatrix} X \\ \gamma I \end{bmatrix} P \right\|^2 = \|Y_a - X_a P\|^2 \end{aligned} \quad (20)$$

where

$$\begin{aligned} Y &= \begin{bmatrix} y_1 & y_2 & \cdots & y_s \end{bmatrix}^t \\ P_0 &= \begin{bmatrix} a_{i_0}^{(1 \dots 1)} & \cdots & a_{i_{n_i}}^{(1 \dots 1)} & b_{i_{11}}^{(1 \dots 1)} & \cdots & b_{i_{1n_i}}^{(1 \dots 1)} & \cdots & b_{ip1}^{(1 \dots 1)} & \cdots & b_{ipn_i}^{(1 \dots 1)} \end{bmatrix}^t \\ X &= \begin{bmatrix} \beta_1^{(1 \dots 1)} & \beta_1^{(1 \dots 1)} y_1(k)_1 & \cdots & \beta_1^{(1 \dots 1)} u_j(k - n_i + 1)_1 & \cdots & \\ & \beta_1^{(r_1 \dots r_m)} & \cdots & \beta_1^{(r_1 \dots r_m)} u_j(k - n_i + 1)_1 & \cdots & \\ & \vdots & & & & \\ \beta_s^{(1 \dots 1)} & \beta_s^{(1 \dots 1)} y_1(k)_s & \cdots & \beta_s^{(1 \dots 1)} u_j(k - n_i + 1)_s & \cdots & \\ & \beta_s^{(r_1 \dots r_m)} & \cdots & \beta_s^{(r_1 \dots r_m)} u_j(k - n_i + 1)_s & \cdots & \end{bmatrix} \end{aligned}$$

and

$$p_0 = \underbrace{\begin{bmatrix} P_0 & P_0 & \cdots & P_0 \end{bmatrix}^t}_{r_1 \cdot r_2 \cdots r_m}$$

It can be observed that Y corresponds to the output data, P to the parameters of the fuzzy model, and X to the fuzzy input/output data. In this case, the factor γ represents the degree of confidence in the initially estimated parameters. By applying the method presented in [9,36], the matrix X_a is ensured to be full rank, even if the membership functions of the fuzzy rules

are defined as triangular and pairwise overlapping. If the matrix is full rank, then P can be calculated as follows:

$$\begin{aligned} J &= \|Y_a - X_a P\|^2 = (Y_a - X_a P)^t (Y_a - X_a P) \\ \nabla J &= -2(Y_a - X_a P)^t X_a = 0 \\ X_a^t (Y_a - X_a P) &= X_a^t Y_a - X_a^t X_a P = 0 \\ P &= (X_a^t X_a)^{-1} X_a^t Y_a \end{aligned} \quad (21)$$

5. Optimal Controller

Any method based on state feedback can be employed to compute the control coefficients. Nevertheless, such methods may lead to systems where the control action exceeds the allowable limits (saturation). Therefore, the optimal selection is the one that achieves a balance between dynamic response speed and control effort. The LQR fulfills these criteria. Its application to linear systems is demonstrated in [37–39]. To extend its use to nonlinear systems, a cost function must be minimized within each fuzzy subsystem. The use of the discrete LQR [40,41] is proposed, given its simplicity in controller design and its applicability to different state-space models. The objective is to minimize the cost function J . To achieve this, appropriate tuning of the weighting matrices Q and R , as well as knowledge of both the extended state reference and the extended state itself, x_{ar} and x_a , is required.

$$J = \sum_{k=0}^{\infty} [(x_a(k) - x_{ar})^t Q (x_a(k) - x_{ar}) + \Delta u(k)^t R \Delta u(k)] \quad (22)$$

where $Q \in \mathbb{R}^{(q+n) \times (q+n)}$ and $R \in \mathbb{R}^{(p \times p)}$ are positive definite symmetric matrices. Thus, if the control objective is to achieve a steady-state output y_r , in the incremental model this is equivalent to reaching the extended reference state [16]. In other words,

$$x_a = \begin{bmatrix} y_r \\ \Delta x_r \end{bmatrix} = \begin{bmatrix} y_r \\ 0 \end{bmatrix} \quad (23)$$

It should be noted that the extended reference state naturally arises without the need for further calculation, as steady-state operation corresponds to a null incremental state. The controller is subsequently described as follows:

$$\begin{aligned} \Delta u(k) &= K(x_{ar} - x_a(k)) = \begin{bmatrix} K_y & K_x \end{bmatrix} \left(\begin{bmatrix} y_r \\ 0 \end{bmatrix} - \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix} \right) \\ &= \begin{bmatrix} K_y & K_x \end{bmatrix} \begin{bmatrix} y_r - y(k) \\ -\Delta x(k) \end{bmatrix} \end{aligned} \quad (24)$$

where $K \in \mathbb{R}^{(p \times (q+n))}$, $K_y \in \mathbb{R}^{(p \times q)}$ y $K_x \in \mathbb{R}^{(p \times n)}$, and the control action is:

$$u(k) = u(k-1) + \Delta u(k) \quad (25)$$

The Equation (25) is obtained from (8). It should be noted that the control action is calculated incrementally (24), which is equivalent to applying an integral action. Furthermore, if the feedback system is stable and steady-state is achieved, the following condition holds:

$$\lim_{k \rightarrow \infty} \Delta u(k) = 0, \lim_{k \rightarrow \infty} \Delta x(k) = 0$$

and, thus,

$$\lim_{k \rightarrow \infty} \Delta K_y (y_r - y(k)) = 0$$

Then, if K_y is of rank q , it is verified that $\lim_{k \rightarrow \infty} y(k) = y_r$. Therefore, the controlled system has a steady-state error equal to zero. Since the goal is to apply the LQR to each of the systems defined in the incremental T-S state-space model, the controller would be defined as follows:

$$\begin{aligned} S^{(i_1 \dots i_m)}: \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\ \Delta u(k) = K^{(i_1 \dots i_m)} ((x_{ar} - x_a(k))) \end{aligned} \quad (26)$$

whose control action is (25).

6. State Observer

For this case, the incremental state $\Delta x(k)$ is not accessible. For this reason, a state observer is required. The incremental state can be estimated using the following expression [35]:

$$\Delta x_e(k) = x_e(k) - x_e(k-1) \quad (27)$$

Furthermore, it is possible to use an incremental state observer, which is described below [15]:

$$\Delta x_e(k+1) = A\Delta x_e(k) + B\Delta u(k) + H(y(k) - y(k-1) - C\Delta x_e(k)) \quad (28)$$

where $H \in \mathbb{R}^{(n \times q)}$. The error estimate would be:

$$\begin{aligned} \epsilon(k+1) &= \Delta x(k+1) - \Delta x_e(k+1) \\ &= (A\Delta x(k) + B\Delta u(k)) - (A\Delta x_e(k) + B\Delta u(k) + \\ &\quad H(y(k) - y(k-1) - C\Delta x_e(k))) \end{aligned} \quad (29)$$

From Equations (5), (6) and (8), we obtain:

$$\begin{aligned} y(k) - y(k-1) &= a_y + Cx(k) - a_y - Cx(k-1) \\ &= C(x(k) - x(k-1)) = C\Delta x(k) \end{aligned}$$

Then, Equation (29) would be:

$$\epsilon(k+1) = (A - HC)\epsilon(k) \quad (30)$$

The method to calculate H has been described in [15] and is obtained as follows:

$$H = AC^t \quad (31)$$

Finally, the observer of the incremental state extended vector is obtained:

$$x_{ae}(k) = \begin{bmatrix} y(k) \\ \Delta x_e(k) \end{bmatrix} \quad (32)$$

If we want to apply the T-S incremental state model, we need a fuzzy state observer that is defined as:

$$\begin{aligned} S^{(i_1 \dots i_m)}: \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\ \Delta x_e(k+1) = A^{(i_1 \dots i_m)} \Delta x_e(k) + B^{(i_1 \dots i_m)} \Delta u(k) \\ + H^{(i_1 \dots i_m)} (y(k) - y(k-1) - C \Delta x_e(k)) \end{aligned} \quad (33)$$

$H^{(i_1 \dots i_m)}$ is calculated in the same way as Equation (31):

$$\begin{aligned} S^{(i_1 \dots i_m)}: \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\ H^{(i_1 \dots i_m)} = A^{(i_1 \dots i_m)} C^t \end{aligned} \quad (34)$$

7. Incremental-2 State Model

As seen in Section 2, it is straightforward to obtain the incremental state model using Equation (11). For this reason, the question arises as to what would happen if this equation were applied again to previously calculated incremental state matrices. By doing so, a new state model is obtained, which we will refer to as the incremental-2 state model. This model naturally allows for the introduction of a second integrator into the control algorithm. If we apply the process described by Equation (11) to the previously obtained incremental model, namely, Equation (13), we obtain the following:

$$\begin{aligned} \begin{bmatrix} y(k+1) \\ \Delta x_a(k+1) \end{bmatrix} &= \begin{bmatrix} I & C_a A_a \\ 0 & A_a \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x_a(k) \end{bmatrix} + \begin{bmatrix} C_a B_a \\ B_a \end{bmatrix} \Delta^2 u(k) \\ y(k) &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x_a(k) \end{bmatrix} \end{aligned} \quad (35)$$

where $A_a \in \mathbb{R}^{((q+n) \times (q+n))}$, $B_a \in \mathbb{R}^{((q+n) \times p)}$ y $C_a \in \mathbb{R}^{(q \times (q+n))}$. In matrix form, it would be

$$\begin{aligned} x_{a^2}(k+1) &= A_{a^2} x_{a^2}(k) + B_{a^2} \Delta^2 u(k) \\ y(k) &= C_{a^2} x_{a^2}(k) \end{aligned} \quad (36)$$

where $A_{a^2} \in \mathbb{R}^{((2q+n) \times (2q+n))}$, $B_{a^2} \in \mathbb{R}^{((2q+n) \times p)}$ and $C_{a^2} \in \mathbb{R}^{(q \times (2q+n))}$, thus obtaining the new extended vector of incremental-2 state, $x_{a^2}(k) \in \mathbb{R}^{(2q+n)}$:

$$x_{a^2}(k) = \begin{bmatrix} y(k) \\ \Delta x_a(k) \end{bmatrix} = \begin{bmatrix} y(k) \\ \Delta y(k) \\ \Delta^2 x(k) \end{bmatrix} \quad (37)$$

The feedback matrix of the incremental-2 state can be designed using the discrete LQR method, which requires for its calculation the state matrices A_{a^2} and B_{a^2} , as well as the matrices $Q \in \mathbb{R}^{((2q+n) \times (2q+n))}$ y $R \in \mathbb{R}^{(p \times p)}$. Following the process described in Section 3, and since the objective of the control is to achieve a steady-state output y_r , the extended reference state of the incremental-2 model would be defined as

$$x_{a^2_r}(k) = \begin{bmatrix} y_r \\ \Delta y_r \\ \Delta^2 x_r \end{bmatrix} = \begin{bmatrix} y_r \\ \Delta y_r \\ 0 \end{bmatrix} \quad (38)$$

and the controller is described as follows:

$$\begin{aligned}\Delta^2 u(k) &= K(x_{a^2r} - x_{a^2}(k)) = \begin{bmatrix} K_y & K_{\Delta y} & K_x \end{bmatrix} \left(\begin{bmatrix} y_r \\ \Delta y_r \\ 0 \end{bmatrix} - \begin{bmatrix} y(k) \\ \Delta y(k) \\ \Delta^2 x(k) \end{bmatrix} \right) \\ &= \begin{bmatrix} K_y & K_{\Delta y} & K_x \end{bmatrix} \begin{bmatrix} y_r - y(k) \\ \Delta y_r - \Delta y(k) \\ -\Delta^2 x(k) \end{bmatrix}\end{aligned}\quad (39)$$

where $K \in \mathbb{R}^{(p \times (2q+n))}$, $K_y \in \mathbb{R}^{(p \times q)}$, $K_{\Delta y} \in \mathbb{R}^{(p \times q)}$ and $K_x \in \mathbb{R}^{(p \times n)}$, and the control action is

$$\begin{aligned}\Delta u(k) &= \Delta u(k-1) + \Delta^2 u(k) \\ u(k) &= u(k-1) + \Delta u(k)\end{aligned}\quad (40)$$

If the feedback system is stable and steady state is approached, the following is true:

$$\lim_{k \rightarrow \infty} \Delta^2 u(k) = 0, \lim_{k \rightarrow \infty} \Delta^2 x(k) = 0$$

and, thus,

$$\lim_{k \rightarrow \infty} K_y(y_r - y(k)) = 0, \lim_{k \rightarrow \infty} K_{\Delta y}(\Delta y_r - \Delta y(k)) = 0$$

So, if K_y and $K_{\Delta y}$ are of rank q , it is verified that

$$\lim_{k \rightarrow \infty} y(k) = y_r, \lim_{k \rightarrow \infty} \Delta y(k) = \Delta y_r$$

Therefore, the controlled system has a steady-state error equal to zero in response to step-type inputs and disturbances, and even to ramp-type inputs. It is important to highlight that the control action (40) has been calculated in an incremental-2 form, which is equivalent to applying a double integral control action. Similarly to the incremental model, for the incremental-2 model, it is necessary to use a state observer, since $\Delta^2 x(k)$ is not accessible. Based on Equation (27), the incremental-2 state can be estimated as follows:

$$\Delta^2 x_e(k) \Delta x_e(k) - \Delta x_e(k-1) \quad (41)$$

With Equations (41) and (28), the observer can be described as

$$\Delta^2 x_e(k+1) = A \Delta^2 x_e(k) + B \Delta^2 u(k) + H \left(\Delta y(k) - \Delta y(k-1) - C \Delta^2 x_e(k) \right) \quad (42)$$

where $H \in \mathbb{R}^{(n \times q)}$ is calculated in the same way as in Equation (31). Thus, the observer of the extended vector of the incremental-2 state would be

$$x_{a^2e}(k) = \begin{bmatrix} y(k) \\ \Delta x_{ae}(k) \end{bmatrix} = \begin{bmatrix} y(k) \\ \Delta y(k) \\ \Delta^2 x_e(k) \end{bmatrix} \quad (43)$$

8. Fuzzy Incremental-N State Model

An iterative algorithm has been proposed to obtain an incremental-N state model that introduces N integrators and can achieve zero steady-state error in response to polynomial-

type inputs and disturbances of order N-1. To do this, the extended state matrices are initialized by assigning them to the traditional discrete-state matrices:

$$\begin{aligned} A_{a^N} &= A \\ B_{a^N} &= B \\ C_{a^N} &= C \end{aligned} \quad (44)$$

Using a loop, the extended state matrices are recalculated N times:

$$\begin{aligned} & \text{for } i = 1 : N \\ & A_{a^N} = \begin{bmatrix} I & C_{a^N} A_{a^N} \\ 0 & A_{a^N} \end{bmatrix} \\ & B_{a^N} = \begin{bmatrix} C_{a^N} B_{a^N} \\ B_{a^N} \end{bmatrix} \\ & C_{a^N} = \begin{bmatrix} I & 0 \end{bmatrix} \\ & \text{end} \end{aligned} \quad (45)$$

In this way, the incremental-N state model would be defined:

$$\begin{aligned} \begin{bmatrix} y(k+1) \\ \Delta x_{a^{N-1}}(k+1) \end{bmatrix} &= \begin{bmatrix} I & C_{a^{N-1}} A_{a^{N-1}} \\ 0 & A_{a^{N-1}} \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x_{a^{N-1}}(k) \end{bmatrix} + \begin{bmatrix} C_{a^{N-1}} B_{a^{N-1}} \\ B_{a^{N-1}} \end{bmatrix} \Delta^N u(k) \\ y(k) &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x_{a^{N-1}}(k) \end{bmatrix} \end{aligned} \quad (46)$$

In matrix form,

$$\begin{aligned} x_{a^N}(k+1) &= A_{a^N} x_{a^N}(k) + B_{a^N} \Delta^N u(k) \\ y(k) &= C_{a^N} x_{a^N}(k) \end{aligned} \quad (47)$$

where $A_{a^N} \in \mathbb{R}^{((Nq+n) \times (Nq+n))}$, $B_{a^N} \in \mathbb{R}^{((Nq+n) \times p)}$, and $C_{a^N} \in \mathbb{R}^{(q \times (Nq+n))}$, thus obtaining the new extended vector of incremental-N state, $x_{a^N}(k) \in \mathbb{R}^{(Nq+n)}$. This would be

$$x_{a^N}(k) = \begin{bmatrix} y(k) \\ \Delta y(k) \\ \Delta^2 y(k) \\ \vdots \\ \Delta^{(N-1)} y(k) \\ \Delta^N x(k) \end{bmatrix} \quad (48)$$

and the reference state is

$$x_{a^{N_r}} = \begin{bmatrix} y_r \\ \Delta y_r \\ \Delta^2 y_r \\ \vdots \\ \Delta^{(N-1)} y_r \\ 0 \end{bmatrix} \quad (49)$$

The controller and control action for the incremental-N would be defined:

$$\begin{aligned}
 \Delta^N u(k) &= K(x_{a^N} - x_{a^N}(k)) \\
 \Delta^{(N-1)} u(k) &= \Delta^{(N-1)} u(k-1) + \Delta^N u(k) \\
 \Delta^{(N-2)} u(k) &= \Delta^{(N-2)} u(k-1) + \Delta^{(N-1)} u(k) \\
 &\vdots \\
 \Delta u(k) &= \Delta u(k-1) + \Delta^2 u(k) \\
 u(k) &= u(k-1) + \Delta u(k)
 \end{aligned} \tag{50}$$

where the incremental-N state feedback matrix K can be obtained, for example, using the discrete LQR method, whose calculation will use the state matrices A_a^N and B_a^N obtained through the recursive algorithm of Equation (45), along with the matrices $Q \in \mathbb{R}^{(Nq+n) \times (Nq+n)}$ and $R \in \mathbb{R}^{(p \times p)}$. For the incremental-N model, as with the previous incremental models, a state observer is required since $\Delta^N x(k)$ is not accessible. Following the approach of Equations (27) and (41), we can estimate the incremental-N state as follows:

$$\Delta^N x_e(k) = \Delta^{(N-1)} x_e(k) - \Delta^{(N-1)} x_e(k-1) \tag{51}$$

Following the approach of the Equations (42) and (51), the observer can be described as

$$\begin{aligned}
 \Delta^N x_e(k+1) &= A \Delta^N x_e(k) + B \Delta^N u(k) \\
 &+ H \left(\Delta^{(N-1)} y(k) - \Delta^{(N-1)} y(k-1) - C \Delta^N x_e(k) \right)
 \end{aligned} \tag{52}$$

where $H \in \mathbb{R}^{(n \times q)}$ is calculated in the same way as in the Equation (31). Thus, the observer of the extended incremental-N state vector is defined:

$$x_{a^N}^e(k) = \begin{bmatrix} y(k) \\ \Delta y(k) \\ \Delta^2 y(k) \\ \vdots \\ \Delta^{(N-1)} y(k) \\ \Delta^N x_e(k) \end{bmatrix} \tag{53}$$

Since the goal is to achieve fuzzy control using the incremental-N state model, the method to be used is explained below. First, after completing the linear identification and T-S identification, the nonlinear system is modeled using the incremental-N T-S state model. Starting from Equation (46), we obtain

$$\begin{aligned}
 S^{(i_1 \dots i_m)}: \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\
 \begin{bmatrix} y(k+1) \\ \Delta x_{a^{N-1}}(k+1) \end{bmatrix} &= \begin{bmatrix} I & C_{a^{N-1}} A_{a^{N-1}}^{(i_1 \dots i_m)} \\ 0 & A_{a^{N-1}}^{(i_1 \dots i_m)} \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x_{a^{N-1}}(k) \end{bmatrix} + \begin{bmatrix} C_{a^{N-1}} B_{a^{N-1}}^{(i_1 \dots i_m)} \\ B_{a^{N-1}}^{(i_1 \dots i_m)} \end{bmatrix} \Delta^N u(k) \\
 y(k) &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x_{a^{N-1}}(k) \end{bmatrix}
 \end{aligned} \tag{54}$$

From this, we can express its matrix form using the T–S fuzzy model, taking Equation (47) as a reference:

$$\begin{aligned} S^{(i_1 \dots i_m)}: \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\ x_{a^N}(k+1) = A_{a^N}^{(i_1 \dots i_m)} x_{a^N}(k) + B_{a^N}^{(i_1 \dots i_m)} \Delta^N u(k) \\ y(k) = C_{a^N} x_{a^N}(k) \end{aligned} \quad (55)$$

where $A_{a^N}^{(i_1 \dots i_m)} \in \mathbb{R}^{((Nq+n) \times (Nq+n))}$, $B_{a^N}^{(i_1 \dots i_m)} \in \mathbb{R}^{((Nq+n) \times p)}$ and $C_{a^N} \in \mathbb{R}^{(q \times (Nq+n))}$, and they are calculated using the iterative method presented in Equations (44) and (45). Then, a discrete LQR controller is used, the calculation of which will use the state matrices A_a^N and B_a^N , as well as the matrices $Q \in \mathbb{R}^{((Nq+n) \times (Nq+n))}$ and $R \in \mathbb{R}^{(p \times p)}$. For this reason, to obtain the controller and the control action, we start from Equation (50), to which the T–S fuzzy model is applied. In this way, the following is obtained:

$$\begin{aligned} S^{(i_1 \dots i_m)}: \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\ \Delta^N u(k) = K^{(i_1 \dots i_m)} (x_{a^N} - x_{a^N}(k)) \\ \Delta^{(N-1)} u(k) = \Delta^{(N-1)} u(k-1) + \Delta^N u(k) \\ \Delta^{(N-2)} u(k) = \Delta^{(N-2)} u(k-1) + \Delta^{(N-1)} u(k) \\ \vdots \\ \Delta u(k) = \Delta u(k-1) + \Delta^2 u(k) \\ u(k) = u(k-1) + \Delta u(k) \end{aligned} \quad (56)$$

Finally, a fuzzy state observer is used because the state $\Delta^N x(k)$ is not accessible. This observer is obtained by applying the T–S fuzzy model to Equation (52):

$$\begin{aligned} S^{(i_1 \dots i_m)}: \text{ If } z_1(k) \text{ is } M_1^{i_1} \text{ and } \dots \text{ and } z_m(k) \text{ is } M_m^{i_m}, \text{ then} \\ \Delta^N x_e(k+1) = A^{(i_1 \dots i_m)} \Delta^N x_e(k) + B^{(i_1 \dots i_m)} \Delta^N u(k) \\ + H^{(i_1 \dots i_m)} \left(\Delta^{(N-1)} y(k) - \Delta^{(N-1)} y(k-1) - C \Delta^N x_e(k) \right) \end{aligned} \quad (57)$$

where $H^{(i_1 \dots i_m)}$ is calculated using the Equation (34).

9. Extremum Seeking Control

Extremum Seeking Control (ESC) is utilized for the design of certain controllers, offering the significant advantage of enabling optimization based on a cost function without requiring explicit knowledge of the system model. In this approach, the system is treated as a black box, and its performance is optimized solely through evaluations of the cost function. This methodology is particularly advantageous for systems characterized by complex or poorly understood dynamics. In the present work, ESC is employed to optimize the weighting matrices Q and R of a discrete Linear Quadratic Regulator (LQR), with the aim of minimizing a specified cost function to enhance both the transient and steady-state performance of the controlled system.

$$J(\theta) = \frac{1}{T} \int_0^T e^2(t, \theta) dt \quad (58)$$

where $\theta = [\theta_1 \ \theta_2 \ \dots \ \theta_{Nq+n+p}] = [R_1 \ \dots \ R_p \ Q_1 \ Q_2 \ \dots \ Q_{Nq+n}]$, $e = y_r - y(k)$, and T is the simulation time. Remember that $Q \in \mathbb{R}^{(Nq+n) \times (Nq+n)}$ and $R \in \mathbb{R}^{(p \times p)}$ are diagonal matrices:

$$Q = \begin{bmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & Q_{Nq+n} \end{bmatrix} \quad y \quad R = \begin{bmatrix} R_1 & 0 & \dots & 0 \\ 0 & R_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & R_p \end{bmatrix}$$

Furthermore, as explained in [31], it is necessary to know the values of h , ω , α , and γ :

$$\begin{aligned} \omega &= [\omega_1 \ \omega_2 \ \dots \ \omega_{Nq+n+p}] \\ \alpha &= [\alpha_1 \ \alpha_2 \ \dots \ \alpha_{Nq+n+p}] \\ \gamma &= [\gamma_1 \ \gamma_2 \ \dots \ \gamma_{Nq+n+p}] \end{aligned} \quad (59)$$

where $0 < h < 1$, $\omega_i = a^i \pi$ and $0 < a < 1$. The values for α_i and γ_i are appropriately selected, as they, together with the cost function, influence the stability and convergence of the method. These values are used in the equations presented in [32]:

$$\begin{aligned} \zeta(k) &= -h\zeta(k-1) + J(\theta(k-1)) \\ \hat{\theta}_i(k+1) &= \hat{\theta}_i(k) - \gamma_i \alpha_i \cos(\omega_i(k)) [J(\theta(k)) - (1+h)\zeta(k)] \\ \theta_i(k+1) &= \hat{\theta}_i(k+1) + \alpha_i \cos(\omega_i(k+1)) \end{aligned} \quad (60)$$

The steps to implement Extremum Seeking Control are as follows:

1. Define a cost function $J(\theta)$.
2. Specify the initial parameters θ .
3. Select the optimizer parameters h , ω , α and γ .
4. Solve the optimizer equations.
5. Obtain the state feedback matrix K using the Q and R matrices provided by the optimizer.
6. Return to step 4 until the optimal values Q and R are found.
7. Verify whether the obtained result corresponds to a global minimum; if not, return to step 3.

It is worth noting that another advantage of this method is that it eliminates the need for manual tuning of the Q and R matrices, ensuring that they are appropriately selected by optimizing the cost function.

10. Thermal Mixing Tank System

In this section, the application of fuzzy control based on the incremental-N state model is illustrated through the thermal mixing tank case study [15,35,36]. The purpose is to evaluate the system's performance and verify the achievement of zero steady-state error in response to step, ramp, and subsequent input and disturbance signals. The thermal mixing tank system, depicted in (Figure 1), is a multivariable and nonlinear system comprising a tank, two inlet valves (for hot and cold fluids), and an outlet pipe. The primary control objective is to regulate the outlet flow rate Q_s and the outlet temperature T_s to their respective setpoints, under the assumption that the outlet temperature is equal to the temperature within the tank.

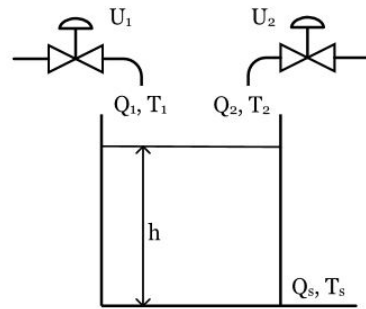


Figure 1. Thermal mixing tank.

The system inputs are two valves: One transports a hot fluid at a constant temperature T_1 with a variable flow rate Q_1 , and the other transports a cold fluid at a constant temperature T_2 with a variable flow rate Q_2 . Both flow rates serve as the manipulated variables of the system, controlled through their respective motorized valves, whose dynamic model is given by

$$\begin{aligned} Q_1 &= \frac{K_1}{1 + \tau_1 s} U_1(s) \\ Q_2 &= \frac{K_2}{1 + \tau_2 s} U_2(s) \end{aligned} \quad (61)$$

Additionally, it is assumed that there are heat losses proportional to the temperature difference between the interior T_s and the exterior T_e of the tank, which act as a disturbance to the system. Thus, the dynamic model is given by

$$Q_1 + Q_2 - Q_s = A \frac{dh}{dt} \quad (62)$$

$$T_1 Q_1 + T_2 Q_2 - T_s Q_s + k_p (T_s - T_e) = A \frac{dT_s h}{dt} \quad (63)$$

$$Q_s = \alpha \sqrt{h} \quad (64)$$

Substituting (64) into (62),

$$Q_1 + Q_2 - \alpha \sqrt{h} = A \frac{dh}{dt} \quad (65)$$

Rewriting (63),

$$T_1 Q_1 + T_2 Q_2 - T_s Q_s + k_p (T_s - T_e) = A \frac{dT_s h}{dt} = AT_s \frac{dh}{dt} + Ah \frac{dT_s}{dt} \quad (66)$$

And substituting (62) into (66),

$$\begin{aligned} T_1 Q_1 + T_2 Q_2 - T_s Q_s + k_p (T_s - T_e) &= T_s (Q_1 + Q_2 - Q_s) + Ah \frac{dT_s}{dt} \\ Q_1 (T_1 - T_s) + Q_2 (T_2 - T_s) + k_p (T_s - T_e) &= Ah \frac{dT_s}{dt} \end{aligned} \quad (67)$$

The model can also be expressed in function of Q_s , starting from (64):

$$Q_s = \alpha \sqrt{h} \rightarrow h = \frac{1}{\alpha^2} Q_s^2 \rightarrow \frac{dh}{dt} = \frac{2}{\alpha^2} Q_s \frac{dQ_s}{dt} \quad (68)$$

Thus, the complete model would be

$$\begin{aligned}
 \frac{dQ_1(t)}{dt} &= \frac{K_1}{\tau_1} u_1(t) - \frac{1}{\tau_1} Q_1(t) \\
 \frac{dQ_2(t)}{dt} &= \frac{K_2}{\tau_2} u_2(t) - \frac{1}{\tau_2} Q_2(t) \\
 \frac{dh}{dt} &= \frac{1}{A} (Q_1 + Q_2 - \alpha \sqrt{h}) \\
 \frac{dQ_s}{dt} &= \frac{\alpha^2}{2Q_s} \frac{dh}{dt} \\
 \frac{dT_s}{dt} &= \frac{1}{Ah} (Q_1(T_1 - T_s) + Q_2(T_2 - T_s) + k_p(T_s - T_e))
 \end{aligned} \tag{69}$$

where the parameters of the thermal mixing tank system are shown in Table 1. Similarly, Table 2 describes the values of the nominal operating point of the system. In (69), it can be seen that Q_s is of order 2 and that T_s is of order 3.

Table 1. Mix tank system parameters.

Parameter	Symbol	Value
Steady state gain of inlet flow rate Q_1	K_1	1 (m ³ /s)
Inlet flow time constant Q_1	τ_1	5
Steady state gain of inlet flow rate Q_2	K_2	1 (m ³ /s)
Inlet flow time constant Q_2	τ_2	4
Mixing tank horizontal section area	A	1 (m ²)
Temperature of inlet flow Q_1	T_1	90 (°C)
Temperature of inlet flow Q_2	T_2	10 (°C)
Mixing tank outside temperature	T_e	15 (°C)
Thermal loss coefficient	k_p	0.01 (m ³ /s)
Discharge coefficient of outlet flow Q_s	α	1 (m ^{2.5} /s)

Table 2. System nominal operating point values.

Parameter	Value
Motorized valve input signal U_1	2
Motorized valve input signal U_2	2
Outlet flow rate Q_s	4 (m ³ /s)
Outlet temperature T_s	50 (°C)

The open-loop simulation of the system was carried out using the Simulink tool from Matlab, with a sampling time of 0.5 s and a total simulation time of 50,000 s. The effect of actuator saturation was explicitly addressed in the simulation framework by constraining the control inputs during the design phase. Specifically, the maximum flow rate for each actuator was limited to 4 (m³/s). Once the system data were obtained, a linear identification was performed using the classical least squares method to estimate an initial set of parameters (P_0) for both the outlet flow rate and the outlet temperature. This identification process is as follows:

$$\begin{aligned}
 Q_s(k+1) &= 0.00007 + 1.8547Q_s(k) - 0.8607Q_s(k-1) + 0.0033U_1(k) \\
 &\quad + 0.0024U_1(k-1) + 0.0038U_2(k) + 0.0021U_2(k-1) \\
 T_s(k+1) &= 0.1838 + 2.4484T_s(k) - 1.9892T_s(k-1) + 0.5373T_s(k-2) \\
 &\quad + 0.1116U_1(k) + 0.0016U_1(k-1) - 0.0748U_1(k-2) \\
 &\quad - 0.1375U_2(k) + 0.0018U_2(k-1) + 0.0952U_2(k-2)
 \end{aligned}$$

The next step is to apply the identification of the T–S fuzzy model. To this end, three fuzzy sets are selected between 1 and 7 (m³/s) for the flow rate, and another three between 20 and 80 (°C) for the temperature. Both membership functions are shown in Figure 2.

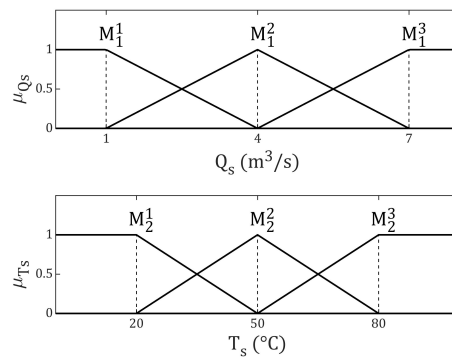


Figure 2. Membership functions for Q_s and T_s .

In this way, the fuzzy model is constructed using a set of 9 rules:

$S^{(1,1)}$: If $Q_s(k)$ is M_1^1 and $T_s(k)$ is M_2^1 , then

$$Q_s(k+1) = 0.0014 + 1.7893Q_s(k) - 0.8034Q_s(k-1) + 0.0070U_1(k) \\ + 0.0055U_1(k-1) + 0.0093U_2(k) + 0.0038U_2(k-1)$$

$$T_s(k+1) = 0.1472 + 2.3728T_s(k) - 1.9048T_s(k-1) + 0.5288T_s(k-2) \\ + 0.5247U_1(k) - 0.0016U_1(k-1) - 0.3395U_1(k-2) \\ - 0.1233U_2(k) + 0.0015U_2(k-1) + 0.0547U_2(k-2)$$

$\vdots$

$S^{(3,3)}$: If $Q_s(k)$ is M_1^3 and $T_s(k)$ is M_2^3 , then

$$Q_s(k+1) = 0.0060 + 1.8593Q_s(k) - 0.8635Q_s(k-1) + 0.0026U_1(k) \\ + 0.0011U_1(k-1) + 0.0037U_2(k) + 0.0009U_2(k-1)$$

$$T_s(k+1) = 0.1894 + 2.4450T_s(k) - 1.9931T_s(k-1) + 0.5449T_s(k-2) \\ + 0.1043U_1(k) + 0.0009U_1(k-1) - 0.0584U_1(k-2) \\ - 0.1893U_2(k) + 0.0109U_2(k-1) + 0.1097U_2(k-2)$$

The next step is to model the nonlinear system using the incremental state T–S model. This is achieved by applying Equation (15) presented in Section 3:

$S^{(1,1)}$: If $Q_s(k)$ is M_1^1 and $T_s(k)$ is M_2^1 , then

$$\begin{bmatrix} y(k+1) \\ \Delta x(k+1) \end{bmatrix} = \begin{bmatrix} I & CA^{(1,1)} \\ 0 & A^{(1,1)} \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix} + \begin{bmatrix} CB^{(1,1)} \\ B^{(1,1)} \end{bmatrix} \Delta u(k) \\ y(k) = \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix}$$

$\vdots$

$S^{(3,3)}$: If $Q_s(k)$ is M_1^3 and $T_s(k)$ is M_2^3 , then

$$\begin{bmatrix} y(k+1) \\ \Delta x(k+1) \end{bmatrix} = \begin{bmatrix} I & CA^{(3,3)} \\ 0 & A^{(3,3)} \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix} + \begin{bmatrix} CB^{(3,3)} \\ B^{(3,3)} \end{bmatrix} \Delta u(k) \\ y(k) = \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix}$$

where $A^{(1,1)}, B^{(1,1)}, \dots, A^{(3,3)}, B^{(3,3)}$ and C are calculated using Equations (2)–(4). So,

$$A^{(1,1)} = \begin{bmatrix} 1.7893 & 1 & 0 & 0 & 0 \\ -0.8034 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.3728 & 1 & 0 \\ 0 & 0 & -1.9048 & 0 & 1 \\ 0 & 0 & 0.5288 & 0 & 0 \end{bmatrix}, \quad B^{(1,1)} = \begin{bmatrix} 0.0070 & 0.0093 \\ 0.0055 & 0.0038 \\ 0.5247 & -0.1233 \\ -0.0016 & 0.0015 \\ -0.3395 & 0.0547 \end{bmatrix}$$

$$\vdots$$

$$A^{(3,3)} = \begin{bmatrix} 1.8593 & 1 & 0 & 0 & 0 \\ -0.8635 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.4450 & 1 & 0 \\ 0 & 0 & -1.9931 & 0 & 1 \\ 0 & 0 & 0.5449 & 0 & 0 \end{bmatrix}, \quad B^{(3,3)} = \begin{bmatrix} 0.0026 & 0.0037 \\ 0.0011 & 0.0009 \\ 0.1043 & -0.1893 \\ 0.0009 & 0.0109 \\ -0.0584 & 0.1097 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

In the same way, the matrix notation of the T-S incremental state model is obtained using Equation (16), thus obtaining

$S^{(1,1)}$: If $Q_s(k)$ is M_1^1 and $T_s(k)$ is M_2^1 , then

$$x_a(k+1) = A_a^{(1,1)} x_a(k) + B_a^{(1,1)} \Delta u(k)$$

$$y(k) = C_a x_a(k)$$

$\vdots$

$S^{(3,3)}$: If $Q_s(k)$ is M_1^3 and $T_s(k)$ is M_2^3 , then

$$x_a(k+1) = A_a^{(3,3)} x_a(k) + B_a^{(3,3)} \Delta u(k)$$

$$y(k) = C_a x_a(k)$$

where:

$$A_a^{(1,1)} = \begin{bmatrix} 1 & 0 & 1.7893 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2.3728 & 1 & 0 \\ 0 & 0 & 1.7893 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.8034 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.3728 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1.9048 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0.5288 & 0 & 0 \end{bmatrix}, \quad B_a^{(1,1)} = \begin{bmatrix} 0.0070 & 0.0093 \\ 0.5247 & -0.1233 \\ 0.0070 & 0.0093 \\ 0.0055 & 0.0038 \\ 0.5247 & -0.1233 \\ -0.0016 & 0.0015 \\ -0.3395 & 0.0547 \end{bmatrix}$$

$$\vdots$$

$$A_a^{(3,3)} = \begin{bmatrix} 1 & 0 & 1.8593 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2.4450 & 1 & 0 \\ 0 & 0 & 1.8593 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.8635 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.4450 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1.9931 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0.5449 & 0 & 0 \end{bmatrix}, \quad B_a^{(3,3)} = \begin{bmatrix} 0.0026 & 0.0037 \\ 0.1043 & -0.1893 \\ 0.0026 & 0.0037 \\ 0.0011 & 0.0009 \\ 0.1043 & -0.1893 \\ 0.0009 & 0.0109 \\ -0.0584 & 0.1097 \end{bmatrix}$$

$$C_a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Once the model is obtained, the feedback matrix K for all the fuzzy rules. To do this, the matrices Q and R are used, which are defined as follows:

$$Q = 0.001 \text{diag}(\begin{bmatrix} 0.5 & 0.2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix})$$

$$R = \text{diag}(\begin{bmatrix} 1 & 1 \end{bmatrix})$$

And so, the controller for each fuzzy rule is defined as

$$S^{(1,1)}: \text{ If } Q_s(k) \text{ is } M_1^1 \text{ and } T_s(k) \text{ is } M_2^1, \text{ then}$$

$$\Delta u(k) = K^{(1,1)}(x_{ar} - x_a(k))$$

$$= \begin{bmatrix} 0.0064 & 0.0122 & 0.3929 & 0.3932 & \dots & 0.5190 \\ 0.0211 & -0.0037 & 1.3570 & 1.3638 & \dots & -0.2250 \end{bmatrix} \begin{bmatrix} y_r - y(k) \\ -\Delta x(k) \end{bmatrix}$$

$$\vdots$$

$$S^{(3,3)}: \text{ If } Q_s(k) \text{ is } M_1^3 \text{ and } T_s(k) \text{ is } M_2^3, \text{ then}$$

$$\Delta u(k) = K^{(3,3)}(x_{ar} - x_a(k))$$

$$= \begin{bmatrix} 0.0185 & 0.0073 & 3.4214 & 3.4035 & \dots & 0.7041 \\ 0.0122 & -0.0110 & 2.1886 & 2.1747 & \dots & -0.9201 \end{bmatrix} \begin{bmatrix} y_r - y(k) \\ -\Delta x(k) \end{bmatrix}$$

Finally, remembering that the state $\Delta x(k)$, we must calculate the matrix H of the state observer for each fuzzy rule using Equation (34) of Section 6:

$$S^{(1,1)}: \text{ If } Q_s(k) \text{ is } M_1^1 \text{ and } T_s(k) \text{ is } M_2^1, \text{ then}$$

$$H^{(1,1)} = A^{(1,1)}C^t = \begin{bmatrix} 1.7893 & 0 \\ -0.8034 & 0 \\ 0 & 2.3728 \\ 0 & -1.9048 \\ 0 & 0.5288 \end{bmatrix}$$

$$\vdots$$

$$S^{(3,3)}: \text{ If } Q_s(k) \text{ is } M_1^3 \text{ and } T_s(k) \text{ is } M_2^3, \text{ then}$$

$$H^{(1,1)} = A^{(1,1)}C^t = \begin{bmatrix} 1.8593 & 0 \\ -0.8635 & 0 \\ 0 & 2.4450 \\ 0 & -1.9931 \\ 0 & 0.5449 \end{bmatrix}$$

Based on the results obtained, a fuzzy controller using the incremental state model was successfully implemented for the thermal mixing tank system. To evaluate the performance of the fuzzy controller on the nonlinear system, changes were applied to the reference inputs of Q_s and T_s , as well as disturbances affecting both variables. Figure 3 show the responses of Q_s and T_s to step changes: One step input is applied to Q_s at 10 s and another to T_s at 150 s, along with step-type disturbances. The disturbance affecting Q_s occurs at 250 s with an amplitude of 0.5 (m³/s), while the disturbance affecting T_s occurs at 350 s with an amplitude of 5 (°C). Moreover, in Figure 4, it can be observed that the system can

withstand excessive changes in the reference input, which confirms its correct operation even in regions far from its nominal operating point.

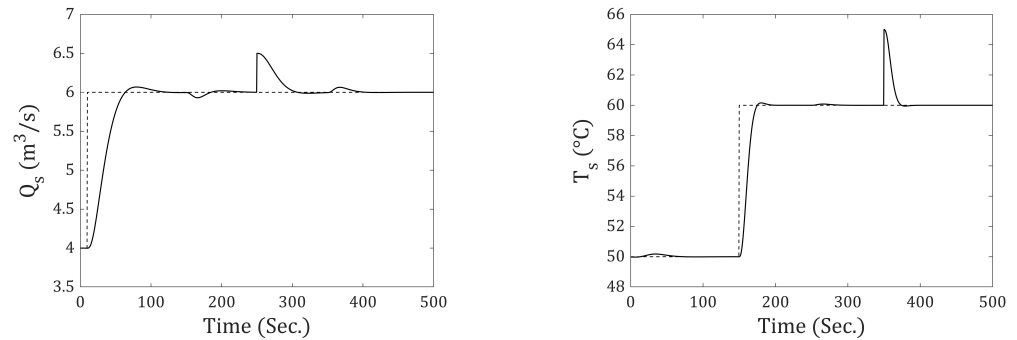


Figure 3. Response of outlet flow rate Q_s and outlet temperature T_s to reference signals and disturbances, both of the step type, using an incremental-1 control.

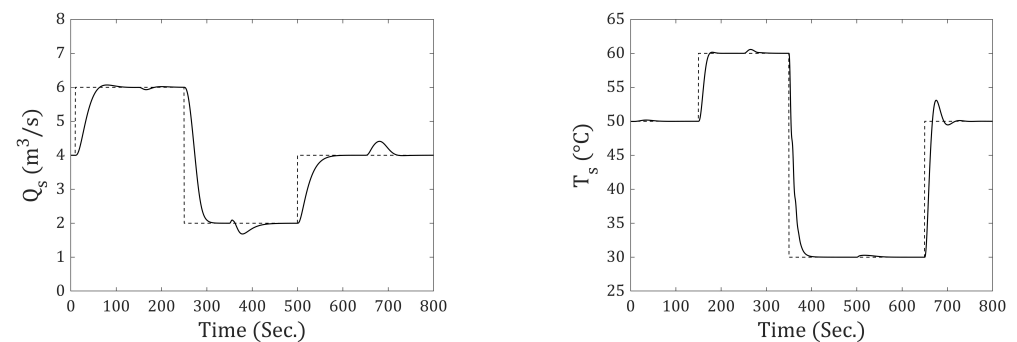


Figure 4. Response of Q_s and T_s to excessive step changes in reference signal using incremental-1 control.

It can be observed that a zero steady-state error has been achieved in response to step-type inputs and disturbances for a nonlinear system using a fuzzy control based on the incremental state model. However, the Q and R values are not optimal; in other words, a better response can still be obtained. A clear example would be to make the system faster while ensuring that the overshoot peaks are not excessively high. To obtain these optimal values, the ESC (Extremum Seeking Control) explained in Section 9 is used, which aims to minimize the cost function shown in Equation (58), corresponding to the MSE (Mean Squared Error).

As illustrated in the Figure 5, the ESC algorithm iteratively updates the controller parameters $\theta(k)$ based on the current value of the cost function $J(\theta(k))$. At each iteration, the algorithm evaluates $J(\theta(k))$ using the updated parameters and subsequently adjusts $\theta(k)$ to minimize the cost. This procedure is repeated until convergence is achieved, i.e., when $J(\theta(k))$ reaches a local minimum, thereby yielding an optimal parameter configuration.

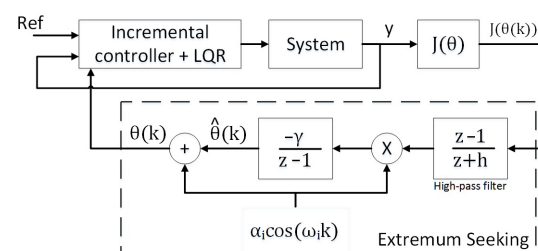


Figure 5. Block diagram of the system.

This optimization is achieved by periodically perturbing the system, usually using a sinusoidal wave, and then estimating the gradient $\nabla J(\theta(k))$. The gradient is computed

by first filtering the discrete-time signal $J(\theta(k))$ using a high-pass filter to remove the DC component. Next, the signal is demodulated by multiplying it with a discrete sinusoidal wave of the same frequency as the perturbation signal. Finally, the result is integrated using a step size γ to obtain a new estimate of the parameter $\hat{\theta}$. It is worth noting that the integrator acts as a low-pass filter.

For the case of the thermal mixing tank system, the following cost function will be used:

$$J(\theta) = \frac{1}{T} \left[\int_0^T e_{Q_s}^2(t, \theta) dt + 0.01 \int_0^T e_{T_s}^2(t, \theta) dt \right]$$

where $\theta = [R_1 \ R_2 \ Q_1 \ Q_2 \ Q_3 \ \dots \ Q_7]$ and the scaling factor 0.01 is introduced to ensure that the output flow rate error (e_{Q_s}) and the output temperature error (e_{T_s}) contribute comparably to the cost function. In general, the magnitude of the temperature signal is approximately 10 times larger than that of the flow rate. As the cost function involves squared errors, this discrepancy is further amplified—potentially causing the temperature error to contribute up to 100 times more than the flow rate error. To address this imbalance, a weight of 0.01 is applied to the temperature term, effectively normalizing both error contributions to a similar scale. Therefore, the chosen scaling factor is not arbitrary; it ensures that both variables influence the optimization criterion in a balanced manner, despite their differing physical units and dynamic ranges. Once the cost function is defined, the initial values for the Q and R matrices are selected:

$$Q = 0.001 \text{diag} \left(\begin{bmatrix} 0.1 & 0.1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \right) \\ R = \text{diag} \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \right)$$

The value of 0.001 was chosen through iterative testing because if it were larger, it caused failures in the controller; on the other hand, if it was too small, a good response was not obtained. Since R is the weighting matrix for the control action of the system and its variables have the same weight, R_1 will change in the same way as R_2 . Similarly, for the Q matrix, only the values of Q_1 , Q_2 and Q_3 are varied, since multiple simulations have shown that variations in the remaining terms do not significantly affect the performance of the control system. In fact, it has been verified that the values from Q_3 to Q_7 can be set identically without degrading control quality. Thus, the configuration is as follows:

$$Q_{\text{initial}} = \begin{bmatrix} R_1 & Q_1 & Q_2 & Q_3 \end{bmatrix} = \begin{bmatrix} 1 & 10^{-4} & 10^{-4} & 10^{-3} \end{bmatrix}$$

This simplifies the simulation run time and makes it less complicated to select appropriate values for ω , α and γ . It should be noted that α_i and γ_i influence the stability and convergence of the method, and that $\omega_i = a^i \pi$ where $0 < a < 1$. The optimizer values are obtained through an iterative search to ensure that the method is stable and able to converge. These values are

$$h = 0.6 \\ \omega = \begin{bmatrix} 2.8274 & 2.5447 & 2.2902 & 2.0612 \end{bmatrix} \\ \alpha = \begin{bmatrix} 0 & 2 \times 10^{-4} & 0.8 \times 10^{-5} & 10^{-5} \end{bmatrix} \\ \gamma = \begin{bmatrix} 10 & 10 & 10 & 10 \end{bmatrix}$$

where a value of $a = 0.9$ was chosen to calculate ω_i . It can also be seen that $\alpha_1 = 0$, since it is intended to keep $R_1 = R_2 = 1$ so that their values remain unchanged throughout the entire iteration. This serves as a reference and allows the focus to be placed solely on the Q matrix. It is worth noting that it has been verified that the optimization is not affected if the value of R_1 remains constant.

Since the goal is to obtain the values that optimize the cost function, it is recommended to apply ESC globally. That is, the system should be subjected to large variations in the reference input for both output flow rate and output temperature across its entire operating range. Table 3 shows the selected points, establishing the path from point 1 to point 11 and then returning back to point 1.

Table 3. Points selected for the optimization of the ESC.

Point	Q_s	T_s
1	4 (m ³ /s)	50 (°C)
2	7 (m ³ /s)	50 (°C)
3	1 (m ³ /s)	50 (°C)
4	1 (m ³ /s)	80 (°C)
5	4 (m ³ /s)	80 (°C)
6	4 (m ³ /s)	20 (°C)
7	1 (m ³ /s)	20 (°C)
8	1 (m ³ /s)	50 (°C)
9	5.5 (m ³ /s)	50 (°C)
10	5.5 (m ³ /s)	65 (°C)
11	5.5 (m ³ /s)	35 (°C)

Figure 6 shows the convergence of $J(\theta)$ and the values obtained for the Q matrix. It can be observed that the value of Q_1 continues to increase; however, the value obtained at iteration number 500 is taken, since $J(\theta)$ no longer decreases. The resulting Q and R values are as follows:

$$Q = 0.001 \text{diag} \left(\begin{bmatrix} 7.5 & 0.18 & 0.22 & 0.22 & 0.22 & 0.22 & 0.22 \end{bmatrix} \right)$$

$$R = \text{diag} \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \right)$$

The feedback matrix K is obtained for each fuzzy rule, which is defined as

$S^{(1,1)}$: If $Q_s(k)$ is M_1^1 and $T_s(k)$ is M_2^1 , then

$$K^{(1,1)} = \begin{bmatrix} 0.0307 & 0.0113 & 1.4218 & \dots & 0.4782 \\ 0.0778 & -0.0046 & 4.1532 & \dots & -0.2984 \end{bmatrix}$$

$$\vdots$$

$S^{(3,3)}$: If $Q_s(k)$ is M_1^3 and $T_s(k)$ is M_2^3 , then

$$K^{(3,3)} = \begin{bmatrix} 0.0674 & 0.0076 & 9.0385 & \dots & 0.7719 \\ 0.0504 & -0.0100 & 6.2131 & \dots & -0.8202 \end{bmatrix}$$

Figures 7 and 8 show the responses of flow rate and output temperature to excessive step-type input throughout the operating range of the nonlinear system, comparing the system when ESC is not used with the values obtained for the matrices Q and R using Global ESC.

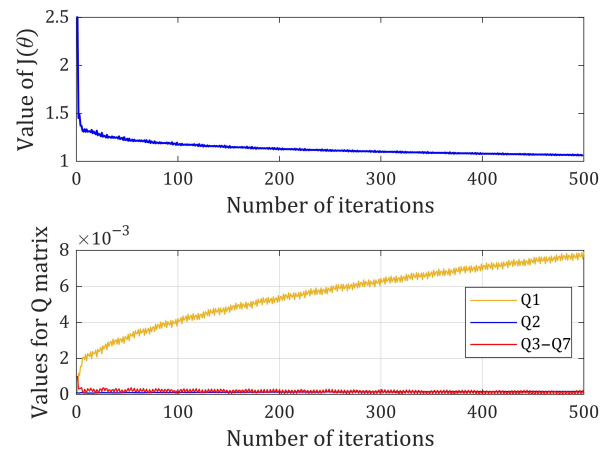


Figure 6. Convergence of the cost function and values for the matrix Q .

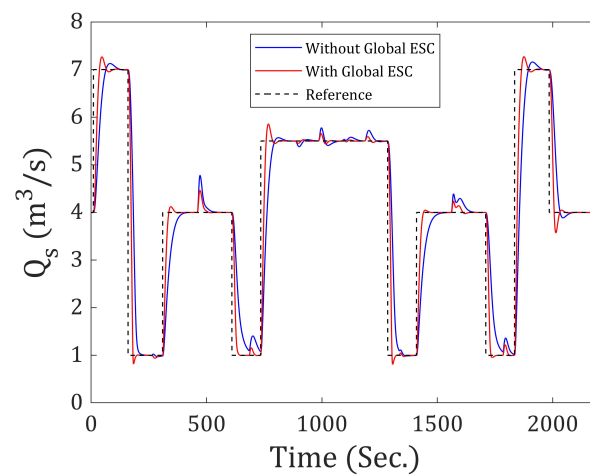


Figure 7. Comparison of the response of Q_s response to excessive step-type reference signals without Global ESC and with Global ESC using an incremental-1 control.

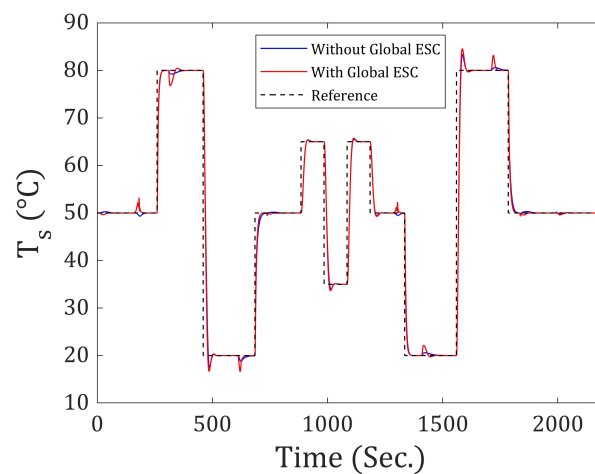


Figure 8. Comparison of the response of T_s response to excessive step-type reference signals without Global ESC and with Global ESC using an incremental-1 control.

As shown in Figure 7, by using Global ESC, a better response is obtained in the output flow rate. It is faster and exhibits a smaller amplitude in the peaks that occur when there is an abrupt temperature change and the flow rate remains constant. On the other hand, in Figure 8, it can be seen that using ESC slightly increases the peaks in the temperature response when it remains constant and the system is subjected to an abrupt change in the

flow rate. However, a faster response is achieved. This can be better understood by looking at Table 4, which shows the cost function $J(\theta)$ and the RMSE (Root Mean Square Error) value for Q_s and T_s , comparing the method used to obtain the matrices Q and R .

Table 4. Value of $J(\theta)$ and RMSE of Q_s and T_s using the matrix values obtained without Global ESC and with Global ESC.

Method for Obtaining Q and R Matrices	$J(\theta)$	RMSE (Q_s)	RMSE (T_s)
Without Global ESC	1.6660	1.1355	6.1335
With Global ESC	1.0691	0.8373	6.0641

This confirms that the application of ESC leads to an improved and faster response under excessive step inputs. Moreover, this method simplifies the determination of the optimal values for the matrices Q and R .

11. Responses for the Incremental-N

In a similar manner, this procedure can be applied to the incremental-N approach in order to analyze the system's behavior at each incremental stage under reference changes and output disturbances, as well as input disturbances and measurement noise.

11.1. Step-Type Reference and Output Disturbance

The results are presented for the system subjected to step-type reference and disturbance inputs. A simulation time of 3500 s was used for all cases. The step input for the output flow rate (Q_s) occurs at 710 s, and for the output temperature (T_s), it occurs at 1200 s. The disturbance for Q_s has an amplitude of 0.5 (m^3/s) and occurs at 1800 s. For T_s , the disturbance has an amplitude of 5 ($^\circ\text{C}$) and occurs at 2800 s.

Figure 9 presents the output flow rate and temperature responses using the traditional discrete-state model (incremental-0). In both cases, it can be seen that the system fails to achieve zero steady-state error. From Figures 10–14, the responses obtained using incremental-1 through incremental-5 are shown. It is demonstrated that all of them achieve zero steady-state error in response to step-type references and disturbances (at the output). The difference among the incrementals lies in the response speed: As the incremental number increases, the response becomes slower. To better understand this behavior, Table 5 presents the IAE and ISE values for the output flow rate and temperature for all incrementals.

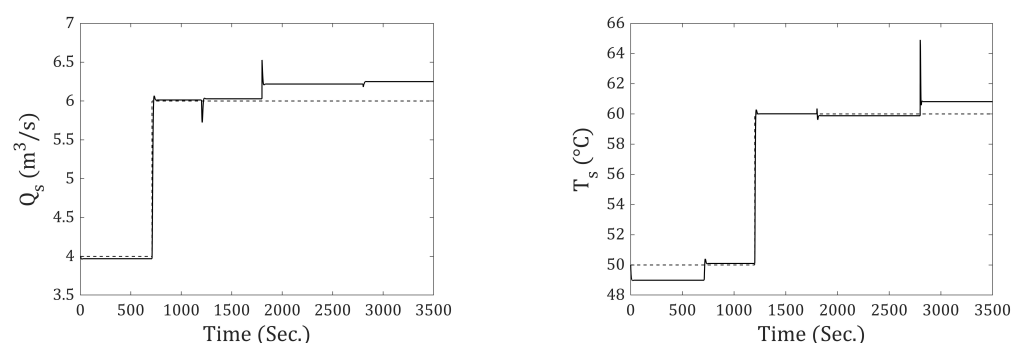


Figure 9. Response of Q_s and T_s to step-type references and output disturbances for incremental-0.

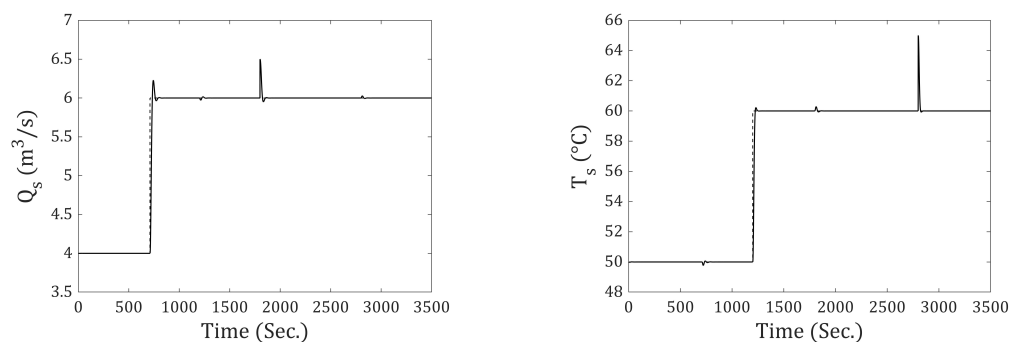


Figure 10. Response of Q_s and T_s to step-type references and output disturbances for incremental-1.

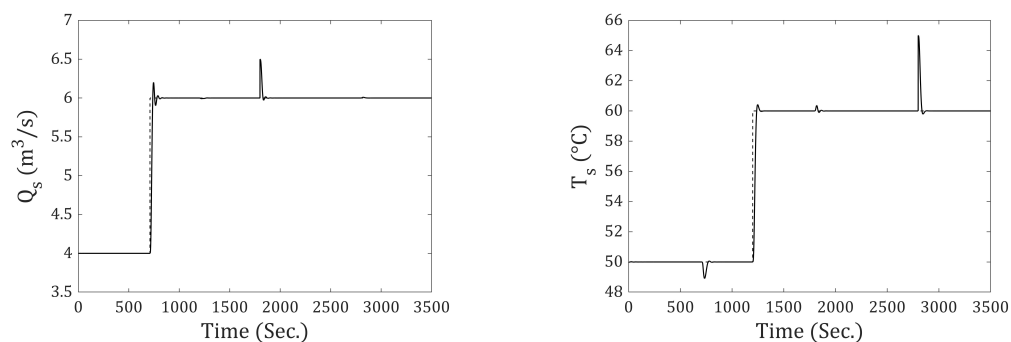


Figure 11. Response of Q_s and T_s to step-type references and output disturbances for incremental-2.

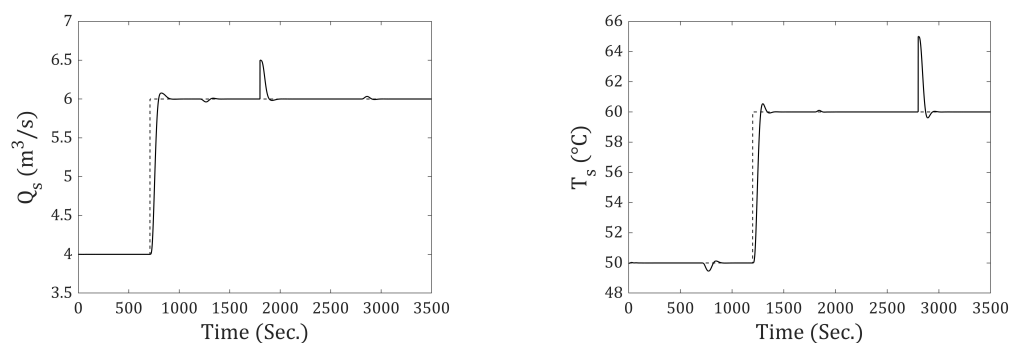


Figure 12. Response of Q_s and T_s to step-type references and output disturbances for incremental-3.

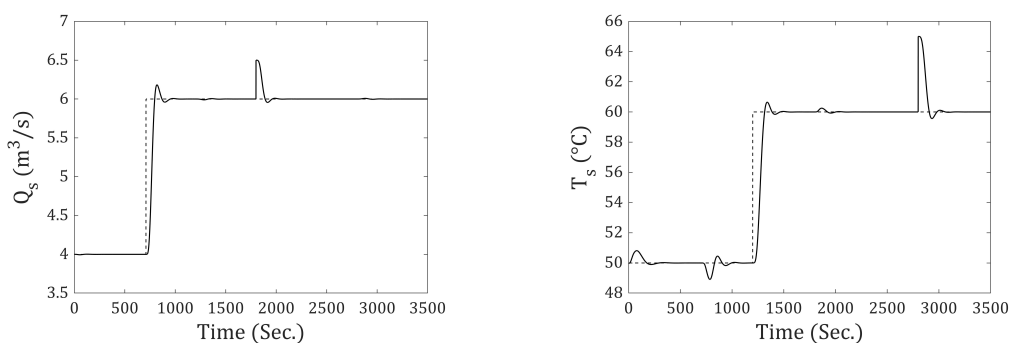


Figure 13. Response of Q_s and T_s to step-type references and output disturbances for incremental-4.

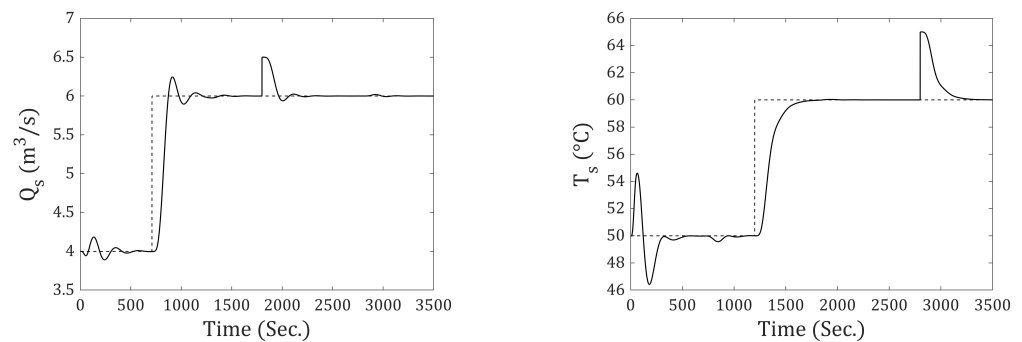


Figure 14. Response of Q_s and T_s to step-type references and output disturbances for incremental-5.

Table 5. IAE and ISE values of Q_s and T_s for all incrementals under step-type references and output disturbances.

Incremental	IAE (Q_s)	IAE (T_s)	ISE (Q_s)	ISE (T_s)
0	457.44	1532	115.73	1628
1	39.34	186	42.1	1075
2	49.24	366	58.92	2075
3	132.37	754	163.44	4530
4	154.48	1328	197.14	7446
1	344.14	3118	401.73	16,094

11.2. Ramp-Type Reference and Output Disturbance

Ramp-type references and disturbances are also commonly used in the field of control, whether to smoothly start a motor or to reach a certain value in a less abrupt manner. Below are the results obtained when the system is subjected to ramp-type reference signals and output disturbances. A simulation time of 5000 s was used for all cases.

The ramp-type reference for the flow rate begins at 710 s and ends at 1110 s, with a slope of $0.005 \text{ (m}^3/\text{s}^2)$. For temperature, the ramp starts at 1410 s and ends at 1810 s, with a slope of $0.025 \text{ (}^\circ\text{C/s)}$. In both cases, the final value reached is maintained until the end of the simulation.

The ramp-type disturbance for the flow rate starts at 2300 s and ends at 2900 s, with a slope of $0.000833 \text{ (m}^3/\text{s}^2)$. For temperature, the ramp begins at 3500 s and ends at 4300 s, with a slope of $0.00625 \text{ (}^\circ\text{C/s)}$. In both cases, the final value reached is maintained until the end of the simulation.

Figure 15 presents the results using incremental-1. This incremental does not achieve zero steady-state error in response to ramp-type references and output disturbances. It can be observed that the response attempts to follow the ramp but consistently exhibits an error. In contrast, when using incremental-2, zero steady-state error is achieved in response to ramp-type references and output disturbances (see Figure 16). The same behavior is observed with incremental-3 (see Figure 17) and beyond.

11.3. Second-Order Type Reference and Output Disturbance

In order to verify the performance of incremental-3, the system is subjected to second-order output references and disturbances. The results are presented for a simulation that spans 5000 s.

The second-order reference for the flow rate starts at 710 s and ends at 885 s, ranging from $4 \text{ (m}^3/\text{s})$ to $6 \text{ (m}^3/\text{s})$. For temperature, it ranges from $50 \text{ (}^\circ\text{C)}$ to $60 \text{ (}^\circ\text{C)}$, starting at 1410 s and ending at 1585 s. In both cases, the final value is maintained until the end of the simulation.

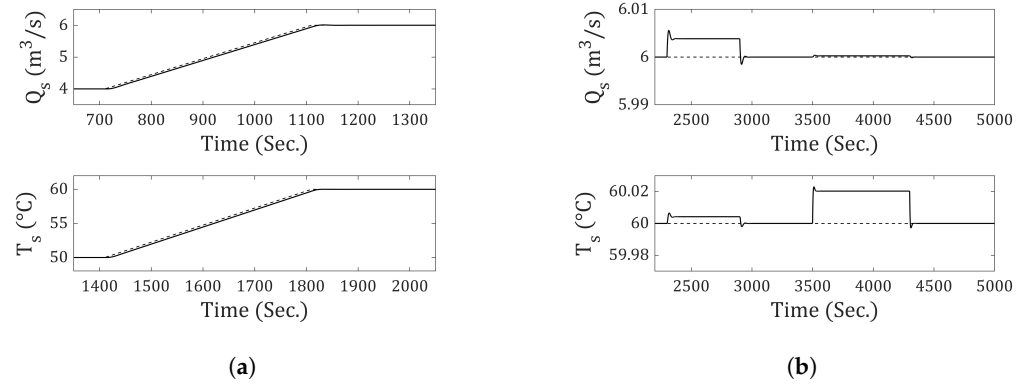


Figure 15. Response of Q_s and T_s to ramp-type references and output disturbances for incremental-1: (a) ramp-type reference, (b) output disturbances.

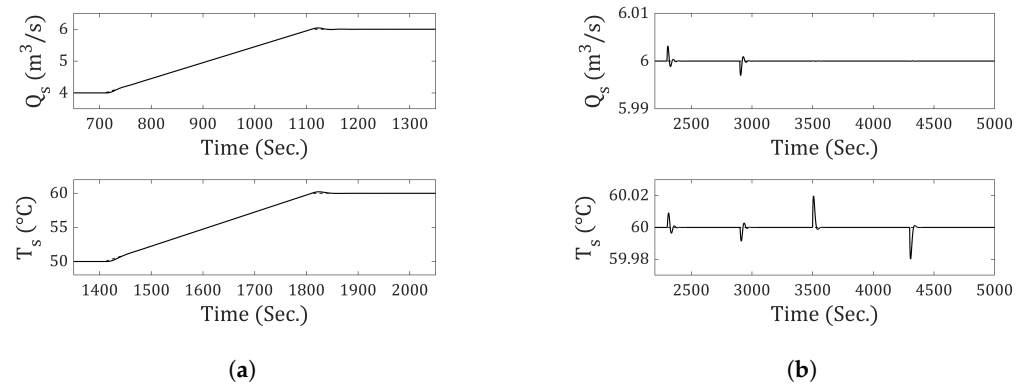


Figure 16. Response of Q_s and T_s to ramp-type references and output disturbances for incremental-2: (a) ramp-type reference, (b) output disturbances.

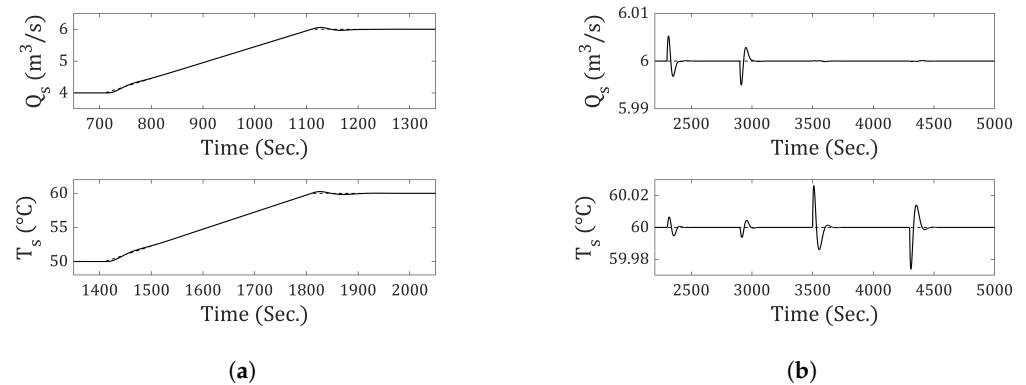


Figure 17. Response of Q_s and T_s to ramp-type references and output disturbances for incremental-3: (a) ramp-type reference, (b) output disturbances.

The second-order output disturbances for the flow rate begin at 2400 s and end at 3000 s, ranging from 0 (m³/s) to 0.5 (m³/s). For temperature, the disturbance ranges from 0 (°C) to 5 (°C), starting at 3400 s and ending at 4200 s. In both cases, the final value is maintained until the end of the simulation.

Figure 18 presents the results obtained using incremental-2. This incremental does not achieve zero steady-state error in response to second-order output references and disturbances. The response attempts to follow the parabolic input but consistently exhibits an error.

In contrast, when using incremental-3, zero steady-state error is achieved for second-order output references and disturbances (see Figure 19). It can be observed that the response successfully follows the parabolic trajectory in both the flow rate and the output temperature.

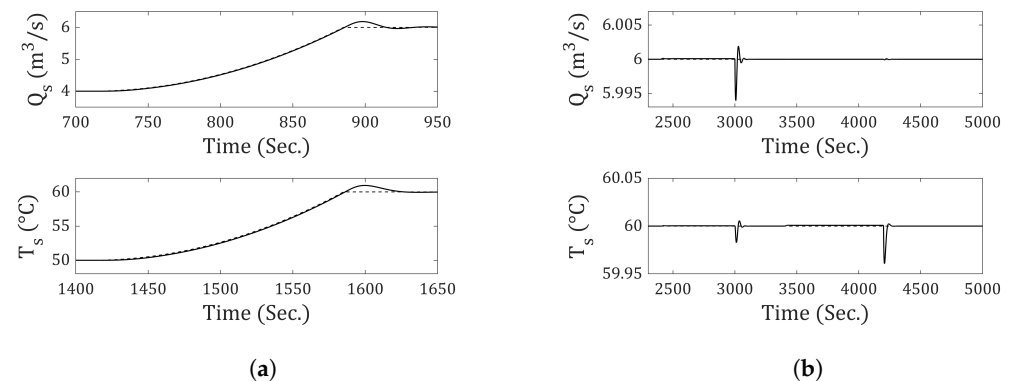


Figure 18. Response of Q_s and T_s to second-order type references and output disturbances for incremental-2: (a) second-order type reference, (b) output disturbances.

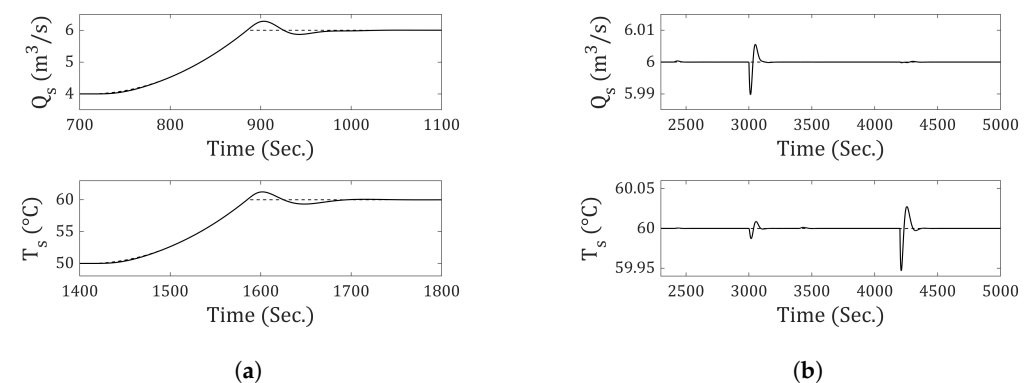


Figure 19. Response of Q_s and T_s to second-order type references and output disturbances for incremental-3: (a) second-order type reference, (b) output disturbances.

11.4. Third-Order Type Reference and Output Disturbance

Similarly, the performance of the incremental-4 will be evaluated by subjecting the system to third-order reference inputs and output disturbances. The results are presented for a simulation time of 5000 s.

The third-order input for the flow rate starts at 710 s and ends at 885 s, increasing from 4 (m³/s) to 6 (m³/s). In the case of temperature, it rises from 50 (°C) to 60 (°C), starting at 1410 s and ending at 1585 s. In both cases, the final value reached is maintained until the end of the simulation.

The third-order output disturbances for the flow rate begin at 2400 s and end at 3000 s, ranging from 0 (m³/s) to 0.5 (m³/s). For temperature, the disturbance goes from 0 (°C) to 5 (°C), starting at 3400 s and ending at 4200 s. In both cases, the final value is held constant until the simulation ends.

Figure 20 presents the results obtained using the incremental-4 approach. This incremental achieves zero steady-state error in response to third-order references and output disturbances. The system output successfully follows the reference in both flow rate and temperature.

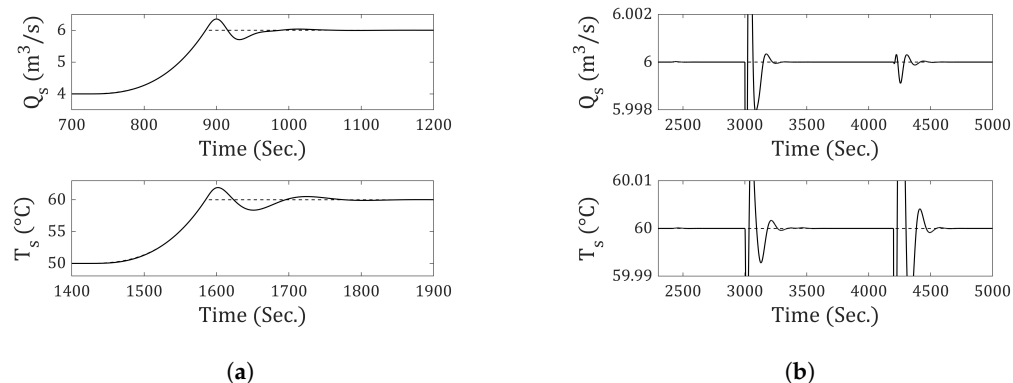


Figure 20. Response of Q_s and T_s to third-order type references and output disturbances for incremental-4: (a) third-order type reference, (b) output disturbances.

11.5. Fourth-Order Type Reference and Output Disturbance

Finally, the performance of incremental-5 is evaluated by subjecting the system to fourth-order reference inputs and output disturbances. The simulation spans a duration of 6000 s.

The fourth-order input for the flow rate starts at 710 s and ends at 985 s, increasing from 4 (m³/s) to 6 (m³/s). In the case of temperature, it rises from 50 (°C) to 60 (°C), starting at 1610 s and ending at 1810 s. In both cases, the final value reached is maintained until the end of the simulation.

The fourth-order output disturbances for flow begin at 3000 s and end at 3600 s, ranging from 0 (m³/s) to 0.5 (m³/s). For temperature, the disturbance ranges from 0 (°C) to 5 (°C), starting at 4000 s and ending at 4800 s. In both cases, the final value reached is maintained until the end of the simulation. Figure 21 presents the results obtained using incremental-5. This incremental achieves zero steady-state error in response to fourth-order output references and disturbances. The system's response accurately follows the reference in both output flow and temperature.

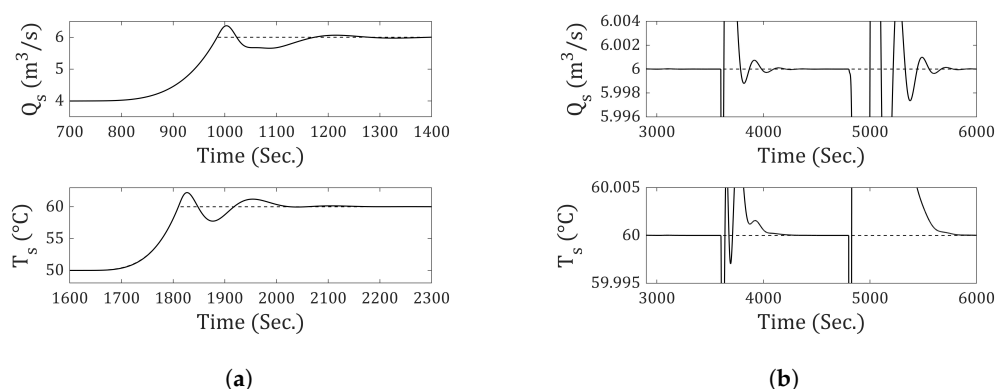


Figure 21. Response of Q_s and T_s to fourth-order type references and output disturbances for incremental-5: (a) fourth-order type reference, (b) output disturbances.

11.6. Input Disturbances

Disturbances can occur not only at the output of the system but also at the input. Therefore, it is important to analyze all possible disturbances that the system may experience.

The system will be analyzed starting from the point where it reaches the state [$Q_s = 6$ (m³/s), $T_s = 60$ (°C)]. Once this state is reached, disturbances will begin to be applied to the motorized valve signals U_1 and U_2 . The step-type input disturbances for U_1 and U_2 occur at 1800 and 2800 s, respectively, each with an amplitude of 0.1. For the remaining disturbances (ramp, second-order, third-order, and fourth-order), the U_1

signal starts at 1800 s and ends at 2200 s, while U_2 starts at 2800 s and ends at 3200 s. All disturbances have an amplitude of 0.1. The simulation time for incrementals-1, 2, and 3 is 3600 s.

Figure 22 shows the responses of Q_s and T_s . It can be observed that under a step-type input disturbance, the system exhibits zero steady-state error. However, as the order of the disturbance increases, the error also increases. Similarly, Figure 23 shows the responses of incremental-2, which exhibit zero steady-state error under a ramp-type disturbance. Conversely, as with incremental-1, the error increases as the order of the disturbance increases. Figure 24 displays the system's response using incremental-3. In this way, zero steady-state error is achieved for disturbances of order N-1 when using incremental-N.

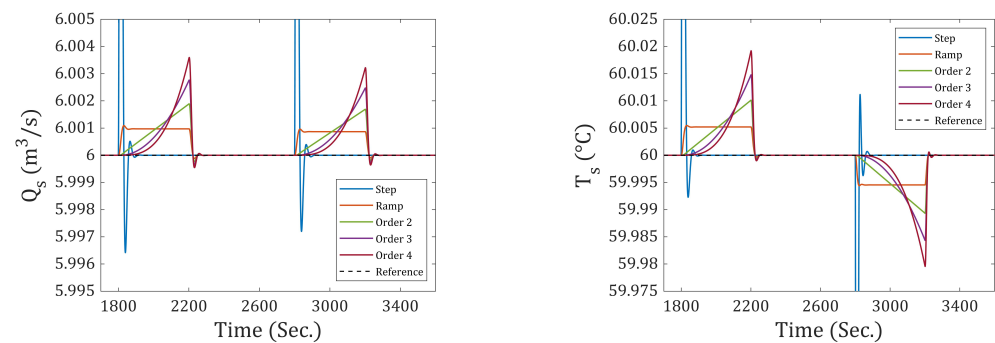


Figure 22. Response of Q_s and T_s to input disturbances for Incremental-1.

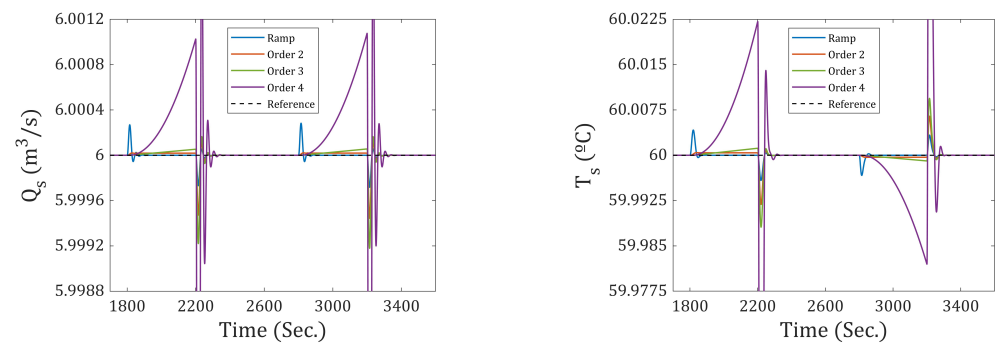


Figure 23. Response of Q_s and T_s to input disturbances for Incremental-2.

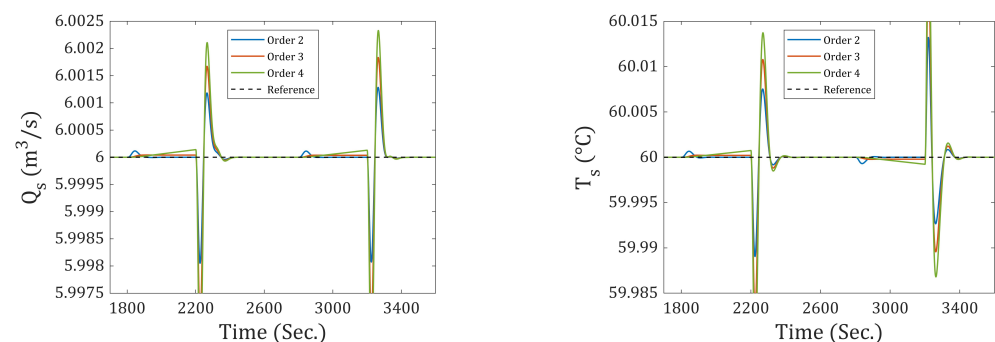


Figure 24. Response of Q_s and T_s to input disturbances for Incremental-3.

11.7. Load Disturbances

Load disturbances occur when changes affect the system's behavior by modifying one of its parameters, such as the discharge coefficient of the outlet flow rate Q_s (α), or the temperature of one of the inlet flow rates (T_1 or T_2). The following section details these possible load disturbances and shows how the system responds to them:

- Variation in the parameter (α)

In this study, a value of (α) equal to 1 was used. In this scenario, this parameter will be disturbed such that at second 200 it decreases to a value of 0.1. This simulates a step-type disturbance, which, in real terms, would represent an object obstructing the outlet duct. For this case, the incremental-1 state model will be used, and the system will start from the point $[Q_s = 6 \text{ (m}^3/\text{s)}, T_s = 60 \text{ (}^\circ\text{C)}]$.

Figure 25 shows how the outlet flow rate decreases due to the obstruction, causing the system to fully open both inlet flow rates and the temperature to approach $50 \text{ (}^\circ\text{C)}$. As time progresses, the flow rate increases, and the temperature also rises until reaching the desired value with zero steady-state error.

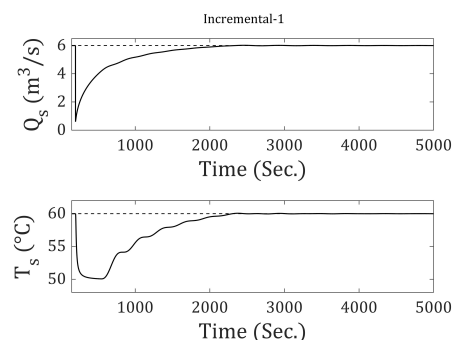


Figure 25. Responses of Q_s and T_s to variations in the parameter (α) using Incremental-1.

- Variation in the inlet flow temperatures (T_1 and T_2)

Another type of load disturbance that may occur is a variation in the temperature of one of the inlet flows. For example, a malfunction in the boiler that heats the flow Q_1 could cause its temperature to change from $90 \text{ (}^\circ\text{C)}$ to $60 \text{ (}^\circ\text{C)}$. This disturbance begins at 600 s and ends at 1200 s. It simulates a ramp-type disturbance, which, in practice, represents a gradual temperature decrease due to the failure. For this case, the incremental-2 state model will be used, starting from the point where the system reaches $[Q_s = 2 \text{ (m}^3/\text{s)}, T_s = 40 \text{ (}^\circ\text{C)}]$ (See Figure 26).

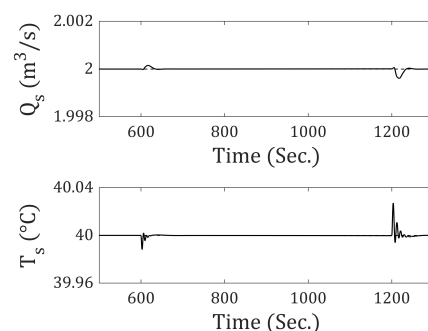


Figure 26. Responses of Q_s and T_s to variations in the parameter T_1 using Incremental-2.

11.8. Noise

In this section, the system will be subjected to noise in both Q_s and T_s in order to analyze its behavior.

Figure 27 shows the responses of incremental-0 to step-type references for Q_s and T_s , along with output noise of $N(0, 10^{-3})$ and $N(0, 10^{-2})$, respectively, where the first term is the mean of the distribution and the second is the variance. Also, Figure 27 illustrates the system's behavior for incremental-1 under the same inputs and noise.

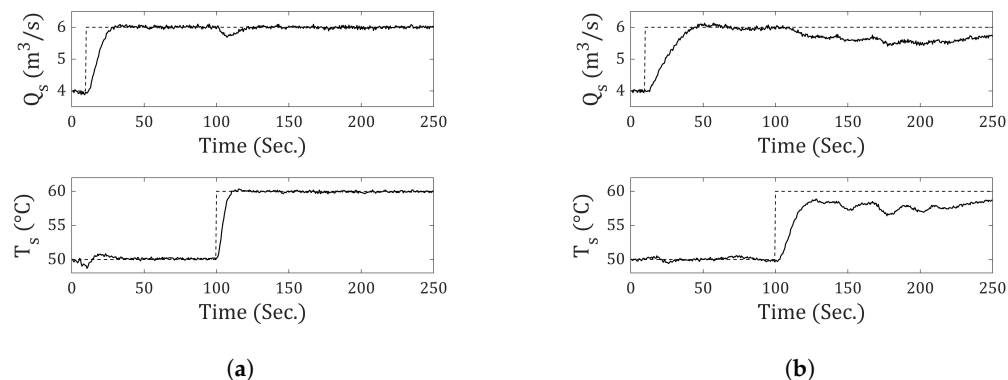


Figure 27. Responses of Q_s and T_s subjected to step-type references and noise effects for Incremental-0 and Incremental-1: (a) incremental-0, (b) incremental-1.

It is clearly observed that Incremental-1 fails under the same noise conditions. This is because, as more incrementals are added, the noise is amplified due to the difference between the current and previous samples, which accumulates and becomes larger for the next incremental. On the other hand, if the output noise is $N(0, 10^{-4})$ for Q_s and $N(0, 10^{-3})$ for T_s (10 times lower than in the previous case), a better response is obtained (see Figure 28).

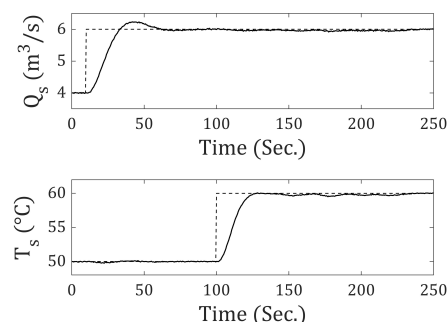


Figure 28. Responses of Q_s and T_s subjected to step-type references and noise effects (reduced variance) for Incremental-1.

Figure 29 shows the response of incremental-2 to ramp-type references for Q_s and T_s , as well as output noise of $N(0, 10^{-5})$ and $N(0, 10^{-4})$, respectively. Also, Figure 29 shows the response of incremental-3 to second-order references for Q_s and T_s , along with output noise of $N(0, 2 \times 10^{-7})$ and $N(0, 2 \times 10^{-6})$, respectively.

Meanwhile, Figure 30 shows the response of incremental-4 to successive third-order references for Q_s and T_s , along with output noise of $N(0, 9 \times 10^{-10})$ and $N(0, 9 \times 10^{-9})$, respectively. Also, Figure 30 presents the response of incremental-5 to successive fourth-order inputs for Q_s and T_s , along with output noise of $N(0, 3 \times 10^{-12})$ and $N(0, 3 \times 10^{-11})$, respectively.

It can be observed that as the number of incrementals increases, the noise variance must be reduced to prevent the system from failing drastically. Additionally, the noise has a greater effect on the output temperature.

11.9. ESC Affected by Noise

As previously shown, when noise is applied to the incremental state model, the system remains stable, although it becomes more sensitive as the number of incrementals increases. To test the ESC under more realistic conditions, it is proposed to use the ESC method with the added challenge of output noise. For the simulation, the same parameters from the ESC described in Section 10 were used, with a simulation time of 2200 s and excessive step-type

inputs. In other words, a Global ESC will be calculated while the system is affected by output noise, which follows a distribution of $N(0, 10^{-4})$ for Q_s and $N(0, 10^{-3})$ for T_s .

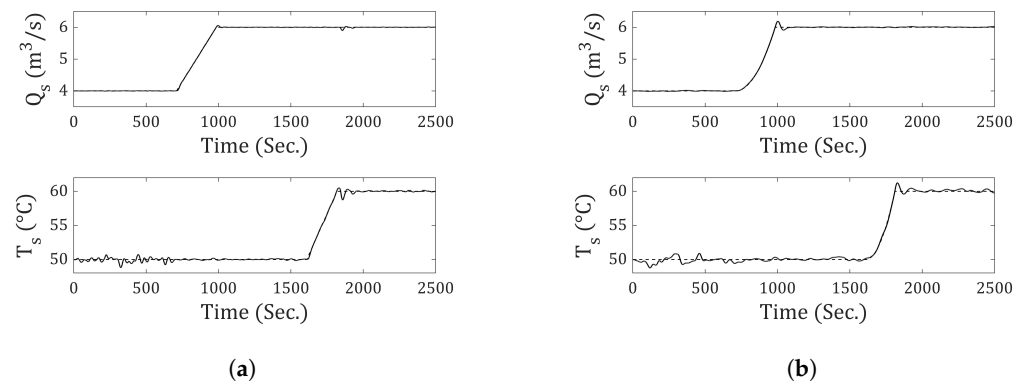


Figure 29. Responses of Q_s and T_s subjected to references and noise effects for Incremental-2 and Incremental-3: (a) ramp-type references for Incremental-2, (b) second-order references for Incremental-3.

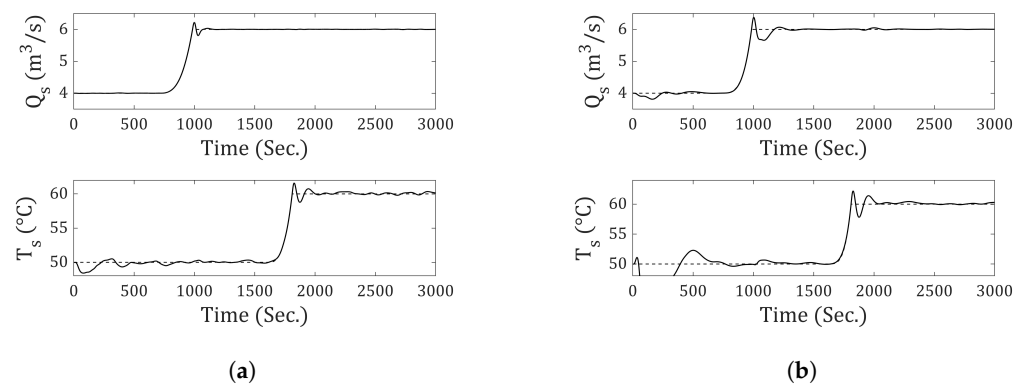


Figure 30. Responses of Q_s and T_s subjected to references and noise effects for Incremental-4 and Incremental-5: (a) third-order references for Incremental-4, (b) fourth-order references for Incremental-5.

Figures 31 and 32 show good system performance under noise, even when subjected to excessive step-type inputs, demonstrating a certain degree of robustness of the fuzzy LQR controller when using the incremental-1 state model. Table 6 shows that the values of $J(\theta)$ and the RMSE of Q_s and T_s are similar in both cases. Additionally, the values of the Q matrix parameters are quite similar (see Table 7).

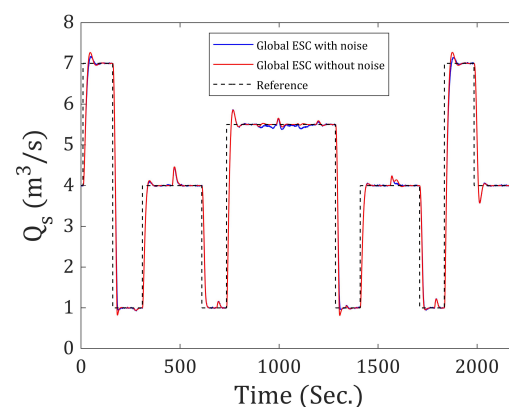


Figure 31. Comparison of the response of Q_s to excessive step-type reference signals using the Global ESC with and without noise for Incremental-1.

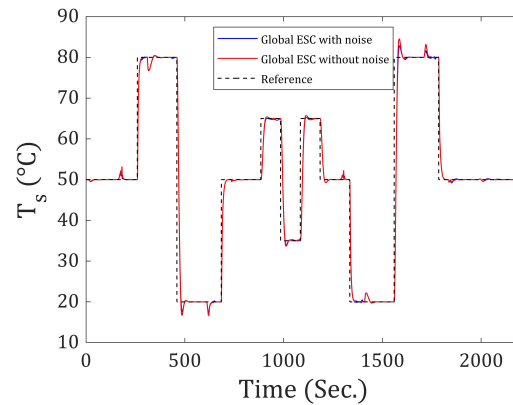


Figure 32. Comparison of the response of T_s to excessive step-type reference signals using the Global ESC with and without noise for Incremental-1.

Table 6. Value of $J(\theta)$ and RMSE of Q_s and T_s for Incremental-1.

Method for Obtaining Q and R Matrices	$J(\theta)$	RMSE (Q_s)	RMSE (T_s)
Global ESC with noise	1.0830	0.8485	6.0797
Global ESC without noise	1.0691	0.8373	6.0641

Table 7. Values of the Q matrix parameters for the ESC With and Without Noise.

Parameter of Q Matrix	Global ESC with Noise	Global ESC Without Noise
Q_1	0.0075	0.0075
Q_2	0.173×10^{-3}	0.179×10^{-3}
$Q_3, \dots, Q_7$	0.209×10^{-3}	0.222×10^{-3}

11.10. Computational Burden and Real-Time Feasibility

The proposed control architecture achieves an effective compromise between control performance, adaptability, and computational feasibility by integrating incremental-N modeling, Takagi–Sugeno fuzzy systems, discrete LQR design, and Extremum Seeking Control (ESC). While PID controllers are widely adopted due to their simplicity and low computational demand, they exhibit significant limitations when applied to multivariable, nonlinear, or strongly coupled systems, such as the thermal mixing tank addressed in this study. Additionally, the tuning process for PID gains in such contexts is often heuristic and lacks robustness.

Model Predictive Control (MPC) offers a more sophisticated alternative by explicitly handling system constraints and nonlinearities through online optimization. However, this advantage comes at the cost of high computational complexity, as MPC requires solving an optimization problem at every sampling instant. Consequently, its real-time implementation becomes impractical for fast dynamic systems or platforms with limited processing capabilities.

In contrast, the methodology proposed in this work exploits the structure of the incremental-N model to manage high-order reference signals and disturbances with improved steady-state accuracy. The fuzzy state-space framework allows for an efficient representation of the system's nonlinear dynamics, while the LQR formulation ensures stable and optimal multivariable control. Furthermore, ESC is employed exclusively as an offline optimization tool to tune the LQR weighting matrices, thereby avoiding any real-time computational overhead. Thus, the proposed approach combines the simplicity and efficiency of LQR with the adaptive tuning capabilities of ESC and the modeling flexibility of fuzzy systems, resulting in a control scheme that is both computationally efficient

and well-suited for real-time applications. It offers a viable alternative to PID and MPC, particularly in scenarios where nonlinearities, multivariable interactions, and high-order dynamics must be addressed without incurring excessive computational cost.

12. Conclusions

This paper proposes the incremental-N state model for the modeling and control of nonlinear multivariable systems using Takagi–Sugeno (T–S) fuzzy models. The proposed approach ensures zero steady-state error in response to input and disturbance signals of order N-1. Furthermore, the use of the incremental-N model eliminates affine terms and addresses the challenge of computing the target state, a difficulty commonly encountered with traditional discrete-time state-space models. Moreover, the proposed approach has shown strong resilience to various adverse conditions, including disturbances at the input and load levels, as well as measurement noise. Importantly, the ESC algorithm preserves its tuning effectiveness even under noisy output conditions, highlighting the robustness and practical applicability of the overall control scheme.

A discrete Linear Quadratic Regulator (LQR) controller was designed based on the fuzzy model, requiring real-time information of the state of the system. As a result, the role of state observers is critical, since any estimation error directly impacts control performance. An optimal observer was developed to estimate the incremental-N state, given that not all states of the system are directly measurable.

The proposed control strategy was evaluated using a thermal mixing tank system as a benchmark. The simulation results confirm the achievement of zero steady-state error under reference and disturbance signals of order N-1 for incremental orders ranging from 1 to 5, while also demonstrating robust performance. However, increasing the incremental order results in slower and more oscillatory system responses, attributed to the additional integrators, which, in discrete time, correspond to poles located at $1 + 0i$ in the Z-plane.

Extremum Seeking Control (ESC) was employed to optimize the Q and R weighting matrices of the discrete LQR controller. This approach significantly streamlines the tuning process, substantially reduces the design effort and computation time, and enables the identification of optimal controller parameters.

Author Contributions: Conceptualization, B.M.A.-H.; Methodology, B.M.A.-H. and G.A.L.A.; Software, G.A.L.A.; Validation, B.M.A.-H. and G.A.L.A.; Formal analysis, B.M.A.-H. and G.A.L.A.; Investigation, B.M.A.-H. and G.A.L.A.; Writing—original draft, B.M.A.-H. and G.A.L.A.; Supervision, B.M.A.-H.; Project administration, B.M.A.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [[CrossRef](#)]
2. Zadeh, L.A. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Trans. Syst. Man Cybern.* **1973**, *SMC-3*, 28–44. [[CrossRef](#)]
3. Mamdani, E.; Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Mach. Stud.* **1975**, *7*, 1–13. [[CrossRef](#)]

4. Sala, A.; Ariño, C. Reduciendo distancias entre el control borroso y el control no lineal: Luces y sombras. *Rev. Iberoam. Autom. Inform Ind. RIAI* **2009**, *6*, 26–35. [\[CrossRef\]](#)
5. Piña, A.J.B. *Síntesis de Sistemas de Control Borroso Estables por Diseño*; Universidad de Huelva: Huelva, Spain, 2009.
6. Ollervides, J.; Santibáñez, V.; Llama, M.; Dzul, A. Aplicación de Control Borroso a un Sistema de Suspensión Magnética: Comparación Experimental. *Rev. Iberoam. Autom. Inform Ind. RIAI* **2010**, *7*, 63–71. [\[CrossRef\]](#)
7. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 116–132. [\[CrossRef\]](#)
8. Jimenez, A.; Al-Hadithi, B.; Matia, F. An Optimal T-S Model for the Estimation and Identification of Nonlinear Functions. *WSEAS Trans. Syst. Control* **2008**, *3*, 897–906.
9. Al-Hadithi, B.M.; Jiménez, A.; Matía, F. A new approach to fuzzy estimation of Takagi–Sugeno model and its applications to optimal control for nonlinear systems. *Appl. Soft Comput.* **2012**, *12*, 280–290. [\[CrossRef\]](#)
10. Peng, X.; van Paassen, A. A state space model for predicting and controlling the temperature responses of indoor air zones. *Energy Build.* **1998**, *28*, 197–203. [\[CrossRef\]](#)
11. Yue, C.; Zhao, Y. An improved aeroservoelastic modeling approach for state-space gust analysis. *J. Fluids Struct.* **2020**, *99*, 103148. [\[CrossRef\]](#)
12. Di Ruscio, D.; Foss, B. On State Space Model Based Predictive Control. *IFAC Proc. Vol.* **1998**, *31*, 301–306. [\[CrossRef\]](#)
13. Gatzke, E.P.; Meadows, E.S.; Wang, C.; Doyle, F.J. Model based control of a four-tank system. *Comput. Chem. Eng.* **2000**, *24*, 1503–1509. [\[CrossRef\]](#)
14. Yu, C.; Chen, J.; Li, S.; Verhaegen, M. Identification of affinely parameterized state-space models with unknown inputs. *Automatica* **2020**, *122*, 109271. [\[CrossRef\]](#)
15. Al-Hadithi, B.M.; Jiménez, A.; Perez-Oria, J. New incremental Takagi–Sugeno state model for optimal control of multivariable nonlinear time delay systems. *Eng. Appl. Artif. Intell.* **2015**, *45*, 259–268. [\[CrossRef\]](#)
16. Adánez, J.M.; Al-Hadithi, B.M.; Jiménez, A. Wind Turbine Multivariable Optimal Control Based on Incremental State Model. *Asian J. Control* **2018**, *20*, 2075–2087. [\[CrossRef\]](#)
17. Sun, B.; van Kampen, E.J. Incremental Model-Based Global Dual Heuristic Programming for Flight Control. *IFAC-PapersOnLine* **2019**, *52*, 7–12. [\[CrossRef\]](#)
18. Zhou, Y.; van Kampen, E.J.; Chu, Q. Incremental model based online heuristic dynamic programming for nonlinear adaptive tracking control with partial observability. *Aerosp. Sci. Technol.* **2020**, *105*, 106013. [\[CrossRef\]](#)
19. Luenberger, D.G. Observing the State of a Linear System. *IEEE Trans. Mil. Electron.* **1964**, *8*, 74–80. [\[CrossRef\]](#)
20. Okongwu, E.H.; Wilson, W.J.; Anderson, J.H. Optimal State Feedback Control of a Microalternator Using an Observer. *IEEE Trans. Power Appar. Syst.* **1978**, *PAS-97*, 594–602. [\[CrossRef\]](#)
21. Jones, L.; Lang, J. A state observer for the permanent-magnet synchronous motor. *IEEE Trans. Ind. Electron.* **1989**, *36*, 374–382. [\[CrossRef\]](#)
22. Wang, W.; Gao, Z. A comparison study of advanced state observer design techniques. In Proceedings of the 2003 American Control Conference, Denver, CO, USA, 4–6 June 2003; Volume 6, pp. 4754–4759.
23. Florián, A. Comparación de desempeño de observadores de estado en sistemas lineales con aplicación a un motor de corriente continua. *Rev. Cintex* **2018**, *23*, 51–59. [\[CrossRef\]](#)
24. Wong, P.K.; Stein, G.; Athans, M. Structural Reliability and Robustness Properties of Optimal Linear-Quadratic Multivariable Regulators. *IFAC Proc. Vol.* **1978**, *11*, 1797–1805. [\[CrossRef\]](#)
25. Palm, R.; Rehfuess, U. Fuzzy controllers as gain scheduling approximators. *Fuzzy Sets Syst.* **1997**, *85*, 233–246. [\[CrossRef\]](#)
26. Amirah, I.N.; Zainah, M.Z.; Pebrianti, D.; Bayuaji, L. Attitude Control of Quadrotor. *ARN J. Eng. Appl. Sci.* **2015**, *10*, 1726–1731.
27. Everett, M. *LQR with Integral Feedback on a Parrot Minidrone*; MIT: Cambridge, MA, USA, 2015.
28. Ahmad, F.; Kumar, P.; Bhandari, A.; Patil, P.P. Simulation of the Quadcopter Dynamics with LQR based Control. *Mater. Today: Proc.* **2020**, *24*, 326–332. [\[CrossRef\]](#)
29. Omeje, C.O.; Salau, A.O.; Eya, C.U. Dynamics analysis of permanent magnet synchronous motor speed control with enhanced state feedback controller using a linear quadratic regulator. *Heliyon* **2024**, *10*, e26018. [\[CrossRef\]](#)
30. Leblanc, M. Sur l'électrification des chemins de fer au moyen de courants alternatifs de fréquence élevée. *Rev. Gen. l'Electricite* **1922**, *12*, 275–277.
31. Krstić, M.; Wang, H.H. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica* **2000**, *36*, 595–601. [\[CrossRef\]](#)
32. Killingsworth, N.; Krstic, M. PID tuning using extremum seeking: Online, model-free performance optimization. *IEEE Control Syst. Mag.* **2006**, *26*, 70–79.
33. Guay, M.; Dochain, D. A proportional integral extremum-seeking control approach. *IFAC Proc. Vol.* **2014**, *47*, 377–382. [\[CrossRef\]](#)
34. Straus, J.; Krishnamoorthy, D.; Skogestad, S. On combining self-optimizing control and extremum-seeking control—Applied to an ammonia reactor case study. *J. Process Control* **2019**, *78*, 78–87. [\[CrossRef\]](#)

35. García-Villaraco, J.M.A. *Nuevas Técnicas para Ampliar el Paradigma Metodológico en el Contexto del Control Borroso*; UPM: Serdang, Malaysia, 2019.
36. Al-Hadithi, B.M.; Barragán, A.J.; Andújar, J.M.; Jiménez, A. Chattering-free fuzzy variable structure control for multivariable nonlinear systems. *Appl. Soft Comput.* **2016**, *39*, 165–187. [[CrossRef](#)]
37. Bryson, A.; Ho, Y.C.; Siouris, G. Applied Optimal Control: Optimization, Estimation, and Control. *Syst. Man Cybern. IEEE Trans.* **1979**, *9*, 366–367. [[CrossRef](#)]
38. Athans, M.; Falb, P.L. *Optimal Control: An Introduction to the Theory and Its Applications*; Dover Publications: Mineola, NY, USA, 2006.
39. Anderson, B.D.O.; Moore, J.B. *Optimal Control: Linear Quadratic Methods*; Dover Publications: Mineola, NY, USA, 2007.
40. Lewis, F.L.; Xie, L.; Popa, D. *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*; CRC Press: Boca Raton, FL, USA, 2007.
41. Åström, K.; Wittenmark, B. *Computer Controlled System Theory and Design*; Prentice Hall: Upper Saddle River, NJ, USA, 1984; Volume 31.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.