Semnan University

Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE)

Journal homepage: **https://mseee.semnan.ac.ir/**

ISSN: 2821-0786

Real-Time Implementation and Simulation of Comparative Fuzzy-PID Controllers for a Fluid Level System

Ali Hosseini Laqa[1]*

Abstract**--This paper develops and compares three control strategies for a water level process: a conventional proportional-integral-derivative (PID) controller, a PID controller embedded in a Smith Predictor structure, and a Fuzzy PID controller. Experiments are carried out on a TE3300 process trainer, where a differential pressure transmitter reports the level, and an electro-pneumatic valve sets the inlet flow. First-order plus time delay (FOPTD) and second-order plus time delay (SOPTD) models are fitted to step-response data recorded on the setup, and the second-order model serves as the common plant against which all three controllers are evaluated. The conventional controller is tuned by the internal model control (IMC) method, and the inner controller of the predictor is independently retuned for the delay-free plant, a step that keeps the comparison focused on control structure rather than on inherited tuning. The Fuzzy PID controller is built on a zero-order Sugeno inference system, and three distinct control surfaces are constructed and tested. All three controllers are subsequently deployed on the physical setup and run in real time through an Open Platform Communications (OPC) interface, which introduces a delay of approximately five seconds between the controller and the plant. The paper reports what this delay compensation is worth in practice and examines how the geometry of the fuzzy control surface balances the error accumulated during a transient against the speed of correction and the resulting valve activity.**

Index Terms**- Fuzzy PID control, Smith Predictor, System identification, Control Surface geometry, Time delay, Water level system.**

I. Introduction

Liquid level processes are among the most common systems in industrial process control, and they represent many practical units such as storage tanks, chemical vessels, and water treatment facilities [1]. Their operation requires accurate modeling and effective regulation to maintain process stability, guarantee safe operation, and satisfy prescribed performance requirements. Because they matter in practice and their dynamics are easy to follow, such systems are widely used for both tutorial purposes and laboratory studies [2].

Despite their apparent simplicity, real liquid level systems are subject to several factors that complicate analysis and controller implementation. Sensor noise, actuator limitations, external disturbances, nonlinear operating conditions, and time delay may all significantly influence closed-loop behavior. These effects are far more visible when the control algorithm runs on a physical laboratory system rather than in simulation alone.

Among the available control strategies, the proportional-integral-derivative (PID) controller remains dominant in industrial practice, and published reviews of PID technology report that most installed control loops are of the PID form [3]. This continued use comes from a simple structure, a small number of clear tuning parameters, and acceptable performance

Received; 2026-06-14 Revised; 2026-08-19 Accept; 2026-08-23

1. Department of Electrical and Computer Engineering, Faculty of Electrical Engineering, Isfahan University of Technology, Isfahan, Iran.
*Corresponding author Email: alihosseinil@alumni.iut.ac.ir

Cite this article as:
Hosseini Laqa, A. (2026) Real-Time Implementation and Simulation of Comparative Fuzzy-PID Controllers for a Fluid Level System. *Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE). Semnan University Press . 6 (3), 1-9.*
DOI: https://doi.org/10.22075/mseee.2026.41616.1284

across a wide range of process dynamics [4], [5]. The performance of a fixed-gain PID controller nevertheless degrades in the presence of dead time, since the delay reduces the phase margin and limits the achievable closed-loop bandwidth [6].

In many process systems, time delay is unavoidable and may originate from fluid transport, signal processing, or communication interfaces. In laboratory environments in particular, the link between the host computer and the physical process adds a delay that the plant itself does not have, and that delay is not negligible compared with the main time constant of the loop. To compensate for such effects, the Smith Predictor [7] employs an internal model of the delay-free plant together with the dead time, thereby permitting the feedback controller to be designed for a delay-free process. Fuzzy logic control offers an alternative strategy, in which rule-based reasoning replaces strict reliance on an analytical model, and it therefore suits processes that are nonlinear or whose parameters change [8], [9].

Although PID control, delay compensation, and fuzzy PID tuning have each been investigated extensively, these strategies are frequently studied in isolation or evaluated mostly in simulation. Comparative studies conducted on a single physical platform, employing models identified from that same platform and subject to the delay imposed by an actual real-time communication interface, are considerably less common. This observation motivates the study reported here, in which the three structures are first compared on a common identified model and then implemented on the setup itself.

In this paper, the modeling, identification, and implementation of three control strategies are presented for the TE3300 liquid level process, namely a conventional PID controller tuned by the Internal Model Control (IMC) method, a PID controller placed inside a Smith Predictor, and a Fuzzy-PID controller based on a zero-order Sugeno inference system. The process is first identified from experimental input-output data and represented by First-Order Plus Time Delay (FOPTD) and Second-Order Plus Time Delay (SOPTD) models. The identified models are then used in two ways. They provide the common nominal plant on which the three structures are compared under identical conditions, and the internal model of the Smith Predictor when the same structures are executed in real time on the physical plant through a communication interface based on the Open Platform Communications (OPC) standard.

The main contributions of this work, in contrast to existing studies on liquid level regulation, are summarized as follows:

1) The TE3300 liquid level process is identified from experimental input-output data acquired in real time, and both FOPTD and SOPTD models are extracted for subsequent controller design.
2) A conventional PID controller is designed by the IMC tuning method and implemented on the physical process, providing the baseline against which the remaining structures are assessed.
3) A Smith Predictor structure is applied to compensate for the delay introduced by the computer–process communication interface, rather than a transport delay inherent to the plant itself.
4) A Fuzzy-PID controller is developed and evaluated under three distinct control surfaces, namely linear, bumpy, and steep, in order to examine the influence of surface geometry on the closed-loop response.

5) The three structures are compared on a common nominal plant under the same input sequence and integration window, with the inner controller of the predictor retuned for the delay-free plant, and are then implemented on the laboratory setup, so that the effect of the control structure is separated from the effect of the model on which it was designed.

The remainder of this paper is organized as follows. Section II reviews the related literature on PID tuning, delay compensation, and fuzzy control. Section III describes the TE3300 experimental setup and its instrumentation. Section IV presents the communication architecture and the real-time implementation. Section V details the process modeling and identification procedure. Section VI addresses the design of the conventional PID, Smith Predictor-PID, and Fuzzy-PID controllers. Section VII compares the three structures on a common nominal plant. Section VIII reports their implementation on the setup. Section IX states the limitations of the study, and Section X concludes the paper.

II. Related Works

Model-based tuning is one of the principal routes for obtaining PID parameters in practice. In [10], it is shown that for a broad class of low-order models, including FOPTD, the IMC design reduces to a PID controller whose parameters are available in closed form and depend on one tuning parameter that sets the balance between closed-loop speed and robustness. The theoretical foundation of this mechanism, together with its treatment of model uncertainty and robust stability, is developed in [11]. The authors in [12] extend the IMC principle to single-loop controller design across a range of process structures. In [13], analytic tuning rules are presented alongside a systematic procedure by which higher-order models may be reduced to first- or second-order plus dead time form.

Because model-based tuning presupposes the availability of a process model, the identification stage is central to the design procedure. In [14], the theory and practice of identifying dynamic models from measured input-output data are treated in detail, including model structure selection and validation. The study in [15] specifically addresses identification from step and transient response experiments, which is the approach adopted in the present work.

Compensation for dead time has received significant attention. The Smith Predictor was first introduced in [7], in which a model of the delay-free plant and of the dead time is placed within the controller so that the feedback law may be designed for a process without delay. The principal limitation of the scheme is its sensitivity to model error, since the compensation is produced entirely by the internal model. The robustness of model-based structures to such a mismatch is analyzed within the IMC in [11]. Later work has modified the structure itself for cases where the original scheme works poorly, and in [16], a modified predictor is developed to integrate processes with long dead time. Although these methods demonstrate clear improvements in the regulation of delayed processes, the reported evaluations are frequently confined to simulation, and the delay is commonly assumed rather than measured on a physical interface.

Fuzzy logic control provides an alternative for processes that are nonlinear, uncertain, or difficult to describe analytically. Building upon the theory of fuzzy sets introduced in [8], the first experimental fuzzy logic controller is reported in [9], where a set of rules written in words is shown to regulate a plant without an explicit mathematical model. In [17] and [18], the structure and design of such controllers are surveyed, including choices for the fuzzification, inference, and defuzzification that determine the resulting control surface, and a complete design procedure is given in [19]. The Sugeno inference structure adopted in the present study originates in [20], in which the output of each rule is a function of the inputs rather than a fuzzy set. In the zero-order case, the output is a constant, which gives a static input-output mapping that is convenient for real-time implementation. This structure additionally permits the gains of a fuzzy PID controller to be directly related to those of a conventional PID controller [19], so that established tuning rules may serve as a starting point before the control surface is shaped.

Level processes have also been addressed by optimization-based and variable-structure methods. In [21], a constrained predictive controller is applied to a coupled-tank apparatus and compared with PID control, with the actuator constraints carried directly in the prediction model. In [22], static and dynamic sliding mode schemes are designed for a two-tank process and verified on an experimental setup, where a boundary layer limits the chattering that the switching term produces in the control signal. Table I compares the three structures examined in this work with these alternatives.

TABLE I
Qualitative Comparison of the Configuration Adopted in this ork with Alternative Control Structures for a Delayed Liquid Level Process

Criterion	Fuzzy-PID ⍟	PID	SP-PID	MPC [21]	SMC [22]
Dead-time compensation	✓	✗	✓	✓	✗
Robust to dead-time mismatch	High	Low	High	Medium	Medium
Response shaping without retuning	✓	✗	✗	✗	✗
Chattering in the control signal	✗	✗	✗	✗	✓
Computational load	Low	Low	Low	High	Low
Dependence on the model	High	Medium	High	High	Medium

Although the individual elements employed in this work are well established, their direct comparison on a single physical setup remains uncommon. In particular, existing studies do not generally combine models identified from the same setup with a delay originating in a real communication interface, nor do they examine the influence of the fuzzy control surface on the measured response under identical operating conditions. These limitations motivate the experimental study presented in the following sections.

III. Experimental Setup and Instrumentation

The TE3300 liquid level process is a closed-loop hydraulic circuit in which water is circulated between a transparent vertical vessel and a lower storage reservoir. Water is delivered from the reservoir to the vessel by a three-speed centrifugal pump and returns to the reservoir through the outlet line, so that the circuit operates continuously. The process and instrumentation diagram and a picture of the assembled setup are shown in Fig. 1. The flow into the vessel is controlled by a pneumatic valve placed in the inlet line, and the opening of that valve therefore sets how fast the level in the vessel changes. Because the vessel is transparent, the level can be observed directly, which provides an independent check on the transmitter reading during an experiment. The process behaves smoothly in the neighborhood of its normal operating level and departs from linear behavior when the level is driven quickly or over a wide range, which is also the case in many industrial tanks and storage vessels.

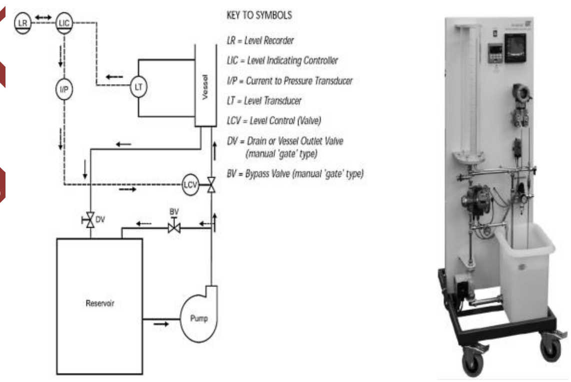

Fig. 1. TE3300 liquid level setup, with (Left) the process and instrumentation diagram and (Right) a photograph of the assembled system.

The setup carries industrial instruments that serve both the control loop and the recording of data. The loop controller is a Honeywell UDC2500 universal digital controller, shown in Fig. 2 (Left), which receives the measured level, compares it with the setpoint, and computes the control action. The level is measured by a differential pressure (DP) transmitter connected across the vessel. Since the hydrostatic pressure at the base of the vessel grows with the water level in the vessel, the transmitter returns that level as a fraction of its calibrated span, and that fraction is transmitted to the controller as the process variable (PV). Throughout this paper, the level is reported as a percentage of the transmitter span, and the controller output is reported as a percentage of its own range.

The operating record is captured by a Honeywell eZtrend QXe paperless recorder, shown in Fig. 2 (Right), which stores the level and the controller output and displays them as trends. These records are the source of the identification data in Section V and of the closed-loop responses in Section VIII. The controller output is transmitted as a 4-20 mA signal to an electro-pneumatic current-to-pressure (I/P) transducer, which converts it into a pneumatic signal in the range 3 to 15 psi. That pressure drives the actuator of the control valve, which opens as the pressure rises, so that an increase of the controller output opens the valve and raises the flow into the vessel.

Fig. 2. Instruments of the TE3300 setup, with (Left) the UDC2500 controller and (Right) the eZtrend QXe recorder.

The hydraulic circuit also carries manual elements through which disturbances are introduced. A manual drain valve at the vessel outlet controls the outflow, and a bypass line with a second manual valve diverts part of the delivered flow. A step change of the drain valve opening is used as the load disturbance. The setpoint is held at 40 % of the transmitter span in every closed-loop run reported here, and within a run, the pump speed and the openings of the bypass and drain valves are not altered, so that the step applied to the drain valve is the only deliberate change made to the circuit while a record is being taken.

IV. Communication Architecture and Real-Time Implementation

The setup operates in real time over an industrial communication link built on the OPC standard, which connects the physical hardware to the computer and lets the control algorithm run in Simulink while the plant stays in the loop. This arrangement is a rapid control prototyping configuration rather than a hardware-in-the-loop one, since the plant is physical and only the controller is executed in software. The communication path is shown in Fig. 3. The network is an Ethernet link running TCP/IP. The UDC2500 is placed on that network at a fixed address; the host computer is given an address in the same range; and data exchange is handled by KEPServerEX V6 acting as the OPC server. A separate UDC channel is created with the Honeywell UDC Ethernet driver and communicates over Modbus TCP on port 502. The settings required to build the same link again are listed in Table II.

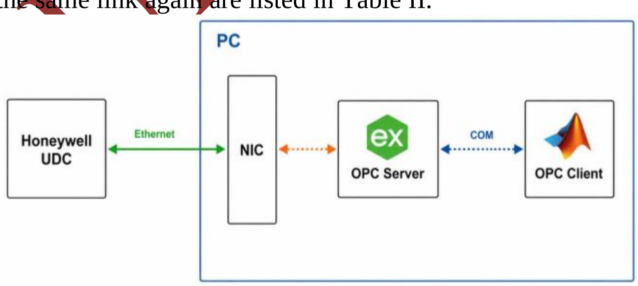

Fig. 3. Data transfer path between the Honeywell UDC2500, the OPC server, and the Simulink client.

TABLE II
Configuration of the OPC Communication Interface

Item	Setting
OPC server	KEPServerEX V6
Channel driver	Honeywell UDC Ethernet
Device model	UDC2500
Controller address	10.100.0.4
Host workstation	10.100.0.1, mask 255.255.255.0
Transport	Modbus TCP, port 502
Connect timeout	3 s
Request timeout	1000 ms
Attempts before timeout	3
Inter-request delay	50 ms
Write optimization	latest value only, duty cycle 10
Process variable tag	UDC float register mapped to the simulator PV
Output tag	simulator OT mapped to OutputWorkingValue
Simulink read	synchronous from cache, 0.5 s
Simulink write	asynchronous, 0.5 s

A second Simulator channel is created within KEPServerEX and acts as the intermediary for Simulink, which is the OPC client. The two channels are joined through the Advanced Tags feature, so that values pass in both directions between the controller registers and the simulation environment. The PV measured by the UDC2500 is mapped from the float register of the controller to the PV tag of the simulator, and the control signal computed in Simulink is written to the OT tag of the simulator, which is linked internally to the OutputWorkingValue register of the controller. Reading and writing use the OPC Configuration, OPC Read, and OPC Write blocks, and both transfers are synchronized at a sample time of 0.5 s. Simulink reads the value held in the server cache and does not wait for the write to be completed, so each transfer adds a delay of the order of one sample period on top of the time the controller needs to complete one cycle. The delay introduced along this path appears in the identified model of Section V and is the quantity that the Smith Predictor of Section VI is designed to compensate.

V. Process Modeling and Identification

A model of the process is required before the controllers can be designed. Because the setup shows practical nonlinear effects such as valve friction and changes in pump delivery, an empirical approach is adopted, and the model is obtained from a step response measured on the plant itself.

The identification experiment is executed in Simulink connected to the plant through the OPC interface described in Section IV. The arrangement used to record the response is shown in Fig. 4. The constant labeled 40 in that figure is the amplitude of the step applied to the control input, expressed as a percentage of the opening of the fill valve, so the valve is driven from the closed position to 40 % of full opening. The resulting change of the liquid level, expressed as a percentage of the transmitter span, is recorded as the process output.

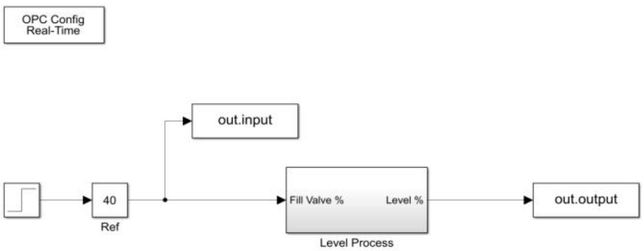

Fig. 4. Simulink arrangement used to record the step response, in which the constant labeled 40 is the step amplitude in percent of valve opening.

Two assumptions are made in the following.

Assumption 1. The transport delay identified in the model is attributed to the communication path between the OPC server and the Simulink environment, and not to the physical movement of the fluid.

Assumption 2. External disturbances, including variations of the pump delivery and changes of the manual discharge valve, are bounded and remain within the operating limits of the setup. This assumption allows the load disturbance of Sections VII and VIII to be considered as a bounded step rather than an arbitrary input.

The measured record is analyzed with the MATLAB System Identification Toolbox, and two model structures are examined. The first is a first-order plus time delay (FOPTD) structure, which describes the process by a static gain, a dominant time constant, and a dead time. The identified FOPTD model is as follows:

$$G(s) = \frac{2.76}{68.9s+1} e^{-5.86s} \tag{1}$$

where the gain is expressed in percent of level per percent of valve opening, and the time constant and the dead time are in seconds. The measured response and this model's response are compared in Fig. 5. The model follows the measurement over most of the step, and the remaining error is largest in the first seconds after the step, where a single pole cannot reproduce the gradual start of the rise.

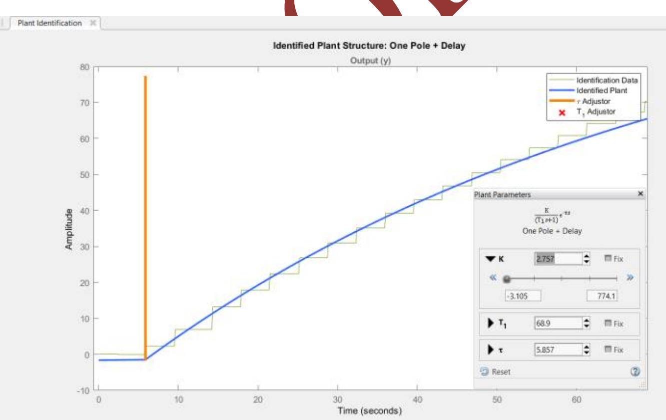

Fig. 5. Measured step response compared with the response of the FOPTD model of (1).

A second-order plus time delay (SOPTD) structure is fitted next in order to describe the initial transient more accurately, which gives

$$G(s) = \frac{1.8}{(0.48s+1)(44.67s+1)} e^{-4.53s} \tag{2}$$

The two poles of (2) represent the inertia of the level response, and the comparison with the measurement is shown in Fig. 6. The second-order model follows the measurement more closely over the whole experiment, and in particular during the early transient.

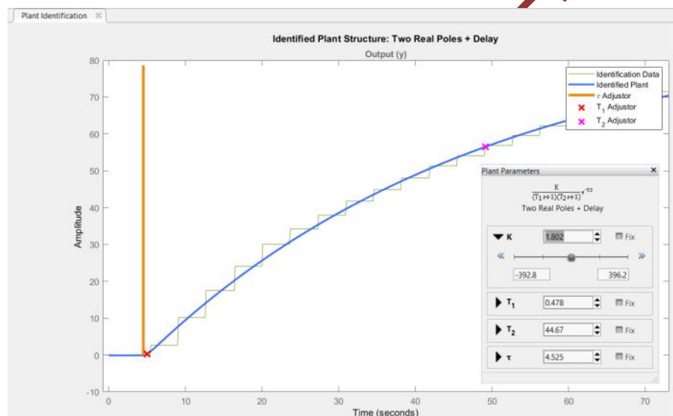

Fig. 6. Measured step response compared with the response of the SOPTD model of (2).

Two remarks should be made before using these models, since the two structures return different parameters for the same record.

Remark 1. The identification record covers approximately one main time constant, which is too short to cleanly separate the static gain from the dominant time. The two structures therefore distribute the same measured rise differently; the first-order model returns a gain of 2.76 with a time constant of 68.9 s, and the second-order model a gain of 1.8 with a dominant time constant of 44.67 s.

Remark 2. The dead time is affected in the same way, and this is where the two values 5.86 s and 4.53 s come from. The first-order model has no second pole with which to reproduce the gradual start of the rise, so the estimator represents that curvature by delaying the response, and the identified dead time absorbs it. The second-order model describes the same curvature with its faster pole and returns a dead time closer to the delay contributed by the interface. The half rule of [13], applied in Section VII-A, states the same thing in the opposite direction, since reducing the second-order model to first-order form moves half of the shorter time constant into the dead time and gives 4.77 s. Neither value is a direct measurement of the interface delay, and Section VII-F shows that the difference between them does not change any of the conclusions of this paper. The second-order model agrees more closely with the measurement and is therefore adopted in Section VII as the nominal plant for all controllers. The first-order model is retained in Section VI-A to tune the conventional controller, where the IMC rules are stated for a first-order form with dead time.

VI. Controller Design

A. Conventional PID Design

The level regulation is first addressed using a conventional controller. Given the dead time identified in Section V and the dominant first-order character of the hydraulic circuit, the internal model control (IMC) tuning method is adopted, since it provides a systematic route from a low-order model with dead time to controller settings and shows the trade-off between response speed and robustness through a single parameter. The two IMC rules that apply to a first-order model with dead time are listed in Table III [23].

Table III
IMC Tuning Rules for a First-Order Model with Dead Time [23]

Case	Model	$K_c K$	τ_I	τ_D
G	$\frac{K}{\tau s+1}e^{-\theta s}$	$\frac{\tau}{\tau_c+\theta}$	τ	—
H	$\frac{K}{\tau s+1}e^{-\theta s}$	$\frac{\left(\tau+\frac{\theta}{2}\right)}{\left(\tau_c+\frac{\theta}{2}\right)}$	$\tau+\frac{\theta}{2}$	$\frac{\tau\theta}{(2\tau+\theta)}$

In Table III, K is the static gain of the process, τ is its dominant time constant, and θ is its dead time, all taken from the identified model. The parameter τ_c is the desired closed-loop time constant and is the only quantity chosen by the designer. The entries K_c, τ_I, and τ_D are the proportional gain, the integral time, and the derivative time returned by each rule. Case G returns a proportional-integral controller, and Case H adds derivative action obtained from a first-order approximation of the dead time. Case G is used here, so that the delay is left to the predictor of Section VI-B rather than being partly absorbed by a derivative term. The derivative time is therefore not defined for Case G.

With the FOPTD model of (1), that is $K = 2.76$, $\tau = 68.9$ s and $\theta = 5.86$ s, the proportional gain follows from Case G as

$$K_c = \frac{\tau}{\left[(\tau_c+\theta)K\right]},\; T_i = \tau \tag{3}$$

The closed-loop time constant is set to $\tau_c = 5$ s, which follows the IMC guideline of [10], in which τ_c is taken of the order of the dead time, which leaves a margin against model error. Substitution gives $K_c = \frac{68.9}{(10.86\times 2.76)} \approx 2.3$ and $T_i = 68.9$ s, which corresponds to an integral coefficient of approximately 0.03 in the parallel form used by the implementation. No derivative action is assigned, so the controller implemented in this work is of proportional-integral form throughout, and it is referred to as a PID controller only in the sense of the family from which its settings are taken. The loop formed by these settings and the model of (1) has a phase margin of about 59°, a gain margin of about 9.3 dB, and a peak sensitivity of about 1.66, which places the design in the range normally regarded as robust. The controller is implemented as a continuous-time block in the parallel form, with the settings $P = 2.3$, $I = 0.03$, and $D = 0$.

B. Smith Predictor-PID Design

The hydraulic circuit itself has no measurable dead time, but a delay of several seconds appears in the loop as soon as the controller runs on the host computer, as shown in Sections IV and V. The link is used by this loop alone and carries no other traffic, so the delay is treated as constant, an assumption examined against the results in Section VII-F. It is not negligible compared with the main time constant of the process, and if left uncompensated, it reduces the stability margins and makes the response slower and more oscillatory.

These effects could be limited by a frequency-domain design or by a rational approximation of the delay. However, a reasonably accurate model of the delayed process is available, which makes the Smith Predictor the more direct choice. In this structure, a model of the plant is identified, a controller is designed for the delay-free part of that model, and the controller is then embedded in the arrangement of Fig. 7, in which the inner feedback path is closed around the delay-free model. The outer feedback path carries the difference between the plant's measured output and the delayed model's output.

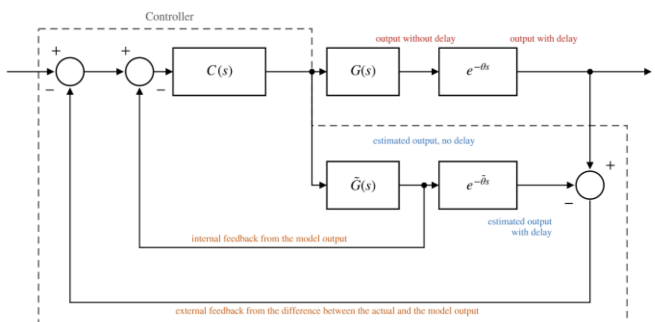

Fig. 7. Smith Predictor structure, in which the dashed boundary encloses the controller, the delay-free internal model, and the internal dead time.

If the internal model is exact and no disturbance is acting, the outer feedback signal is zero, and the feedback is generated solely by the model. When a load disturbance is applied or the model differs from the plant, the difference between the measured output and the delayed model output is no longer zero, and this difference corrects the inner loop reference. Two limitations of the structure should be noted. The predictor cannot reject a load disturbance faster than the dead time permits, and the original scheme requires modification when the plant contains an integrator or is unstable [16]. The equivalent controller seen by the plant is as follows:

$$C^*(s) = \frac{C(s)}{\left\{1+C(s)\tilde{G}(s)\left[1-e^{-\tilde{\theta}s}\right]\right\}} \tag{4}$$

where $\tilde{G}(s)$ is the delay-free internal model and $\tilde{\theta}$ is the internal dead time, both taken from the identified process model. Written in the standard second-order form, the SOPTD model of (2) becomes:

$$G(s) = \frac{0.0839}{s^2 + 2.11s + 0.0466} e^{-4.53} \tag{5}$$

which keeps the static gain of 1.8 of the identified model. The predictor is implemented with this transfer function together with the controller of Section VI-A, and the resulting arrangement in Simulink is shown in Fig. 8. The controller coefficients are kept at the values of Section VI-A, so that any change in the closed-loop behavior follows from the predictor rather than from a change of tuning.

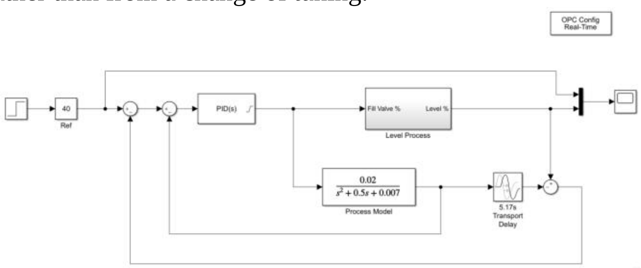

Fig. 8. Simulink implementation of the Smith Predictor with the conventional controller.

The internal model used in the comparison of Section VII is the transfer function given in (5). The predictor was also implemented on the setup, where the internal model corresponded to the first-order model of (1) rather than to (2). This difference and its consequences for the experimental records are discussed in Section VIII.

C. Fuzzy-PID Controller Design

In the third design, the conventional controller is replaced by a fuzzy one while the rest of the closed loop, including the Smith Predictor, is left unchanged. The purpose is to examine whether rule-based reasoning improves the response in the presence of process nonlinearity and the communication delay. The complete arrangement is shown in Fig. 9, and the internal structure of the fuzzy block is shown in Fig. 10. The controller acts on the control error, formed from the reference and the measured level, and on the change of that error from one sample to the next.

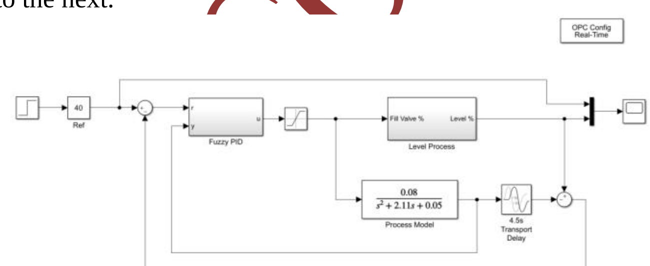

Fig. 9. Simulink implementation of the Smith Predictor with the Fuzzy PID controller.

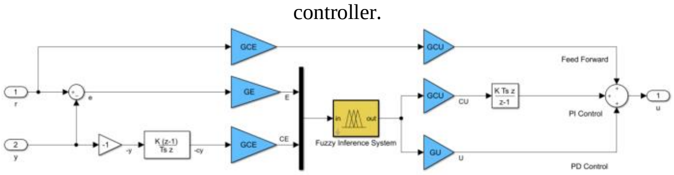

Fig. 10. Internal structure of the Fuzzy PID block, with the feedforward, PI, and PD paths and the scaling gains, following [24].

The inference system is described first, since the remaining relations follow from it. Both inputs, the scaled error and the scaled change of error, are defined on the normalized universe [−100, 100], and the output is defined on [−200, 200]. In the nominal design, each input carries three overlapping triangular membership functions labeled Neg, Zero, and Pos, with corner points at [−200, −100, 0], [−100, 0, 100], and [0, 100, 200]. The outer corners are placed beyond the ends of the universe so that Neg and Pos hold a grade of one there, and the sets are spaced so that the grades of any two neighboring sets add up to one at every point. A zero-order Sugeno system is used, so that the output of every rule is a constant rather than a fuzzy set. The AND in the rule condition is evaluated with the product, the OR with the probabilistic sum, and the output of the system is the weighted average of the rule outputs. The rules are written as follows:

$$R^i : IF\ x_1\ is\ M_1^i\ AND\ x_2\ is\ M_2^i\ THEN\ y = b^i \tag{6}$$

where i is the rule index, x_1 and x_2 are the two inputs of the controller, that is, the scaled error and the scaled change of error, M_j^i is the membership function assigned to the jth input in the ith rule, y is the output of the inference system, and b^i is the constant output of the ith rule. Since each of the two inputs carries three membership functions, the rule base of the nominal design is complete with nine rules. The rule outputs are assigned the values listed in Table IV, where the sign of each entry follows the sign of the required correction, and the magnitude grows when the error and its change point are in the same direction. The rule base and the scaling structure follow [25].

TABLE IV
Rule Base of the Zero-Order Sugeno Inference System. The Entries Represent the Constant Rule Outputs BI

$e / \Delta e$	Neg	Zero	Pos
Neg	−200	−100	0
Zero	−100	0	+100
Pos	0	+100	+200

The values of the rule outputs, −200, −100, 0, +100, and +200, are not valve positions, and the range they span is chosen for the following reason. It is the output of the inference system, which is converted into a valve opening by the output gains GCU and GU and by the saturation block that follows the controller in Fig. 9. That block is set to 0% and 100 %, the closed and fully open positions of the valve. The range of the rule outputs is therefore deliberately wider than the valve can follow, so that a large error drives the valve to its limit through the saturation block rather than being cut off inside the inference system itself. The gains are listed in Table V.

For an input pair (x_1, x_2), the firing strength w^i of the ith rule is obtained from the membership grades of its two conditions, and the output of the block at the kth sample is the weighted average of the n rule outputs,

$$out_k = \frac{\left(\sum_{i=1}^{n} w_k^i b^i\right)}{\left(\sum_{i=1}^{n} w_k^i\right)} \tag{7}$$

Eq. (7) is the general form and produces a nonlinear mapping from the two inputs to out_k. When the membership functions are arranged so that the resulting control surface is planar, which is the linear surface examined in Section VII, the weighted average in (7) reduces to a linear combination of the two inputs. With the error at the *k*th sample denoted by e_k and its change by $\frac{(e_k - e_{k-1})}{T_s}$, and with the two inputs scaled by the gains *GE* and *GCE*, this combination is

$$out_k = GEe_k + GCE\frac{e_k - e_{k-1}}{T_s} \tag{8}$$

where T_s is the sample period, equal to 0.5 s in this work. Equation (8) is therefore not a second definition of the fuzzy output, but the particular case of (7) obtained with a linear surface. It is the form used in the remainder of this subsection, since it allows the fuzzy controller to be compared directly with a conventional one.

The block of Fig. 10 distributes out_k over three paths. The PI path accumulates it, the PD path scales it, and a feedforward path adds a bias set by the reference. The two accumulated contributions are

$$pi_k = pi_{k-1} + GCUout_k, \quad pd_k = GUout_k \tag{9}$$

where *GCU* and *GU* are the output gains of the two paths, and the control signal applied to the plant at the *k*th sample is

$$u_k = GCE \times GCU \times r_k + pi_k + pd_k \tag{10}$$

in which r_k is the reference. The first term is the feedforward contribution. It adds a bias to the control signal proportional to the reference, which accelerates the initial response of a process with slow open-loop behavior and leaves a smaller steady-state offset for the integral path to remove. Its gain is the product of the input gain on the change of error and the output gain of the PI path, so it is fixed once those are chosen and is not tuned separately.

The relations above may now be compared with the conventional discrete laws. The standard proportional-derivative and proportional-integral actions are written as

$$u_k^{PD} = K_p\left[e_k + T_d\frac{e_k - e_{k-1}}{T_s}\right], \; u_k^{PI} = K_p\left[e_k + \frac{T_s}{T_i}\sum_{i=0}^{k} e_i\right] \tag{11}$$

In (11), K_p is the proportional gain, T_i the integral time, and T_d the derivative time of the conventional controller, and T_s is the sample period defined below (8). These are the same three quantities that the tuning rules of Table III and (14) denote by K_c, τ_I, and τ_D. Substituting (8) into (9) and collecting the terms of (10) that multiply the error, its accumulated value, and its change, and matching them term by term with (11), gives a direct correspondence between the four fuzzy gains and the conventional gains,

$$\begin{aligned} K_p &= GU \times GE + \frac{GCU \times GCE}{T_s} \\ K_i &= \frac{GCU \times GE}{T_s} \\ K_d &= GU \times GCE \end{aligned} \tag{12}$$

The factor T_s appears in the first two expressions because the sum in (9) adds the output of the block at every step, whereas the change of error in (8) is divided by the sample period. The link between the two sets of gains is useful in practice, since the fuzzy gains can be initialized from a conventional tuning, such as the one obtained in Section VI-A, after which the shape of the control surface is adjusted without returning to the gains.

Three control surfaces are examined in the experiments, obtained by changing the shape of the membership functions and, in one case, the size of the rule base. The linear surface is the nominal design just described and is planar. The bumpy surface keeps the nine rules of Table IV but replaces the triangular sets by Gaussian sets of width 30 with centers at −100, 0, and 100, which makes the surface non-monotonic near the origin. The steep surface carries only two membership functions per input, a Z-shaped set for Neg and an S-shaped set for Pos on the same universe, so its rule base reduces to the four rules of Table IV, whose conditions use Neg or Pos only, and the resulting surface rises sharply for small errors and saturates early. The configurations and the gains of the three cases are collected in Table V, and the surfaces are shown in Fig. 11.

TABLE V
Configuration of the Three Fuzzy Control Surfaces

Surface	Membership functions on the error and the change of error	Rules	GE	GU	GCU
Linear	three triangular, [−200 −100 0], [−100 0 100], [0 100 200]	9	1	5	0.2
Bumpy	three Gaussian, width 30, centers −100, 0, 100	9	1	15	0.5
Steep	two sigmoidal, Z-shaped, and S-shaped on [−100, 100]	4	1	5	0.03

The input gains are the same in the three cases, so the membership functions and the output gains listed in Table V are the only quantities that differ between them. The comparison of Section VII does not use these values but the gains derived from the mapping of (12), which are given in Section VII-A.

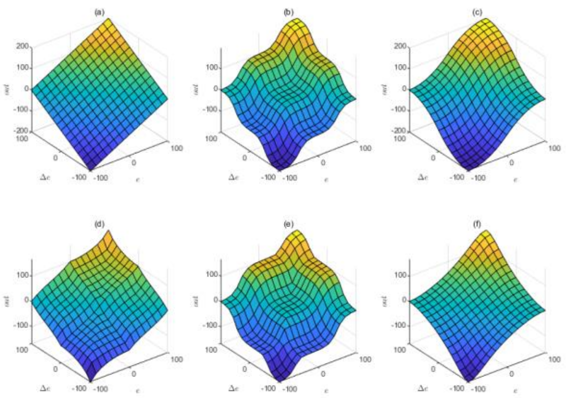

Fig. 11. Control surfaces of the Fuzzy PID controller for the (a) linear, (b) bumpy, and (c) steep cases with zero-order Sugeno inference and the (d) linear, (e) bumpy, and (f) steep cases with Mamdani inference.

VII. Comparison on a Common Nominal Plant

The responses measured on the setup were each taken with the internal model and the test sequence used in that particular run, as described in Section VIII, so they do not isolate the effect of the control structure. A meaningful comparison requires all controllers to regulate the same nominal plant, each tuned for the plant that its own loop encounters, with every configuration driven by the same input sequence and evaluated over the same integration window. Such a comparison is presented in this section.

The nominal plant is the second-order model of (2), which follows the measured transient more closely than the first-order model. Every configuration uses it unchanged, both as the plant and, where a predictor is present, as the internal model of that predictor.

A. *Controller Design on the Common Plant*

The conventional controller is designed for the plant whose own loop meets, i.e., the plant with the dead time still present. The two-pole model is first reduced to a first-order form by the half rule of [13], in which half of the smaller time constant is carried into the dead time,

$$\tau' = \tau_2 + \tau_1/2, \quad \theta' = \theta + \tau_1/2 \tag{13}$$

In (13), the time constant that is halved is the smaller of the two in (2), and a prime marks a parameter of the reduced first-order model rather than a derivative. The reduction returns $\tau' = 44.91$ s, and $\theta' = 4.77$ s. Case G of Table III then applies, and with $\tau_c = 5$ s it gives the settings listed in Table VI.

The inner controller of the predictor meets a different plant. Because the predictor takes the dead time outside the loop closed around that controller, it must be designed for the delay-free model rather than inherit the settings of Section VI-A. Those settings are detuned, that is, deliberately made slower and less aggressive so that the loop keeps its margins while the dead time is present, and the inner loop of the predictor no longer meets that dead time. Designing this controller again for the delay-free model is referred to as retuning throughout this paper. The delay-free model carries two real poles, so the corresponding internal model control rule applies [10],

$$K_c K = \frac{(\tau_1+\tau_2)}{\tau_c}, \quad \tau_I = \tau_1+\tau_2, \quad \tau_D = \frac{\tau_1\tau_2}{(\tau_1+\tau_2)} \tag{14}$$

The derivative time τ_D returned by (14) is 0.475 s, which is shorter than the 0.5 s sample period of the interface. It is retained because the fuzzy controllers implement the same derivative action through the backward difference of (8), and a configuration without it is also reported so that its contribution can be assessed directly.

The fuzzy controllers are not tuned separately. Their gains are obtained by inverting the link of (12) for the settings of the retuned inner controller, which, for GE = 1, gives the relations (15), in which the quadratic is solved first for GCE, and the other two gains follow from it,

$$\begin{aligned} &GCU = K_i T_s, \\ &GU = K_p - K_i GCE \\ &K_i GCE^2 - K_p GCE + K_d = 0 \end{aligned} \tag{15}$$

The quadratic is solved for its smaller root, and the relations return GCE = 0.480, GU = 4.963, and GCU = 0.0556. Every fuzzy configuration, therefore, carries the same equivalent proportional, integral, and derivative action as the retuned predictor, and the only quantity that differs between them is the shape of the control surface. This is what allows the effect of the surface to be separated from the effect of the tuning, which the experimental records of Section VIII cannot do.

Table VI
Controller Settings on the Common Nominal Plant

Controller	Design model	Rule	K_c	$T_i(s)$	$T_d(s)$
Conventional	first order after (13)	Case G	2.55	44.91	—
Predictor inner, PI	delay-free second order	(14)	5.02	45.15	—
Predictor inner, PID	delay-free second order	(14)	5.02	45.15	0.475
Fuzzy, all surfaces	equivalent to the row above	(15)	5.02	45.15	0.475

The two designs differ because the delay does. The conventional loop crosses over at 0.103 rad/s with a phase margin of 60.6° and a peak sensitivity of 1.61. The inner loop of the predictor, which meets no delay, crosses over at 0.201 rad/s with a phase margin of 84.6° and a peak sensitivity of 1.07. Both lie within the range ordinarily regarded as robust, so neither controller is favored by being tuned closer to its own stability limit.

B. *Test Protocol*

Two scenarios are considered, and both start from the operating point rather than from an empty vessel. Filling the vessel from empty is an actuator-limited operation in which every controller keeps the valve fully open for most of the transient, so that such a test reflects the plant rather than the controller and counts against the faster design the integral action accumulated during the rise. Each run, therefore, holds the level at 40 % of span for 250 s before any measurement is taken, and the integration window opens at the instant of the event and closes 200 s later.

In Scenario A, the setpoint is stepped from 40% to 50% of the span. In Scenario B, the setpoint is held constant, and a load step equivalent to −10% of valve opening is applied at the plant input, representing an opening of the manual drain valve, and would move the level by 18% of the span if it were not compensated. The controller output is limited to 0%-100% valve opening, as in the setup, and each controller is evaluated at the 0.5 s sample period of the interface. Each integral action uses conditional integration, so that no configuration is penalized for windup that another avoids.

Performance is quantified by four integral indices together with two quantities that the integral indices do not capture. The integral of the absolute error (IAE) and the integral of the squared error (ISE) weight the error uniformly in time, the second giving more weight to large deviations. The integral of the time-weighted absolute error (ITAE) and the integral of the time-weighted squared error (ITSE) weight the error by the time elapsed since the event, so that a slow approach to the setpoint weighs more heavily. The level is expressed as a percentage of the transmitter span, so IAE is expressed in percent-seconds, ISE in percent-squared-seconds, and the two time-weighted indices carry one further power of seconds. The first of the two additional quantities is the total variation (TV) of the controller output, the sum of the absolute changes from one sample to the next. It replaces the accumulated magnitude of the output used elsewhere in the literature, which, on a level loop, is dominated by the steady valve position and therefore reflects the operating point of the valve rather than the amount of control activity. The second is the form of the transient, reported as the overshoot and the settling time t_s in Scenario A and as the peak deviation and the recovery time t_s in Scenario B, both measured against a band of 0.2 % of span, which is 2 % of the setpoint step. All quantities are computed over the single window defined above, so that the columns of Tables VI and VII are directly comparable.

C. Setpoint Response

The responses to the setpoint step are shown in Fig. 12, and the corresponding indices are listed in Table VII. The predictor is examined first. Carrying the conventional settings unchanged into the predictor makes the response worse rather than better, and the IAE increases from 110.0 to 140.7. Those settings were detuned for a dead time that the inner loop no longer encounters, so the resulting loop is slower than the structure permits. Retuning the inner controller for the delay-free model reverses this result. The retuned predictor removes the 6.8% overshoot of the conventional loop, settles in 21.6 s instead of 30.8 s, a reduction of about 30%, and reduces the IAE to 93.4. The improvement is therefore obtained from the predictor together with a design matched to the plant seen by its controller, and not from the structure alone.

This speed is obtained at the cost of valve activity. The retuned predictor moves the valve more, and the total variation increases from 28.8 to 45.2. Delay compensation, therefore, provides a faster and less oscillatory transient at the expense of actuator activity, a trade-off that the integral indices alone do not reveal.

The control surfaces are examined next. With the tuning held common by (15), the steep surface gives the smallest IAE, ISE, and settling time among all configurations tested, 81.6, 629.1, and 16.6 s, respectively, since its membership functions produce a high slope near the origin and therefore a strong action on small errors. The linear surface follows the retuned predictor closely, as expected, because a planar surface reduces the fuzzy controller to a linear one. The remaining difference of about 6% in IAE is caused by the feedforward term in (10), which is active only when the setpoint is changing, and Scenario B, in which the setpoint is constant, shows the two responses agreeing to four significant figures. The bumpy surface does not settle within the window. Its Gaussian sets produce an almost flat region around the origin, so the control action is small when the error is small. The low total variation in this case indicates that the controller has stopped acting, rather than that it is using the valve efficiently.

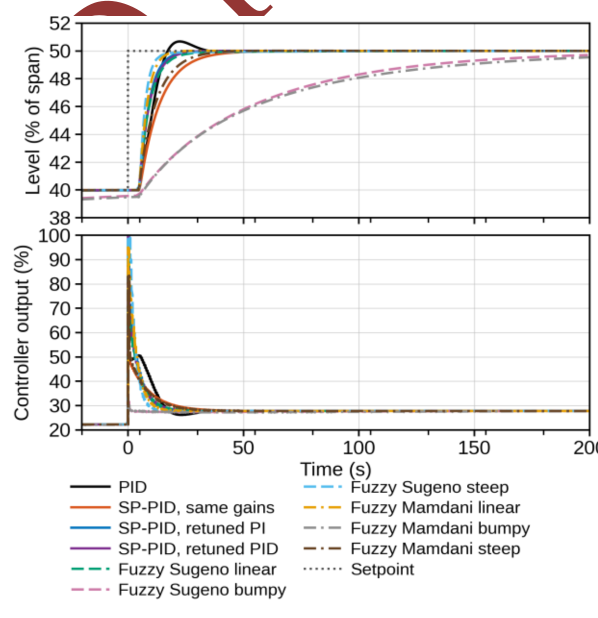

Fig. 12. Scenario A, a setpoint step from 40 % to 50 % of span, with (Top) the level and (Bottom) the controller output.

TABLE VII
Scenario A, Setpoint step. The Best Entry in Each Column is Shown in Bold.

Configuration	IAE	ISE	ITAE	ITSE	TV	OS (%)	t_s (s)
Conventional PID	110.0	834.9	839	3823	28.8	6.8	30.8
SP-PID, conventional gains	140.7	950.7	1446	5528	20.3	0.0	40.6
SP-PID, retuned PI	93.4	714.9	580	2776	45.2	0.0	21.6
SP-PID, retuned PID	94.7	709.4	598	2786	72.2	0.1	22.8
Fuzzy, Sugeno, linear	100.6	715.6	872	2870	72.2	0.0	26.2
Fuzzy, Sugeno, bumpy	571.0	3188	28026	84172	5.7	0.0	> 200
Fuzzy, Sugeno, steep	**81.6**	**629.1**	654	**2076**	72.2	0.0	**16.6**
Fuzzy, Mamdani, linear	84.1	671.1	**444**	2394	57.3	0.1	17.4
Fuzzy, Mamdani, bumpy	612.4	3381	32677	96401	**2.9**	0.0	> 200
Fuzzy, Mamdani, steep	89.4	674.9	643	2458	72.2	0.0	20.2

The two bumpy configurations give the smallest total variation in Table VII, and the smaller of the two is set in bold only to keep the convention of the table. Neither of them settles within the window, so this entry marks a controller that has almost stopped acting rather than one that uses the valve efficiently.

D. Load Disturbance Response

The responses to the load step are shown in Fig. 13, and the indices are listed in Table VIII. The improvement obtained from the predictor is smaller here than in the setpoint test. Retuning the inner controller for the delay-free model reduces the peak deviation from 3.39% to 2.97% and the IAE from 174.4 to 169.6, improvements of about 12 % in peak deviation and 3% in IAE, against the 15% in IAE and the 30% in settling time obtained for the setpoint step. The bold total variation in Table VIII corresponds to a bumpy configuration and should be read in the same way as the corresponding entry of Table VII.

This difference is consistent with the structure of the scheme. The predictor improves setpoint tracking because the controller acts on a predicted output rather than on a delayed measurement. However, it cannot detect a load disturbance earlier than the dead time permits, since the disturbance reaches the measurement through the same delay. This limitation is stated in Section VI-B, and the present result confirms it rather than reporting a uniform improvement that the theory does not support.

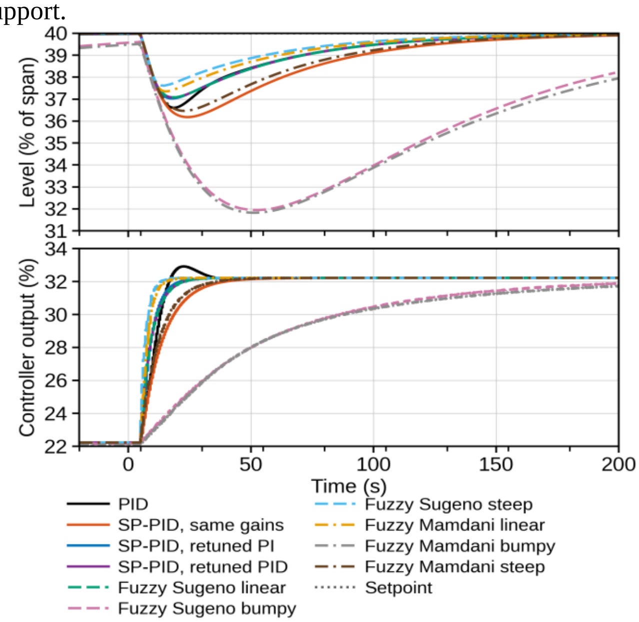

Fig. 13. Scenario B, a load step equivalent to −10 % of valve opening, with (Top) the level and (Bottom) the controller output.

TABLE VIII

Scenario B, Load Disturbance. The Best Entry in Each Column is Shown in Bold.

Configuration	**IAE**	**ISE**	**ITAE**	**ITSE**	**TV**	**Peak (%)**	**t_r (s)**
Conventional PID	174.4	327.4	9025	10993	11.5	3.39	143.8
SP-PID, conventional gains	253.3	610.7	14387	24730	10.0	3.82	167.1
SP-PID, retuned PI	169.6	296.9	8956	10432	10.0	2.97	143.9
SP-PID, retuned PID	169.6	295.8	8965	10473	10.0	2.92	143.6
Fuzzy, Sugeno, linear	169.5	295.6	8963	10467	10.0	2.92	143.6
Fuzzy, Sugeno, bumpy	983.8	5822	85521	443646	9.7	8.05	> 200
Fuzzy, Sugeno, steep	**125.2**	**168.6**	**6373**	**5495**	10.0	**2.39**	**127.6**
Fuzzy, Mamdani, linear	141.0	212.0	7241	7055	11.9	2.64	133.5
Fuzzy, Mamdani, bumpy	1021	6151	89617	474012	**9.6**	8.21	> 200
Fuzzy, Mamdani, steep	144.9	221.7	7487	7491	10.0	2.64	135.0

E. Effect of the Inference Mechanism

The three surfaces are also implemented as Mamdani systems with centroid defuzzification. The rule base of Table IV and the membership functions of Table V are kept unchanged, so that the comparison shows the effect of the inference mechanism alone. The six resulting surfaces are shown in Fig. 11.

The inference mechanism changes the surface in one systematic way. Centroid defuzzification returns the center of the area of the combined output set, which cannot reach the largest and the smallest rule outputs even when a single rule fires alone, so the Mamdani surfaces flatten out earlier and more smoothly than the Sugeno ones. What this flattening is worth depends on the membership functions. On the triangular sets, it helps in both tests. In Scenario A, the Mamdani version gives an ITAE of 444 compared with 872. It settles in 17.4 s instead of 26.2 s, and in Scenario B, it reduces the IAE from 169.5 to 141.0 and the peak deviation from 2.92 % to 2.64 %, since a moderate flattening acts on the large error produced by the disturbance without changing the slope of the surface near the origin. On the steep sets, it does not help because the flattening removes the high slope that makes that surface effective, and the Sugeno version gives better values in both scenarios for every index except the total variation and the overshoot, where the two are equal, and the ITAE of the setpoint test, where they are within 2 % of each other.

The inference mechanism is therefore not neutral, although its influence is smaller than that of the control surface. Between the best and the worst surface, the IAE changes by a factor of seven, while changing the inference mechanism on a fixed surface changes it by at most one-fifth.

F. Sensitivity to the Dead Time

The dead time used in the internal model is the only quantity in this study that the measurements leave open, since the identification gives 4.53 s for the second-order model and 5.86 s for the first-order model. The cost of this ambiguity can be measured. The internal model of the predictor is held at 4.53 s while the dead time of the plant is varied, and the setpoint step of Scenario A is applied at each value. The results are shown in Fig. 14 and listed in Table IX.

Across the range in question, from 4.5 s to 5.175 s, the IAE of the predictor rises from 93.4 to 95.6, an increase of 2.3 %, while the conventional loop over the same range rises from 109.5 to 122.5, an increase of 11.9 %. The predictor is thus not only faster at the nominal value but markedly less sensitive to getting that value wrong, and the ambiguity in the identified dead time changes none of the conclusions of this section. Degradation sets in beyond about 5.5 s, when the mismatch is no longer small compared with the dead time itself.

The same measurement bounds the effect of network jitter, that is, the variation of the delay from one transfer to the next, which Section IV assumes to be negligible. The transfers are synchronized at 0.5 s, the inter-request delay of the server is 50 ms, and no other device uses the link; hence, no transfer can be delayed by more than the interval between two transfers, and the round-trip time varies by at most one sample period. A mismatch of this magnitude changes the IAE of the predictor from 93.4 to 93.9, that is, by 0.5 %. The delay is therefore taken as constant because the closed loop is insensitive over the range that the interface can produce, and not because of an assumption about the protocol itself.

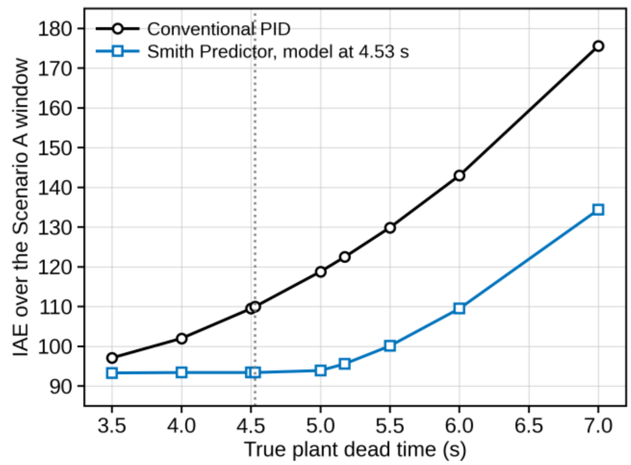

Fig. 14. Integral of the absolute error over the Scenario A window against the plant dead time, with the internal model held at 4.53 s.

TABLE IX
Sensitivity to a Mismatch Between the Plant Dead Time and the Internal Model, Which is Held at 4.53 s

Plant dead time (s)	Conventional PID, IAE	SP-PID retuned, IAE
3.50	97.1	93.3
4.00	102.0	93.4
4.50	109.5	93.4
4.53	110.0	93.4
5.00	118.8	93.9
5.175	122.5	95.6
5.50	129.9	100.1
6.00	143.0	109.5
7.00	175.6	134.4

The plant, controller settings, input sequence, and integration window are all given above so that this comparison can be repeated without access to the setup.

VIII. Experimental Implementation

The three structures were also implemented in the setup and executed in real time through the OPC interface, with read and write operations synchronized at 0.5 s intervals. This section reports the results of those runs. It is not a second comparison. The runs were performed at different times and under conditions that were not held common between them, as described below. The quantitative statement, therefore, rests on Section VII, while the records presented here show that each structure regulates the level of the physical process under the delay imposed by the interface.

A. PID and Smith Predictor-based PID

The conventional controller was implemented first, and the same controller was then placed inside the predictor. Two differences between the two runs must be stated. The internal model used in the predictor run is the transfer function shown in Fig. 8, $\frac{0.02}{(s^2+0.5s+0.007)}$ with a transport delay of 5.17 s. Its static gain is 2.86 and its dominant time constant is approximately 69 s, so that it corresponds to the first-order model of (1) rather than to the second-order model of (2), the latter being the model used in the fuzzy runs. In addition, the conventional run contained only a setpoint step, whereas the predictor run contained a setpoint step followed by a step opening of the drain valve. The block of Fig. 9 carries the coefficients of (5) rounded to two decimals, $0.08/(s^2 + 2.11s + 0.05)$, with a transport delay of 4.5 s, so that its static gain is 1.6 against the 1.8 of (2) and its internal dead time is 4.5 s against 4.53 s. Each run is therefore reported on its own; no performance index is computed from either, and the comparison of Section VII is unaffected. The two responses are shown in Fig. 15.

Both records show the level brought to the setpoint and maintained there. In the conventional run, the level reaches the setpoint after an overshoot followed by a slow return, which is what a loop closed around a delayed measurement produces. In the predictor run, the approach is faster, the deviation is smaller, and the disturbance applied later in that run is rejected without lasting oscillation. The difference goes in the same direction as the results of Section VII.

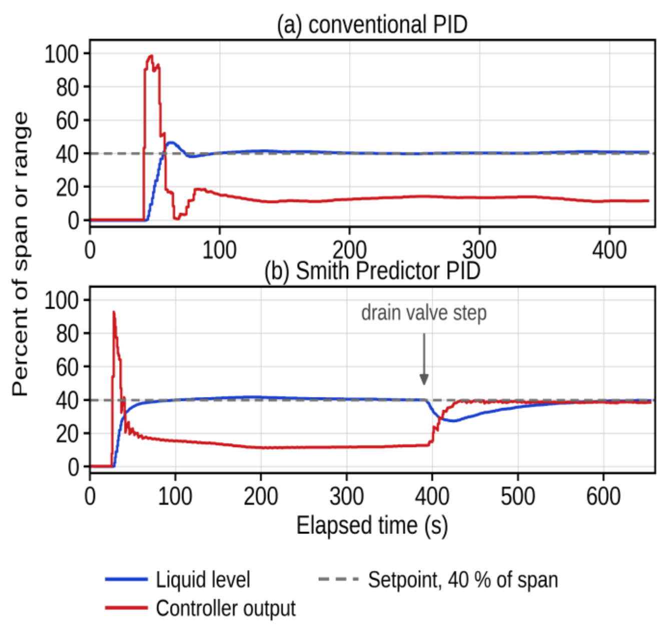

Fig. 15. Measured closed-loop response on the setup for (a) the conventional PID controller under a setpoint change and (b) the Smith Predictor-based PID controller under a setpoint change followed by a load disturbance. The blue trace is the level, and the red trace is the controller output.

B. Fuzzy PID with Different Control Surfaces

In the second set, the conventional controller is replaced by the fuzzy controller while the operating conditions and the predictor are left unchanged, and the three surfaces of Fig. 11 are compared. The measured responses are shown in Fig. 16.

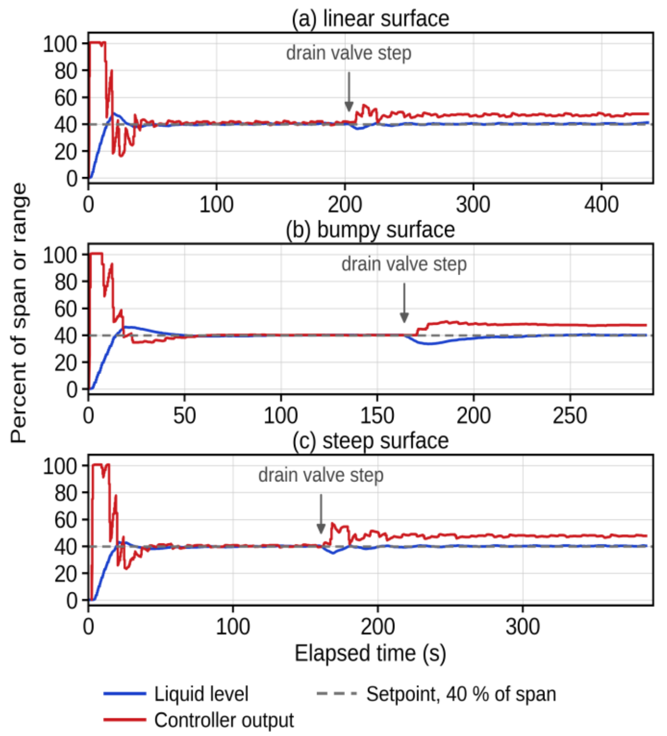

Fig. 16. Measured closed-loop response of the Fuzzy PID controller on the setup under a load disturbance for the (a) linear, (b) bumpy, and (c) steep surfaces of Fig. 11. The blue trace is the level, and the red trace is the controller output.

The three surfaces behave clearly differently. The linear surface gives a smooth approach with a moderate deviation from the setpoint. The bumpy surface moves the valve less, and its controller output trace is quieter. The steep surface removes the error fastest and shows the largest valve activity near the setpoint. The same behavior is seen on the nominal plant in Section VII. Taken together, these records show that all three structures operate on the physical process through the interface, and that each behaves in the direction predicted by the model study. The quantitative comparison is in Section VII, where the operating conditions are common to all configurations and can be reproduced from the parameters given there.

IX. Limitations

Several limitations of the present study should be stated. The identification record covers approximately one main time constant, so the static gain and the time constant of both models cannot be separated cleanly, which explains the different values returned by the two structures. The delay is treated as constant. The results in Section VII show that the variation admitted by the interface is small compared with the tolerance of the predictor. However, a synchronous protocol would remove this assumption rather than bound it. The gains of the fuzzy controller were selected by hand on the setup and differ among the three surfaces, so the experimental records in Section VIII compare surface and tuning together rather than surface alone. Section VII separates the two by deriving a common set of gains from (15), and the effect is visible in the bumpy surface, which performs well on the setup with the larger output gain used there and poorly on the common plant without it. The comparison in Section VII is carried out on a model and therefore excludes valve friction and changes in pump delivery that the identification could not represent. The experimental records of Section VIII were obtained under conditions that were not held common between runs, and the raw logs are no longer available, so that no performance index is computed from them. A fourth control surface, built from sigmoidal membership functions and the four-rule base of the steep case, was also implemented during the experimental work and is not reported here.

X. Conclusion

This paper presents the modeling, identification, and implementation of three controllers for the TE3300 liquid level process. In a laboratory setup, the path between the computer and the plant introduces a delay that the hydraulic circuit itself does not have, and in this setup, that delay is not negligible compared with the main time constant of the loop. First-order and second-order plus time delay models were fitted to a step record taken through that path. The three structures were then compared on the second-order model under conditions common to all of them, with the inner controller of the predictor retuned for the plant that it actually meets rather than inheriting settings detuned for a delay that the predictor removes. That comparison quantifies the benefit of delay compensation for this process and the effect of the shape of the fuzzy control surface. The same three structures were subsequently implemented on the setup, where each regulated the level through an interface, introducing several seconds of delay. Future work will repeat the identification over a longer record, measure the variation of the delay, and include it in the predictor, and automatically tune the gains of the fuzzy controller.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this article.

Authors Contribution Statement

The author has contributed to the whole research and preparation of the manuscript.

AI disclosure

An AI-based language model, ChatGPT (OpenAI, San Francisco, CA, USA), was used solely to improve readability and expression. Final responsibility for the accuracy and integrity of the content rests entirely with the authors.

Data Availability

All data generated or analyzed during this study are included in this published article.

References

[1] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle III, *Process Dynamics and Control*. 3rd ed. Hoboken, NJ: Wiley, 2011.

[2] B. W. Bequette, *Process Control: Modeling, Design, and Simulation*. Upper Saddle River, NJ: Prentice Hall, 2003.

[3] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, Jul. 2005.

[4] K. Ogata, *Modern Control Engineering*. 5th ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[5] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. 2nd ed. Research Triangle Park, NC: Instrument Society of America, 1995.

[6] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ: Princeton Univ. Press, 2008.

[7] O. J. M. Smith, "Closer control of loops with dead time," *Chem. Eng. Prog.*, vol. 53, no. 5, pp. 217–219, May 1957.

[8] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[9] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Mach. Stud.*, vol. 7, no. 1, pp. 1–13, Jan. 1975.

[10] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control. 4. PID controller design," *Ind. Eng. Chem. Process Des. Dev.*, vol. 25, no. 1, pp. 252–265, Jan. 1986.

[11] M. Morari and E. Zafiriou, *Robust Process Control*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[12] Q.-G. Wang, C. C. Hang, and X.-P. Yang, "Single-loop controller design via IMC principles," *Automatica*, vol. 37, no. 12, pp. 2041–2048, Dec. 2001.

[13] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning," *J. Process Control*, vol. 13, no. 4, pp. 291–309, Jun. 2003.

[14] L. Ljung, *System Identification: Theory for the User*. 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[15] R. Isermann and M. Münchhof, *Identification of Dynamic Systems: An Introduction with Applications*. Berlin, Germany: Springer, 2011.

[16] K. J. Åström, C. C. Hang, and B. C. Lim, "A new Smith predictor for controlling a process with an integrator and long dead-time," *IEEE Trans. Autom. Control*, vol. 39, no. 2, pp. 343–345, Feb. 1994.

[17] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller–Part I," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404–418, Mar./Apr. 1990.

[18] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller–Part II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 419–435, Mar./Apr. 1990.

[19] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Menlo Park, CA: Addison-Wesley, 1998.

[20] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.

[21] N. K. Poulsen, B. Kouvaritakis, and M. Cannon, "Nonlinear constrained predictive control applied to a coupled-tanks apparatus," *IEE Proc. Control Theory Appl.*, vol. 148, no. 1, pp. 17–24, Jan. 2001.

[22] N. B. Almutairi and M. Zribi, "Sliding mode control of coupled tanks," *Mechatronics*, vol. 16, no. 7, pp. 427–441, Sep. 2006.

[23] I.-L. Chien and P. S. Fruehauf, "Consider IMC tuning to improve controller performance," *Chem. Eng. Prog.*, vol. 86, no. 10, pp. 33–41, Oct. 1990.

[24] J.-X. Xu, C. C. Hang, and C. Liu, "Parallel structure and tuning of a fuzzy PID controller," *Automatica*, vol. 36, no. 5, pp. 673–684, May 2000.

[25] J. Jantzen, "Tuning of fuzzy PID controllers," Dept. Automation, Tech. Univ. Denmark, Lyngby, Denmark, Tech. Rep. 98-H 871, 1998.