A Fuzzy Water Level Control and itsImplementation for the STM32 ARM Microcontroller

Ridha Mahjoub[1], Ali Hmidene[2]
[1, 2]*Department of Electrical engineering, High Institute of technological studies, BP135, Sousse, Tunisia*

Abstract: It is very useful for developers and researchers to have at their disposal a hardware platform or a prototype of a real system that is often difficult to access. This equipment is usedto validate the theoretical results obtained because the simulation is not always faithful to the real world. In this paper, we describe the design and realization of a water storage model in which the water level is read and then controlled. The process modelling is found by mathematical equations, by the RLS estimation method and by a neural model. For the retained model and the real process, we apply two approaches of control: The first one is based on the use of an anti-windup PI control, and the second one is based on a fuzzy-PI controller. The control law algorithms are embedded in the STM32 F4-Discovery ARM microcontroller unit to control the water level in the storage model.

Keywords: Design, Realization, Water level, Real process, Neural modelling, ARM microcontroller, PI control, Fuzzy control, Simulation, Implementing.

I. Introduction

Water is a vital factor in our life, which requires that it is permanent and that we have to control its level and monitor its consumption. Managing water storage systems is important in many sectors such as in agriculture, chemistry and liquid manufacturing. One prominent aspect to be looked into is the level of water in the tank used in a manufacturing process system. The importance is in maintaining the water level to meet the consumer's own needs. In maintaining the level of a water tank, a lot studies have been done to automatically control the level of water tank systems [1], [2].

The most accepted water level controller is the classic PI controller which is widely used in both laboratory and industrial sectors. This is due to the controller ability to fit the majority of water tank systems with their simplicity despite the size, the tank construction, the monitoring system and the location at the plant. However, the difficulty in tuning the PI parameters and the moderate performance of the level control requires an extension method to optimize the performance [3]. During numerous years, the most important obstacle against the diffusion of the modern methods to control systems has been caused hardware development. Indeed, all solutions for a better classic PI controller are difficult to achieve [2]. Today and since we have computer and microcontroller powerful calculations we need to create new control algorithms.

Some numeric water level control, based on the exploitation of numerical computation units, has been proposed in the literature. However, most of them are interested in the functioning of the control algorithm, without addressing the design and identification of the platform study. Some other work has been limited to basic control and some published work can be representatively classified into three categories, which include classic control such as open loop control, and PI control [3], [4], advanced control like anti-windup PI control [5], [6], and unconventional control such as predictive [7], neural and fuzzy [1]control. Several applications have been put forward in this topic, including the use of an ATMEGA microcontroller for switching on/off the pump to limit the water between two levels [8], [9], the implementation of a PI controller on a PIC microcontroller (series 18) for water level control [4], the use of an ARDUINO board to implement fuzzy water-level control [2], [10], the implementation of a fuzzy controller on DSP [11], and the use of FPGA for automated multiple water tanks in on/off switching mode [9].

The main contribution of our work is the building of a complete test prototype that will be used in what follows in the implementation and validation of results. Another contribution is the process modelling by three methods to retain the best one that will serve as a model faithful to the real process during the simulation tests.A major contribution is the simulation and practical validation of an advanced control law and an unconventional control law.

The rest of the paper is organized as follows. In section 2, we describe the various elements constituting the system such as the process, the electronic blocks and the software part.

In section 3, we describe the modelling of the system by using three methods: mathematical identification, parametric identification (RLS) and neural networks. In section 4, two control laws are studied: an anti-windup PI controller and a fuzzy-PI controller. In section 5, we present the simulation results. We implement the two aforementioned laws on an STM32 F4-Discovery ARM microcontroller with high performances and low costs, as given in section 6. Section 7 provides an experimental test in regulation mode. Finally, in section 8, we present our conclusion.

II. System Description

The water storage model is shown in Fig. 1, starts with the pump, which receives a control signal and transforms it to a flow of water in the tank. In the output of the process, the water level is measured by using a differential pressure sensor[12].

The process components are:

- A top-tank, a hand adjustable valve, an output orifice and a down-tank.
- A DC pump for filling the top-tank.
- A series chopper for supplying the pump.
- A data reading: reference voltage, pressure sensor and measuring board.
- A control part: STM32 F4-Discovery ARM microcontroller unit.

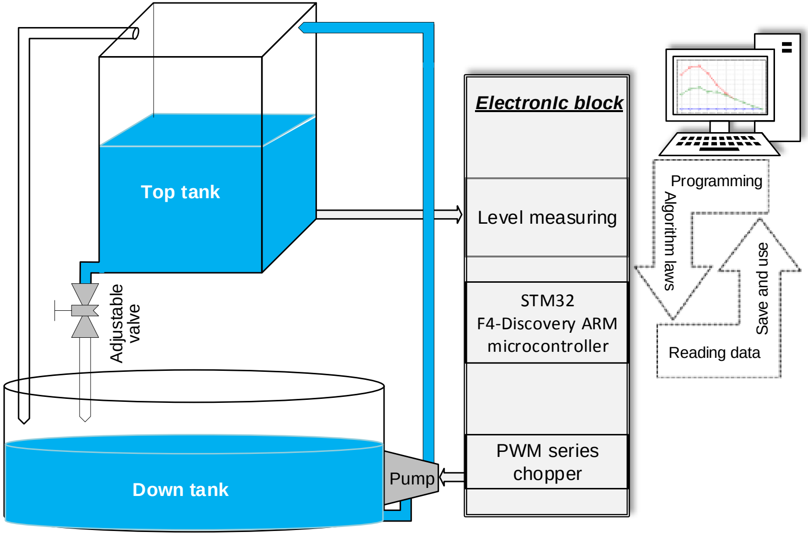


Fig. 1.Level-control model

II.1Electronicpart

It organizes data reading, conversion and adaptation between the process and the microcontroller. The electronic part, as shown in Fig. 2, is composed of the following elementary boards:

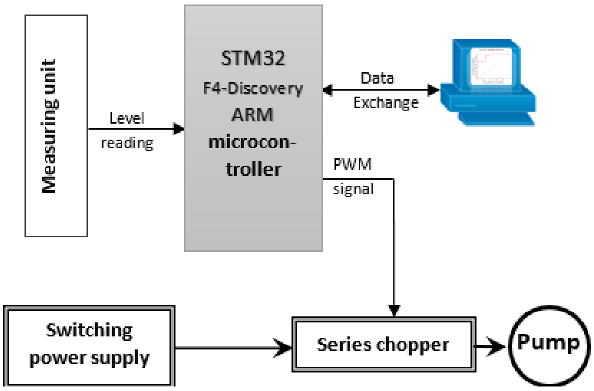


Fig. 2.Block diagram of electronic part

The measuring board measures the water level. The controller must have a well-adapted voltage with the measured water level. A differential pressure sensor, referenced HONEYWELL 26PCAFA6D, measures the pressure in the depth of the top-tank in comparison to the atmospheric pressure. To adapt the output voltage of the sensor to the scale of the ADC converter of the microcontroller, it is necessary to add an amplifier to allow the controller to read the image of the water level in the top-tank [12].

For a watercolumnheight equal to30cm, we obtain an output voltage, from the sensor, equal to 7.285mV, thuswe can calculatethe amplification gain as follows: $G = \frac{3}{7.2855 \times 10^{-3}} = 411.78$ (1)

In orderto verifythesensitivity of the sensorandthe linearity of its responseover thepressure change, a calibration operationis necessary. Thisis achievedby varying thewaterandrecordingthe values of thevoltage at the outputofthe measuring boardlevel. The results are recorded in TABLE1.

TABLE 1. Table of Measurements

H (cm)	0	3	9	12	15	21	27	30
Vout (V)	0	0.31	0.89	1.2	1.5	2.12	2.68	3

The calibration curve in Fig. 3is almostlinear. We canvarythecommandselectedproportionallyto the water levelwithout having tocompensate for themeasurement deliveredby thesensor.

The power part is a MOSFET series chopper, which uses a control signal based on Pulse Width Modulation (PWM). The series chopper delivers a periodic signal whose duty cycle determines the average value of the voltage across the pump.

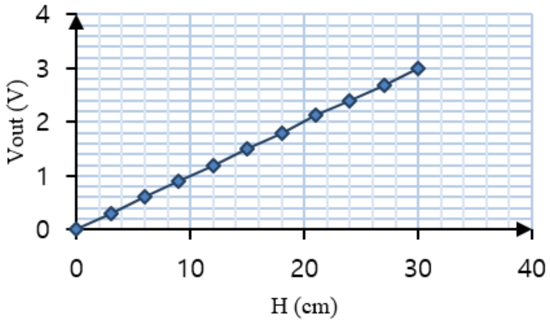


Fig. 3.Sensor calibration curve

II.2 Softwarepart

The master unit ensures the process control that contains the ARM microcontroller and its peripherals developed by ST-Microelectronics (STM). In its program memory,the content of a hexadecimal file obtained after compiling the source code with the MDK-ARM compiler is written. For the exchange of data, this master unit presents the possibility of being connected to the computer or to the industrial units having the necessary ports and communication protocols.

III. Process Modelling

The modelling of the process consists in representing its dynamic behavior with the help of a parameterized mathematical model conceiving a merely empiric model, is exclusively based on the results of measurements done on the process. This model will be used for a proofreader's training, within a system of order or as a simulator of the process[12].

III.1Mathematical model

The knowledge model is constructed from physical, chemical and biologic laws (or other types of processes), while applying the general laws founded on principles (laws of mechanics, electromagnetism, thermodynamics, quantum physics, etc.), govern the intervening phenomena within the process [2], [12].

The development of a mathematical model for the liquid tank is based on the principles of the fluid level variation, as depicted in Fig. 4.

The parameters in Fig. 4 are:

Q_i input flow, Q_o output flow, H liquid level in the tank, A_T basis area of the tank, V liquid volume, ρ mass density, A_g section of the leaking valve, S_i speed of the input liquid and S_o speed of the output liquid.

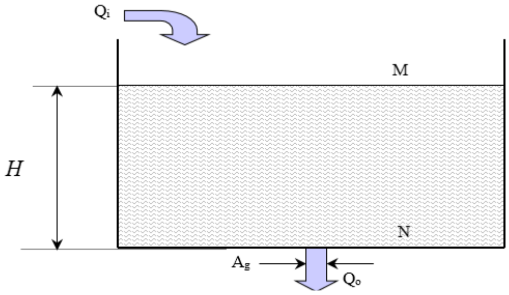


Fig. 4. Diagram of liquid tank with leakage

Therefore, for every point M situated in height H of the exhaust opening, we have:
Using the Laplace transform, we obtain:

$$H = \frac{1}{A_T s}\left(Q_i - A_g\sqrt{2gH}\right) \text{ (2)}$$

The nonlinear model is presented in Fig. 5.

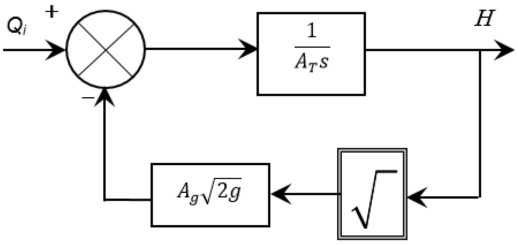


Fig. 5. Block diagram of nonlinear model

III.2Process linearization

Equation (2) is nonlinear. We try to achieve the linearization of the mathematical model in order to determine a linear transfer function and to know the system order.
An operating point is fixed and characterized by input flow Q_{iM} and height H_M, knowing that the level depends on the flow. The latter cannot be fixed. Subsequently, during the numeric application the transfer function will only be treated arbitrarily.
For every point M situated in height H of the exhaust opening, we have: $H = h + H_M$ and $Q_i = q_i + Q_{iM}$, where h and q_i reflect the small variations around the considered operating point.
The transfer function of the linearized system is:

$$G(s) = \frac{h(s)}{q_i(s)} = \frac{b}{a+s} = \frac{K}{1+\tau s} \text{ (3)}$$

With:
$a = k\frac{A_g}{A_T}\sqrt{\frac{g}{2H_M}}$ when k=1, $k \in [0,1]$: the opening coefficient of the valve, $b = \frac{1}{A_T}$, $K = \frac{b}{a}$ and $\tau = \frac{1}{a}$.

III.3RLS estimation

The system is identified by the response to a control signal modulated in amplitude, width and length, so that the parameters of the model can be close to those of the system.
Alternatively, as an equation to the differences we have:

$$y(k) + ay(k-1) = bu(k-1) \text{ (4)}$$

Where: u(z) is the numeric value of the set point that corresponds to the q flow and y(z) is the numeric value of the voltage equivalent to height h.
As the model only represents an approximation of the first order of the real system, a corrective term is added. The relation is finally described by:

$$y(k) = -ay(k-1) + bu(k-1) + e(k) \text{ (5)}$$

In general, the output of process $y(k)$ in a determinist approach can be written under the following matrix shape:

$$y(k) = \theta^T(k)\phi(k) \text{ (6)}$$

where $\phi(k)$ and $\theta(k)$ are defined respectively by: $\theta^T(k) = \begin{bmatrix} a & b \end{bmatrix}$ and $\phi^T(k) = \begin{bmatrix} -y(k-1) & u(k-1) \end{bmatrix}$.
We define the predicted output a priori of the process by:

$$\hat{y}(k) = \hat{\theta}^T(k)\phi(k-1) \text{ (7)}$$

where $\hat{\theta}(k)$ is the vector of the estimated value of parameters.
In practice, the measures done on the process are always blemished of noise (modelling error, measurement error, etc.). A gap, called the prediction error, between the process output and the predicted output by the adjustable model exists. The prediction error is defined by:

$$e(k) = y(k) - \hat{y}(k) \text{ (8)}$$

The *RLS* identification algorithm that allows determining the parameters of the process is given by the following stages:

$$\hat{\theta}(k+1)=\hat{\theta}(k)+P(k+1)\phi(k+1)e(k+1) \quad (9)$$

$$P(k+1)=P(k)-\frac{P(k)\phi(k+1)\phi(k+1)^{T}P(k)}{1+\phi(k+1)^{T}P(k)\phi(k+1)} \,(10)$$

$$e(k+1)=y(k+1)-\hat{\theta}^{T}(k)\phi(k+1)\,(11)$$

The *RLS* identification algorithm permits a sequential treatment of the experimental data, which that is available to every instant. This enables the evaluation of the parameters in real time with one relatively weak calculation time [12], [15].

III.4Determination of linear model parameters

To determine the parameters of the process, we achieve the convenient tests, which permit estimating parameters *a* and*b* according to the linear model described by (3). The adjustable valve is maintained to mid-opening (k=0.5).

The RLS estimation method consists in:
1. Applying a set point vector, which is modulated in order to control the flow of the pump;
2. Recording the vector of the set point and that of the system response (the water level) in a file;
3. Exploiting the descended values in a program allowing the estimation of parameters a and b of the discrete model, as a are function of static gain K and time constant τ of (3), where T_e represents the sampling time.

with: $$\begin{cases} a=-e^{-\frac{T_e}{\tau}}=-0.99 \\ b=K(1-e^{-\frac{Te}{\tau}})=6.17 \end{cases} \qquad \text{when } T_e=0.2s\,.$$

4. Finally verifying the identification of the process by the application of the same control signal to the obtained model and to the real system.

The results of identification are presented in Fig. 6.

The results got for the identification of the process by theRLS method show that there exist some obvious constraints because the process is estimated by a first-order system. According, we need to find a better modelling.

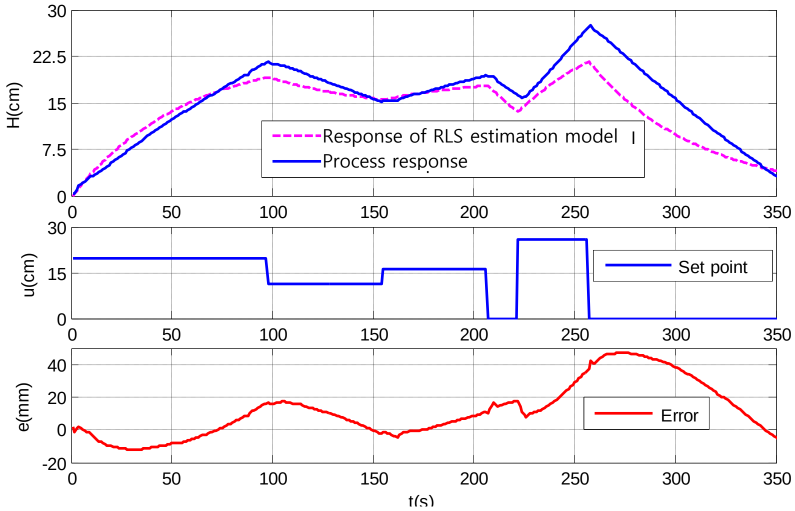


Fig. 6. Validation test of the linear model

III.5Neural model

The first stage consists in assembling the knowledge of the behavior of the process from experimental tests. Thereafter, an algorithm of training is set in motion. This algorithm adjusts the parameters of the neural model to minimize a definite cost function from the gap between the output of the process and the values generated by the network (prediction error). At the end of the real model identification, we keep definitely obtained neural model. The validation is done according to themodel's performances for the foreseen use (control or simulation)[16].

We make the hypothesis that the process can be described correctly, in the domain of validity wished, by a representation of the shape given by (12).

$$y_d(k) = \varphi\left[u(k), u(k-1), u(k-2), y_d(k-1), y_d(k-2)\right] \quad (12)$$

This neural model, after the success of the operation of training in accordance with the tabular diagram, as shown in Fig. 7, presents the same behaviour as the real process[12].

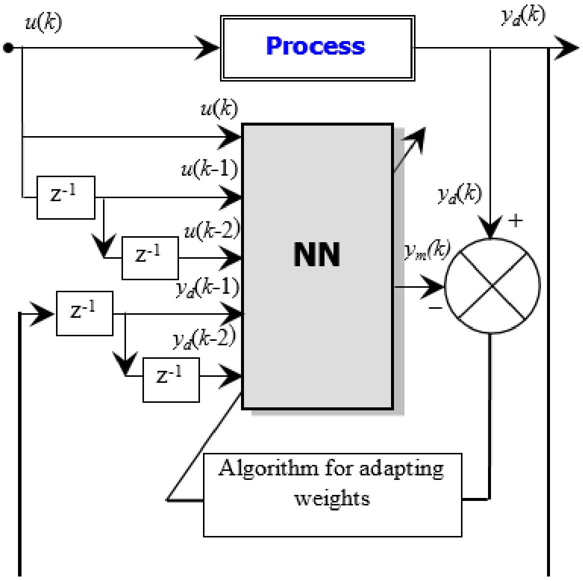


Fig. 7. Bloc diagram of the neural method

The validation of the retained model is made by applying the same order on the neural model input and the real process. The results of neural identification are presented in Fig. 8.

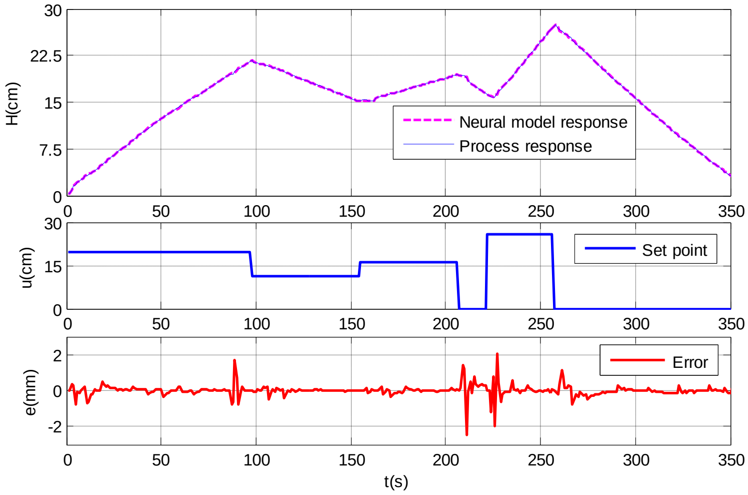


Fig. 8. Validation test of neural model

The response of the neural model compared to that of the real process, as illustrated in Fig. 8, allows us to record an error not exceeding 1.35%. However, with the RLS method, this error is multiplied by more than 10 times (18%). It is clear that the retained model will be the neural one. This model will facilitate a lot the choice of the controller parameters and allow relieving the real process on which these parameters are tested and kept.

IV. Synthesis of Control Laws

Today, a number of different controllers are used in industry and in many other fields. In a quite general way, those controllers can be divided into two main groups:

- Conventional controllers
- Unconventional controllers

As conventional controllers, we have to know the mathematical model of the process in order to design a controller. Unconventional controllers use new approaches to the controller design where the knowledge of a mathematical model of a process is not generally required. Examples of unconventional controller are the fuzzy controller and the neural or fuzzy-neural controller [2], [3], [17].

IV.1Anti-windup control using a PI controller

Many industrial processes are nonlinear and therefore complicated to be described mathematically. However, it is known that a good nonlinear process can be satisfactorily controlled using PID controllers providing that controller parameters are tuned well. The practical experience shows that this type of control has a lot of sense since it is simple and based on three basic behaviour types: proportional (P), integrative (I) and derivative (D). Instead of using a small number of complex controllers, a larger number of simple PID controllers are used to control simpler processes in an industrial assembly in order to automate a certainly more complex process. The PID controller and its different types, such as P, PI and PD controllers, are today the basic building blocks in control of various processes. In spite their simplicity, they can be used to solve even very complex control problems, especially when combined with different functional blocks, filters (compensators or correction blocks), selectors, etc. A continuous development of new control algorithms ensures that the PID controller is not obsoleteand that this basic algorithm will have its part to play in the process control in the foreseeable future. It can be expected that it will be a backbone of many complex control systems [13].

The transfer function of a classic PI controller is given by (13).

$$c(s) = k_p\left(1+\frac{k_i}{s}\right) \quad \text{when} \;\; k_i = \frac{1}{T_i} \, (13)$$

Using the backward approximation $s = \frac{1-z^{-1}}{T}$ when T is the sampling period, the discrete form of (13) is:

$$c(z) = \frac{u(z)}{e(z)} = k_p\left(1+\frac{T\cdot k_i}{1-z^{-1}}\right) (14)$$

The controller output is:

$$u(z) = u_p(z)+u_i(z) \text{ when: } u_p(z) = k_p.e(z) \text{ and } u_i(z) = k_p\frac{T\cdot k_i}{1-z^{-1}}.e(z) \; (15)$$

The differential equation is:

$$u_p(k) = k_p e(k) \text{ and } u_i(k) = u_i(k-1)+T\cdot k_i\cdot k_p\cdot e(k) \;\; : \;\; u(k) = u_p(k)+u_i(k) \; (16)$$

The added anti-windup circuit compares the output of the controller and the saturation block, and according to the sign of this error, it acts on the integrator to bring the output of the controller to the saturation limit [5], [6].

An anti-windup PI controller, shown in Fig. 9, is based on the conditional integration method. We can write the following equations:

$$e(k) = h_{sp}(k)-h(k) \text{ and } u_p(k) = k_p e(k) \;\; (17)$$

$$u_i(k) = u_i(k-1)+k_i T\left[k_p e(k)+e_{sat}(k-1)\right] \; (18)$$

with: $e_{sat}(k-1) = k_m\left[v(k-1)-u(k-1)\right]$ and $u(k) = u_p(k)+u_i(k)$

$$v(k) = \begin{cases} u_{\max} & \text{if} \quad u(k) > u_{\max} \\ u_{\min} & \text{if} \quad u(k) < u_{\min} \\ v(k) & \text{if} \quad u_{\min} \le u(k) \le u_{max} \end{cases} \qquad (19)$$

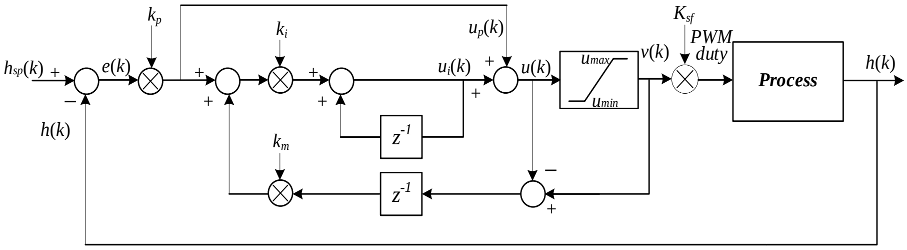


Fig. 9. Block diagram of anti-windup PI controller

In the linear range, the error is integrated and the difference between the saturated and unsaturated control signals is used to generate a feedback signal to properly control the integral state in the saturation range such that it may seem advantageous to choose a very large value for the anti-windup gain k_m because the integrator can be limited quickly. If an anti-windup gain is chosen, which is too big, a spurious error can cause an input saturation, which many accidentally reset the integrator.

IV.2Fuzzy-PI controller

In linguistic models, also called fuzzy Mamdani models, the antecedents and consequences of rules are fuzzy propositions. For our application, the control variables for the regulation of the system are [1], [2]:

- Error *e* representing the difference between a predetermined command and the real output
- The variation of error $\Delta e = (e(k) - e(k-1))$ between two periods of sampling time
- Output Δu representing the variation in set point *u*

By applying the bilinear transformation $s = \frac{2}{T} \cdot \frac{1-z^{-1}}{1+z^{-1}}$, the discrete form of (13) is:

$$c(z) = k_p + k_p \frac{k_i T}{2} \cdot \frac{1+z^{-1}}{1-z^{-1}} = K_p + k_p \frac{K_i}{1-z^{-1}} \quad (20)$$

with $K_p = k_p \left(1 - \frac{k_i T}{2}\right)$ and $K_i = k_p k_i T$ .The discrete set point is:

$$u(z) = z^{-1}u(z) + K_p\left[e(z) - z^{-1}e(z)\right] + K_i e(z) \quad (21)$$

We deduce the recurrent form of the set point as follows:

$$u(k) = u(k-1) + K_p\left[e(k) - e(k-1)\right] + K_i e(k) \Rightarrow \Delta u(k) = u(k) - u(k-1) = K_p \Delta e(k) + K_i e(k) \quad (22)$$

with, $e(k) = h_{sp}(k) - h(k)$ et $\Delta e(k) = e(k) - e(k-1)$.

The fuzzy-PI controller has $k_i.e(k)$ and $k_p.\Delta e(k)$ for inputs and $\Delta u(k)$ as an output for which we add $u(k-1)$ to have the output. Output $u(k)$ is multiplied by scaling factor K_{sf} to have the final output of the controller (*duty* of the PWM signal). The functional diagram of the fuzzy-PI controller designated for the control of the system is depicted in Fig. 10[17], [18].

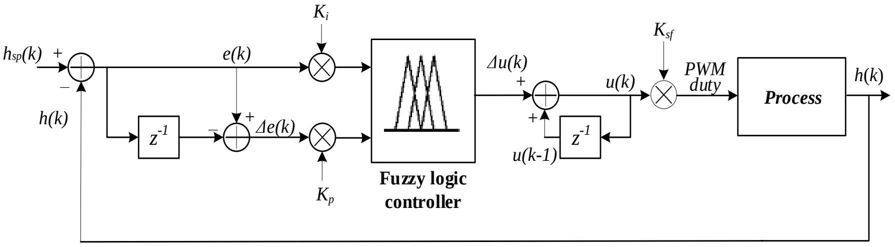


Fig. 10. Block diagram of fuzzy-PI controller

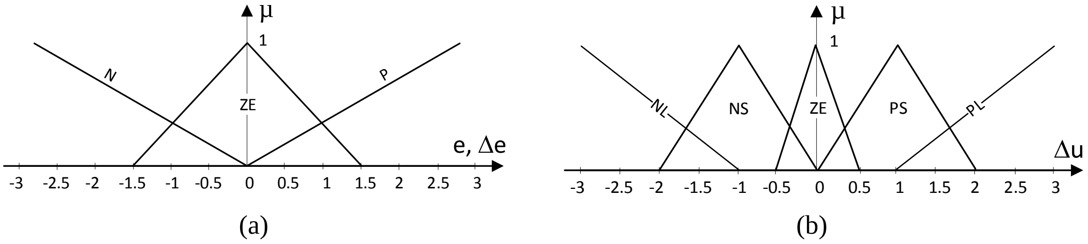
IV.2.1 Fuzzification

Fuzzification is the process, that determines the degree of membership of the input values according to fuzzy sets (linguistic variables). In order to define the variables of control, we must specify the fuzzy subsets associated to these variables as well as their functions of adherence [17], [19].

- Three fuzzy sets are defined for input values $e(k)$ and $\Delta e(k)$: Negative (N), Zero Equal (ZE), Positive (P).

- Five fuzzy sets are defined for output value $\Delta u(k)$: N: Negative, Negative Large (NL), Negative Small (NS), Zero Equal (ZE), Positive Small (PS) and Positive Large (PL).

The membership functions of inputs $e(k)$ and $\Delta e(k)$ and output $\Delta u(k)$ are given by Fig. 11.



(a) (b)

Fig. 11. Membership functions: a) inputs e and Δe. b) output Δu.

IV.2.2 Fuzzy inference

The basis of rules is constituted by nine rules that treat all combinations for the quoted variables. This rule base was initially introduced by Mac Vicar Whelan [3], [18].

The rules have the following generic shape:

$$\text{If } (e \text{ is } A) \text{ and } (\Delta e \text{ is } B) \text{ then } (\Delta u \text{ is } C) \quad (23)$$

where A, B and C designate the fuzzy subsets. Equation (23) represents the different rules. TABLE2 gives the associated numbers.

TABLE 2. Table of rules

	Δu		
e / Δe	N	Z	P
N	NG	NP	ZE
Z	NP	ZE	PP
P	ZE	PP	PG

The operator (minimum) for the fuzzy inference and the operator (maximum) for the aggregation of the rules construct the premises and the findings of the rules.

IV.2.3 Defuzzification

The reverse operation of moving from a linguistic variable representation to a physically applicable numerical variable is calleddefuzzification. Several methods exist to obtain a precise value from a fuzzy subset. The two most common subsets are the centroid calculation and the average of the maxima [3], [19].

The used defuzzification method is the center of gravity method because experience has proved that it gives good results. The general mathematical formula, which is used to obtain the centroid point, is:

$$z_{COA} = \frac{\int_z \mu_A(z) \cdot z \cdot dz}{\int_z \mu_A(z) \cdot dz} \quad (24)$$

where Z$_{COA}$ is the centroid of area, and μ_A is the aggregated output member function.

V. Simulation Results

Before proceeding to the implementation of control laws on the ARM microcontroller, we will exploit the neural model (the most faithful to the real process) to simulate both types of control.

V.1Fuzzy control

To control the water level, we want the controller output to be a number (net value) and not a fuzzy set. For the PWM value, we do not want the controller to generate a large PWM duty, for example. What we want to know is the exact value of the PWM duty to be generated. The result of defuzzification must be a numerical value, which determines the duty factor of the PWM signal used to drive the process [1], [3], [13], [14], [17].

Mainly a fuzzy-PI controller will control the process as a simulating mode, as described in Fig. 12.

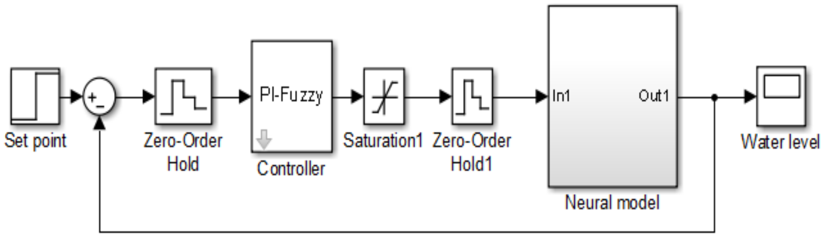


Fig. 12.Simulink schema of the fuzzy control

The order of the system driven by a fuzzy-PIcontroller, for a set point of 866 and which corresponds to a desired level of 26*cm*, leads to the result shown in Fig. 13.

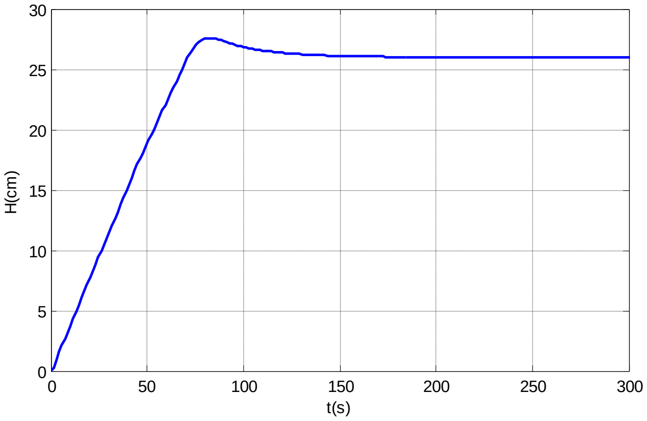


Fig. 13.Step response of neural model controlled by fuzzy-PI controller

This simulation response shows that the fuzzy-PI control of the neural model gives a rise time of 62s, and an overshoot of 6.53% and it reaches a steady state after 140s.

V.2Comparative study

Generally, we use a PI controller to impose the performances in servitude (stabilization of the system and annulment of the static error) [3], [13], [20].

If there are large transport delays present in the controlled process, error prediction is required. However, mode D cannot be used for prediction because every information is delayed until the moment when a change in the controlled variable is recorded. Accordingly, it is better to predict the output signal using the mathematical model of the process in a broader sense (process + actuator).

In this case, proportional parameter K_P and integral parameter $T_i$are obtained using Ziegler and Nichols tuning. However, in order to obtain the same rise time and the same overshoot as those given by the fuzzy-PI controller, these parameters are slightly modified following a success of simulation tests on the neural model, given the following parameters: $K_p = 1.8$ and $T_i = 55s$ .The controller output is multiplied by a scaling factor to have the *duty* of the PWM signal.Also, the anti-windup PI controller will control the neural model as a simulating mode, as described in Fig. 14.

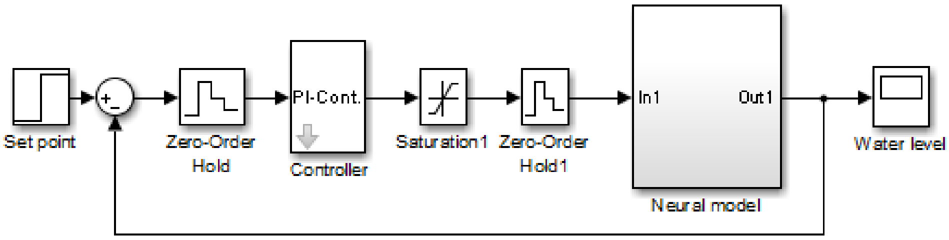


Fig. 14.Blok diagram of anti-windup PI controller

Fig. 15 shows two responses, presented on the same plane, of the neural model controlled by an anti-windup PI controller and by a fuzzy-PI controller.

For the same rise time and overshoot, these responses justify that the fuzzy-PI controller gives a settling time of 140s, whereas the controller-PI one is 230s. This allows the process to reach its stationary state more quickly with the presence of the fuzzy-PI controller.

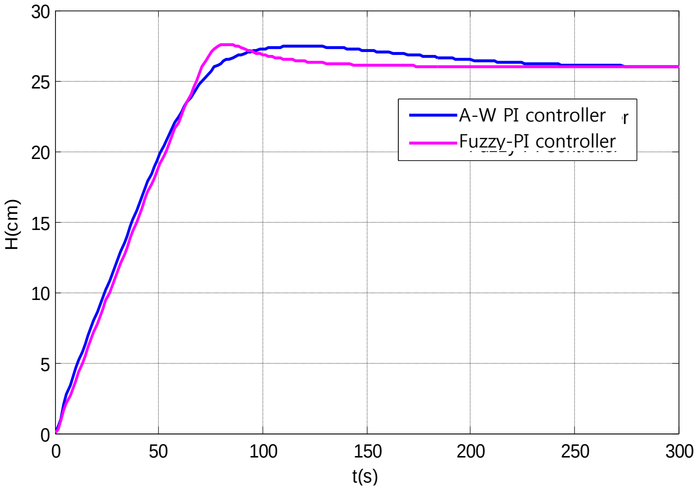


Fig. 15. Response of neural model for two types of controllers

VI. Implementing Algorithm Laws

The controller programs are implemented in the STM32 F4-Discovery ARM microcontroller, where the input port receives a signal from the measurement board and the output port generates the PWM duty to control the pump via the power board [7], [19].

As a result, we avoid damage caused by excessive control entries by defining an upper limit and a lower limit. However, since our control system incorporates a PI controller, the input continues to increase after reaching the target quantity, which can lead to excessive overshoot.

In experimental tests, the PI controller does not always work in the linearity zone: The digital correspondent of the PWM duty (coded on 12 bits) limits the output of the digital controller between two values, Min and Max. The windup phenomenon is observed when the controller output reaches the upper limit. Yet the controller continues to integrate the error and then provides a large control value even though it has exceeded the limit. The consequence is the appearance of very marked oscillations, thus increasing the stabilization time of the system. To remedy this limitation, we add an anti-windup circuit, which compares the output of the controller and the saturation block, and according to the sign of this error, it acts on the integrator to bring the output of the controller to the saturation limit.

The tests on the real process, controlled by an anti-windup PI controller (blue response) and by a fuzzy-PIcontroller (purple response), give the responses represented in Fig. 16. The applied set point corresponds to a desired level of 26*cm*, and the valve is maintained to mid-opening.

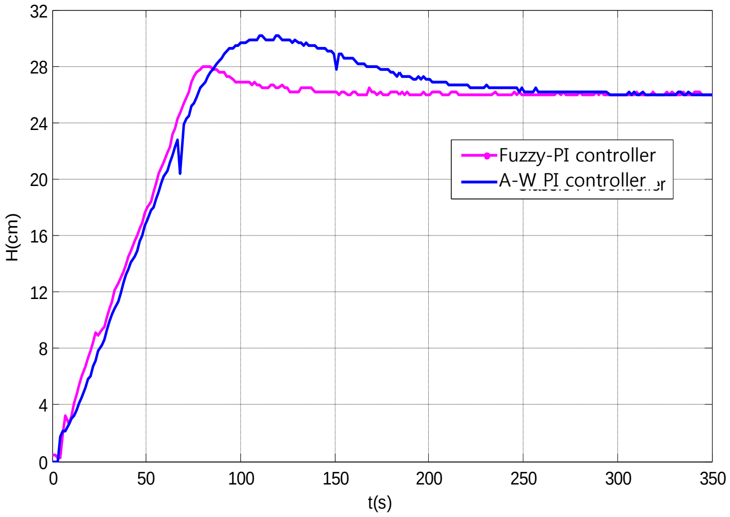


Figure 16. Response of real process for two types of controller

Fig. 16 justifies the interpretation from Fig. 15, not only for the rise time but also for the fuzzy-PI controller, which has less overshoot.

VII. Experimental Test in Regulation Mode

When the system reaches a steady state at 26*cm*, we make a total opening of the valve. The consequence is a transitory decrease in the water level in the tank.

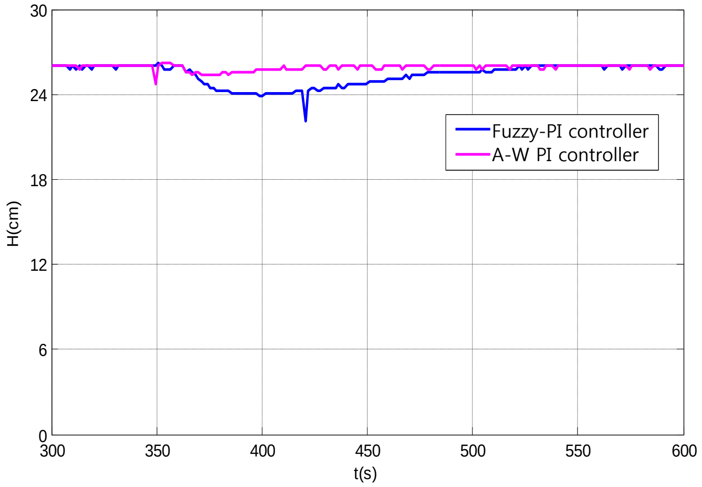


Fig. 17.Applying a disturbancesat t=350s

Fig. 17 shows that the fuzzy- PIcontroller reacts quickly. At the end of 80*s* the level of water in the tank joins its initial value, whereas the response of the anti-windup PI controller is slower and it requires 180*s* to come back to its steady state. The experimental results indicate, unlike the PI controller, that the fuzzy- PI controller is more robust against disturbances and that it rejects errors well.

VIII. Conclusion

In this paper, we have presented the design and realization of a water storage prototype and used several methods to find its correct model. We have carried out computer simulation to apply two control laws on this model, before moving on to the real process. Finally, we have made a comparative study of the controller performances. Although this method saves us a lot of time, it reduces the wear and depreciation of the equipment. Nevertheless, it requires the correct exploitation of the aforementioned technological means and some scientific knowledge. This work has been done by our own means and it has taken an important time. This prototype will be considered as a platform to implement and validate our future research results.

References

[1] Reshmi B. “WATER LEVEL CONTROLLER BY FUZZY LOGIC”, *International Journal of Innovative research in Advanced Engineering* (IJIRAE), Vol. 2, pp. 250-261, 2015.

[2] Nur M.M.S., Munirah M.S, Mohd B. B., Md N. O. and SitiH.J.“Fuzzy Takagi-Sugeno Method in Microcontroller Based Water Tank System”, *International Journal of Robotics and Automation* (IJRA), pp. 1-7, 2018

[3] Davood M.S.S., Hamidreza A. and Faridoon S. ″Comparative study between tank’s water level control using PID and fuzzy fogic controller”, *Advances in Intelligent Systems and Computing*, pp. 41-53, 2013.

[4] Ardavan M. ″Implementation of PID Controller by Microcontroller of PIC (18 Series) and Controlling the Height of Liquid in Sources”, *Advances in Robotics & Automation*, Vol. 5, Issue 3, 2016.

[5] Xin-lan L., Jong-Gyu P., and Hwi-BeomS.“Comparison and Evaluation of Anti-Windup PI Controllers”, *Journal of Power Electronics* (JPE), Vol.11, No.1, pp. 45-50, 2011.

[6] Kyohei S., Yoshihisa I.“A Design of an Improved Anti-Windup Control Using a PI Controller Based on a Pole Placement Method”, *International Journal of Simulation: Systems, Science & Technology*, pp. 1.1-1.7, 2016.

[7] Rihab K., Hichem S. and FaouziB.“Application of Model Predictive Control for a thermal process using STM32 ARM Microcontroller”, *ICCAD’17*, Hammamet, Tunisia, IEEE, pp. 146-151, 2017.

[8] Md. J.I., and Shaikh K.M. ″Microcontroller Based Water Level Detection and Pump Control Using Ultrasound″, *Journal of Multidisciplinary Engineering Science and Technology* (JMEST), Vol. 3, Issue

4, pp. 4565 – 4568, 2016.

[9] Lukman A.A., Blessing O.A., James A., Abdulzeez O.A., Muhammed B.M., and AbdulazeezF.S.“Automated Multiple Water Tanks Control System Using ATMEGA and FPGA Technology”, *IEEE International conference on Sciences and Techniques of Automatic control & computer engineering*, pp. 346-353, Nigeria,2019.

[10] Fayçal C., Rachid T., Abderrahmen B. and Mohammed A.B.“The Application of Fuzzy Control in Water TankLevel Using Arduino”, *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 4, pp. 261-265, 2016.

[11] Mourad T., Anis S. and Mohamed B.“Implementation of a Fuzzy Logic Control System using a TMS320F2812 DSP”, *International conference on Sciences and Techniques of Automatic control & computer engineering*, pp. 1087-1098, 2009.

[12] Ridha M., “Conception et realization d’unemaquette de stockaged’eaudans unsite isolé”, *International conference on innovative materials, manufacturing, and advanced technologies, Monastir- Tunisia, http://www.ijartech.com/sessionIMMAT2019.php, October 2019.*

[13] Abirami S., Lina R.“Parallel Distributed Compensator Design For Non Linear Tank”, *International Journal of Novel Research in Electronics and Communication*, Vol. 2, pp. 1-7, 2015.

[14] Elamari A E. “Design of heuristic fuzzy logic controller for liquid level control”, *Fifth International Conference on Intelligent Systems, Modelling and Simulation*, pp. 131-136, (2018).

[15] Honghong D., Jie J., RuifengD.“Two-stage recursive least squares parameter estimation algorithm for output error models”, *Mathematical and Computer Modelling*, ELSEVIER, Vol. 55,Issues 3-4 pp. 1151-1159, 2012.

[16] Juri B., Eduard P.“ Model based control of a water tank system”, *19th World Congress The International Federation of Automatic Control*, ELSEVIER, Vol. 47,Issues 3 pp. 10838-10843, 2014.

[17] Shilpa S., S. V. Halse “Simulink Modeling for Liquid Level Control Using Fuzzy Logic Controller”,*International Journal of Scientific Research Engineering & Technology* (IJSRET), Vol. 4, pp. 77-81, 2015.

[18] Abhijit N. “REAL-TIME CONTROL IMPLEMENTATION OF SIMPLE MECHATRONIC DEVICES USINGMATLAB/SIMULINK/RTWPLATFORM”, American Society for Engineering Education, pp. 1-12, 2011.

[19] Bernard W., Edward H. “Implementation of the FITA Fuzzy Inference System on the specific microcontroller platform”, ELSEVIER,48-4, pp. 165-169, 2015.

[20] Sumit K., Mrs. Pooja N.“COMPARATIVE ANALYISIS OF P, PI, PID AND FUZZY LOGIC CONTROLLER FOR TANK WATER LEVEL CONTROL SYSTEM”,*International Research Journal of Engineering and Technology* (IRJET), Vol. 3, pp. 1174-117, 2017.