USING FUZZY LOGIC CONTROLLER FOR A TWO- TANK LEVEL CONTROL SYSTEM

Omar W. Abdul-Wahab

ABSTRACT

This paper presents a fuzzy logic controller for a two-tank level control system, which is a process with a dead time. The fuzzy controller is a proportional-integral (PI-like) fuzzy controller which is suitable for steady state behavior of the system. Transient behavior of the system was improved without the need for a derivative action by suitable change in the rule base of the controller. Simulation results showed the step response of the two-tank level control system when this controller was used to control this plant and the effect of the dead time on the response of the system.

الخلاصة

هذا البحث يقدم مسيطرمنطق ضبابي للسيطرة على نظام مستوى الماء في خزانين. المسيطر الضبابي هو مسيطر تناسبي- تكاملي والذي يكون مناسباً لسلوك النظام في الحالة المستقرة. أما فيما يخص سلوك الحالة الإنتقالية للنظام فقد تم تحسينه دون الحاجة إلى التأثير التفاضلي وذلك بواسطة إجراء تغيير مناسب في قاعدة بيانات المسيطر. النتائج التمثيلية بينت استجابة نظام الخزانين لإشارة الإدخال بإستخدام هذا المسيطر وكذلك بينت تأثير زمن التأخير على تلك الاستجابة.

KEYWORDS
PI-like fuzzy logic controller, two-tank level control system.

INTRODUCTION

The vast majority of conventional control techniques have been devised for linear-time invariant systems which are assumed to be completely known and well understood. In most practical instances, however, the systems to be controlled are nonlinear and the basic physical processes in it are not completely known a prior. These types of model uncertainties are extremely difficult to manage even with the conventional adaptive techniques (Abonyi, Nagy, and Szeifert 2005).

The past few years have witnessed a rapid growth in the use of fuzzy logic controllers for the control of processes that are complex and badly defined. Most fuzzy controllers developed till now have been of rule-based type (Driankov, Hellendoorn, and Reinfrank 1996), where the rules in the controller attempt to model the operator's response to particular process situations. An alternative approach uses fuzzy or inverse fuzzy model in process control (Babuska, Sousa, and Verbruggen

1995) because it is often much easier to obtain information on how a process responds to particular inputs than to record how, and why, an operator responds to particular situations.

In recent applications, many authors started applying more sophisticated fuzzy logic controller structures in sake of a more robust fuzzy logic controller and overcoming some of the difficulties associated with fuzzy logic controller design, in general. Self organizing (Sutton and Jess 1991), self tuning (Isomursu and Rauma 1994) and adaptive (Pave and Chelaru 1992) (Fei and Isik 1992) fuzzy logic controllers have become very popular. Some of the tuning methods assume an existence of the initial fuzzy logic controller model and the availability of the plant model. However, most of the late design methods do not require any plant model at all (Ghanayem, and Reznik 1996).

THE STRUCTURE OF THE PI-LIKE FUZZY LOGIC CONTROLLER

In practice, full PID control sometimes is not desired. Instead, partial PID control in the form of PI or PD control is more effective and appropriate. This is because the derivative term tends to amplify noise and hence should be avoided if the system output is rather noisy. On the other hand, the integral term can cause slower system response and larger system overshoot (Ying 2000). PI-like fuzzy controller is known to be more practical than PD-like fuzzy controller because it is difficult for the latter to remove steady state error (Karasakal, Yesil, Guzelkaya and Eksin 2005). An equation giving a conventional PI-controller is

$$u(t) = K_P \times e(t) + K_I \times \int e(t)\,dt \tag{1}$$

where K_P and K_I are the proportional and integral gain coefficients, respectively. Differentiating Eq. (1) with respect to t yields

$$\frac{du(t)}{dt} = K_P \times \frac{de(t)}{dt} + K_I \times e(t) \tag{2}$$

The corresponding discrete-time form is

$$\Delta u(kT) = K_P \times \Delta e(kT) + K_I \times e(kT) \tag{3}$$

where T is the sampling interval.

For the PI-like fuzzy controller, the inputs are the error $e(kT)$ and the change-of-error $\Delta e(kT)$ and the output is the change-of-control $\Delta u(kT)$. To obtain the control output variable $u(kT)$, the change-of-control output $\Delta u(kT)$ is added to $u((k-1)T)$. This takes place outside the fuzzy controller and is not reflected in its rule base. A block diagram of this controller is given in **Fig. (1).**

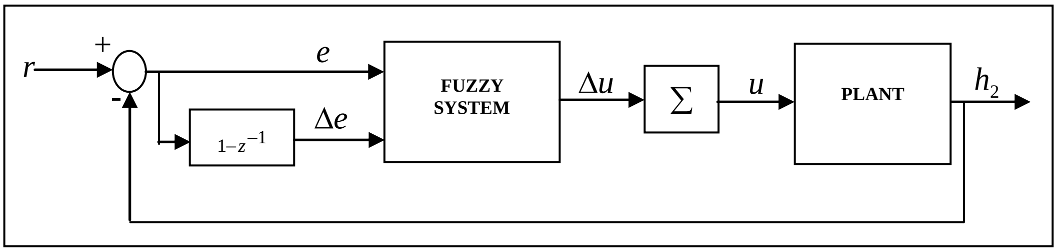

Fig (1): Block diagram of the PI-like fuzzy logic controller.

FUZZIFICATION

This controller uses singleton fuzzification method. The reason is that when singleton fuzzification is used, combining the inputs with rule premises is reduced to computing the values of the input membership functions at the current input values. Other fuzzification methods add computational complexity to the inference process and the need for them has not been well justified. (Passino and Yurkovich 1998).

KNOWLEDGE BASE

The knowledge base of the fuzzy logic controller consists of the following parameters:

Membership Functions

For each input variable ($e(kT)$ and $\Delta e(kT)$), five fuzzy sets have been used. They are Negative Large (NL), Negative Small (NS), Zero (Z), Positive Small (PS), and Positive Large (PL). The membership function of each fuzzy set is triangular with width 2 and overlapped with the two adjacent membership functions by 0.5. The outermost two membership functions (μ_{NL} and μ_{PL}) are saturated. The equations of the three input membership functions $\mu_{NS}, \mu_Z, \mu_{PS}$ are

$$\mu_i(x) = \begin{cases} 1 - \dfrac{|x-c|}{0.5} & \text{if } c-0.5 \le x \le c+0.5 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $i = NS, Z, PS$, $x = e(kT), \Delta e(kT)$ and c is the center of the input membership function. The equations of the outermost membership functions μ_{NL} and μ_{PL} are

$$\mu_{NL}(x) = \begin{cases} 1 & \text{if} \quad x < -1 \\ 1 - \dfrac{x+1}{0.5} & \text{if} \quad \text{-}1 \le x \le -0.5 \\ 0 & \text{if} \quad x > -0.5 \end{cases} \qquad \mu_{PL}(x) = \begin{cases} 0 & \text{if} \quad x < 0.5 \\ 1 - \dfrac{1-x}{0.5} & \text{if} \quad 0.5 \le x \le 1 \\ 1 & \text{if} \quad x > 1 \end{cases} \tag{5}$$

For the output variable $\Delta u(kT)$, the same fuzzy sets were used with the same membership functions. The only difference is that the outermost two membership functions (μ_{NL} and μ_{PL}) are not saturated. The equations of the output membership functions are

$$\mu_i(\Delta u(kT)) = \begin{cases} 1 - \dfrac{|\Delta u(kT) - c|}{0.5} & \text{if } c-0.5 \le x \le c+0.5 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $i = NL, NS, Z, PS, PL$ and c is the center of the output membership function.

The input and output membership functions are shown in **Fig. (2).**

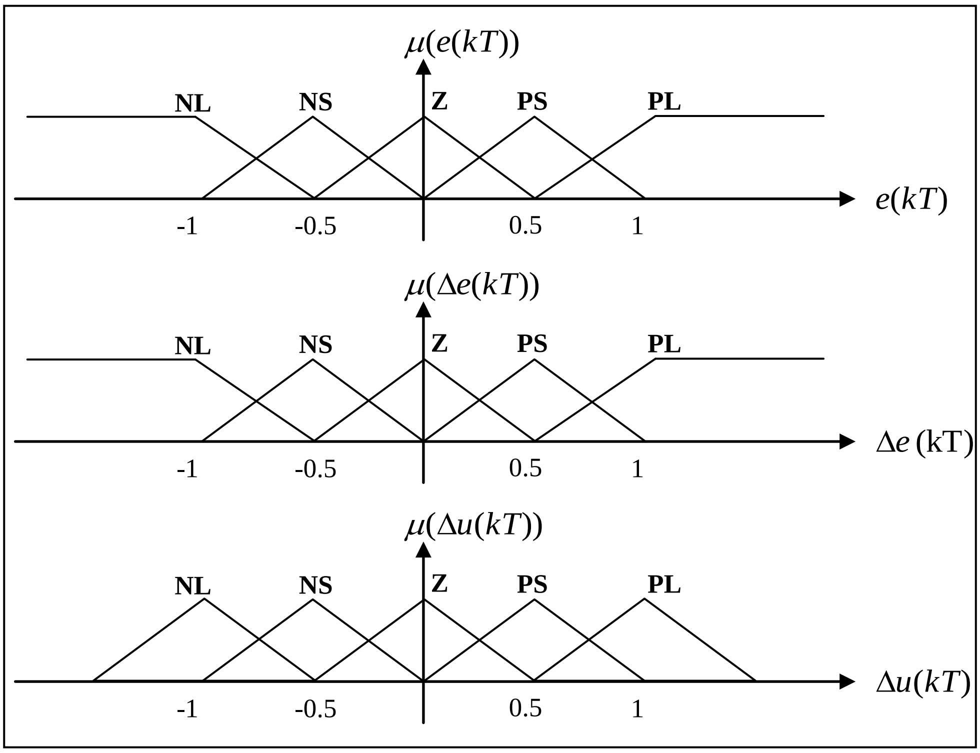

Fig (2): The input and output membership functions.

Scaling Factors

According to Eq. (3), the input scaling factors are K_P (for $\Delta e(kT)$) and K_I (for $e(kT)$). $\Delta u(kT)$ is multiplied by the output scaling factor $K_{\Delta u}$. Different values for the input scaling factors were used in the simulation to show their effect on the sensitivity of the controller with respect to the optimal choice of the operating areas of the input signals. Also different values for the output scaling factor were used to show their effect on the stability and oscillation of the system (Reznik 1997).

RULE BASE

Since the inputs of the PI-like fuzzy controller are the error $e(kT)$ and the change-of-error $\Delta e(kT)$, a standard PD-like fuzzy controller rule base was used. To form the rule base, each fuzzy set was given an index. The indices of the fuzzy sets NL, NS, Z, PS, PL are 0,1,2,3, and 4, respectively. The rule base was generated according to the following equation:

$$Rule(i,j) = \begin{cases} 0 & \text{if} \quad i+j-2 \le 0 \\ i+j-2 & \text{if} \quad 0 \le i+j-2 \le 4 \\ 4 & \text{if} \quad 4 \le i+j-2 \end{cases} \tag{7}$$

where i and j are the indices of the input membership functions and $Rule(i,j)$ is the index of the output membership function. The resultant rule base is shown in **Table (1)**.

Table (1): Rule base of a standard PD-like fuzzy controller.

$\Delta e(kT)$

		NL	NS	Z	PS	PL
$e(kT)$	NL	NL	NL	NL	NS	Z
	NS	NL	NL	NS	Z	PS
	Z	NL	NS	Z	PS	PL
	PS	NS	Z	PS	PL	PL
	PL	Z	PS	PL	PL	PL

Because of the absence of the derivative action in this fuzzy controller (since the overall action of this controller after integrating the output of the fuzzy system is a PI-like fuzzy controller), an overshoot (or undershoot) may occur at some operating points. To solve this problem, the rule base can be modified so that when the output of the plant moves towards the set point and becomes near it, a reverse action is given for the change-of-control output. This is done by replacing the rule (IF $e(kT)$ is PS and $\Delta e(kT)$ is NS THEN $\Delta u(kT)$ is Z) with the rule (IF $e(kT)$ is PS and $\Delta e(kT)$ is NS THEN $\Delta u(kT)$ is NS) (to reduce overshoot) and replacing the rule (IF $e(kT)$ is NS and $\Delta e(kT)$ is PS THEN $\Delta u(kT)$ is Z) with the rule (IF $e(kT)$ is NS and $\Delta e(kT)$ is PS THEN $\Delta u(kT)$ is PS) (to reduce undershoot). The modified rule base is shown in **Table (2)**.

Table (2): The modified rule base.

$\Delta e(kT)$

		NL	NS	Z	PS	PL
$e(kT)$	NL	NL	NL	NL	NS	Z
	NS	NL	NL	NS	**PS**	PS
	Z	NL	NS	Z	PS	PL
	PS	NS	**NS**	PS	PL	PL
	PL	Z	PS	PL	PL	PL

INFERENCE ENGINE

The inference step is taken by computing, for each rule, the implied fuzzy set. This is done by clipping the membership function of the consequent part of each rule according to the following equation:

$$\mu_{\text{implied}\,(i,j)}(\Delta u(kT)) = \mu_{\text{premice}\,(i,j)}(e(kT),\Delta e(kT)) * \mu_{\text{consequent}(i,j)}(\Delta u(kT)) \qquad (8)$$

where $\mu_{\text{implied}\,(i,j)}(\Delta u(kT))$ is the membership value for the implied fuzzy set of rule (i,j), $\mu_{\text{premice}\,(i,j)}(e(kT),\Delta e(kT))$ is the membership value for the premise of rule (i,j) which represents the certainty that rule (i,j) holds for the given inputs, and $\mu_{\text{consequent}(i,j)}(\Delta u(kT))$ is the membership function for the consequent part of rule (i,j). By using the minimum operation as a t-norm for the intersection operation, $\mu_{\text{consequent}(i,j)}(\Delta u(kT))$ is clipped to produce $\mu_{\text{implied}\,(i,j)}(\Delta u(kT))$.

DEFUZZIFICATION

Since the inference step is done by computing the implied fuzzy set for each rule, center of gravity defuzzification technique was used to compute the final controller output $\Delta u(kT)$ (Passino and Yurkovich 1998).

$$\Delta u(kT) = \frac{\sum_{i=0}^{4}\sum_{j=0}^{4} center\,(Rule(i,j)) \times Area(\mu_{\text{implied}\,(i,j)}(\Delta u(kT)))}{\sum_{i=0}^{4}\sum_{j=0}^{4} Area(\mu_{\text{implied}\,(i,j)}(\Delta u(kT)))} \tag{9}$$

TWO-TANK LEVEL CONTROL SYSTEM

The two-tank level control system is shown in **Fig. (3)**. The fuzzy logic controller has to keep a constant level in the second tank. The system is nonlinear with a dead time, so it is not easy to design a conventional controller. Let the levels in tank 1 and tank 2 be h_1 and h_2, respectively, the cross-sectional areas be D_1 and D_2, and the cross-sectional areas of each pipe be A_1 and A_2. L corresponds to the dead time. Then the plant model can be described with the following equations (Reznik 1997):

$$\begin{aligned} \dot{h}_1(t) &= \frac{q_1(t-L) - q_2(t)}{D_1} \\ q_2(t) &= A_1\sqrt{2gh_1(t)} \\ \dot{h}_2(t) &= \frac{q_2(t) - q_3(t)}{D_2} \\ q_3(t) &= A_2\sqrt{2gh_2(t)} \end{aligned} \tag{10}$$

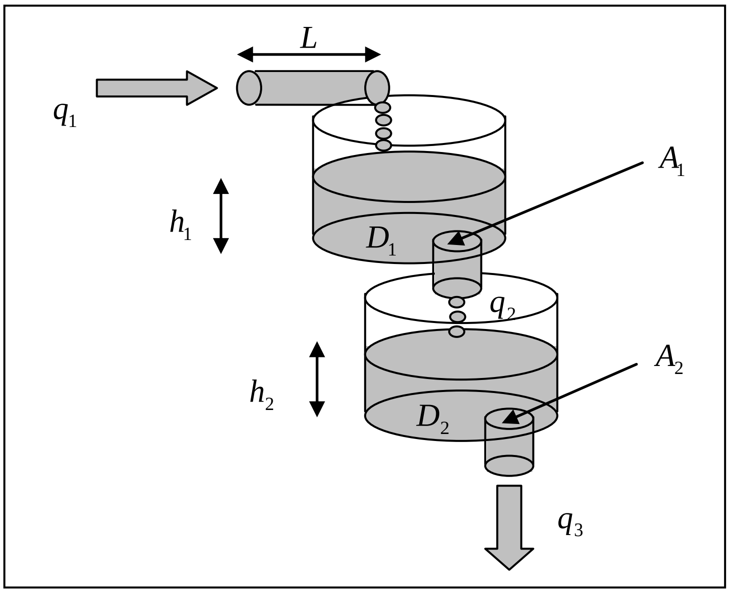

Fig (3): The two-tank level control system.

SIMULATION RESULTS

The simulation of the two-tank system needs to find the corresponding discrete-time form of Eqs (10). Euler method was used to approximate the differential equations in Eqs. (10) (i.e., the first and third equations) (Astrom and Wittenmark 1997) and (Severance 2001). The result is the following difference equations:

$$\begin{aligned}
&h_1((k+1)h) = h_1(kh) + h \times \frac{q_1(kh-L) - q_2(kh)}{D_1} \\
&q_2(kh) = A_1\sqrt{2gh_1(kh)} \\
&h_2((k+1)h) = h_2(kh) + h \times \frac{q_2(kh) - q_3(kh)}{D_2} \\
&q_3(kh) = A_2\sqrt{2gh_2(kh)}
\end{aligned} \qquad (11)$$

where h is the integration step size. The values of the parameters in Eqs (11) were taken as follows:

$L = 0.4\,\text{sec}$, $g = 9.8\,{}^{m}\!/_{\text{sec}^2}$, $A_1 = A_2 = 0.1\,m^2$, $D_1 = D_2 = 1\,m^2$, $h = 0.2\,\text{sec}$.

Initially, the levels in the two tanks were assumed to be 0 meters, i.e. , $h_1(0) = h_2(0) = 0\,m$.

For the controller to simulate the way that a digital control system would be implemented on a computer in the laboratory, the value of the sampling interval T must be an integer multiple of the integration step size h. In this simulation, $T = h = 0.2\,\text{sec}$.

Two Step Changes

The step response of the system has been investigated by applying two step changes to the reference input. The value of the reference input is

$$r = \begin{cases} 2 & \text{when} \quad 0 \le kT < 100\,\text{sec} \\ 4 & \text{when} \quad 100 \le kT \le 200\,\text{sec} \end{cases}$$

Table (1) was used for the rule base of the controller. **Fig. (4)** through **Fig. (10**) show the response of the system for different values of the scaling factors. The effect of the scaling factors on the overall response of the system is summarized in **Table (3)**.

Table (3): The effect of the scaling factors on the overall response of the system.

	Increasing K_I	Increasing K_P	Increasing $K_{\Delta u}$
Overshoot/Oscillation	Increases	Decreases	Increases
Speed of response	Increases	Decreases	Increases

Large Step Change and Overshoot

Fig (11) shows the step response of the system for a step change of 5 m. As shown, the response has an overshoot of about 5%. To reduce (or even eliminate) this overshoot, the modified rule base given by **Table (2)** was used. **Fig. (12)** shows the new step response. In both cases, the values of the scaling factors were: K_I=0.01, K_P=0.4, $K_{\Delta u}$=1.

Effect of Dead Time Parameter *L* on the Response of the System

Processes with large dead times present a special challenge for a controller—**any** controller (http://www.expertune.com/artdt.html 2008). Dead time is the delay from when a controller output signal is issued until when the measured process variable first begins to respond. The presence of dead time is never a good thing in a control loop. For any process, as dead time becomes larger, the control challenge becomes greater and tight performance becomes more difficult to achieve (http://www.controlguru.com/wp/p51.html 2008).

Fig. (13) shows the step response of the system for a step change of 5 m for different values of the dead time parameter L. These values are measured in seconds. This figure shows that increasing the dead time drives the system towards instability.

In each figure, the dotted line represents the reference input signal.

CONCLUSIONS

The simulation results showed that a PI-like fuzzy controller can stabilize the two-tank level control system which has a dead time with zero steady-state error. They show the effect of changing the input scaling factors on the sensitivity of the system and the effect of changing the output scaling factor on the stability and oscillation of the system. The overshoot and oscillation of the system were reduced or eliminated by a suitable change in the rule base that led to a change in the control surface so that the controller would become more reactive in the neighborhood of the set-point. Also the effect of the dead time on the response of the system was shown so that large dead time can make the system unstable.

REFERENCES

Abonyi J., Nagy L., and Szeifert F., “Adaptive Sugeno Fuzzy Control: A Case Study”, http://citeseer.ist.psu.edu/448436.html, 2005.

Astrom Karl J. and Bjorn Wittenmark, "Computer-Controlled Systems", Prentice-Hall, 1997.

Babuska R., Sousa J., and Verbruggen H. B., “Model-Based Design of Fuzzy Control Systems”. Proceedings of the EUFIT’95 837-841, 1995.

Driankov D., Hellendoorn H., and Reinfrank M., “An Introduction to Fuzzy Control”,Springer-Verlag,1996.

Fei J. and Isik C., “Adaptive Fuzzy Control Via Modification of Linguistic Variables”, IEEE International Conference on Fuzzy Systems, New York, USA, pp. 399-406, 1992.

Ghanayem O., and Reznik L., “A Universal Adaptation Procedure for Fuzzy Controller Design With The Application to Power System Stability Control”, Proceedings of the 35th Conference on Decision and Control, Kobe, Japan, 1996.

Isomursu P. and Rauma T, “A Self Tuning Fuzzy Logic Controller for Temperature Control of Superheated Steam”, Proceedings of the Third IEEE Conference on Fuzzy Systems, IEEE Word Congress on Computational Intelligence, New York, USA, vol 3, pp. 1560-1563, 1994.

Karasakal O., Yesil E., Guzelkaya M., Eksin I., " Implementation of a New Self-Tuning Fuzzy PID Controller on PLC, Turk J Elec Engin, VOL.13, NO.2, 2005.

Passino K. M. and Yurkovich S., “Fuzzy Control”, Addison Wesley, 1998.

Pave L. and Chelaru M., “Neural Fuzzy Architecture for Adaptive Control”, IEEE International Conference on Fuzzy Systems, pp. 1115-1122, New York, USA, 1992.

Reznik L., “Fuzzy Controllers”, Newnes, 1997.

Severance Frank L., "System Modelling and Simulation", John Wiley and Sons, Ltd, 2001.

Sutton R. and Jess I. M., "Real-Time Application of A Self Organizing Autopilot to Warship Taw Control", IEEE International Conference in Control'91, pp. 827-832, Edinburgh, UK, March 1991.

Ying H., "Fuzzy Control And Modeling", Institute of Electrical and Electronics Engineers, Inc, New York, 2000.

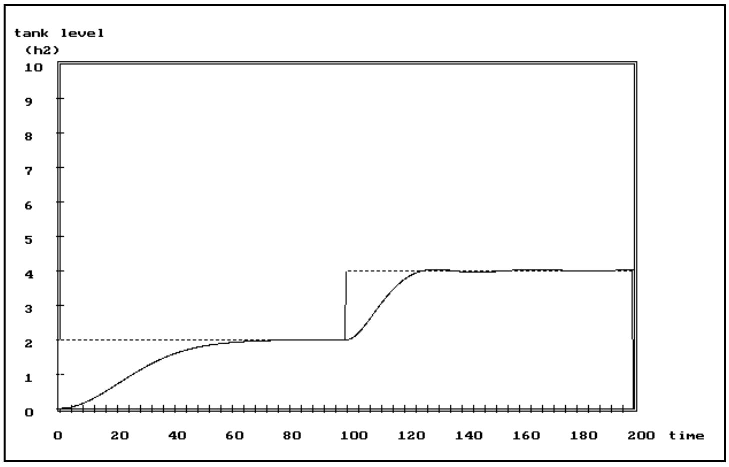

Fig (4): Two step changes response with K_I =0.01, K_P =0.55, $K_{\Delta u}$ =1.

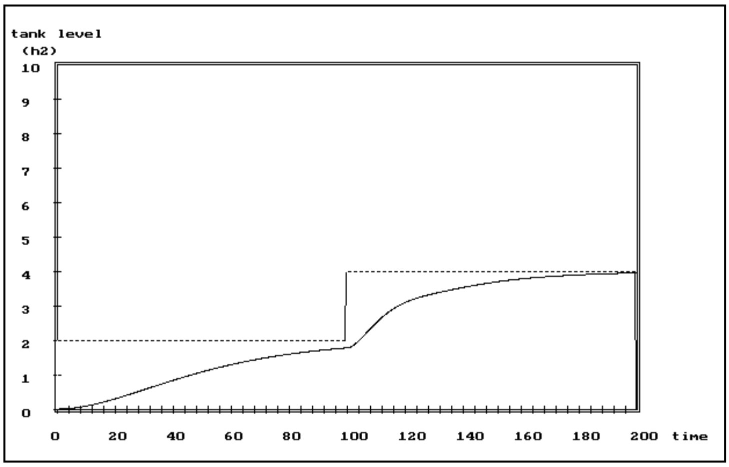

Fig (5): Two step changes response with K_I =0.005, K_P =0.55, $K_{\Delta u}$ =1.

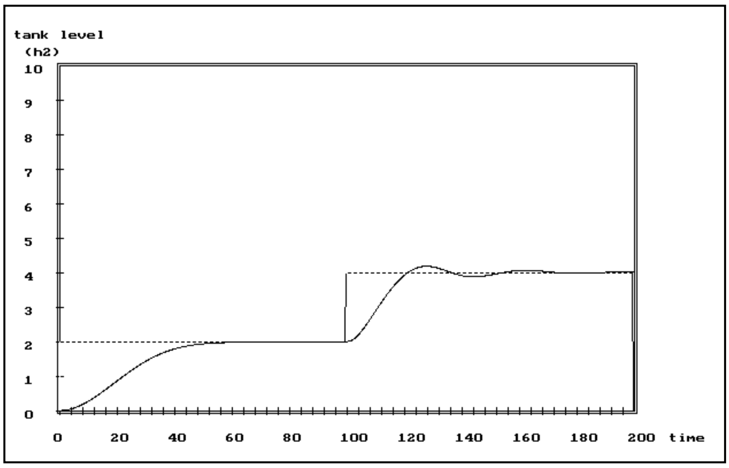

Fig (6): Two step changes response with K_I =0.012, K_P =0.55, $K_{\Delta u}$ =1.

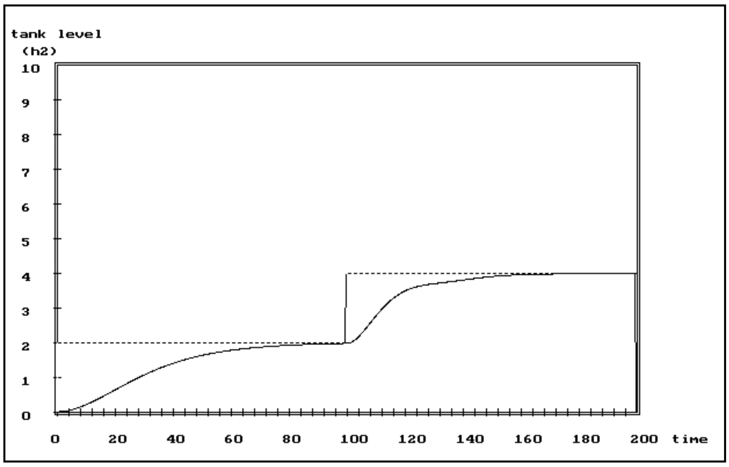

Fig (7): Two step changes response with K_I =0.01, K_P =0.8, $K_{\Delta u}$ =1.

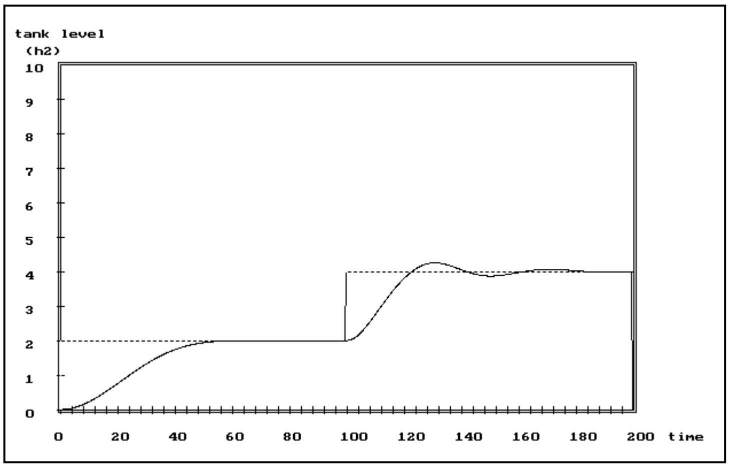

Fig (8): Two step changes response with K_I =0.01, K_P =0.4, $K_{\Delta u}$ =1.

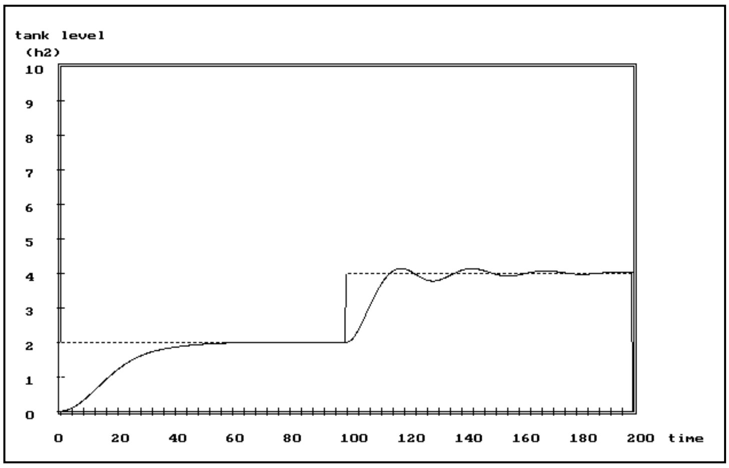

Fig (9): Two step changes response with K_I =0.01, K_P =0.55, $K_{\Delta u}$ =2.

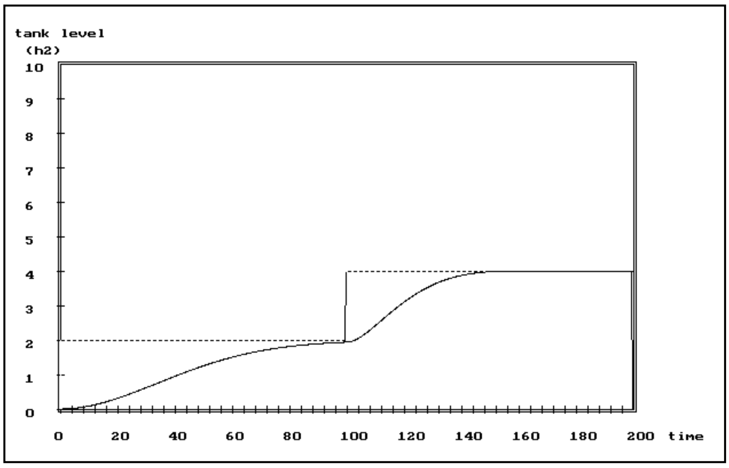

Fig (10): Two step changes response with K_I =0.01, K_P =0.55, $K_{\Delta u}$ =0.5.

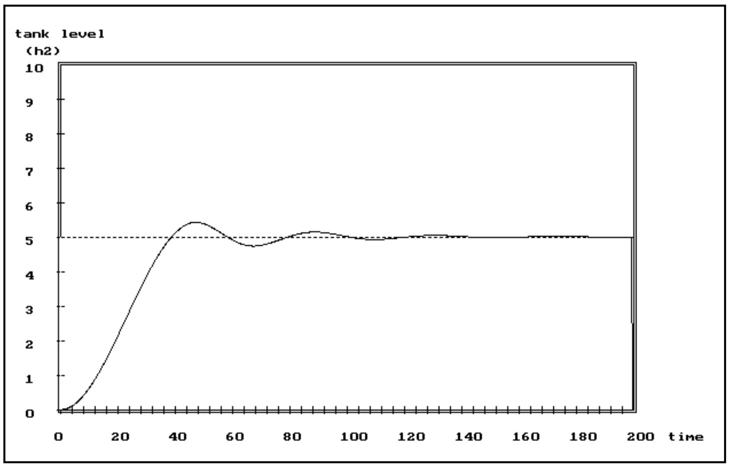

Fig (11): Large step change response using **Table (1)** for the rule base.

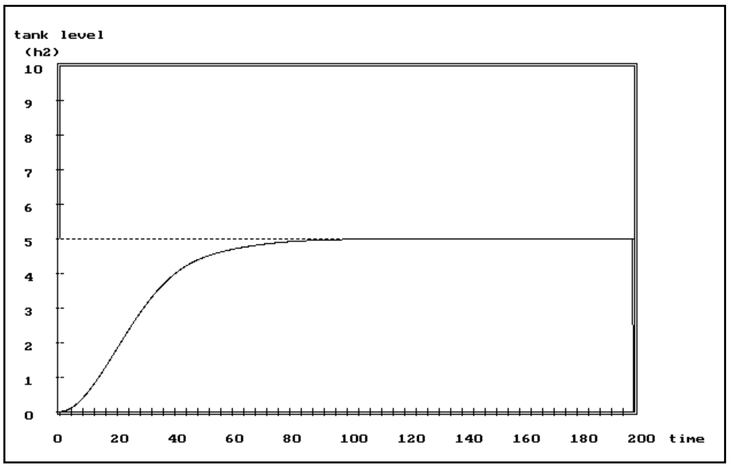

Fig (12): Large step change response using **Table (2)** for the rule base.

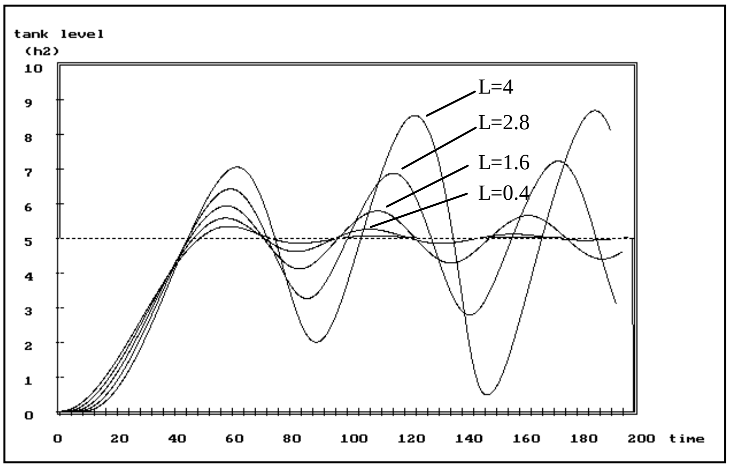

Fig (13): Effect of dead time parameter *L* on the response of the system.