

Simulated Design of Water Level Control System

Bourdillon. O. Omijeh¹ M. Ehikhamenle¹ Elechi Promise²

1.Department of Electronic and Computer Engineering, University of Port Harcourt

2.Department of Electrical and Computer Engineering, Rivers State University of Science & Technology, Port Harcourt, Rivers State, Nigeria

Abstract:

In this paper, the modelling and simulation of a water tank level controller using fuzzy logic approach has been achieved. This project is aimed at the design and simulation of a fuzzy logic based controller that will provide a stabilized output response. In order to successfully achieve this project, emphasis was made on two areas; the foundational knowledge of fuzzy logic and the fuzzy inference system, and the definition of the tank system model and its parameters. The design was implemented using Fuzzy Logic Toolbox package and SIMULINK environment which can be found in MATLAB software. For the purpose of analysis, the controller was simulated using a variety of rules in order to test the effect of the rules on the fuzzy logic controller. Results will show that fuzzy logic can realize faster results, superior features, and better end product performance with respect to overshoots, oscillations and response time.

Keywords: Fuzzy logic, Control System Design, Matlab, Simulink

I. INTRODUCTION

Water level control systems are used to control the level and flow of water circulation systems for a variety of processes (*Dhiman and Kumar, 2011*). These processes may include; water dam, petrochemical process plants, water treatment plants, pharmaceutical and food processing plants, overhead tanks etc (*Srinivas and Durga 2012*). The device that performs the controlling action is a valve which is attached to a controller whose main objective is to maintain a desired set point level as well as accept a new one desired by the control engineer. Fuzzy logic is widely used in the control of many processes, including that of the control of the water level in a tank. the fuzzy logic controller applies human reasoning and experience in defining the behaviour of the controller.

Due to the complexity (nonlinearities and uncertainties) inherent in liquid (water) tank systems, level control using conventional control schemes (i.e PI, PD, PID) will not give efficient results. This is due to two reasons:

- Difficulty in designing a precise mathematical model of the control system
- Poor control performance due to the fact that parameters of conventional controllers are required to be tuned over a period of time to get a stabilized output.

This paper is aimed at the modelling and simulation of a water tank system based on the control of a *fuzzy logic expert system* to stabilize the water level.

II. RELATED WORKS

In the work published by *Abdullah (2008)* on "Water Level in Tank using Level Sensor and PID Controller", the writer proposed the control of the water level of a tank using a level sensor and a PID controller. He was able to tune the response of the controller by adjusting three multipliers; k_p , k_i and k_d , where k_p is the main drive in the control loop which reduces a large part of the overall error, k_i reduces the final error in a system and k_d helps remove the overshoot but has no effect on the final error. This form of control produced small overshoots in the response of the controller.

Chin (2008) modelled the control of a multiple tank system using a fuzzy logic controller. He used the lamella filtration system of Bukit Sebukar water treatment plant as a case study. Results showed a smooth and low error water level control system.

This work was achieved by studying the control system theory, fuzzy logic and the fuzzy inference system and the SIMULINK environment, a simulation workbench of MATLAB. These areas are discussed in detail in the preceding sections.

III. DESIGN APPROACH

The system consists of a tank, a valve, a fuzzy logic controller, the inputs (level, rate) and the outputs (scopes). The tank system is modelled using SIMULINK and consists of two pipes; one flowing in and the other flowing out denoted by *inflow* and *outflow* respectively. The fuzzy controller controls the valve that changes the diameter of the inflow pipe but the outflow rate depends on the diameter of the outflow pipe (which is constant) and the pressure (which depends on the level) of the water. Thus, by keeping the water level at the desired set-point, we ensure that the tank works properly. Fig 1.1 shows the design approach of the work

Designing the fuzzy logic controller to work with only the level as its input will produce an output (water level) that oscillates about the desired level, hence there's a need to add an additional input. This input will be the rate which is derived by differentiating the current level of the tank. Both the rate and level (which is the difference between the current level and the desired set-point) will serve as inputs to the fuzzy logic controller.

The rules used by the fuzzy logic controller will be written using the fuzzy logic toolbox. One of the great advantages of the Fuzzy Logic Toolbox is the ability to take fuzzy systems directly into SIMULINK and test them out in a simulation environment. These rules will be integrated into the SIMULINK model and adjusted to acquire the best results.

3.1 The Water Tank Level Control System

The system shown in Fig 3.1 consists of 16 blocks out of which 3 are subsystems; fuzzy logic controller with ruleviewer, valve and water tank.

The input of the system consists of a square wave that is generated using the signal generator block. This block is gotten from the 'sources' folder found in the simulink library browser. The signal is added to another signal coming from the *constant* block carrying a value of 1 by the use of a *sum* block. This makes the signal cycle about the 1 position. This signal acts as the desired *level* input and the error signal (desired level minus actual level) is fed to the multiplexer block having two inputs and one output.

The second input to the multiplexer is the rate of outflow derived by differentiating the output level and passing it through a *saturation* block which limits the input signal to the lower and upper saturation values which are -0.1 and 0.1 respectively. The multiplexer doesn't perform any mathematical operation on the two signals passing through it. It only overlaps the two signals and conveys them to the next block.

The output of the multiplexer is fed to the *fuzzy logic controller with ruleviewer* block. The fuzzy rules written with the FIS editor are loaded into the *fuzzy logic controller with ruleviewer* by double-clicking on the block and writing the name of the *.fis* file which the rules were saved with. This file must be saved in the workspace in order for the block to make use of it. Saving the file to the workspace is done by opening the FIS editor containing the rules, pressing 'ctrl+T' and saving the work to the workspace. Another interesting feature of the *fuzzy logic controller with ruleviewer* block is that it displays the whole fuzzy inference process when the simulation starts running.

The signal leaves the fuzzy controller block and enters the *valve* subsystem. The second input of the valve subsystem consists of a constant block with a value of 0.5, serving as the maximum inflow of the tank. The valve multiplies the control signal by the maximum inflow of the tank thus performing the control action.

The manipulated variable (inflow of the tank) enters the *tank* block and leaves as the controlled level, the outflow and the overflow signal. The controlled level signal is split into two, one signal goes to the comparison scope while the other is fed back to the level input of multiplexer. The signal that is fed back is also differentiated and set as the second input (rate) of the multiplexer. The *outflow* signal goes to the scope which measures the flow rate while the overflow signal goes to an alarm (scope) which indicates when the level has been exceeded.

The S-Function block provides access to S-functions from a block diagram. This block serves as a visual display of the tank's fluctuation in level as the simulation is running. The S-function named as the S-function name parameter can be a Level-1 M-file or a Level-1 or Level-2 C MEX-file S-function. The S-Function block allows additional parameters to be passed directly to the named S-function. The function parameters can be specified as MATLAB expressions or as variables separated by commas.

3.2 MODELLING THE WATER TANK SUBSYSTEM

The design of this project was based on an open tank system that has a cross-sectional area of *area*. Water is pumped into the tank at the top at rate of flow of *q* in cubic metres per second (m^3/s). Water is flowing out of the tank through a hole in the bottom of the tank of area *outletArea* as seen in Fig 3.2. The rate of flow of water through the hole is according to the Bernoulli equation given by:

$$q_{out} = outletArea\sqrt{2gh} \dots\dots\dots(1)$$

where *h* is level of tank and *g* is the acceleration of gravity. Conservation of mass yields the equation:

$$area\frac{dh}{dt} = q_{in} - q_{out} = q_{in} - outletArea\sqrt{2gh} \dots\dots\dots(2)$$

Therefore resolving the differential equation in the function of *h(t)* we have:

$$h(t) = h(0) + \int_0^t \frac{1}{area} (q_{in}(t') - q_{out}(t')) dt'$$

$$= h(0) + \int_0^t \frac{1}{area} (q_{in}(t') - outletArea \sqrt{2gh(t')}) dt' \dots\dots(3)$$

Where $area = 1m^2$, $outletArea = 0.05m^2$ and $g = 9.8ms^{-2}$. Converting the following mathematical model to a control block model and making a few adjustments to the system will result to the system depicted in **Fig 3.3**

This subsystem was created by selecting the 'Subsystem' block from the commonly used blocks folder found in the simulink library browser. The mask parameters were first assigned by right-clicking on the block and selecting 'mask subsystem' option. The following window popped-up.

The five parameters inputted include:

1. Height of the tank in m with variable name *ht*
2. Bottom area in m^2 with variable name *area*
3. Outpipe cross-section in m^2 with variable name *outletArea*
4. Overflow distance from the top in m with variable name *overflowLimit*
5. Initial Level height in m with variable name *initialHt*

The actual values of the mask parameters were then entered by double-clicking on the subsystem block. These values are:

1. $ht = 2$
2. $area = 1$
3. $outletArea = 0.05$
4. $overflowLimit = 0$
5. $initialHt = 0.5$

After all the parameters were set as seen in **Fig 3.4**, the subsystem was created .

The block labelled 'tank volume 1' is another subsystem that performs the function of a limited integrator. These parameters are:

1. Lower boundary = 0
2. Upper boundary = $Ht*area$
3. Initial condition = $initialHt*area$

The subsystem of the *tank volume* block is shown in **Fig 3.5** below.

The *Mux* block is a multiplexer that combines the input and the feedback signal without performing any mathematical operations on the two signals. The two signals pass through the *Fcn* block gotten from the user-defined blocks found in the simulink library browser with the following equation inputted in the expression dialog box.

$$u[2]*(((u[1]>lb)+(u[2]>=0))>0) * (((u[1]<ub)+(u[2]<=0))>0))$$

The signal then passes through the integrator whose initial condition is set from the mask parameters of the *tank volume 1* subsystem. After the signal is being integrated, it passes through the *saturation* block whose upper and lower limits are also set from the mask parameters.

When the water passes through the *tank volume 1* block, it goes through the *gain* block. This block multiplies the signal by a factor of $\frac{1}{area}$. At this point, what we have is the level of the water in the tank. This

can be checked by dimensional analysis. This level is tapped out and sent to out port 1 labelled 'water level'. The level is tapped out again and goes to relay whose switch-on and switch-off points are set as $ht - overflowLimit$ and $overflowLimit$ respectively. The relay serves as a switch that triggers an alarm when the level exceeds the overflow limit. For the purpose of this project, the overflow signal is sent to a scope and can be viewed while the simulation is running.

The signal moves to the $\sqrt{2gh}$ block which consists of a function block that is gotten from the user-defined functions folder found in the simulink library browser. This block together with the 'outletArea' block converts the level to the outflow volume in m^3 . This signal is split into two parts. one part is taken as feedback

to the water tank system while the other goes to out port 3 labelled 'outflow'.

3.4 CREATING THE FUZZY LOGIC CONTROLLER

Developing the Fuzzy logic Controller involves several stages. The first stage involves defining all the inputs and outputs of the fuzzy logic controller. This is done using the *FIS editor*. The second stage involves setting the membership function for each of the inputs and outputs. This is achieved by the use of the *Membership Function editor*. The final stage involves defining the rules that govern the fuzzy logic controller. These rules are set using the *Rule editor*. The aforementioned stages are discussed in detail in the preceding sections.

The FIS Editor

The fuzzy logic controller makes use of two inputs and one output. The first input is the level of the tank denoted as *level* while the second is the rate of change of the level denoted as *rate*. According to the Rules written in the *Rule Editor* the controller takes the action and governs the opening of the Valve which is the Output of the controller and is denoted by *valve*.

In order to start the FIS editor, 'fuzzy' is typed in the MATLAB command window and the enter button is pressed. The FIS editor pops up with the mamdani style of fuzzyfication set as default. The FIS editor opens with only one input and one output therefore in order to add a second input variable, the 'add variable' option is selected from the edit option in the toolbar. The input and output blocks are selected and their names are written starting from the inputs as shown in **Fig 3.7**.

The Membership Function Editor

The Membership Function Editor is the tool that lets you display and edit all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system. The second stage involves setting the membership functions of the two inputs and the output. The input and output sets are created as shown below.

(a) Level

Double-clicking on *level* block found on the FIS editor opens the Membership function window. The name of the input is assigned by typing the name in the name box as seen in **Fig 3.8**. The range along with the other parameters are set with accordance to **table 3.1**.

(b) Rate

The membership function window for the *rate* input parameter is gotten by double-clicking on the *rate* block found in the FIS editor. The parameters seen in **table 3.2** are set accordingly as seen in **Fig 3.9** below.

(c) Valve

The parameters are set using the same method described in the previous section. Using triangular membership function types for the output, the Range (and the Display Range) is set to (-1 1), to cover the output range. The *close fast* membership function will have the parameters (-1.0,-0.9,-0.8), the *close low* membership function will be (-0.6,-0.5,-0.4), the *no change* membership function will be (-0.1,0,0.1), the open slow membership function will be (0.2 ,0.3,0.4), the open fast membership function will be (0.8 0.9 1.0). The system should look like **Fig 3.10**.

3.4.3 The Rule Editor

Constructing rules using the graphical Rule Editor interface is fairly self-evident. Based on the descriptions of the input and output variables defined with the FIS Editor, the Rule Editor allows you to construct the rule statements automatically, by clicking and selecting one item in each input variable box, one item in each output box, and one connection item. Choosing none as one of the variable qualities will exclude that variable from a given rule. Doing so will give rise to the following rules.

1. if (level is ok) then (valve is no change) (1)
2. if (level is low) then (valve is open_fast) (1)
3. if (level is high) then (valve is closed_fast) (1)
4. if (level is ok) and (rate is positive) then (valve is close_slow) (1)
5. if (level is ok) and (rate is negative) then (valve is open_slow) (1)

IV. RESULTS AND DISCUSSION

4.1 Comparative Analysis

For the purpose of analysis, the results obtained from the fuzzy logic controller were compared with those obtained using three, four, five and six rules.

4.1.1 Results Obtained Using Three

The three rules that were chosen initially are:

if(level is low) then (valve is open_fast)
if(level is ok) then (valve is no_change)

if(level is high) then (valve is close_fast)

Fig 4.1 shows the plot of two overlapped signals; the desired level and the actual level of the water tank. The step function represents the desired level while the second waveform represents the control response of the controller. From the plot it can be observed that the use of only three rules pertaining to the level will not give desired results since the water level tends to oscillate about the desired position. Two oscillations were observed; the first had an overshoot of 0.4354 meters while the second had an overshoot of 0.2689 meters. The rise time was 6 seconds and settling 25.3 seconds.

4.1.2 Results Obtained Using Four Rules

Addition of a fourth rule:

if(level is ok) and (rate is negative) then (valve is open_slow)

changed the control response to that shown in Fig 4.2 above. The introduction of a fourth rule relating the level and rate to the valve percentage opening, produced an overshoot of 0.4354 meters, a rise time of 6 seconds and a settling time of 25.3 seconds. This result showed a reduced number of oscillations but was still not desirable due to the overshoot, hence a fifth rule was introduced.

4.1.3 Results Obtained Using Five Rules

Addition of a fifth rule:

if (level is ok) and (rate is positive) then (valve is close_slow)

changed the control response to that shown in Fig 4.3 above. The introduction of a fifth rule relating the level and rate to the valve percentage opening, produced no overshoot, a rise time of 2.6 seconds and a settling time of 11.9 seconds.

4.1.4 Results Obtained Using Six Rules

A sixth rule was added in order to check if the response of the fuzzy logic controller could be further optimized.

Addition of:

if (level is ok) and (rate is zero) the (valve is no_change)

showed no change in the response of the controller as shown in Fig 4.4, hence the number of rules that gave the best response is five.

Table 4.1 shows the performance characteristics of the different rules used. It can be observed that using three (3) and four (4) rules for the fuzzy controller, gave unwanted overshoots and an increased response time depicted by the rise time. These results may not be tolerated for sensitive processes requiring fast response. However, the use of five (5) rules produced no overshoots with a reduced rise time which represents an improved performance.

4.2 The Ruleviewer and Tank Demo

During simulation, the ruleviewer shows the process of the fuzzy inference system in real-time. It shows one calculation at a time in great detail and can be used to manually deduce the output of the fuzzy controller for every input value assigned to it. Fig 4.5 shows the percentage open of valve when the level is 0.0734 m and rate is -0.032 m/s.

The tank demo is an animation of the physical tank showing the desired and actual level of the tank. The rectangle to the left of the tank's exterior represents the desired level whereas the rectangle in the tank's interior shows the actual level of the water tank. This is depicted in the Fig 4.6 below.

Comparison of the data collected when using different number of rules shows that the best response of the fuzzy logic controller was achieved by using five rules. It can be observed that the use of five rules gave the best performance response with respect to overshoots, oscillations, rise and settling time. The results show that fuzzy logic adapts quickly to time delays and provides a stable

V. Conclusion

In this project, a unique fuzzy logic controller using five rules and a straight forward implementation was used to solve the problem of overshoots in level control, commonly encountered by conventional controllers. The result shows significant improvement in maintaining performance in terms of oscillations and overshoots produced.

REFERENCES

- Abdullah, A.B. (2008): "A Research Project on Water Level in Tank using Level Sensor and PID Controller", Pahang: University of Malaysia.
- Chin, G.C. (2008): "A Research Project on Design and Modelling of Multiple Tank Control for Fluid Circulation System Using Fuzzy Controller", Melaka: Technical University of Malaysia.
- Dharamniwas, Ahmad, A., Redhu, V. and Gupta, U. (2012): " Liquid Level Control by Using Fuzzy Logic Controller", IJAET, Vol. 4, Issue 1, pp. 537-549.
- Dhiman and R. Kumar, B. (2011): " Optimization of PID Controller for Liquid Level Tank System Using Intelligent Techniques", Canadian Journal on Electrical and Electronics Engineering, Vol. 2, p. 531.

Distefano, J.J., Stubberud, A.R. and Williams J.(1995), **Theory and Problems of Feedback and Control Systems**, 2nd edition, McGraw-Hill Publishers, USA, p. 156.

Jang, J.S. and Gulley N. (1997): **Matlab: Fuzzy Logic Toolbox**, 1st edition, The Mathworks Inc, USA, pp. 1-68.

Grundberg, S. (2007): *"A Report on Tank Level Control"*, Umea University.

Srinivas, P. and Durga, R (2012): *"Comparative Analysis of Conventional PID Controller and Fuzzy Controller with Various Defuzzification Methods in a Three Tank Level Control System"*, International Journal of Information Technology, Control and Automation, Vol. 2, pp. 75-76.

Wu, D. , Karray, F. and Sond, I. (2005): *" Water Level Control by Fuzzy Logic and Neural Networks"*, IEEE Conference on Control Applications, pp. 1- 6.

Mathworks 2007, MATLAB 7.5.0. (R2007b),

AUTHOR'S BIOGRAPHY

Bourdillon .O. Omijeh holds a B.Eng degree in Electrical/Electronic Engineering, M.Eng and Ph.D in Electronics/Telecommunications Engineering from the University of Port Harcourt & Ambrose Alli University (A.A.U) Ekpoma, respectively. His research areas include: Artificial Intelligence, Robotics, Embedded Systems Design, Modeling and Simulation of Dynamic systems, Intelligent Metering Systems, Automated Controls, Telecommunications and ICT. He has over thirty (30) technical papers & publications in reputable National & International peer reviewed Journals. He has authored some Electrical/Electronic Engineering Text books; and also, has developed over ten(10) engineering application Software. He is a member, Institute of Electronics and Electrical Engineers (MIEEE), Corporate Member, Nigeria Society of Engineers; and also, a registered practicing Engineer with COREN. He is currently a Senior Lecturer & pioneer HOD, Department of Electronic and Computer Engineering, University of Port Harcourt, Nigeria; and also, a consultant to companies & Institutions. He is happily married with Children. E-mail: bourdillon.omijeh@uniport.edu.ng



Ehikhamenle M. graduated with BEng (Second class upper honors) Electrical/Electronics Engineering from Ambrose Alli University Ekpoma in 2007. He obtained his Master degree in Electrical/Electronics Engineering from University of Benin (UNIBEN), specializing in Electronics and Telecommunications in 2010. He was Head of Department Essential services, Works and Services Directorate Ambrose Alli University Ekpoma in 2012. He is presently a lecturer in Electronic and Computer Engineering Department University of Port-Harcourt (UNIPORT). He has taught courses in electronics and telecommunications Engineering. His research interest include: Electronic, telecommunication, power electronic CAD, ICT and Control Systems



Elechi Promise received his B.Eng and M.Eng degrees in Electrical/Electronic Engineering from University of Port Harcourt, Choba, Rivers State, Nigeria in 2006 and 2011 respectively. He is a member of the Nigerian Society of Engineers (NSE) and Nigerian Institution of Electrical/Electronic Engineers (NIEEE). His current research interest is on radio propagation for Mobile communications, GSM technology, Microwave propagation, Signal analysis and ICT. He is currently a Lecturer in the department of Electrical and Computer Engineering, Rivers State University of Science and Technology, Port Harcourt, Nigeria.



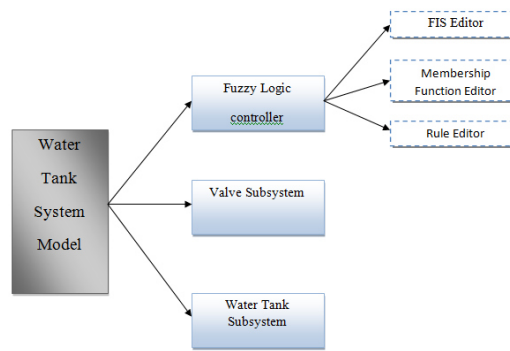


Fig 1.1: Block Diagram Showing Design Approach

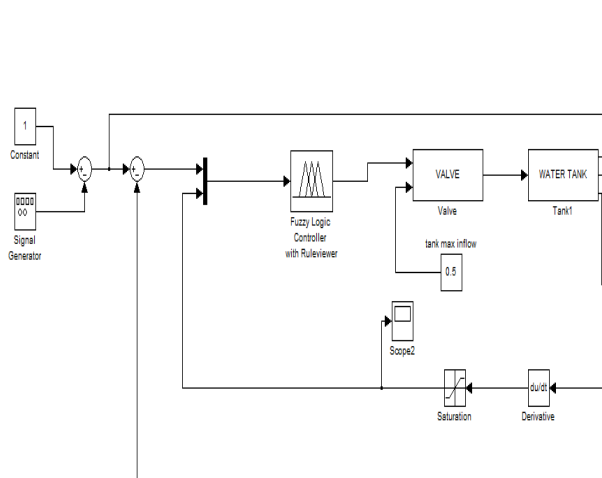


Fig 3.1: The Complete Water Tank Level Control System

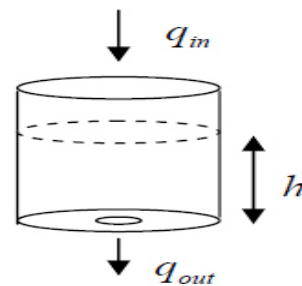


Fig 3.2: The Water Tank System (source: Grundberg (2007))

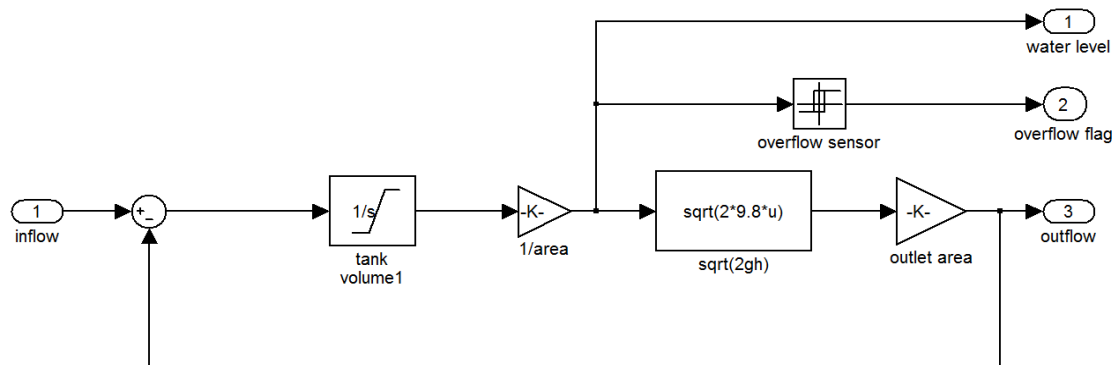


Fig 3.3: The Water Tank Subsystem (source: MATLAB 7.5.0)

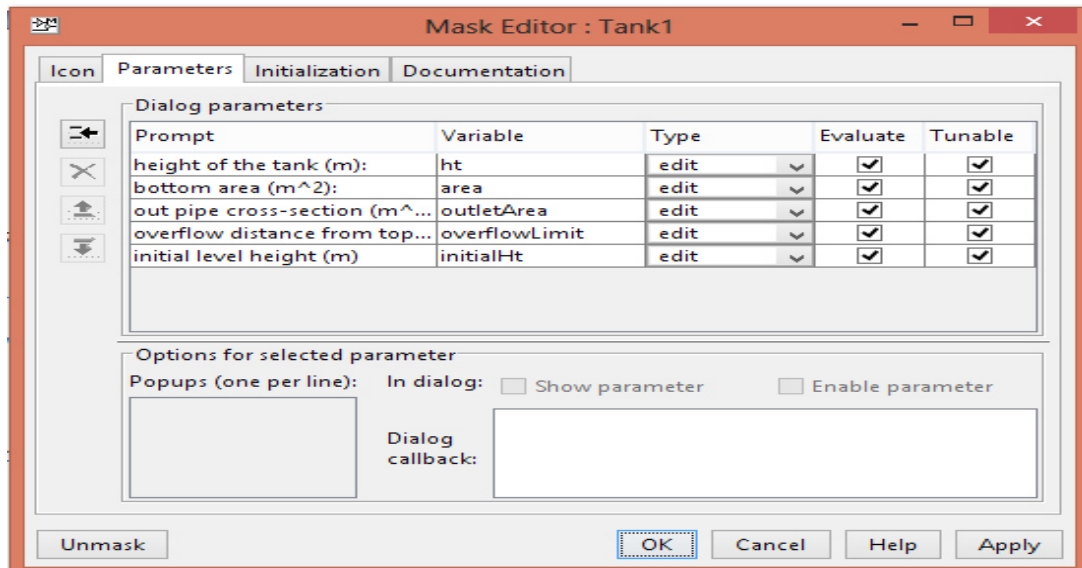


Fig 3.4: The Mask Editor (source: MATLAB 7.5.0)

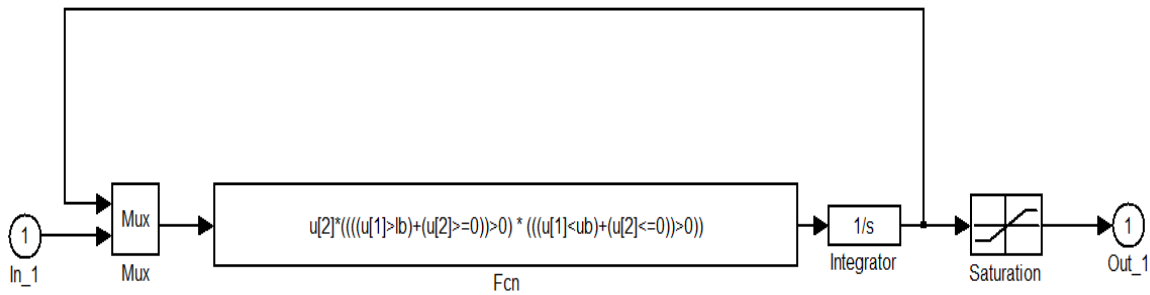


Fig 3.5 The tank Volume subsystem (source: MATLAB 7.5.0)

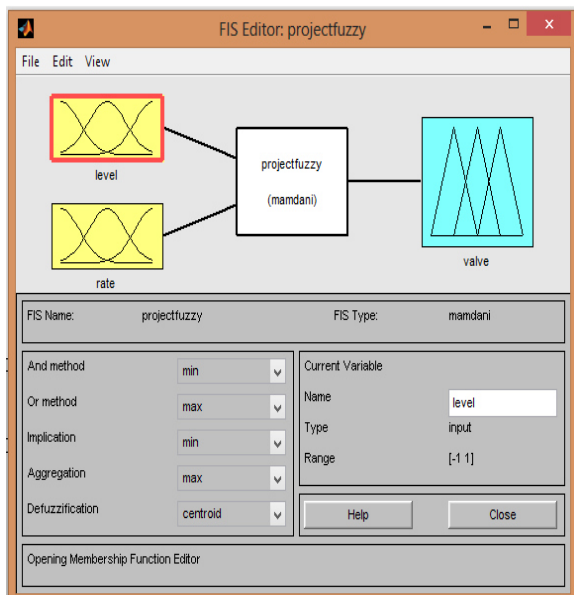


Fig 3.7: The Projectfuzzy FIS Editor

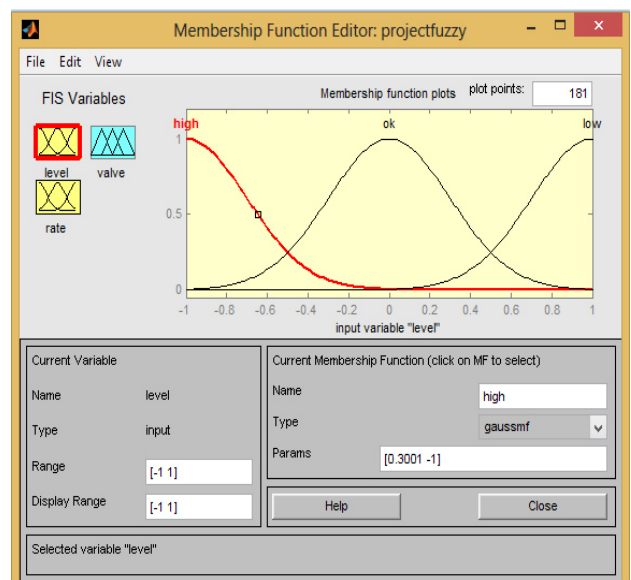


Fig 3.8 Level MF

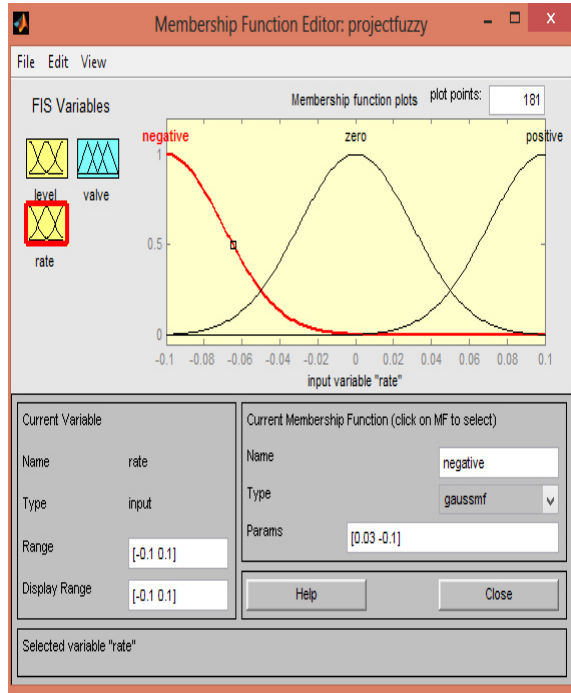


Fig 3.9 Rate MF

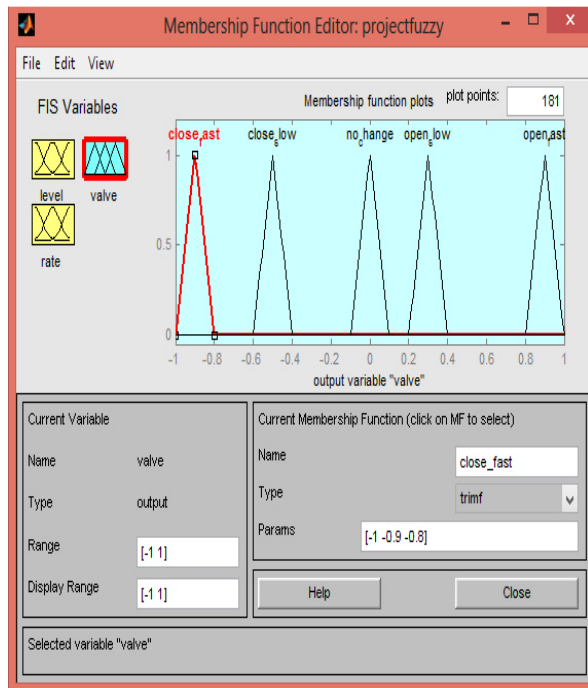


Fig 3.10 Valve MF

Table 3.1 Level Input Parameters

Fuzzy Variable	MF used Range (-1,1)	Crisp Input Used
high	Gaussian	(0.3,-1)
ok	Gaussian	(0.3,0)
low	Gaussian	(0.3,1)

Table 3.2 Rate Input Parameters

Fuzzy Variable	MF used Range (-1,1)	Crisp Input Used
negative	Gaussian	(0.03,-0.1)
zero	Gaussian	(0.03,0)
positive	Gaussian	(0.03,0.1)

Table 3.3 Valve Output Parameters

Fuzzy Variable	MF used (Range: -1 to 1)	Crisp Input Used
close_fast	triangular	(-1.0,-0.9,-0.8)
close_slow	triangular	(-0.6,-0.5,-0.4)
no_change	triangular	(-0.1,0,0.1)
open_slow	triangular	(0.2,0.3,0.4)
open_fast	triangular	(0.8,0.9,1)

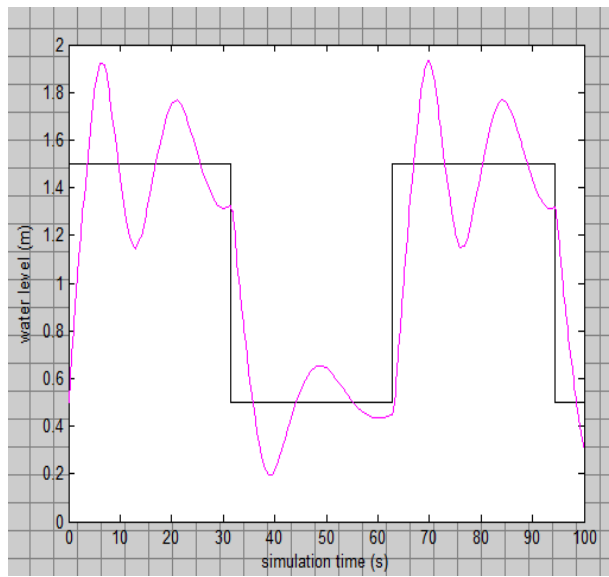


Fig 4.1: Control Response of Fuzzy Logic Controller Using 3 Rules

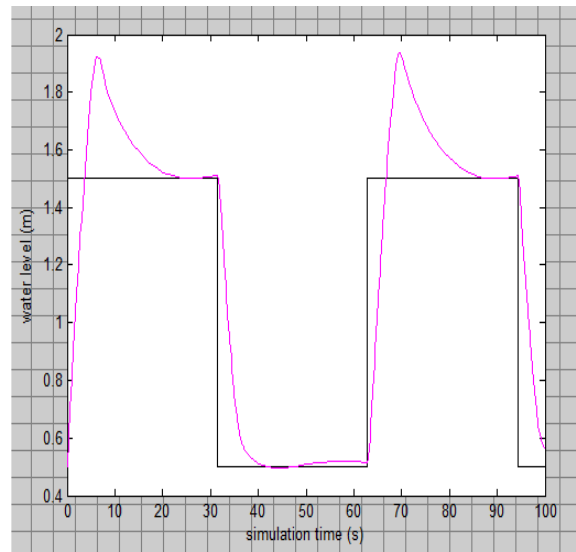


Fig 4.2: Control Response of Fuzzy Logic Controller Using 4 Rules

seconds.

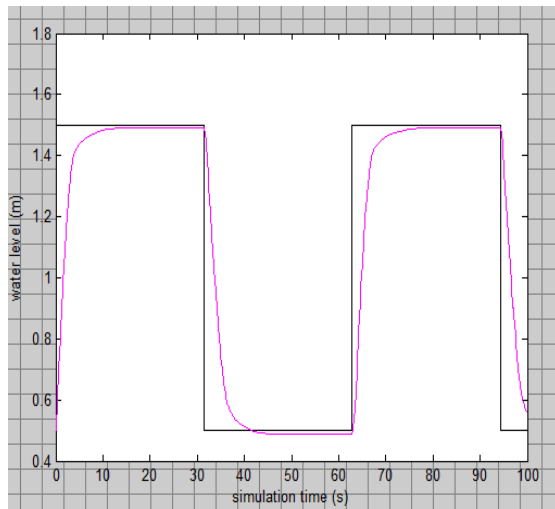


Fig 4.3: Control Response of a Fuzzy Logic Controller Using Five Rules

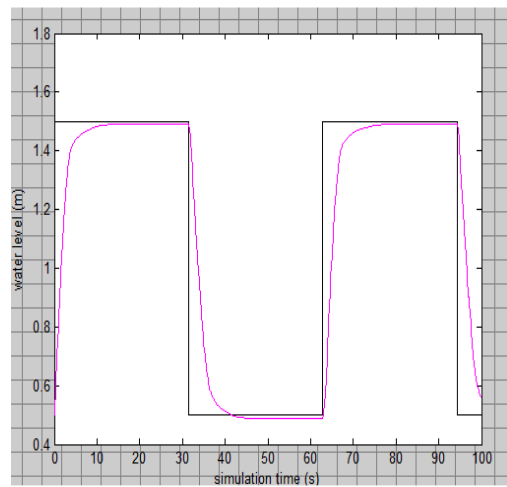


Fig 4.4: Control Response of a Fuzzy Logic Controller Using Six Rules

Table 4.1: Simulation Results

No. of rules used	Rise Time (s)	Settling Time (s)	Overshoot (m) (max)
3	6	25.3	0.4354
4	6	25.3	0.4354
5	2.6	11.9	Not present
6	2.6	11.9	Not present

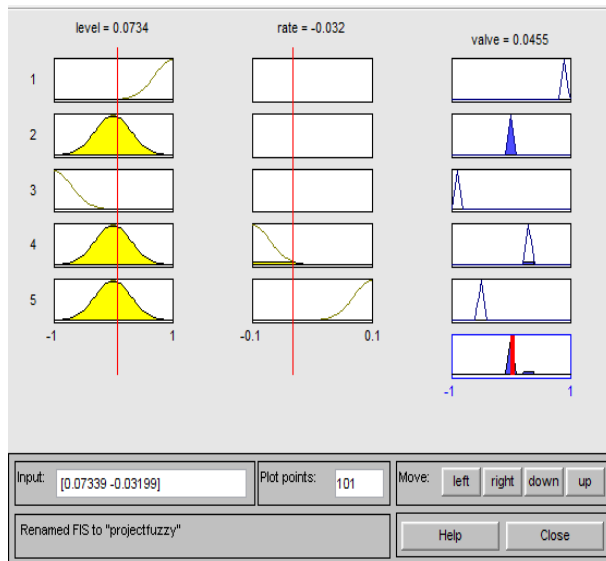


Fig 4.5: Response of Fuzzy Logic Controller Using Ruleviewer

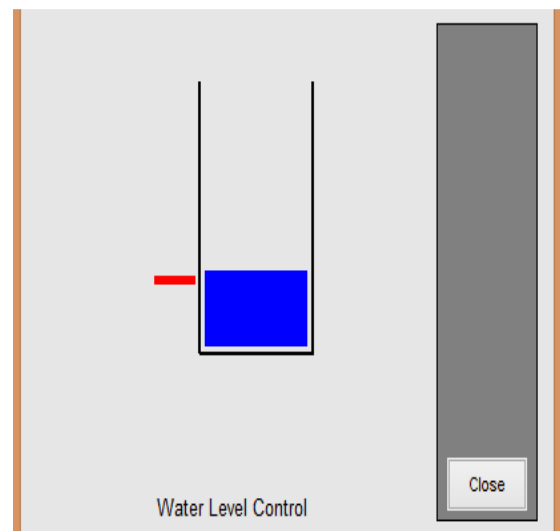


Fig 4.5: Tank Demo

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:

<http://www.iiste.org>

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: <http://www.iiste.org/journals/> All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: <http://www.iiste.org/book/>

Academic conference: <http://www.iiste.org/conference/upcoming-conferences-call-for-paper/>

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library, NewJour, Google Scholar

