

IMPLEMENTATION OF FUZZY LOGIC BASED TEMPERATURE-CONTROLLED HEAT EXCHANGER

By

Thomas A. Nwodoh

*Department of Electronic Engineering
University of Nigeria, Nsukka
and*

Ihekeremma A. Ejimofor

*Department of Computer Science
Nnamdi Azikiwe University, Awka*

ABSTRACT

This paper discusses the performance analysis of a heat exchanger using simulations for an Adaptive Network based Fuzzy Inference System toolbox developed with MATLAB. The plant transfer function is derived based on process reaction curve obtained from a heat exchanger pilot plant and then the model is used to represent the heat exchanger throughout the analysis. This research then compares the control performance of PID (Proportional Integral and Derivative) and Fuzzy logic controllers. Conclusions are made based on these control performances. The results show that the control performance for a Fuzzy controller is quite similar to PID controller but comparatively gives a better response.

Key Words: Heat exchanger, Fuzzy logic controller, Proportional Integral and Derivative Control.

INTRODUCTION

A Heat Exchanger is a device for efficient heat transfer from one medium to another, whether the media are separated by a solid wall so that they never mix, or the media are in direct contact. The heat exchanger relates generally to controlling temperature of a fluid. This research uses fuzzy logic to control the rate at which the output (leaving water) temperature in the exchanger is to a predefined set point.

Fuzzy logic has become one of the most successful of today's technologies for developing sophisticated control systems. The reason is very simple: Fuzzy logic addresses applications perfectly as it

resembles human decision making with an ability to generate precise solutions from certain or approximate information. It fills an important gap in engineering design method left vacant by purely mathematical approaches (e.g. linear control design), and purely logic-based approaches (e.g. Expert systems) in system design. While other approaches require accurate equations to model real-world behaviors, fuzzy design can accommodate the ambiguities of real-world human language and logic. It provides both an intuitive method for describing systems in human terms and automates the conversion of those system specifications into effective models.

The objective of this research is to model the problems – overshoot, undershoot and non-linearity and proffer solutions associated with Heat Exchanger temperature control via conventional Proportional Integral and Derivative (PID) and maintain the temperature at the set point.

2. LITERATURE REVIEW

Noie [1] investigated the thermal performance of a heat exchanger consisting of 90 thermosyphons arranged in six rows in a test rig. The variable parameters which were being altered were the air velocity and the inlet temperature to the evaporator section. A computer program was developed to analyze the thermosyphon heat exchanger using the E - NTU method. In order to verify its accuracy and conformity, the experimental results were compared to those predicted by the simulation program. The temperature across the evaporator section was varied in the range of 100-250 °C while the inlet temperature to condenser section was nearly constant 25°C. The experimental results showed the minimum effectiveness of the thermosyphon took place at $C_h = C_c$. The conclusion is that equal value of air face velocities in evaporator and the condenser sections should be avoided.

Niu et al. [2] studied a heat exchanger system combining chilled ceiling with desiccant cooling for maintaining the indoor air humidity within a comfort zone and to reduce the risk of water condensation on chilled panels. The results reveal that chilled ceiling combined with desiccant cooling might conserve up to 44% of primary energy use compared to a conventional constant volume all-air system.

Zhang et al. [3] conducted a study on a

thermodynamic model built with an air moisture removal system incorporated a membrane-based total heat exchanger to estimate the energy use annually. The outcome suggested that the independent air moisture removal could save 33% of primary energy.

3. ANALYSIS OF PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER

One of the most powerful but complex controller mode of operations combines the proportional, integral, and derivative modes. The analytic expression for a standard PID controller is:

$$U = K_p + K_D \frac{de}{dt} + k_I \int e(t)dt + P_0$$

where,

U = The controller output (0-100%)

K_p = Proportional gain constant

K_D = Derivative gain constant

K_I = Integral gain constant

e_t = Error i.e. PV – SP or SP – PV, depending on whether it is a direct acting or reverse acting controller

P_0 = bias or feed forward.

The proportional constant K_p expresses a linear relationship between the controller output and the error. Thus, over some range of errors above set point, each value of error has a unique value of controller output in one-to-one correspondence.

The integral constant K_I relates the changes in the rate of controller output with changes in error. Thus, a large value of K_I means that a small error produces a large rate of change of output and vice versa. For example, in case of the heat exchanger, the controller output value initially begins to change very rapidly, but as the valve opens,

the error decreases and shows the valve-opening rate.

The derivative constant K_D relates the extent of the controller output to the rate at which the error is changing and not on the value of the error. The derivative action is also known as rate of anticipating control.

An important characteristic of the proportional Integral and Derivative control mode is that it produces a permanent residual error in the operating point of the controlled variable when a change in load occurs. This error is referred to as offset [4]. It can be minimized by a large constant K_p , which also reduces the proportional band. The proportional band is defined as that positive and negative error for which the output will be driven to 0% and 100%. To see how offset occurs, consider a system under nominal load with the controller at 50% and the errors as shown in Figure 1.

The effect of process and control system lag is shown as simple delays in the controller output change and in the error reduction when the controller action occurs.

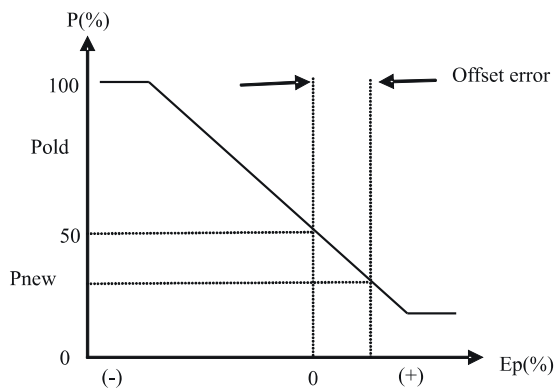


Fig. 1: Offset Error in a Proportional Integral and Derivate Controller

If the process lags are too large, the integral action causes a considerable overshoot of the error and output before settling to the operation point. Also, the error can oscillate about zero or even be cyclic. This is shown in figure 2, where proportional band is seen as a dashed band. The effect of the integral action can be viewed as a shifting of the whole proportional band [4].

Other PID weaknesses include: the presence of strong disturbances (non-linearity), time-varying parameters of the process and presence of dead times. The reason for these is that a PID controller assumes the process to behave in a strictly linear fashion. While this simplification can be made in a stable condition, strong disturbances can push the process operation point far away from the set point. The same happens if a process changes its parameters over time.

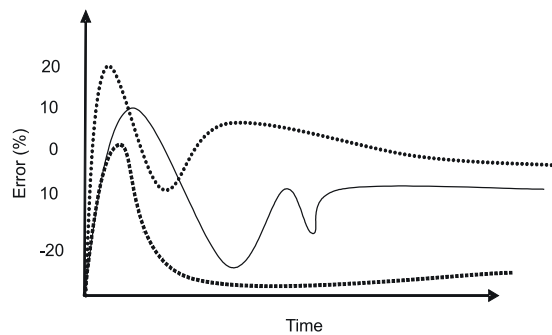


Fig. 2: Overshoot and Cycling in PID Controller

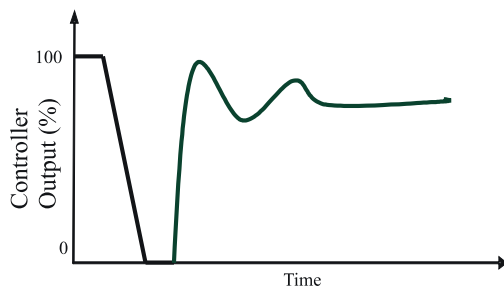


Fig. 3: Cycling in PID Controller

4. DESIGN OF A FLC FOR A HEAT EXCHANGER

The Heat Exchanger for Water for Injection (WFI) system was chosen to study the response and control action of a Fuzzy logic controller. The WFI system is shown in Figure 4. It consists of a Distiller, WFI storage and distribution tank, the circulation pump, and the Heat exchanger. The Distiller

generates distilled water and operates to keep the WFI storage tank within acceptable production limits. The circulation pump operates on a continuous basis and circulates WFI throughout the plant.

The unused distilled water is returned to the storage tank via the Heat Exchanger. The primary function of the Heat exchanger is to maintain the temperature of the return WFI at the set point. This is done by regulating the flow of plant steam into the Heat exchanger via a steam flow control valve. Thus, the Heat exchanger controls the process variable i.e. temperature via the control variable i.e. steam flow control valve position.

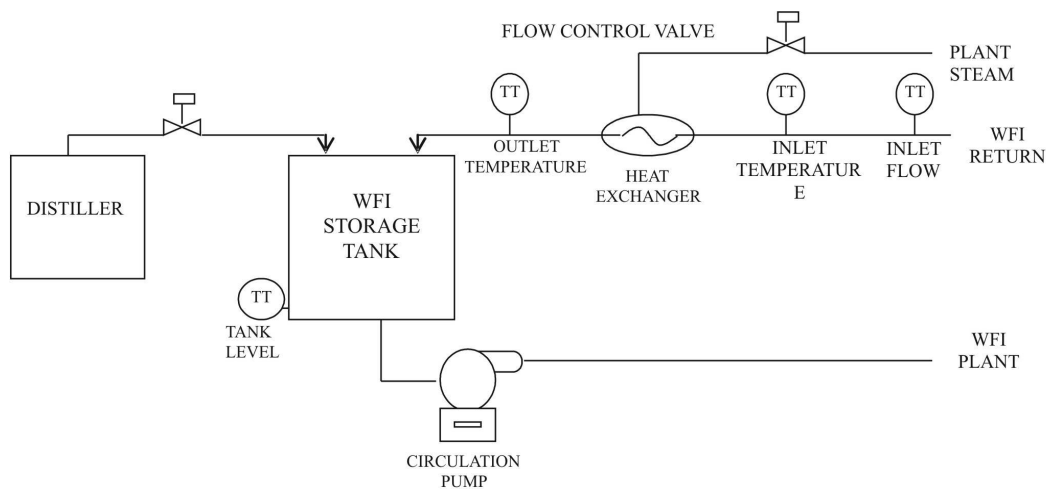


Fig. 4: Heat exchanger in a WFI System

4.1 Design Steps

Step 1: System Functional Requirements

The functional requirement of the system is to control the temperature of the return WFI

and minimize the effects of process upsets e.g. overshoots, undershoots due to set point and load changes.

Step 2: System Parameters

The principle of the Heat-exchange process is based on the control of one process variable namely, water temperature. The PID controller reads the water temperature at the outlet of the Heat-exchanger, applies a mathematical algorithm, and produces a control variable that manipulates the position of the steam flow control valve, which in effect control the steam flow into the Heat exchanger.

Step 3: System Parameters in terms of fuzzy sets

In order to build a fuzzy controller that represents the relationship between the inputs and the output, each variable must first be decomposed into a set of control regions called fuzzy sets. The input variable is temperature and the output variable is the control output applied to the steam flow control valve. The input variable temperature is used indirectly in the form of "Temperature Error" where,

$$\text{Error} = \text{Set point} - \text{Process variable} \quad (2)$$

The next step was to decide which fuzzification function should be used. Temperature loops are well known to be slow acting loops in comparison to pressure loops or pH loops and hence trapezoid and triangular functions were chosen. More complex functions are possible but require greater computing overhead to implement. The variable "Temperature Error" is decomposed into fuzzy regions using MATLAB. These regions were given unique names called labels, within the domain of the variable.

The number of labels associated with a variable corresponds to the accuracy of the

control action needed. The tighter the control action needed, the more the number of labels should be used. Also, each label should overlap somewhat with its neighbors. This overlap, in fact, is what gives a fuzzy controller its smooth, stable control surface. The labels used to describe the 'Temperature fuzzy' regions are as follows:

pb = Error is positive and big

ps = Error is positive and small

z = Error is zero

ns = Error is negative and small

nb = Error is negative and big

For the 'Error Rate', the labels used are:

p = Rate is Positive

ze = Rate is zero

n = Rate is negative

For the 'control output', the following were used:

Vh = Control output is very high

high = Control output is high

med = Control output is medium

low = Control output is low

vl = Control output is very low

The labels are self-explanatory, and it is clear that the more positive the "Error" the further away is the actual temperature from the set point the more the steam needed by the Heat exchanger. The input variable temperature is also used indirectly in the form of difference in error or temperature rate of change of error "Temperature rate".

Where "Temperature rate" = $\frac{\text{Previous Error} - \text{Current Error}}{\Delta t}$ (3)

The labels p, ze, and n are used to describe the rate of change of error "Temperature Rate" and this is analogous to the derivative action in a PID controller. The variable

“Temperature Rate” is decomposed into fuzzy regions. The output variable “Control” is also decomposed into fuzzy regions.

Step 4: Formulation of Control Rules

Here are a list of intuitive rules that govern the heat-exchanger application are made. Unlike the conventional control method, which yields a mathematical model, the rules are developed in language of IF – THEN statements. In this model five labels, representing two in the positive direction, two in the negative direction, and a zero for the ‘Temperature Error’ are used, three labels, one in both directions and a zero for the

‘Temperature rate’. The rules that describe the ‘Control’ to be taken on each combination of the control variables ‘Temperature Error’ and ‘Temperature Rate’ are explained below.

Here “Error” = Derived set point - Actual temperature at the outlet of the heat exchanger = SP - PV (out) (4)

and
 “Temperature Rate” = Previous error - Current error (5)

Table 1: Rule Base 1

Rule	Antecedent Block	Consequent Block
Rule 1.1	If Error is nb and Rate is n	THEN output to valve is high
Rule 1.2	If Error is nb and Rate is ze	THEN output to valve is vh
Rule 1.3	If Error is nb and Rate is p	THEN output to valve is vh
Rule 1.4	If Error is ns and Rate is n	THEN output to valve is high
Rule 1.5	If Error is ns and Rate is ze	THEN output to valve is high
Rule 2.1	If Error is ns and Rate is p	THEN output to valve is med
Rule 2.2	If Error is z and Rate is n	THEN output to valve is med
Rule 2.3	If Error is z and Rate is ze	THEN output to valve is med
Rule 2.4	If Error is z and Rate is p	THEN output to valve is med
Rule 2.5	If Error is ps and Rate is n	THEN output to valve is med
Rule 3.1	If Error is ps and Rate is ze	THEN output to valve is low
Rule 3.2	If Error is ps and Rate is p	THEN output to valve is low
Rule 3.3	If Error is pb and Rate is n	THEN output to valve is low
Rule 3.4	If Error is pb and Rate is ze	THEN output to valve is vl
Rule 3.5	If Error is pb and Rate is p	THEN output to valve is vl

As can be seen from Table 1, there are 15 rules covering all possible states of the defined process variables. The rule numbers are of the form “Rule r.c”, on purpose, where “r” and “c” are row and column numbers of

the rules when tabulated in a matrix form called Rule – Matrix.

The Rule – Matrix 1 is shown in Table 2

$$\text{Rate} = \text{Previous} - \text{Current} \quad (6)$$

Table 2: Rule Matrix

	nvb	nb	z	ps	pb
n	high	Vh	Vh	High	High
ze	med	Med	med	Med	Med
p	low	Low	Low	Vl	Vl

Step 5: Selection of a Method of Defuzzification

The “center of gravity” or the “centroid” method of defuzzification was chosen, since it weighs the effect of each input variable towards the calculation of the output [5]. Input fuzzy sets and rules are converted into an output fuzzy set, and then into a crisp output for controlling the steam control valve. All the rules that have any truth in their premises will fire and contribute to the output fuzzy set – the one that will represent the control variable controlling the steam valve. The process of defuzzification is explained below:

Suppose that at a given time, the system sensor determine the “Error” and “Error Rate”. The degree of fulfillment is determined for rules that fires. The rules have somehow to be combined to form a single system output.

The following three-step procedure [6], shows how.

- For each premise expression connected by an AND, take the minimum of the truth of the expressions as the truth level of the premise.
- Truncate the output fuzzy set being built at the truth level of the premise.

- Copy the newly modified fuzzy set into the output variable’s fuzzy set. If that region is not empty, combine it with the current contents by taking the maximum of the new fuzzy region and the currently existing fuzzy region at each point in the domain (along the horizontal axis). In order words, if the region is not empty, then OR the outputs together.

This is simplified and illustrated using MATLAB. A value for the control output is determined. This value is used to adjust the position of the steam control valve. After that, the temperature sensor will make new measurements, starting the cycle over gain.

5. SYSTEM SIMULATION

MATLAB was used to generate the Membership Functions. The control outputs, for given ‘Temperature errors’ and ‘Temperature Rates’ were also determined using MATLAB.

The labels used are:

Temperature Error: nb, ns, z, ps, pb.

Temperature Rate: p, ze, n.

Control: vh, high, med, low, vl.

Where, the labels are as defined in table 4.2.

For the ‘Temperature error’, the MATLAB program used in generating the Membership Functions is:

```
Temperature_error = [-36 : 0.1: 36];
nb= trapzoid ( Temperature_error, [-36 -36
-10 -5]);
ns = triangle ( Temperature_error, [-10 -2
0 ]);
z = triangle ( Temperature_error, [-1 0 1 ]);
ps = triangle ( Temperature_error, [0 2 10 ]);
```

```

pb = trapezoid ( Temperature_error, [5 10 36
36 ]);
Temperature_error = [nb;ns;z;ps;pb];
plot ( Temperature_error, T_error);
title ( 'Temperature Error Membership
Functions' );
xlabel ( 'Temperature Error' )
ylabel ( 'Membership' )

```

When RUN in MATLAB environment, the resulting 'Temperature Membership functions' are as shown in Figure 5.

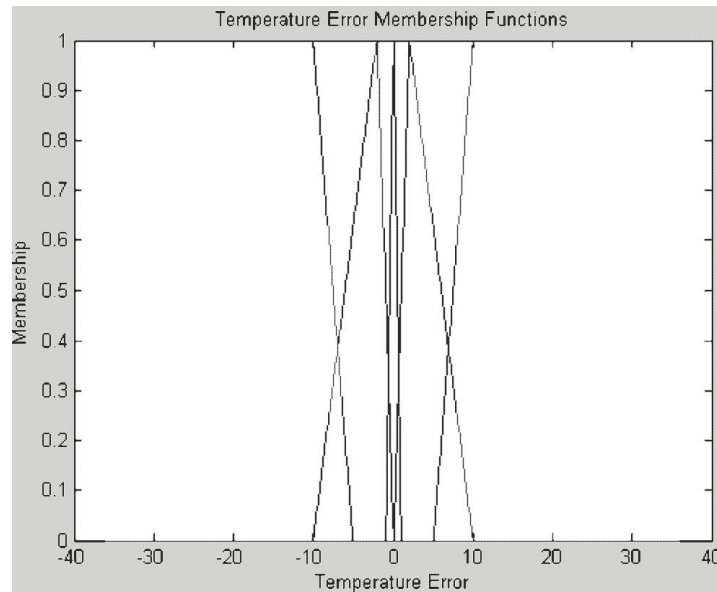


Fig. 5: Temperature Error Membership Functions

For the Temperature Rate, the MATLAB program is:

```

Temperature_Rate = [ -40 :0.1 :40 ];
p= trapzoid( Temperature _Rate, [-40 -40 -2
0 ]);
ze= triangle ( Temperature_Rate, [-1 0 1]);
n = trapzoid ( Temperature _Rate, [ 0 2 40
40 ]);
T_Rate = [p;ze;n];
plot ( Temperature_Rate,T_Rate);

```

```

title('Temperature Rate Membership
Functions' )
xlabel ('Temperature Rate')
ylabel ('Membership')

```

When RUN, the resulting 'Temperature Rate Membership Functions' are as shown in Figure 6

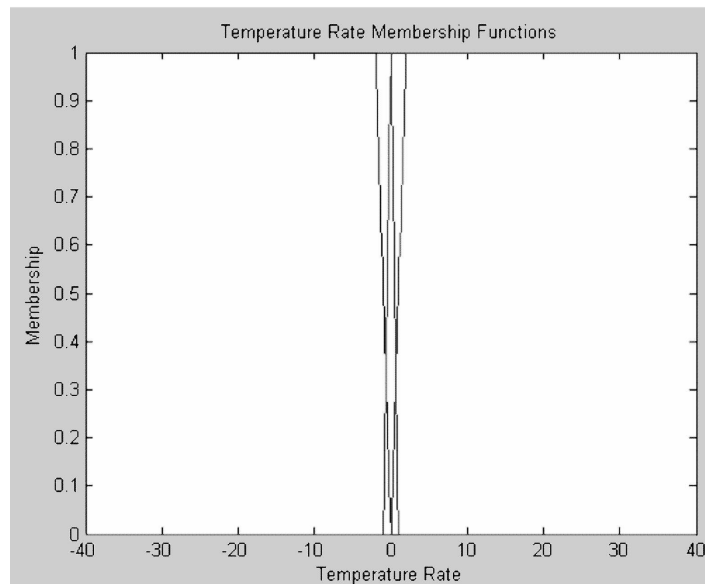


Fig. 6: Temperature Rate Membership Functions

The MATLAB program that is used to generate the output Control Membership Functions are:

```
Control = [-4.5: 0.05: 1];
vh = triangle (Control, [0 1 1 ] );
high = triangle (control, [-1 0 1]);
med = triangle (Control, [-3 -2 -1 ] );
low = triangle (control, [-4.5 -3.95 -3 ]);
vl = triangle (control, [-4.5 -4.5 -3.95 ]);
```

The resulting 'Output Current Membership Functions' are as shown in Figure 7.

5.1 Centroid Defuzzification

The next step was to create a Mamdani fuzzy

system that uses centroid defuzzification. Suppose a Temperature error of -8.1 degC and Temperature Rate of $0.3^{\circ}\text{C}/\text{min}$ are input to the fuzzy controller. First, the degree of fulfillments of the antecedent membership functions is calculated.

The MATLAB program is:

```
Temperature_error = -8.1;
Temperature_Rate = 0.3;
DOF1 = interp1 ( Temperature_error', T_
error' error' );
DOF2 = interp1(Temperature_Rate', d_error'
error' );
```

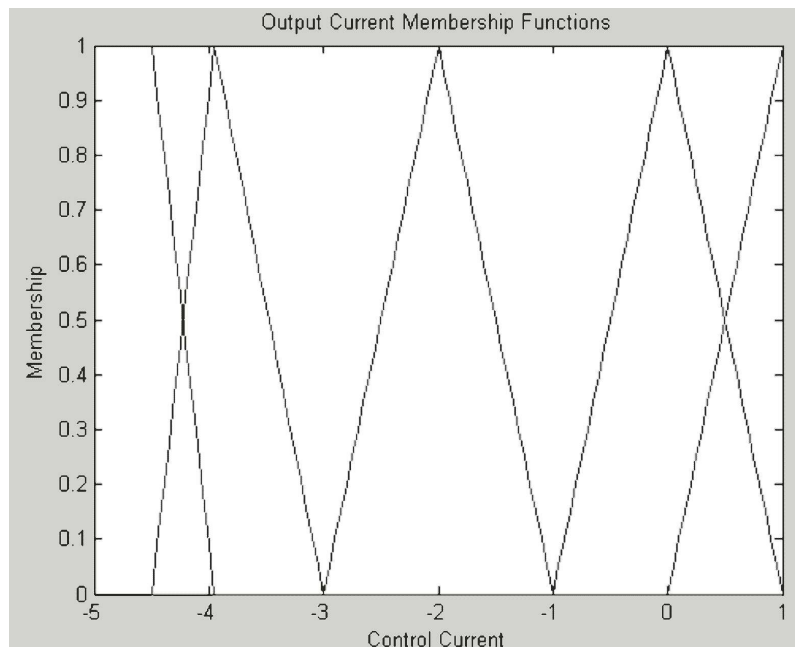


Fig. 7: Output Current Membership Functions

Next, the fuzzy relation operations inherent in the 15 rules were performed as shown using MATLAB:

```

antecedent_DOF= [min (DOF(1), DOF2(1))
min (DOF1(1), DOF2(2))
min (DOF1(1), DOF2(3))
min (DOF1(2), DOF2(1))
min (DOF1(2), DOF2(2))
min (DOF1(2), DOF2(3))
min (DOF1(3), DOF2(1))
min (DOF1(3), DOF2(2))
min (DOF1(3), DOF2(3))
min (DOF1(4), DOF2(1))
min (DOF1(4), DOF2(2))
min (DOF1(4), DOF2(3))
min (DOF1(5), DOF2(1))
min (DOF1(5), DOF2(2))
min (DOF1(5), DOF2(3))
min (DOF1(1), DOF2(2))]
    
```

When RUN, in MATLAB environment, the resulting DOF for the pair of inputs is:

```

antecedent_DO=0, 0.6200, 0.1500, 0, 0.2375,
0.1500, 0, 0, 0, 0, 0, 0, 0, 0
    
```

The MATLAB program for the control action is:

```

consequent = [ control ( 5, :),control (5,
:),control (4, :),control (4, :),control (4,
:),control (3, :),control (3, :),control (3,
:),control (3, :),control (3, :),control (2,
:),control (2, :),control (2, :),control (1,
:),control (1, :)];
    
```

The MATLAB program for the generation of 'Consequent of Fuzzy Rules' is:

```

Consequent = product (consequent,
antecedent_DOF);
plot ( Control, Consequent)
axis ( [min(Control) Max(Control) 0 1.0])
title ( 'Consequent of Fuzzy Rules')
xlabel ( 'Control Current')
ylabel ( 'Membership' )
    
```

When RUN, in MATLAB environment, the resulting ‘Consequent of Fuzzy Rules’ is as shown in Figure 8.

From this figure, it can be seen that this output is not a single fuzzy set, then the fuzzy output sets was aggregated to form a single fuzzy output set.

```
The MATLAB program for that is:
aggregation =max (Consequent);
plot (Control, aggregation)
axis ([min (Control) max(Control) 0 1.0])
title ( ‘Aggregation of Fuzzy Rule Outputs’)
xlabel ( ‘Control Current’)
```

ylabel (‘Membership’)

The resulting ‘Aggregation of Fuzzy Rule Outputs’ when RUN is as shown in Figure 9.

Next is to find the crisp output current. To find the crisp output current the output fuzzy set is defuzzified as shown using MATLAB program:

```
output = centroid ( Control, aggregation);
c_plot (Control, aggregation, output, ‘Crisp
Output Value for Current’)
```

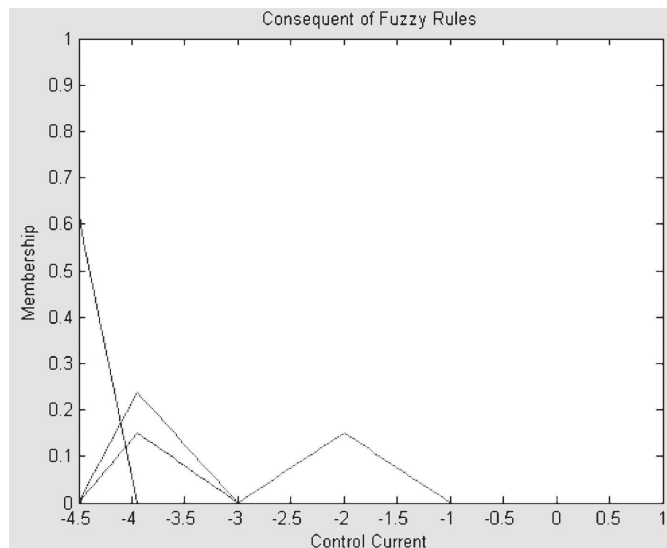


Fig. 8: Consequent of Fuzzy Rules

```
axis ([min( Control) max(Control) 0 1.0])
xlabel ( ‘ Control Current’ );
```

The ‘Crisp Output Value’ for the current is as shown in figure 11. For this inputs, a current of -3.402mA would be sent to the control valve. This value is used to adjust the position of the steam control valve. After

that, the temperature sensor will make new measurements, starting the cycle over again. Six other cases of ‘Temperature Errors’ and ‘Temperature Rates’ were input to the fuzzy controller, one pair of inputs at a time.

MATLAB program for ‘Centroid Defuzzification’ was used to compute the various ‘Crisp outputs’ and the results were recorded accordingly as shown in Table 3. The ‘Temperature Rate’ stays constant

through out the cases. It is not likely that this would happen in a real system, but for purposes of illustration that case was assumed.

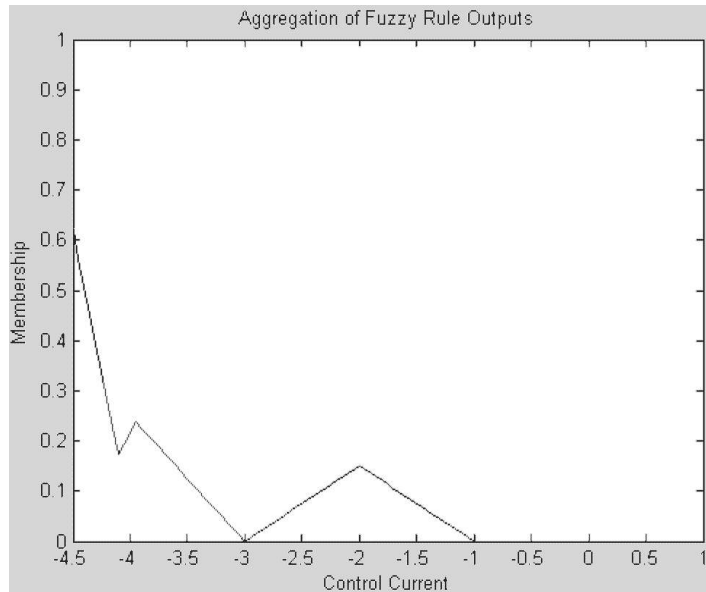


Fig. 9: Aggregate of Fuzzy Rule Outputs

6. RESULTS

The results obtained from the simulation are presented in Table 3.

Table 3: Simulation results

Error (°C)	Rate (°C/min)	Control Output
-8.1	0.3	-3.402
-5.0	0.3	-2.143
-3.0	0.3	-0.839
0	0.3	0.000
3.0	0.3	1.039
5.0	0.3	2.239
8	0.3	2.613

The next step was to use the standard PID instruction, available as a user-configurable block for testing PID control. The independent gains equation was used, that is output

$$= U = K_P + K_D \frac{de}{dt} + k_I \int e(t)dt + P_0$$

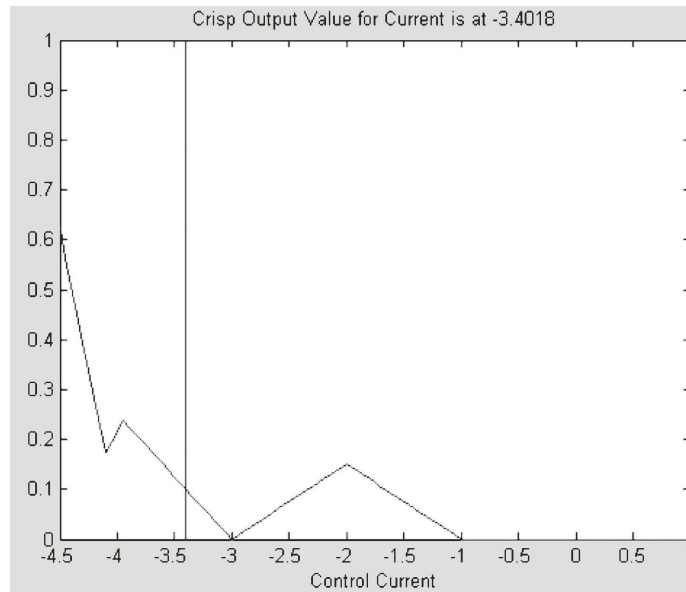


Fig. 10: Crisp output current

The temperature loop was tuned by adjusting the loop tuning constants, K_p , K_i and K_d . The software written in ladder logic for achieving fuzzy control was used. Using Rule Base 1 and inserting values, I have Rule Base 1 Data as shown in Table 4:

Table 4: Rule Base 1 Data

	nvb	nb	z	ps	pb
n	0	1	1	0	0
ze	-2	-2	-2	-2	-2
p	-3.95	-3.95	-3.95	-4.5	-4.5

The response of the PID Controller, Rule Base 1 for a set point of 90°C are shown in Figures 11 and 12 respectively.

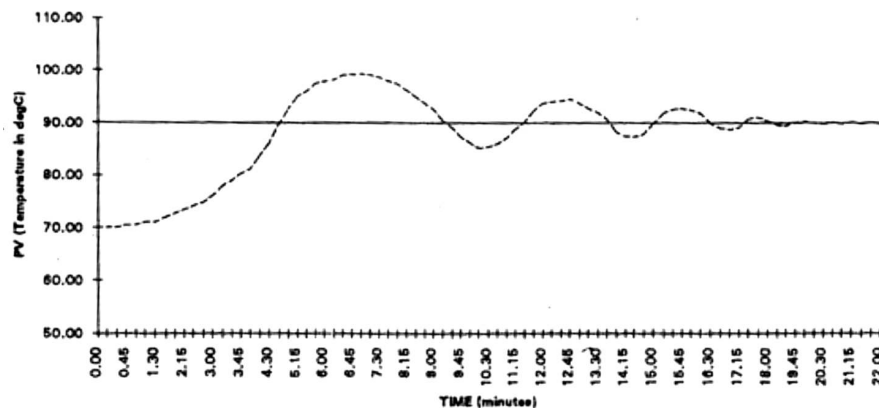


Fig. 11. Response of PID Controller

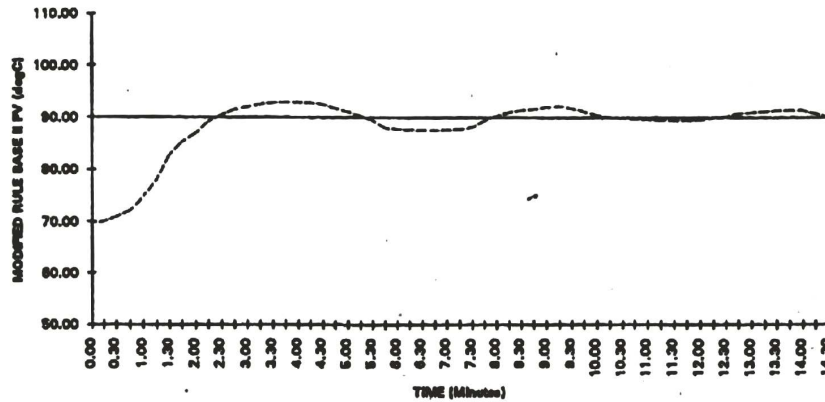


Fig.12: Response of Rule Base 1

A heuristic approach was followed to modify the rules. Based on experience, the Rule Base 1 was modified as shown in Table 5. Since the system was having slow response rate as long as 'Error' was large, a large control signal to the steam control valve was applied. The modified data is shown in Table 5.

n b z
p s

pb

n	1	1	1	1	1
ze	-1	-1	-1	-1	-1
p	-3.95	-3.95	-3.0	-3.0	-3.0

The response for the modified Rule Base 1 is shown in Figure 13. As seen from Figure 13, the response of the modified Rule Base 1 is far more superior to the PID Control response.

Table 5:

Modified Rule Base 1 Data for a faster response

nvb

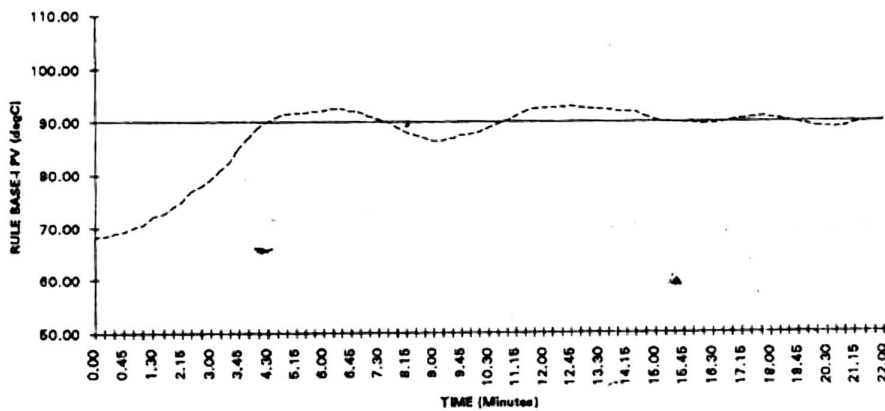


Fig. 13: Response of modified Rule Base 1

A comparison of PID controller and three fuzzy strategies is provided in table 6. The following response evaluation factors are considered.

Time Constant: The time constant is defined as time required to complete 63.2% of the total response.

Peak Error: It is the maximum deviation of the process variable from the setpoint.

Settling Time: In the event of a transient input or a change in set point, the settling time is the time required for the process control loop to bring the process variable back to within the allowable setpoint range.

Table 6: Comparison of PID and Fuzzy Strategies

	Time constant	Peak Error	Settling time
PID Control	4.15 mins.	9.3°C	12.15 mins
Rule base 1	3.30 mins	2.5°C	12.30 mins
Modified Rule Base I	3.15 mins	2.3°C	5.30 mins

6.1 Analysis of Results

As can be seen from Table 6, as the error increases from negative values to zero and then to positive values, the control output also changes in sympathy to positive values which agrees with the initial predictions. When a 'Crisp output' of -3.402mA was obtained, the negative current implies that the actual temperature of the heat exchanger is above the desired one, the controller will change its output direction that closes the steam control valve, to lower the temperature.

The positive values of output current implies that the actual temperature of the heat exchanger is below the desired one, the

controller is in the direction that opens the steam control valve to allow in more steam to increase the temperature.

At zero control output, the output direction remains unchanged. It was observed from figures 10 - 13 that though there was no significant change in the time to reach the set point, the overshoot was drastically reduced by Rule Base 1. Cycling of the process variable around the set point as observed in the PID response was also considerably reduced by Rule Base 1. The modified Rule Base 1 had similar results. There was no significant change in the time required to reach the set point. The overshoot was further

reduced in comparison to Rule Base 1. This was due to the fact that the Rule Base 1 was modified. Though there was practically no cycling around the set point, the process variable lingered above the set point. This is the result of the centroid method of deriving rules.

CONCLUSIONS

The problems associated with the conventional PID controller were studied for a Heat Exchanger in a WFI system. Multiple variables affecting the stability of the controller were found to be the main problem causing the inconsistent behavior. Rule-based strategies were designed and implemented. The Rule base 1 utilizes temperature error at the heat exchanger outlet and the rate of change of this error, as the input to the FLC. Though the time required to reach the set point had no significant improvement, the overshoot was drastically reduced in comparison with a PID controller. Rule base 1 was further modified for a faster response.

REFERENCES

1. S.H. Noie, Investigation of thermal performance of an air-to-air thermosyphon heat exchanger using – NTI method, *Applied Thermal Engineering* 26, 2006, 559 -567.
2. J. L. Niu, L.Z. Zhang, H.G. Zuo, Energy savings potential of chilled ceiling combined with desiccant cooling in hot and humid climates, *Energy and Buildings* 34(2002) 487-495.
3. L. Z. Zhang, D. S. Zhu, X. H. Deng, B. HUa, Thermodynamic modeling of a novel air dehumidification system, *Energy and Building* 37, 2005,279-286.
4. Johnson, C. D., (1987), *Process Control Instrumentation Technology* John Wiley and Sons Inc. NY: 3rd Edition.
5. Lee C.C. (1990), *Fuzzy Logic in Control System*, *IEEE Trans. On Systems, Man & Cybernetics*. 20(2) Mar/April Edition.