Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol II
IMECS 2009, March 18 - 20, 2009, Hong Kong

Multi-Dimensional Supervisory Fuzzy Logic Time Control DEV Processing System for Industrial Applications

M. Saleem Khan, Khaled Benkrid

Abstract—**This research paper presents the design model of a fuzzy logic time control discrete event DEV system under the control of multi-agents based supervisory control in local and distributed environment for industrial application of a processing plant regarding the specific product of certain quality and amount. The designing concepts of fuzzy logic time control discrete event system are summarized in various modes of operation and a new supervisory approach using multi-agents is implemented in local and distributed control modes. This research work will enhance the capability of fuzzy logic time control systems in process automation with potential benefits of multi-dimensional control and supervision.**
MATLAB-simulation is adapted to establish the design goal of multi-agents based supervisory fuzzy logic time control processing system.

Index Terms— **Multi-agents based Fuzzy logic control, supervisory control, fuzzy logic system in distributed environment, fuzzy processing control, multi dimensional fuzzy control.**

I. Introduction

The issues regarding the degree of vagueness of variables in process control tasks can be handled properly using systematic framework of fuzzy logic control. During the last 20 years, fuzzy logic control strategies contributed a lot in the field of control for manufacturing of consumer products and to achieve the successful industrial tasks [5].
Fuzzy logic time control (FLTC) discrete event (DEV) systems have the merits versus conventional control to operate event based processes in specific time [3]. This design algorithmic approach makes the system efficient and completely under the control of time to avoid time delay evolved in feedback control mechanism. The classical fuzzy control system structures like
fuzzy SISO, fuzzy MIMO, cascaded control system with fuzzy main control, fuzzy feed forward and feed backward control, variable control structure with fuzzy weighing, fuzzy feed forward and feed backward adaptive control need to be structured using concepts of fuzzy logic time control discrete event system.

Manuscript received December 30, 2008.
F. A. M. Saleem Khan is with the GC University Lahore Pakistan, working as an assistant professor in the field of Electronics. Currently availing a research fellowship at The School of Electronics & Engineering in Edinburgh University UK (e-mail mskgcu@yahoo.com, s.khan@ed.ac.uk)
S. B. Khaled Benkrid is with the School of Electronics & Engineering in Edinburgh University UK, working as a lecturer and research supervisor at The University.(e-mail K.Benkrid@ed.ac.uk)

Multi-agents based supervisory fuzzy control system in local and distributed environment using any one of its basic structure will support the issues of controllability and efficiency under the distributed management control of the system operation [8].

II. BASIC STRUCTURE OF THE PROPOSED MODEL

A general scheme for the organisation of industrial process automation task based on fuzzy logic time control DEV, linked with supervisory control, multi-agents information processing through supervisory control and via communication control arrangement is shown in Fig.1.
The supervisory check point, monitoring and control feedback help to avoid undesired or unpermitted process and decide to take appropriate actions such as shut-down or triggering of redundancies or reconfiguration schemes through its feedback to the system inputs.
Multi-agents based information, gathered from various external sources is used to control the system or to take appropriate actions through supervisory control or via communication link. In industrial environment agents based information may be regarded as the non operation of a certain part of the plant, smoke or fire detector, product quality issues and unavailability of the product contents up to certain level needs to reduce the volume.
This sort of facility can manage the system in local and distributed environment. Therefore, the processed variables detail is also communicated to make the system under the control of distributed management.

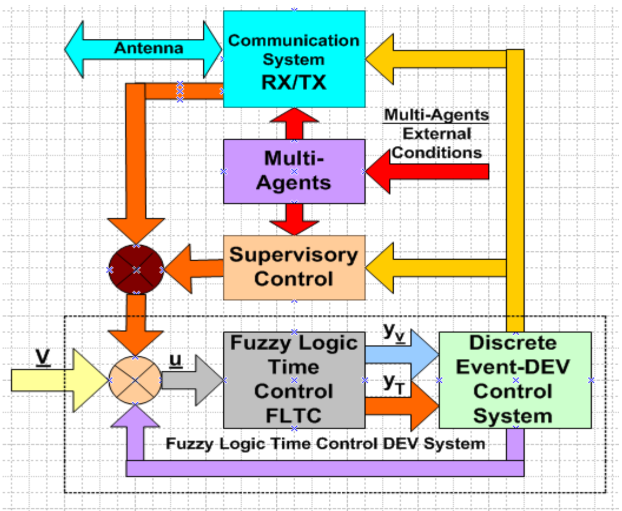
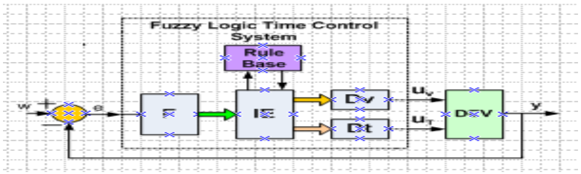
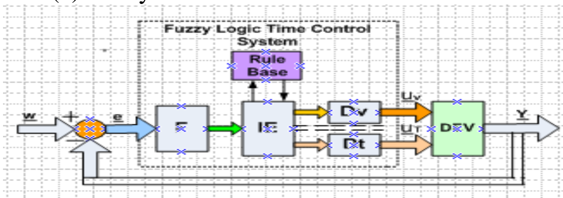


Fig. 1 Block diagram of multi-dimensional supervisory fuzzy logic time control DEV system.

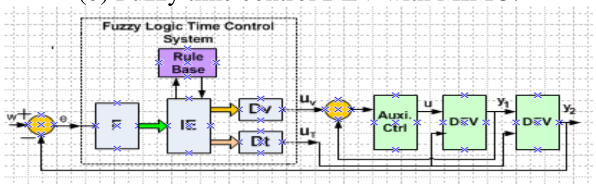
III. BASIC LEVELS OF FUZZY LOGIC TIME CONTROL DEV DESIGN



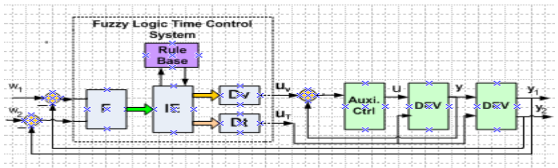
(a) Fuzzy time control DEV-SISO controller



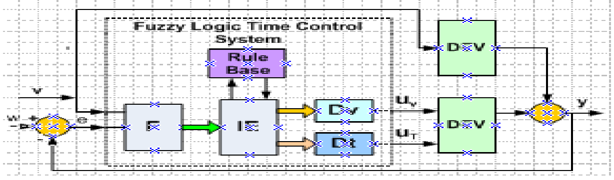
(b) Fuzzy time control DEV with MIMO.



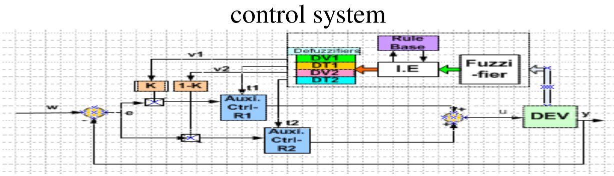
(c) Cascaded control system with fuzzy time control DEV as main controller.



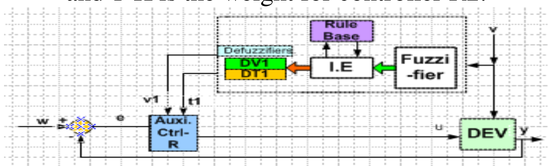
(d) Cascaded control system with fuzzy time control DEV as a main controller for two alternative control variables y1&y2.



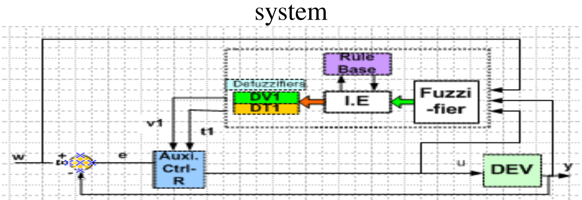
(e) Fuzzy time control DEV- feed forward and feedback control system



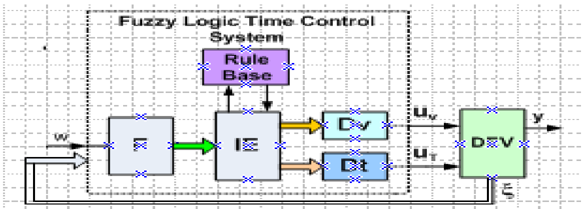
(f) Variable control structure with fuzzy time control DEV weighing- K is the weight for controller R1 and 1-K is the weight for controller R2.



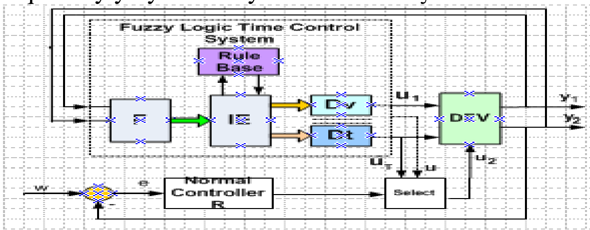
(g) Fuzzy time control DEV- feed forward adaptive control system



(h) Fuzzy time control feedback adaptive control. Fuzzy time control self tuning of classical controllers



(i) Fuzzy time control DEV system of a non measureable quantity y by auxiliary measurement ξ.



(j) A feedback control for normal operating conditions with normal controller and for special operating conditions with a fuzzy time control DEV controller.

Fig. 2 Control system structures with different fuzzy logic time control DEV component. Parts (a) to (j) show the different control arrangements

Any one of the control strategies shown in Fig.2 can be implemented as a fuzzy logic time control DEV component depending upon the requirements and the number of inputs and controlled outputs used.

Fig. 2(a) & (b) show SISO and MIMO direct fuzzy time control DEV systems. In these cases a general fuzzy-rule-base system is required for the design of fuzzy controller. The defuzzification follows via the center of singletons and output attributes must be distributed over the universe of discourse.[8]

Fig.2(c) & (d) show the fuzzy DEV cascaded control systems. These systems are used to improve the control performance with an auxiliary classical controller and a main fuzzy DEV controller [9].

Fig. 2(e) represents fuzzy DEV feed forward and feedback control. In this system external disturbance is measured and added into fuzzified value to improve the performance and proportional nonlinear feed forward control action is applied [8].

Fig.2 (f) shows a variable control structure with fuzzy DEV weighting. This type of the system can be used to avoid the change in transfer behaviour of plant, control deviations and disturbances [10].

Fig.2 (g) & (h) show the fuzzy DEV adaptive control system used to adapt the control behaviour of the change properties of the processes under control and adapts their signals [11].

Fig. 2(g) shows the fuzzy DEV feedforward adaptive control and Fig.2 (h) shows feedback adaptive control with fuzzy self tuning of classical controllers. These controllers can be adjusted using tuning rules based on open loop transfer function measurement [12], [13].

Fig.2 (i) shows fuzzy DEV control of non measureable quantities using an alternate sensor signal instead of original sensor to avoid the economic constraints [8].

Fig.2(j) shows a feedback optional control used in normal operating conditions along with a normal controller and in special operating conditions with a fuzzy DEV controller[8].

In our process control design, we used three inputs variables and, six outputs: three for controlled output variables and, three for their timing control as shown in Fig.3.

Fig. 2(b) represents fuzzy logic time control DEV with multiple inputs-multiple outputs (MIMO), this structure is used in the proposed design model.

IV. DESCRIPTION OF MULTI-AGENTS BASED SUPERVISORY FUZZY LOGIC TIME CONTROL DEV INDUSTRIAL PROCESSING SYSTEM

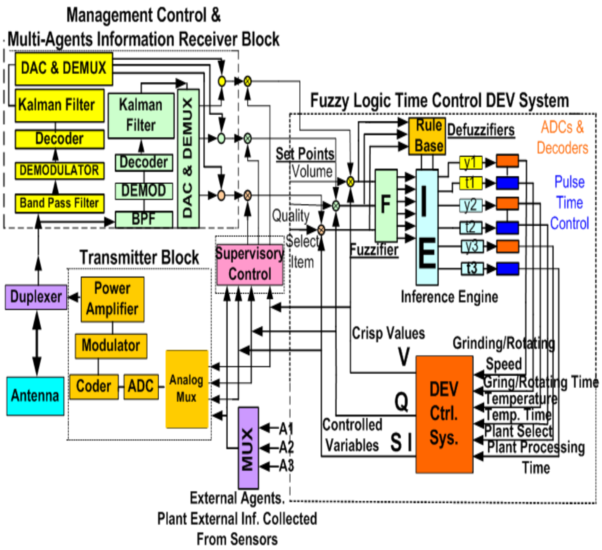


Fig.3 Main diagram of the proposed model

The proposed design work deals with industrial application of a novel multi-agents based processing system consisting of three input variables: "Volume", "Quality" and "Select item". These are used in a processing plant: to maintain the certain grinding/rotating motor speed for a particular time, to maintain heating/cooling temperature for a certain time, and to select a particular plant for a certain processing time. Here different plants are used for different quality production.

There are six output variables of the processing system, therefore six defuzzifiers are used for: grinding/rotating motor speed, grinding/rotating time, heating/cooling temperature, heating/cooling time, plant selection, and plant processing time.

The crisp values of defuzzifier outputs for grinding/rotating motor speed, heating/cooling temperature, and plant selection are given to three different ADCs and decoders for the selection of discrete events of DEV system under time control.

The time control of the events is set using pulse time control of grinding/rotating time, heating/cooling time and plant processing time.

The DEV control system information about product volume, product quality and item selection is gathered from various sensors and sent to: communication system for distributed management record to involve in the processing control , supervisory control to involve in control processing at local level if it is needed. The supervisory control also receives the multi-agents multiplexed information to over ride the main control processing in case of emergency if it is needed. At local level the action of multi-agents is through supervisory control. The multiplexed information of multi-agents is also communicated for the distributed control.

Input set points of processing plant are fully under the control of: Plant operator manually, plant feed back, supervisory control at local level, and distributed management through communication.

The communication transmitting block consists of: analog multiplexer (4to1), analog to digital converter (ADC), coder for specific code generation for each information, digital modulator, and power amplifier according to specific antenna requirement. The receiving block consists of two different bandwidth sections: one receiving section is used to pass the control information of distributive management, and another section to receive the multi-agents information in distributed mode. Each section consists of: band pass filter, demodulator, decoder, Kalman filter, digital to analog convertor and analog demultiplexer. Duplexer is needed for the antenna to use both transmitting and receiving sections.

A. Design Algorithm Of Supervisory Control System

The complete Supervisory control is linked with local and distributed environment as shown in Fig.4. The communicated information from distributed management is shown as, a1, b1 and c1 to control volume, quality and item select respectively.

a2, b2 and c2 are the controls sent by multi-agents. All of these signals are collected from the receiver.

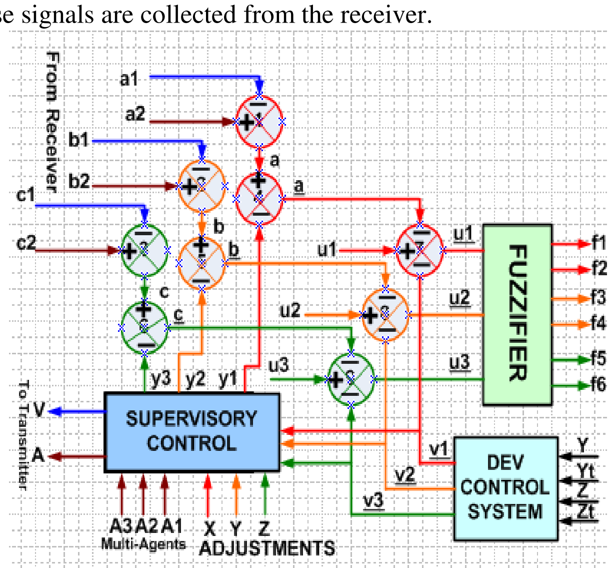


Fig. 4 Supervisory control arrangement

Three inputs, set points of the main fuzzy control circuit for volume, quality and, item select are u1, u2 and u3.The fuzzifier inputs are $\underline{u1}$, $\underline{u2}$ and $\underline{u3}$. The feedback of the main DEV control is, in the form of $\underline{v1}$, $\underline{v2}$ and $\underline{v3}$. Multi-agents information given to supervisory block are A1, A2, and A3.Where the other inputs to supervisory control are $\underline{v1}$, $\underline{v2}$ and $\underline{v3}$. The outputs of supervisory control for local environment are y1, y2 and y3.

The outputs: V is the information about controlled variables, A is about multi-agents and given to transmitter for distributed control.

In fig. 4 nine adders-subtractors, numbered as 1, 2,3,4,5,6,7,8 and 9 are used at various points. We can establish the relations of various signals according to Fig.4.

The mathematical relation for the volume input of fuzzifier is,

$\mathbf{\underline{u1} = u1 - \underline{v1} - \underline{a}}$

where, $\mathbf{\underline{a} = a - y1}$, or

$\mathbf{\underline{u1} = u1 - \underline{v1} - a + y1}$, where **a** is the output of subtractor 1 and $\mathbf{a = a2 - a1}$, or

$\mathbf{\underline{u1} = u1 - \underline{v1} - a2 + a1 + y1}$**................ (i)**

Similarly, the expressions for other inputs, quality and item select are,

$\underline{U2} = u2 - \underline{v2} - b2 + b1 + y2$................ **(ii)**
$\underline{U3} = u3 - \underline{v3} - c2 + c1 + y3$................ **(iii)**
These expressions show that how the different factors effect the inputs of the main control fuzzifiers.

B. Control Strategies of Supervisory Arrangement.

The supervisory control unit can control the main fuzzy time control DEV system in multi dimensions.
Various control strategies can be adapted according to the need of supervisory system under the priority circumstances.
1. No control of supervisory system at local level, it allows the main fuzzy system to work under its set points values. In this case no distributed control is given.
$y1 = y2 = y3 = 0$ and $V = A = 0$, according to (i), (ii) and (iii), we get

$\underline{u1} = u1 - \underline{v1}$
$\underline{U2} = u2 - \underline{v2}$
$\underline{U3} = u3 - \underline{v3}$

2. When supervisory control has its own adjustments and considering multi agents to effect at local level without giving permission to distributed environment. In this case
$V = A = 0$, $a = b = c = 0$ and **$\underline{a} = \underline{b} = \underline{c} = 0$,** as, **$a1 = a2 = b1 = b2 = c1 = c2 = 0$.** Using (i), (ii) and (iii), we get

$\underline{u1} = u1 - \underline{v1} + y1$
$\underline{U2} = u2 - \underline{v2} + y2$
$\underline{U3} = u3 - \underline{v3} + y3$

3. When distributed environment is permitted and local supervisory arrangement is not adapted.
In this case, **$y1 = y2 = y3 = 0$**

$\underline{u1} = u1 - \underline{v1} - a2 + a1$
$\underline{U2} = u2 - \underline{v2} - b2 + b1$
$\underline{U3} = u3 - \underline{v3} - c2 + c1$

4. When partial distributed environment is permitted and local supervisory control in not allowed
In this case **$y1 = y2 = y3 = 0$** and, either **$A = 0$** or **$V = 0$**
When **$A = 0$**

$\underline{u1} = u1 - \underline{v1} + a1$
$\underline{U2} = u2 - \underline{v2} + b1$
$\underline{U3} = u3 - \underline{v3} + c1$

5. When partial distributed environment is permitted and local supervisory control in not allowed
In this case **$y1 = y2 = y3 = 0$, When $V = 0$**

$\underline{u1} = u1 - \underline{v1} - a2$
$\underline{U2} = u2 - \underline{v2} - b2$
$\underline{U3} = u3 - \underline{v3} - c2$

The numbers of these five strategies are considered as priority levels (P1, P2, P3, P4, and P5) in the design of Supervisory control unit.

C. Design of Supervisory Control Unit

The interconnection of supervisory control and link of supervisory control with the main fuzzy control DEV are shown in fig.5.
Fig. 6 gives the complete supervisory control block arrangement, showing 23 inputs and 5 outputs. Five inputs for the priority arrangements according to the designing scheme, three inputs for the multi-agents interference, three inputs from DEV control system, three inputs for the provision of supervisory adjustments, three inputs from set points main input terminals, six inputs from the receiving circuit: three inputs from distributed management; for volume, quality, and select item control interference and three inputs from multi-agents based communicated information; for the control

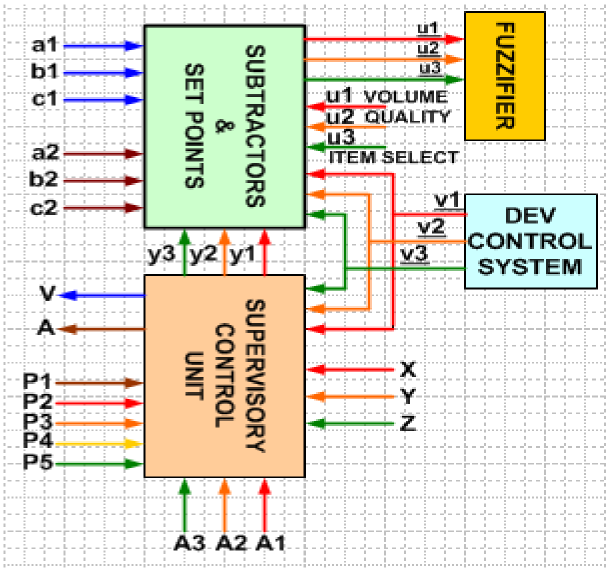


Fig. 5 Supervisory control blocks

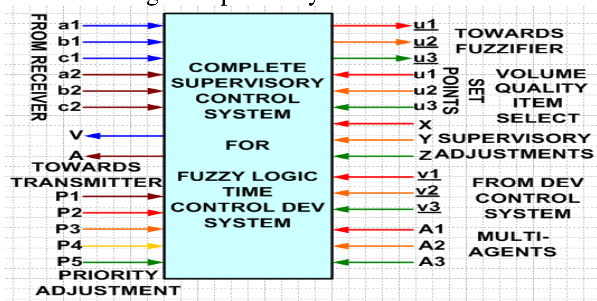


Fig. 6 Complete supervisory control block

interference. The three outputs of the supervisory control block are provided to the main fuzzy control inputs and the two outputs, one for the communication of multi-agents based control information and another for sending information to distributed management.
The design of control block is according to the scheme shown in Table 1.
The description of each strategy is given with priority arrangements P1, P2, P3, P4, and P5. This block, includes the sum and subtract arrangements under the control of priority information.

TABLE I
DESIGN DESCRIPTION OF SUPERVISORY CONTROL ARRANGEMENT

Description	P5	P4	P3	P2	P1	$\underline{u3}$	$\underline{u2}$	$\underline{u1}$
Fuzzy system works under its set points values, $y1 = y2 = y3 = 0$ and $V = A = 0$	0	0	0	0	1	$u3 - \underline{v3}$	$u2 - \underline{v2}$	$u1 - \underline{v1}$
Adjustment of supervisory control and multi-agents at local level but communication mode is not allowed. $V = A = 0$, $a1 = a2 = b1 = b2 = c1 = c2 = 0$	0	0	0	1	0	$u3 - \underline{v3} + A3 + Z$	$u2 - \underline{v2} + A2 + Y$	$u1 - \underline{v1} + A1 + X$
Distributed environment is permitted and local supervisory arrangement is not adapted. $y1 = y2 = y3 = 0$	0	0	1	0	0	$u3 - \underline{v3} - c2 + c1$	$u2 - \underline{v2} - b2 + b1$	$u1 - \underline{v1} - a2 + a1$
Distributed environment is permitted partially and local supervisory control in not allowed. $y1 = y2 = y3 = 0$, $A = 0$	0	1	0	0	0	$u3 - \underline{v3} + c1$	$u2 - \underline{v2} + b1$	$u1 - \underline{v1} + a1$
Distributed environment is permitted partially and local supervisory control in not allowed $y1 = y2 = y3 = 0$, $V = 0$	1	0	0	0	0	$u3 - \underline{v3} - c2$	$u2 - \underline{v2} - b2$	$u1 - \underline{v1} - a2$

The supervisory control unit of the processing control fuzzy DEV system was tested according to the design description given in Table 1 using MATLAB simulation.

V. PRELOADED MEMORY BASED FUZZY LOGIC TIME CONTROL PROCESSING SYSTEM

Two types of design strategies are discussed. One strategy is regarding the use of preloaded memories, about the system information in digital form, whereas another strategy is more simple and based on the design algorithm for a fuzzy system using comparisons and less, hardware burden and time delay. The proposed industrial processing system has three inputs: volume, quality and select item. Therefore, fuzzy control system consists of three fuzzifiers, one for each input variables as shown in fig. 7.

Each fuzzifier consists of: a memory module of fuzzy sets input memory, an incrementer, and an inverter.

The binary coded input information of each input variable provides the address to its memory module. Each location of the memory consists of 8-bits information; 3-bits for the sequence number of the first fuzzy set and 5-bits for the membership value corresponding to the input variable value. The memory module maintains the preloaded information for the every value of input variable. An incrementer increases the sequence number adding one in the value of sequence number of 3- bits information.

An inverter in the fuzzifier is used to invert the membership function value to obtain the second fuzzy set information. Each fuzzifier gives the two active fuzzy sets information because each value of input variable lies in a region where it has two mapping values information of the two membership functions in that region.

The first active fuzzy set information consists of 3-bits function sequence number and 5-bits for the mapping value of the membership function in that region, and the second active fuzzy set information consisting of; 3-bits for function sequence number, and 5-bits for the mapping value of the next function in the same region.

The fuzzification of the three input variables gives the six active fuzzy sets information.

As shown in fig.7, active rule selector consists of three 4x2 multiplexers. These multiplexers select, three active fuzzy sets information, in a sequence maintained by an address counter.

The 3- bits sequence numbers information for each input is given to rule memory.

In this system, six output variables are used, therefore, six preloaded ROMs; rule memories are used.

The output of each rule memory is the rule consequent, which gives the singleton value of the output function related with the particular combination of membership functions sequence number corresponding to the values of input variables.

The degree of applicability provides the Min-AND value from the three mapping values of the three selected fuzzy sets.

Six defuzzifiers are used because in this system six output variables are used. Each defuzzifier consists of: one multiplier; to multiply the singleton value of a rule with the degree of applicability, two accumulators; one to take the sum of the eight products values for the eight rules applied, and another to take the sum of eight values of degree of applicability for the eight rules. Each defuzzifier works with the mathematics of centroid method [1].

This memory based fuzzy time control processing system is complex and needs more hardware, time delay involved in memory addressing and preloading. A simple design base algorithm, discussed in the next section reduces the complexity, time delay and memory preloading burden.

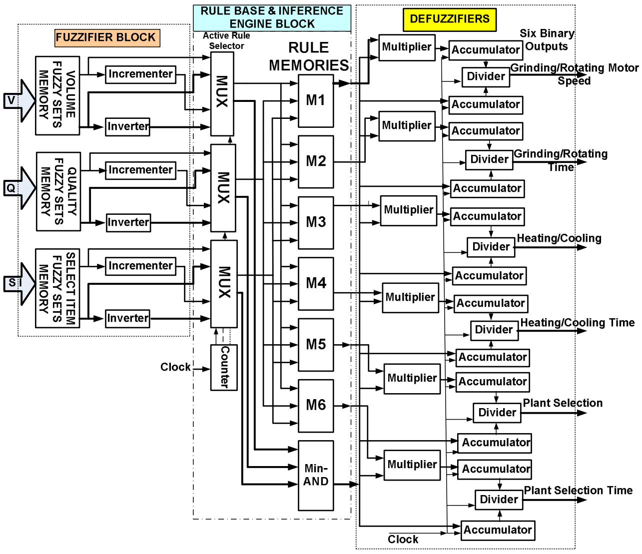


Fig. 7 Fuzzy time control processing system (ROM-Memory based arrangement)

VI. SIMPLIFIED DESIGN ALGORITHM OF FUZZY LOGIC TIME CONTROL DEV INDUSTRIAL PROCESSING SYSTEM

This simplified design algorithm is used to design the fuzzifier, inference engine, rule base and defuzzifier for the industrial processing system according to the control strategy of the processing plant to achieve the quantity and quality of the selected product.

A. Fuzzifier

The set points of fuzzifiers use the information of three input variables, "Volume", Quality" and, "Select item". Their membership functions, range and occupied region description are shown in Table I.

TABLE II
INPUT VARIABLES MEMBERSHIP FUNCTIONS AND RANGES

Range of MF	Volume	Volume MF	Quality	Quality MF	Item No Selection	Item No. Select. MF	Region Occupied
0-20	Very small	f_1 [1]	Very low	f_2 [1]	Very low	f_3 [1]	1
0-40	Just small	f_1 [2]	Just low	f_2 [2]	Just low	f_3 [2]	1,2
20-60	Just medium	f_1 [3]	Just medium	f_2 [3]	Just medium	f_3 [3]	2,3
40-80	Above medium	f_1 [4]	Above medium	f_2 [4]	Above medium	f_3 [4]	3,4
60-100	Just large	f_1 [5]	Just high	f_2 [5]	Just high	f_3 [5]	4,5
80-100	Very large	f_1 [6]	Very high	f_2 [6]	Very high	f_3 [6]	5

For each input variable, six membership functions are used as shown in Fig.8, Fig.9 and in Fig.10.

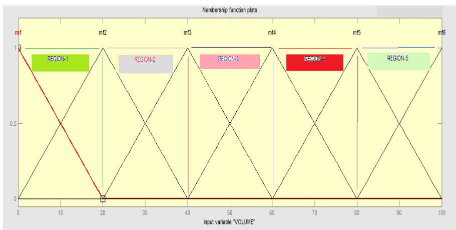


Fig. 8 Plot of membership functions for input variable, "VOLUME"

The six membership functions, f_1 [1], f_1 [2], f_1 [3], f_1 [4], f_1 [5] and f_1 [6] are used to show the various ranges of input fuzzy variable "VOLUME" in a plot consisting of five regions as shown in Fig.8.
The six membership functions, f_2 [1], f_2 [2], f_2 [3], f_2 [4], f_2 [5] and f_2 [6] are used to show the various ranges of input fuzzy variable "QUALITY" in Fig.9. This plot also consists of five regions.

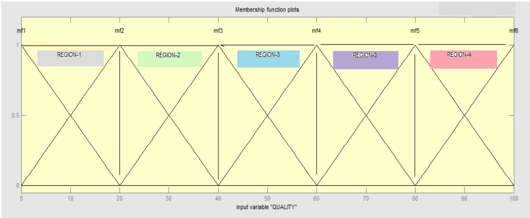


Fig. 9 Plot of membership functions for input variable, "QUALITY"

f_3 [1], f_3 [2], f_3 [3], f_3 [4],) f_3 [5] and f_3 [6] are the membership functions used to show the various ranges of input fuzzy variable "ITEM SELECTION" in a plot consisting of five regions in Fig.10.

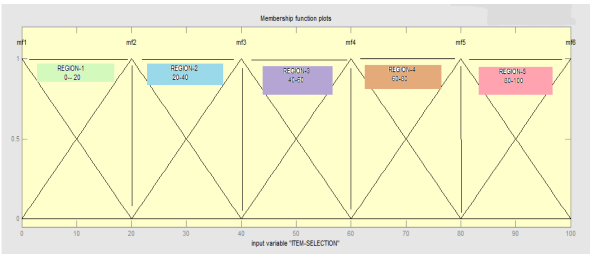


Fig. 10 Plot of membership functions for input variable, "ITEM SELECT"

The numbers of membership functions and the range values for each variable may be taken different according to the need. In this system, these are taken same for simplification.

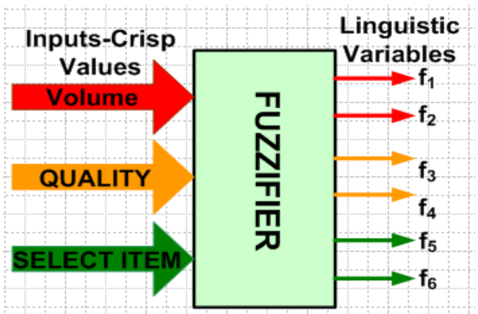


Fig. 11 Fuzzifier Block

A fuzzifier is used to convert the input crisp value into the linguistic fuzzy values. The output of fuzzier gives the linguistic values of fuzzy set. For the three input variables, three fuzzifiers are used. Each fuzzifier consists of : a multiplier: which converts the input voltage range 0-5volt into the crisp value 0-100 by multiplying the input with 20, comparators; used to decide the region occupied by the input variable value, subtractors; used to find the difference of crisp value from the end value of each region, multiplexer; using the address information from the region selection and inputs from the five subtractors, multiplex the five values because this system is designed for the five predefined regions, divider; used to divide the difference value in each selected region by 20 to find the mapping value of membership function with the input variable value in that region, a second fuzzy set subtractor; used to find the active value of the second fuzzy set by subtracting the first active fuzzy set value from 1 as shown in Table III.

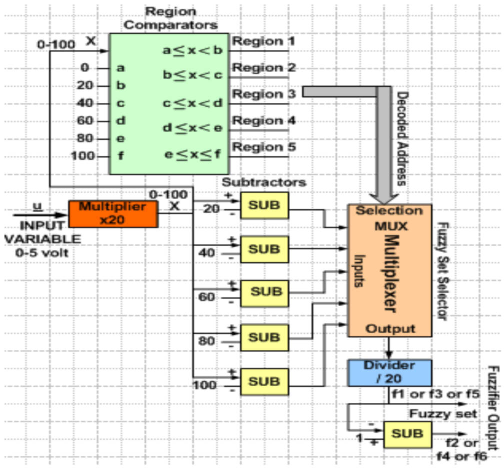


Fig. 12 Fuzzifier Design

TABLE III
FUZZIFIERS RESULTS

Input variables	Input voltage u	Values x=20u	Region selection	Fuzzy set calculation
Volume	3.25volt	65	60≤x<80 Region 4	f_1=80-65/20=0.75 f_2=1-f_1=1-0.75=0.25
Quality	4.24volt	85	80≤x<100 Region 5	f_3=100-85/20=0.75 f_4=1-f_3=1-0.75=0.25
Item selection	1.8volt	36	20≤x<40 Region 2	f_5=40-36/20=0.2 f_6=1-f_5=1-0.2=0.8

B. Inference Engine

If any one of the input variables lies in one region and the other two lie in any one of the five regions than 72 rules are required for the complete simulation of control system, but if we need any one of the five regions for all the three variables, the number of rules for complete simulation will be exceeded to 216 rules.
Number of active rules = $\mathbf{m^n}$
where m = maximum number of overlapped fuzzy sets and n = number of inputs.
For this design, m = 6 and n = 3, so the total number of active rules are 216.
The total number of rules is equal to the product of number of functions accompanied by the input variables in their working range.
In this case only 8 rules are required for the particular values of three variables because each value of three variables in a region corresponds to mapping of two functions. The corresponding mapping values of f_1 [4], f_1 [5], f_2 [5], f_2 [6], f_3 [2] and, f_3 [3] were used to establish the 8 rules. Here f_1 [4] means the corresponding mapping value of fourth membership function for the first variable volume in its region and the similar definitions are for the others.
The inference engine consists of eight AND operators, these are not the logical ANDs but select minimum value input for

the output. This inference engine accepts six inputs from fuzzifier and applies the min-max composition to obtain the output R values. The min-max inference method uses min-AND operation between the three inputs [2]. Fig.13 shows this type of inference process

$R1= f_2 \wedge f_4 \wedge f_6 = f_1[5] \wedge f_2[6] \wedge f_3[3] = 0.25 \wedge 0.25 \wedge 0.8 = 0.25$
$R2= f_1 \wedge f_4 \wedge f_6 = f_1[4] \wedge f_2[6] \wedge f_3[3] = 0.75 \wedge 0.25 \wedge 0.8 = 0.25$
$R3= f_2 \wedge f_3 \wedge f_6 = f_1[5] \wedge f_2[5] \wedge f_3[3] = 0.25 \wedge 0.75 \wedge 0.8 = 0.25$
$R4= f_1 \wedge f_3 \wedge f_6 = f_1[4] \wedge f_2[5] \wedge f_3[3] = 0.75 \wedge 0.75 \wedge 0.8 = 0.75$
$R5= f_2 \wedge f_4 \wedge f_5 = f_1[5] \wedge f_2[6] \wedge f_3[2] = 0.25 \wedge 0.25 \wedge 0.2 = 0.2$
$R6= f_1 \wedge f_4 \wedge f_5 = f_1[4] \wedge f_2[6] \wedge f_3[2] = 0.75 \wedge 0.25 \wedge 0.2 = 0.2$
$R7= f_2 \wedge f_3 \wedge f_5 = f_1[5] \wedge f_2[5] \wedge f_3[2] = 0.25 \wedge 0.75 \wedge 0.2 = 0.2$
$R8= f_1 \wedge f_3 \wedge f_5 = f_1[4] \wedge f_2[5] \wedge f_3[2] = 0.75 \wedge 0.75 \wedge 0.2 = 0.2$

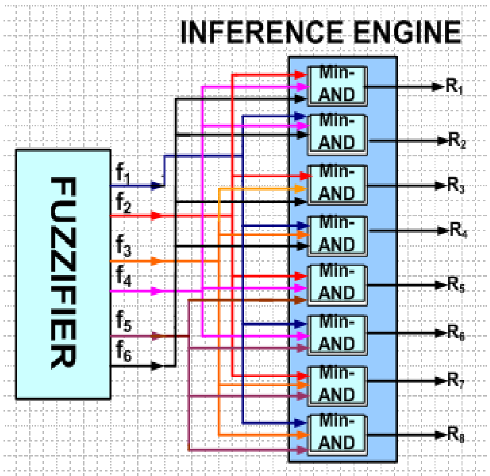


Fig. 13 Block Diagram of Inference Process

C. Rule Selector.

The rule selector receives three crisp values of volume, quality and item-select. It provides singleton values of output functions under algorithm rules applied on design model. For three variables, eight rules are required to find the corresponding singleton values S1, S2, S3, S4, S5, S6, S7, and S8 for each variable according to the division of regions. These rules are listed in Table IV

TABLE IV
ILLUSTRATION OF RULES APPLIED ON DESIGN MODEL

Rule NO	Volume	Quality	Item Select	Grinding/ Rotating Speed	Grin. / Rot. Time	Temp.	Temp. Time	Plant Select	Plant Select. Time	Singleton Values
1	J. Large	V.H	B.M	V. Fast	Long	V.H	Long	M	V. L	S1
2	J. Large	V.H	J.L	V. Fast	Long	V.H	Long	Low	V. L	S2
3	J. Large	J.H	B.M	Fast	Long	H	Long	M	Long	S3
4	J. Large	J.H	J.L	Fast	Long	H	Long	Low	Long	S4
5	A.M	V.H	B.M	V. Fast	M	V.H	M	M	V. L	S5
6	A.M	V.H	J.L	V. Fast	M	V.H	M	Low	V. L	S6
7	A.M	J.H	B.M	Fast	M	H	M	M	Long	S7
8	A.M	J.H	J.L	Fast	M	H	M	Low	Long	S8

The rule base takes in three crisp input values, divides the universe of discourse into regions with each region containing two fuzzy variables, fires the rules, and gives the output singleton values corresponding to each output variable [2]. Here we have three input variables, so we need three region detectors and decoders.

The rule base consists of: three region detectors and decoders, comparators and lookup tables LUTs to give the rule base information to the defuzzifiers. Fig.14 shows the main block of rule base and Fig.15 shows the arrangement of one region detector and decoder.

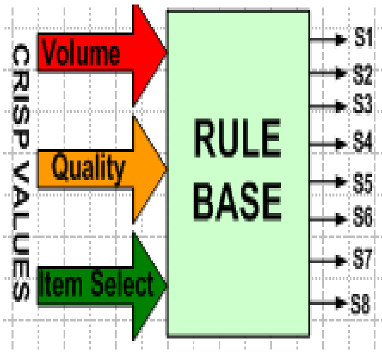


Fig. 14 Rule base block diagram

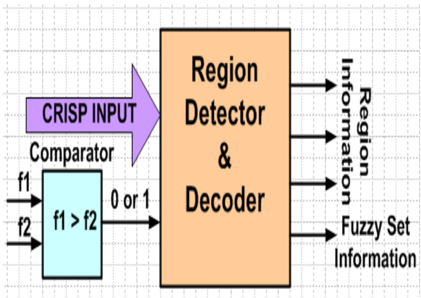


Fig. 15 Region detector and decoder

The region detector and decoder provide the information about region occupied by the variable and the fuzzy set.

The comparators and LUTs block shown in Fig.16, takes in the information from three region detectors and decoders, and provides the rule based information to the defuzzifiers.

The defuzzifiers work mathematically with these information and give the crisp value for each output variable according to the expression $\sum$ **Si * Ri** / $\sum$ **Ri.**

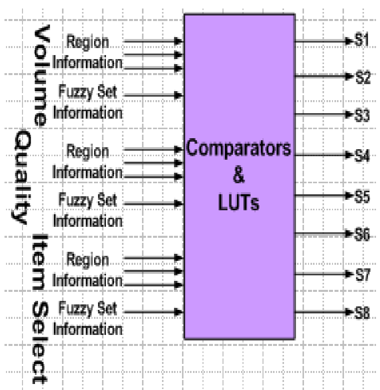


Fig. 16 Block of comparators and LUTs

D. Deffuzifier

This system consists of six defuzzifiers: three are used to control the actuators; grinding/rotating motor, heating/cooling system, and plant selection, and three defuzzifiers are used to maintain the time constraint for power providing to the actuators. This time limit makes the system more efficient and versatile to save the time, energy and engagement in the delay response of feed back circuit. The membership functions of the six output variables are shown in fig.17 to fig.22, and the detail of each plot is given in Table V.

TABLE V
OUTPUT VARIABLES MEMBERSHIP FUNCTIONS

MFs	Range	Grind/ Rot. Motor Speed	Grind/ Rot. Time	Heating/ Cooling Temp.	Heating/ Cooling Time	Plant Selection	Plant Processing Time
MF1	0-5	STOP	NO	STOP	NO	NO	NO
MF2	0-50	LOW	SMALL	LOW	SMALL	LOW	SMALL
MF3	40-60	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM
MF4	50-90	FAST	LONG	HIGH	LONG	HIGH	LONG
MF5	70-100	V. FAST	V. LONG	V. HIGH	V. LONG	V. HIGH	V. LONG

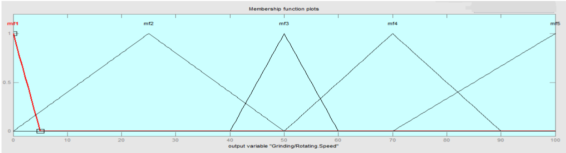


Fig.17 Plot of membership functions for output variable, “Grinding/Rotating Speed”

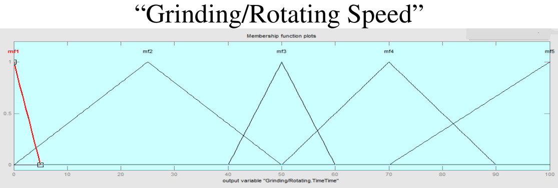


Fig. 18 Plot of membership functions for output variable, “Grinding/Rotating Time”

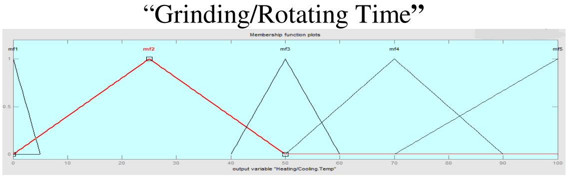


Fig. 19 Plot of membership functions for output variable, “Heating/Cooling Temperature”

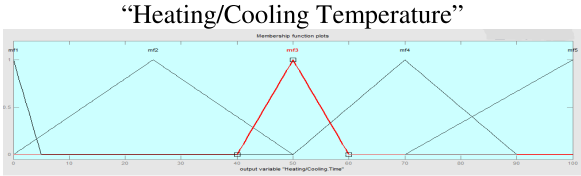


Fig. 20 Plot of membership functions for output variable, “Heating/Cooling Time”

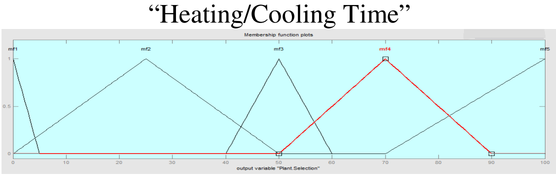


Fig. 21 Plot of membership functions for output variable, “Plant Selection”

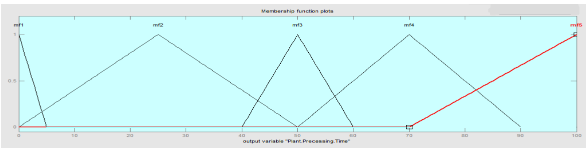


Fig. 22 Plot of membership functions for output variable, “Plant Processing Time”

The defuzzification process provides the crisp value outputs after estimating its inputs. In this system 16 inputs are given to each of eight defuzzifiers. Eight values of R1, R2, …….., R8 from the outputs of inference engine and eight values S1, S2, ………,S8 from the rule selector are shown in fig. 23

Each defuzzifier estimates the crisp value output according to the center of average (C.O.A) method using the mathematical expression , ∑ Si * Ri / ∑ Ri ,where i = 1 to 8.
Each output variable membership function plot consists of five functions with the same range values for simplification.

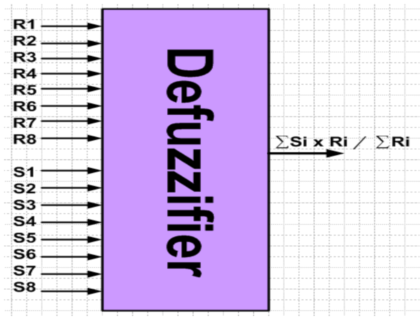


Fig. 23 Defuzzifier block

Fig. 24 shows the design arrangement of a defuzzifier. One defuzzifier consists of : one adder for ∑ Ri, eight multipliers for the product of Si * Ri, one adder for ∑ Si * Ri, and one divider for ∑ Si * Ri / ∑ Ri. Finally a defuzzifier gives the estimated crisp value output.

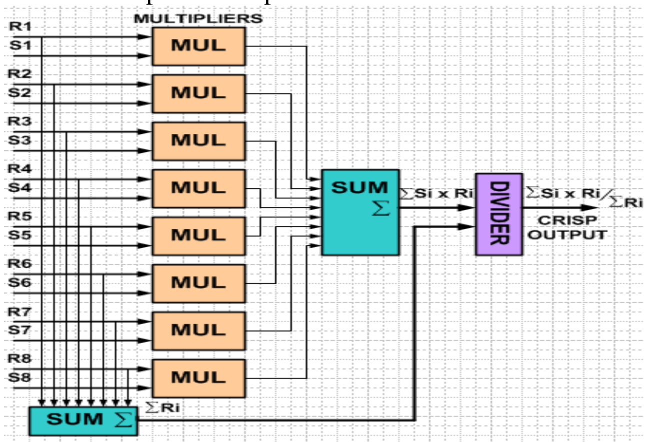


Fig. 24 Defuzzifier design

VII. RESULTS DISCUSSION

According to the results of inference engine
∑Ri=R1+R2+R3+………….+R8
= 0.25+0.25+0.25+0.75+0.2+0.2+0.2+0.2=2.3
Using mathematical expression ∑ **Si * Ri** / ∑ **Ri** the crisp values for output variables were determined and the results were found according to the MATLAB simulation as shown in Fig.25. These results are compared in Table VII and found correct according to the design model.
MATLAB simulation was adapted according to the arrangement of membership functions for eight rules as given in Table VI.

TABLE VI
ARRANGEMENT OF MEMBERSHIP FUNCTIONS FOR SIMULATION

Rule NO.	Volume	Quality	Item Select	Grinding/ Rotating Speed	Grind. / Rot. Time	Temp. Heating/ Cooling	Temp. Time	Plant Select	Plant Time
1	MF5	MF6	MF3	MF5	MF4	MF5	MF4	MF3	MF5
2	MF5	MF6	MF2	MF5	MF4	MF5	MF4	MF2	MF5
3	MF5	MF5	MF3	MF4	MF4	MF4	MF4	MF3	MF4
4	MF5	MF5	MF2	MF4	MF4	MF4	MF4	MF2	MF4
5	MF4	MF6	MF3	MF5	MF3	MF5	MF3	MF3	MF5
6	MF4	MF6	MF2	MF5	MF3	MF5	MF3	MF2	MF5
7	MF4	MF5	MF3	MF4	MF3	MF4	MF3	MF3	MF4
8	MF4	MF5	MF2	MF4	MF3	MF4	MF3	MF2	MF4

ISBN: 978-988-17012-7-5

In Fig.25 the same values of input variables, Volume=65, Quality=85 and ITEM-SELECT=36 are shown.

Various values of input and output variables match the dependency scheme of the system design. The simulated values were checked using MATLAB-Rule viewer as shown in fig. 25.

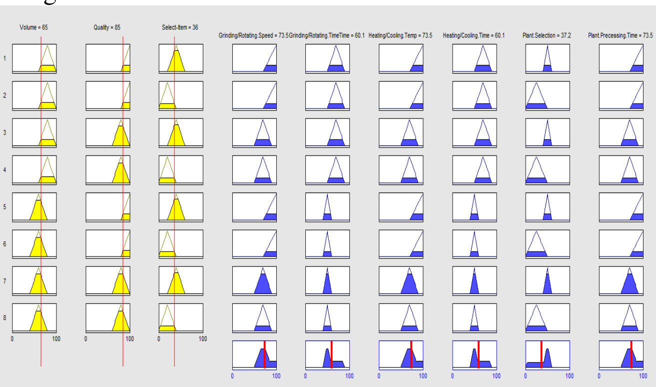


Fig. 25 MATLAB- Rule viewer and simulation result for the industrial processing fuzzy time control system

The correctness of results shows the credibility of the simplified design work for processing system using fuzzy time control system.

TABLE VII
COMPARISON OF SIMULATED AND CALCULATED RESULT

Result	Grinding/Rotating Motor Speed	Grind. /Rot. Time	Heating/Cooling Temperature	Heating/Cooling Time	Plant Selection	Plant Processing Time
Design values	72.8	59.7	72.8	59.7	36.7	72.8
MATLAB simulation	73.5	60.1	73.5	60.1	37.2	73.5
% error	0.95	0.67	0.95	0.67	1.34	0.95

A. Simulation Graphs Discussion

This system was simulated for the given range of input variables. The given value of: Volume = 65 lies in region 4 of the range 40-60, Quality = 85 lies in region 5 of the range 80-100, and Item Selection = 36 lies in region 2 of the range 20-40. The eight rules were applied for MATLAB simulation according to this range scheme.

In this design model, grinding/rotating time depends upon the selected value of volume, grinding/rotating speed and plant processing time depend on the value of quality, and the type or the number of plant selection depends upon input variable item select.

The simulated and calculated results are according to the dependency scheme.

Fig. 26(a) shows that the grinding/rotating motor speed is inversely proportional to quality and it does not depend upon the volume.

Fig. 26(b) represents that the grinding/ rotating time is directly proportional to the design range of volume and it does not depend upon the selected value of quality.

Fig. 26(c) supports the design view and shows that the heating/cooling temperature is directly proportional to quality and it does not depend upon volume.

Fig. 26 (d) shows that heating/cooling time is directly proportional to volume and it does not depend on quality.

Fig. 26(e) shows that plant selection does not depend upon volume and quality.

Fig. 26 (f) explains that plant processing time depends upon the selected value of quality input and it does not depend on volume of the product

Fig. 26 (g) shows the grinding/rotating motor speed does not depend upon the values of volume and select item.

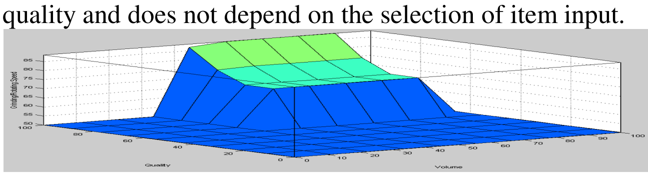
Fig. 26 (h) supports the design and shows that plant processing time does not depend upon select item input but it is directly proportional to the needed value of quality.

Fig. 26(i) shows that the selection of a plant depends on select item but it does not depend upon quality.

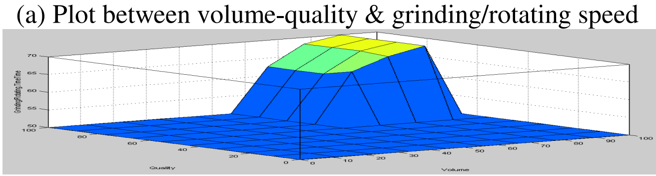
Fig. 26(j) explains that heating/cooling temperature is directly proportional to the quality and it does not depend on select item.

Fig. 26(k) shows that grinding/rotating time does not depend upon the quality and item select inputs.

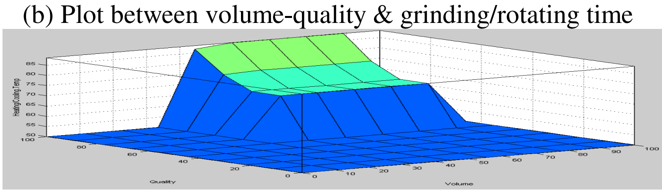
Fig. 26(l) supports the design view and shows that grinding/rotating motor speed is directly proportional to the quality and does not depend on the selection of item input.



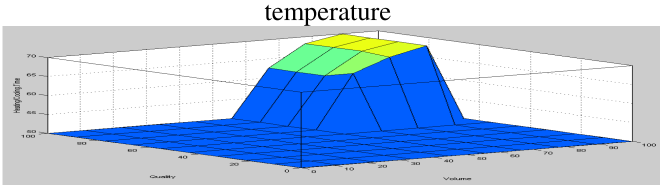
(a) Plot between volume-quality & grinding/rotating speed



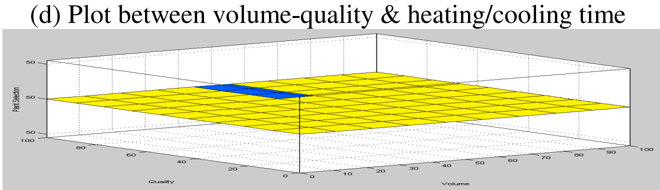
(b) Plot between volume-quality & grinding/rotating time



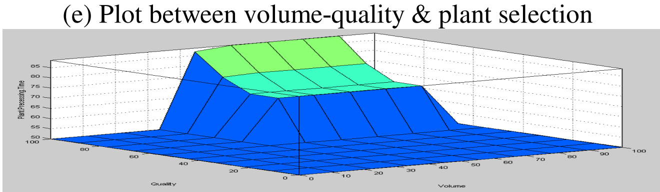
© Plot between volume-quality & heating/cooling temperature



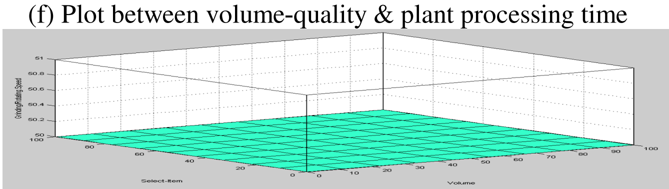
(d) Plot between volume-quality & heating/cooling time



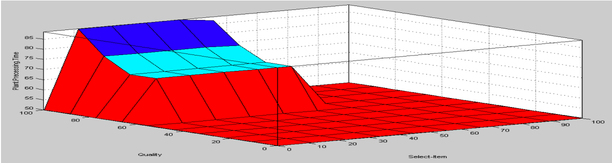
(e) Plot between volume-quality & plant selection



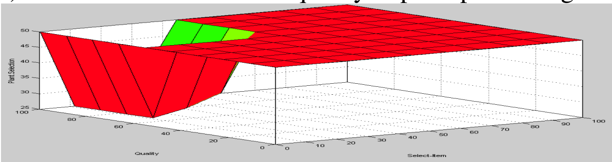
(f) Plot between volume-quality & plant processing time



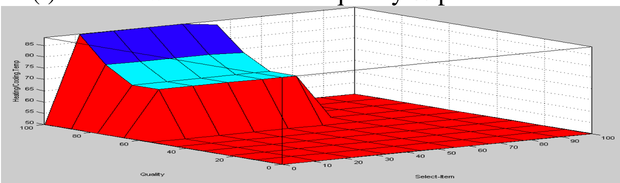
(g) Plot between volume-select item & grinding/ rotating speed



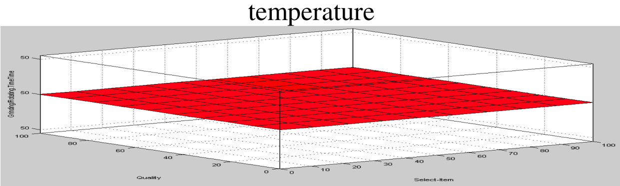
(h) Plot between select item-quality & plant processing time



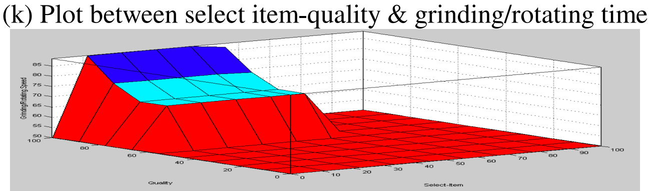
(i) Plot between select item-quality & plant selection



(j) Plot between select item-quality & heating/cooling temperature



(k) Plot between select item-quality & grinding/rotating time



(l) Plot between select item-quality & grinding/rotating speed

Fig. 26 parts (a) to (l) MATLAB simulation plots

VIII. CONCLUSION AND FUTURE WORK

Design model of multi-dimensional supervisory fuzzy logic time control processing system provided the results completely in agreement with the simulation results during the testing of various parts of the control system.

The supervisory control unit in local multi-agents based environment effectively controlled the main fuzzy logic time control processing system according to the design scheme. The algorithmic design approach makes the system efficient and completely under time control. This advanced system is better than the other systems because it provides the time control management for a system without the complexity and enhances the modes of control in local and distributed environment under multi-agents based supervisory control. The functionality of the proposed system in distributed environment is being carried out and in future it will help to design the state of the art advanced control system using high tech micro-electronics based system for the various industrial applications

REFERENCES

[1] M.Y. Hassan, Waleed F.Sharif. "Design of FPGA based PID-like Fuzzy Controller for Industrial Applications", IAENG International Journal of Computer Science, 34:2, IJCS_34-2-05.

[2] Shabiul Islam, Shakowat, "Development of a Fuzzy Logic Controller Algorithm for Air-conditioning System", ICSE2006 Proc2006 IEEE.

[3] M.Saleem Khan,"Fuzzy Time Control Modeling of Discrete Event Systems", ICIAR-51, WCECS 2008, pp.683-688.International Conference on Intelligent Automation and Robotics. U.S.A. 2008

[4] Y.Y. Chen and T.C Tsao,"A description of the dynamic behaviour of fuzzy systems", IEEE TransVol.19, July 1989, pp. 745-755

[5] W. Pedryez, J.V. de Oliveia,"Optimization of Fuzzy Models", IEEE Trans. Syst. Man, Cybern, Vol. 26, August. 1996, pp. 627-636

[6] B. P. Zeigler, P. Herbert," Theory of Modeling and Simulation, Integrating Discrete Event and Continuous Complex Dynamic Systems", IEEE Press,1994

[7] M. Sugeno, Tanaka,"Successive identification of a fuzzy model and its application to prediction of a complex system", fuzzy sets syst. Vol. 42, 1991, pp. 315-334.

[8] Rolf Isermann, "On Fuzzy Logic Applications for Automatic Control, Supervision, and Fault Diagnosis", IEEE Trans. On system, man and cybernetics, vol. 28, NO.2 March 1998.

[9] S. Germann and R. Isermann, "Nonlinear distance and cruise control for passenger cars," in Proc. IFAC-Workshop Advance Automotive Control, Ascona, Switzerland, Mar.13-17, 1995, pp. 203-208.

[10] R. Isermann, Digital control Systems. New York. Springer - Verlag. 1989, vol. 1-2.

[11] R. Isermann, K. H. Lachmann, Adaptive Control Systems. Hertfordshire, U.K, Prentice-Hall, 1992.

[12] B. M. Pfeiffer and R. Isermann, "Fuzzy logic applications in learning digital control systems," in Proc. 3rd European Congr. Fuzzy Intelligent Techniques Soft Computing. Aachen, Germany, 1995, pp. 847-852.

[13] B. M Pfeiffer and R. Isermann, "Selftuning of classical controllers with fuzzy logic," Math. Comput. Simul., vol. 37, pp. 101-110, 1994.

F.A. Mr. M. Saleem Khan is an Assistant Professor at The GC University Lahore Pakistan. He is currently availing a research fellowship at The School of Electronics & Engineering, University of Edinburgh, UK and completed his PhD thesis in the field of control systems design, simulation and analysis in local and distributed environment.

He contributed his services on various projects in the field of Advanced Electronics and Communication. His research interests include control systems design and industrial applications. He promoted a large team of Electronics researchers and organized this field in his country. Mr. Khan had also been served as a senior scientific officer in a classified defence research organization in his country.

S.B. Dr. Khaled Benkrid is a faculty member of The School of Electronics & Engineering, University of Edinburgh, UK.

He is HEA fellow, IEEE senior member, MACM/SIGDA.

His research interests are: Custom hardware design, Electronic design automation, construction of optimized compilers, image and video processing and high performance parallel computing. Mr. Benkrid is supervising the research scholars in the fields of advanced Electronics.