www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 6 June, 2013 Page No. 1739-1754

A LITERAL SYNTHESIS ALGORITHM FOR HIGH PROCESS REGULATION IN ICOFLS

*** Okafor KC, Udeze CC, Okafor CM**

Electronics Development Institute, Awka, National Agency for Science and Engineering Infrastructure, NASENI Federal Ministry of Science and Technology, Nigeria.

arissyncline@yahoo.com,

Abstract

This paper presents an implementation algorithm for Intelligent Changeover Fuzzy Logic Switching System (ICOFLS) for domestic load management. The algorithm achieves high quality regulation through utilization of fuzzy logic controller in self-managing the three entities viz: Phase lines, Generator system and Inverter system. The MATLAB Simulink fuzzy logic blockset was used in this research for high process regulation. With mamdani fuzzy inference structure for the system algorithm developed, the fuzzy plots were generated for ICOFLS. In this context, this paper proposes a fuzzy logic control (FLC) Altera Stratix IV GX FPGA which has been designed and fabricated with 0.8μm CMOS technology for process real-time control for the ICOFLS. The high regulation factor, simplicity, flexibility and adaptability of FLC FPGA makes ICOFLS dual knowledge based viz: inference control rules implement and output scale regulation. The synthesis algorithm and experimental results of ICOFLS model show that the system demonstrates resilence, stability, cost effectiveness and environmental sustainability.

[Keywords:ICOFLS,load management,MATLAB,Simulink,Mamdani,VLSI,Regulation]

INTRODUCTION

Power instability in developing countries have continued to create a need for automation of electrical power generation or alternative sources of power to back up the utility supply as observed in literature. The original philosophy of ICOFLS for domestic load management is based on the importance of fuzzy logic technique compared with other power methodologies in the design of changeover systems. We began this work in [1] and now presents a synthesis algorithm as well as the VLSI-FPGA implementation framework for the system in this research. According to [2], there is a necessity to reach advance control technologies that is capable of :

i. Managing uncertainty and expert knowlegde

ii. Accomodating significant changes in the plant and its environment

iii. Incoorperating techniques for learning either uncertain information, or a changing environment, and methods of combining existing knowlegde with a learning process.

The traditional approach to building intelligent changeover systems requires a prior model of the system. The quality of the model, that is, loss of precision from linearization and/or uncertainties in the system's parameters negatively influences the quality of the resulting control [3]. At the same time, methods of soft computing such as fuzzy logic possess non-linear mapping capabilities, do not require an analytical model and can deal with uncertainties in the system's parameters [3]. Hence, using fuzzy logic approach to handle the intelligent power management will justify [2] . Although fuzzy logic deals with imprecise information, the information is processed in sound mathematical theory [4].

Control systems can be implemented with either logic controllers , linear–feedback controller or fuzzy logic controllers. Proportional integral derivative (PID) controller is used as an industrial process controller, but it offers a major constraint in the selection of controller gains. In case of ICOFLS temperature regulation, it may not produce satisfactory results when used as a temperature process controller because the temperature process has the characteristics of non-linearity, disturbances-large inertia and time variations.

The work in [5] presents the design and construction of an automatic phase change-over switch that switches electrical power supply from public supply to generator in the event of a power outage or insufficient voltage. The system uses an electronic control circuit involving integrated circuits, transistor and electromechanical devices. A similar discusson in [6] explained a cost effective approach to implementing a change over system. In the work, digital integrated circuits and microcontroller were used to reduce the component count as well as improve the speed of the system. The system also has some desirable features like liquid crystal display (LCD) which makes the system user friendly, an alarm system for indicating generator failure, automatic phase selector for selecting most appropriate phase, over-voltage and under-voltage level monitoring. Figure 1 shows the expanded circuit diagram of microcontroller based automatic change over with solid state relays.

A representative sample of fuzzy based system implementations was studied in [7,8,9,10,11,12,13]. These focused on fuzzy logic controller as a better option than the PID and conventional logic for control systems implementations.

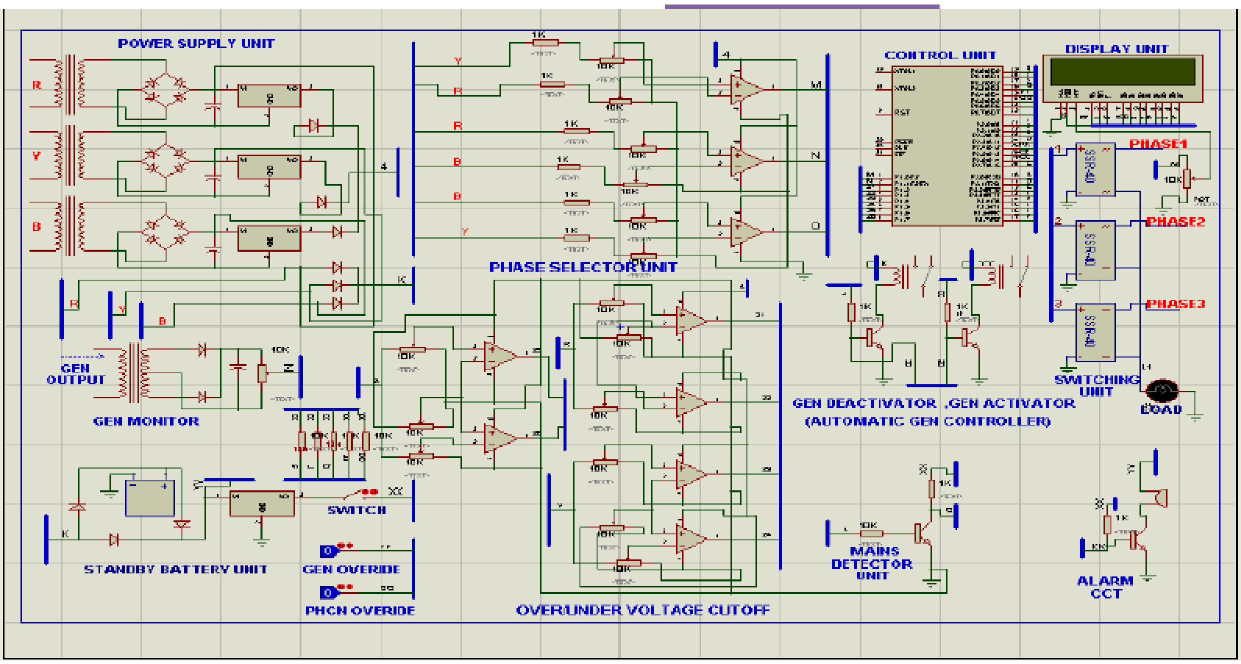

Figure 1: Expanded schematics of microcontroller based automatic change over [6]

However, following the limitations of errors resulting from imprecision, lack of parallel processing, high latency and poor resilence in [5], and [6], this research investigates futher into finding the an optimal method to achieveing high quality regualtion in ICOFLS model. In this context, by introducing fuzzy logic synthesis process algorithm and setting up the optimum values for the controller gains, a speed high ICOFLS model is shown to exhibit high performance characteristics.With its high regulation index, the model can control loads over a wide range with appreciable accuracy using fuzzy logic controller in vector control .

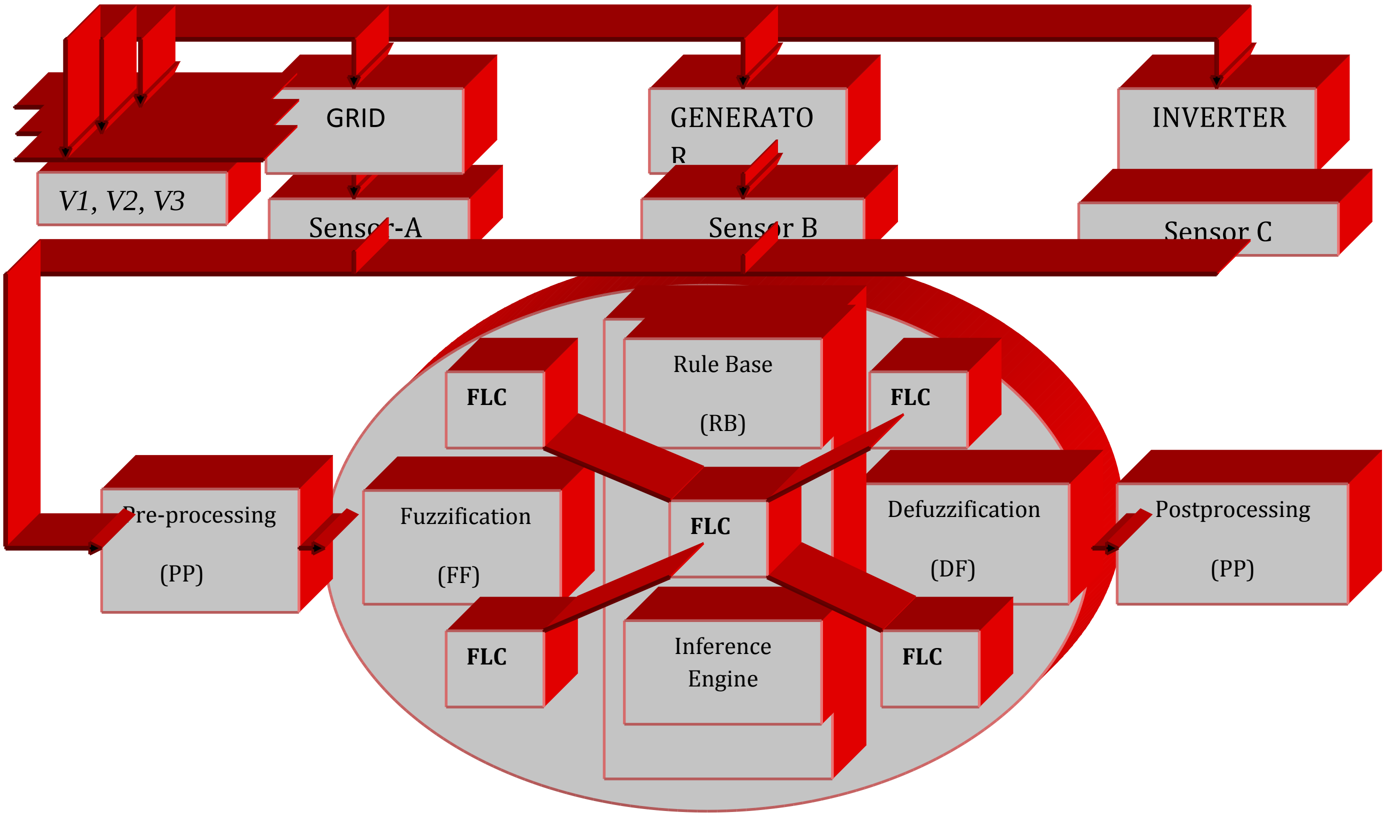

Figure 2: ICOFLS Architecture

With the vision of a developing, adaptive, and self-reconfigurable ICOFLS, firstly the system architecture is shown in Figure 2. The model consists mainly of the fuzzy logic controller block and three smart sensor interfaces, (Nepa, Generator and Inverter) The fuzzy logic controller block contains a reference to a fuzzy logic inference system. The inference system has three linguistic variables for each plant which are the two inputs (error signal and error derivative) and the output (control signal). The fuzzy logic inference system for the fuzzy proportional-derivative controller contains a set of fuzzy logic rules that define the behavior of the system in relation between the error signal, error derivative signal and the control signal of the controller. The first input to the fuzzy logic inference system is the error signal which is the difference between the desired process variable parameter and the actual process variable parameter. The error derivative signal is achieved by differentiating the error signal before passing it to the fuzzy logic controller block. Since the fuzzy logic controller block expects three inputs, a multiplexer is used to combine the error signal and the error derivative signals as input into the block.

Materials and Methods

The methodology adopted in this paper in view of the modelling, involves both literal rule base description (algorithm) Fuzzy and Process modelling with MATLAB simulink blockset design tools. This study employs a literal algorithm to design the ICOFLS fuzzy logic controller and optimizes the inference rules, membership functions while scaling gains of this controller by using parameters of the literal algorithm (LA). The resultant optimal fuzzy logic controller is used in the process and control variable units which is now implemented with fuzzy logic FPGA chip with other hardware components. The performance of the LA Optimized Fuzzy Logic controller is presented afterwards. Outlined below are the step by step modelling approaches in the ICOFLS model.

Step 1. Fuzzification of Inputs (Process variables)

Here, the inputs (the process variables) were obtained and then we determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. Fuzzification of the input amounts to either a table lookup or a function evaluation in the rule base and inference engine.

Step 2. Application of the Fuzzy Operator

After the process variables have been fuzzified, the degree to which each part of the antecedent has been satisfied for each rule is ascertained. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number will then be applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth value.

Step 3. Application of the Implication Method in the Rule Editor

Before applying the implication method, the rule's weight was taken care of in the FIS. Every rule has a *weight* (a number between 0 and 1), which is applied to the number given by the antecedent in the rule editor. From time to time the weight of one rule in the ICOFLS is varied relative to the others by changing its weight value to something other than 1. Once proper weighting has been assigned to each rule, the implication method is implemented. In this context, a consequent is a fuzzy set represented by a membership function which weights appropriately the linguistic characteristics that are attributed to it. The consequent is reshaped using a function associated with the antecedent in the rule editor.

Step 4. Aggregation of All Outputs

The ICOFLS decisions are based on the results of testing all the rules in an FIS. The rules are combined in order to make intelligent decisions. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the defuzzification phase.

Step 5. Defuzzification

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number which serves as a control signal for driving the output interface systems. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number (drive signal). However, the aggregate of a fuzzy set encompasses a range of output controlled values, and so must be defuzzified in order to control the dashboard or other output interface units.

The centroid calculation, (the ICOFLS defuzzification method) which returns the center of area under the curve in the FIS model was carefully setup. There are five built-in methods supported by the MATLAB fuzzy blockset: centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum.

This work leverages on the open and easily modifiable fuzzy inference system (FIS) structure of the Fuzzy Logic Toolbox. In this context, within the basic constraints of the ICOFLS process, this work customized the fuzzy inference process for the ICOFLS application to achieve a highly available and reliable system. Figure 3 depicts the Mamdani Inference System used to model the ICOFLS while figure 4 shows the FIS for the system.

SYSTEM ASSUMPTIONS

Before going into the modelling of ICOFLS, the following assumptions were considered:

- The modeling and design is carried out in the context of power supply ranges (220-480v)
- The utility power supply for domestic installation is 220v, 50Hz.
- The voltage level may fall to 0v and may rise to 480v (or more) but however, the frequency is steady.
- The auxiliary or emergency power supply is a generator set or inverter system.
- The utility power supply could be single phase,(two wire) or three phase,(four wire)
- The generator supply is 220v, 50Hz. It may exceed or fall below rated value, though the latter case is often and the former, seldom. The generator could be fueled with petrol or gasoline.
- Voltage regulation viz no voltage, under voltage and over voltage (in absolute sense) are taken into account when they exceed predefined (recommended) values in the fuzzy set.
- The model of ICOFLS will require as little human intervention and maintenance in context.

ICOFLS LITERAL ALGORITHM

For the architecture in figure 1, this work presents routine system logic presented thus:

If there is GRID
Then Check the Line voltages
If there is voltage in any of the lines
Then check line 1
If there is voltage in line 1
Then switch load

Switch ON the battery charger ………………….i

Monitor the charging level

If the full charge voltage level of the battery is reached then cut-off the battery charger from GRID

Else

Continue charging…………………………ii

Else

Check line 2

If there is voltage in line 2

Then switch load

Switch ON battery charger…………………………i

Else

Check line 3

If there is voltage in line 3

Then switch load

Switch ON battery charger…………………………i

Else

Check the oil level of the generator……………………………………………………..ii

If not OK then Display the oil tank State and sound an Alarm

Refill the oil tank

Else

Check the fuel level

If not Ok then Display the fuel tank state

Refill the fuel tank

Else

Check the water level[14]

If not Ok then Display the water level State

Refill the water tank

Else

Check systems temperature

If not Ok Then Display system fault and sound alarm

Else

Switch load to Generator

Go back and keep monitoring the line voltages

MAMDANI FUZZY INFERENCE FRAMEWRK (M-ICOFLS)

In implementing the literal algorithm, this paper adopts fuzzy Logic Toolbox in MATLAB software designed to work in Simulink environment. After creating the fuzzy systems using the GUI tools, the system is then ready to be embedded directly into a simulation. The FIS Editor displays general information about a fuzzy inference system. Mamdani-type inference, as we have defined it for the Fuzzy Logic toolbox, expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification.

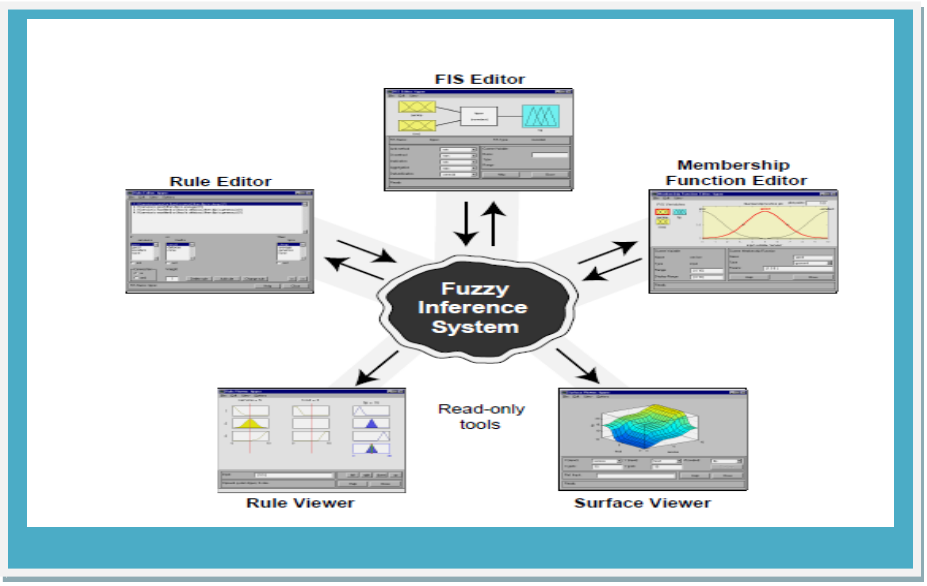

Figure 3: Mamdani Inference System [14]

In the Fuzzy Logic Toolbox, there are five parts of the fuzzy inference process viz:

1. Fuzzification of the input variables
2. Application of the fuzzy operator (AND or OR) in the antecedent
3. Implication from the antecedent to the consequent,
4. Aggregation of the consequents across the rules.
5. Defuzzification.

Table 3.1 shows the look up table for the linguistic variable - phase voltages. Table 3.2 shows Table Based Controller (GRID)

Table 3.1: Linguistic variables for Phase Voltages (PV)

Change in Phase Voltage

Phase Voltages		***-240***	***-200***	***0***	***200***	***240***
	-240	*-480*	*-440*	*-240*	*-40*	*0*
	-200	*-440*	*-400*	*-20*	*0*	*40*
	0	*-240*	*-200*	*0*	*200*	*240*
	-200	*-40*	*0*	*200*	*400*	*440*
	240	*0*	*40*	*240*	*440*	*480*

Table 3.2: Table Based Controller (GRID)

S/N	***Phase Voltage (PV)***	***Change in PV***	***O/P***	***States***
1.	*Neg*	*PoS*	*Zero*	*LS*
2.	*Neg*	*Zero*	*NM*	*MS*
3.	*Neg*	*Neg*	*NH*	*Hs*
4.	*Zero*	*Pos*	*PM*	*MS*
5.	*Zero*	*Zero*	*Zero*	*LS*
6.	*Pos*	*Pos*	*PM*	*NS*
7.	*Pos*	*Zero*	*PM*	*NS*
8.	*Pos*	*Neg*	*Zero*	*LS*

A two-input, one-output, and nine-rule phase voltage rule base algorithm is presented in this section. The basic structure of this case is shown in Table 3.2. Zero, Positive (Pos), Negative (Neg), Negative high (NH), Negative Medium (NM), Positive big (PB) and positive medium (PM) are labels of fuzzy sets. Information flows from left to right, from two inputs to a single output. The parallel nature of the rules is one of the more important aspects of fuzzy logic systems. Also see appendix 1 for the general rule base conditions

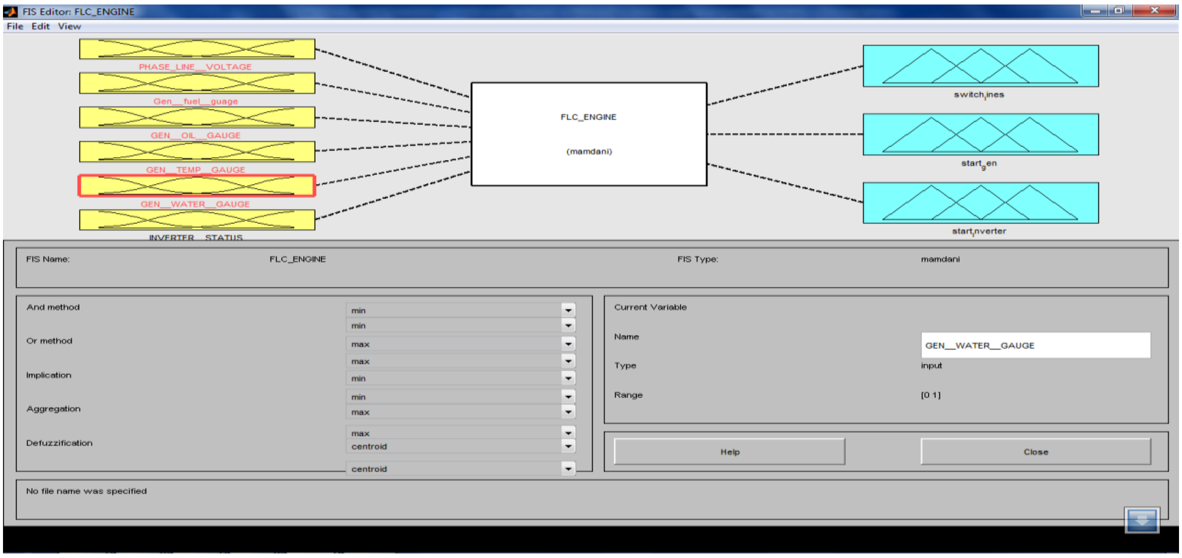

Figure 4: FIS Editor Model of ICOFLS with FLC Engine

FPGA FLOWCHART INTEGRATION

By using the reconfigurability of the FPGAs, a reconfigurable computing system [15] can reduce execution time by hardwiring the computationally intensive parts of the algorithm.
Most of the fuzzy logic applications with the physical systems require a real-time operation to interface high speed constraints [15.] The simple way to implement figure 1 is to realize it as a software program on general purpose computers, the flowchart in figure 5 explains the functionl processes. A High density programmable logic device FPGA can be used to integrate large amounts of logic in a single IC. FPGA becomes one of the most successful of technologies for developing the systems which require a real time operation. Semi-custom and full-custom application specific integrated circuit (ASIC) FPGA provides flexibility and is used with tighter time-to-market schedules. Figure 5 depicts an implementation schematic capture.

To define the behavior of the FPGA, a hardware description language (HDL) with a schematic design is developed. In our context, we propose a very high speed integrated circuit hardware description language (VHDL) for schematic description. Then, using an electronic design automation tool (Warp cypress), a technology-mapped netlist is generated. The netlist is fitted to the actual FPGA architecture using a process called place-and-route, during the compilation. The synthesis map is validated alongside with the place and route results via timing analysis, simulation, and other verification methodologies. Once the design and validation process is complete, the binary file (jedec) is generated for configuring the FPGA. This file is transferred to the FPGA via a serial interface (JTAG) or to an external memory device like an EEPROM [16]. In this work Altera Stratix IV GX FPGA is proposed for our implementation. Fuzzy dedicated circuits are characterized by [17]:

- The number of inputs and outputs.
- The number and shapes of membership functions.
- Inference techniques including operators, consequences, and size of the premises.
- Defuzzification method.
- The number of fuzzy logic inferences per second, FLIPS.
- Physical size.
- Power consumption.
- Software available to support the design.

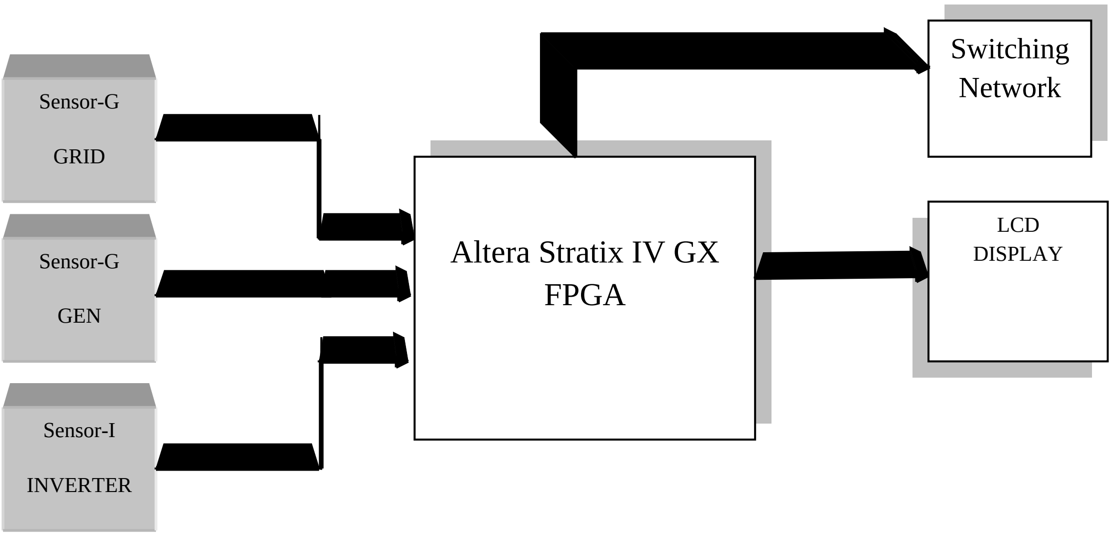

Figure 5: A VLSI implementation schematic capture

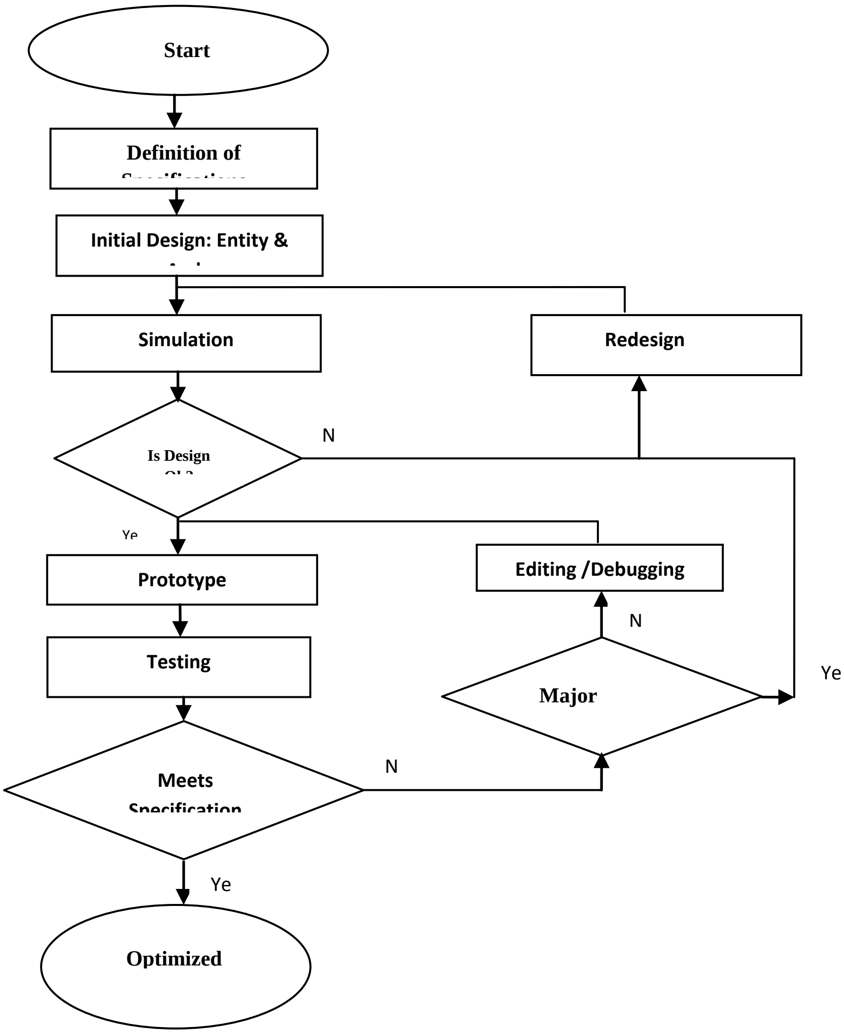

Figure 6: A Development FPGA Design Process Framework

RESULTS AND DISCUSSION

The MATLAB SimEvent fuzzy blockset version 7.4.0 is used for FLC model results. Also, Figure 6.1- 6.18: shows the MATLAB fuzzy plots of the system under analysis. In this work, the MATLAB rule viewer and surface viewer were used for the result analysis. The Rule Viewer is a MATLAB based display of the fuzzy inference diagram shown in figure 3.1. Used as a diagnostic, it shows which rules are active and how individual membership function shapes are influencing the results. The Surface Viewer is used to display the dependency of one of the outputs on any one or two of the inputs—that is, it generates and plots an output surface map for the system.

Figure 6.1 shows the membership functions for the ICOFLS in this work for MATLAB and Protus ISIS. For the phase voltage, the plot shows the low voltage state (Mf_Lv), Medium voltage state (Mf_Mv), Normal voltage state (Mf_Nv), High voltage state (Mf_Hv) with a fuzzy range [0 420]. The Fuzzy inference engine uses the rule base to generate appropriate responses to be defuzzified and passed on to the output dashboard. Also, the membership functions of other process variables (fuel gauge, gen_oil, water_gauge, inverter status) were defined and represented in the rule base.

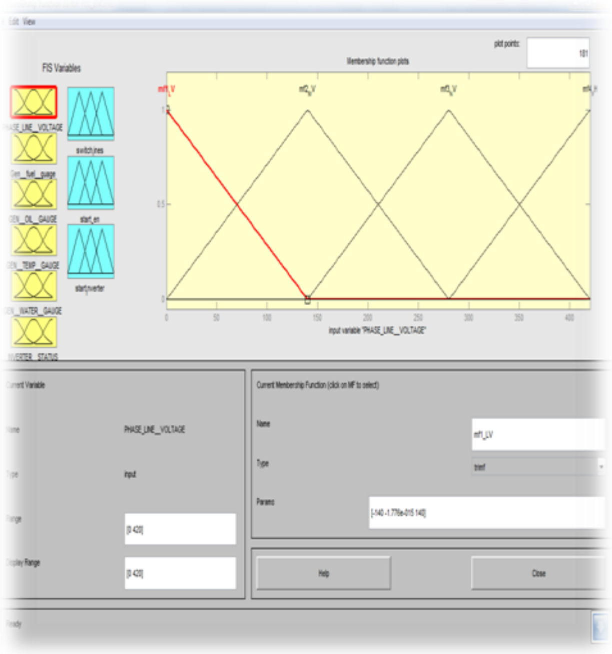

Figure 6.1: Membership function of ICOFLS with FIS variable

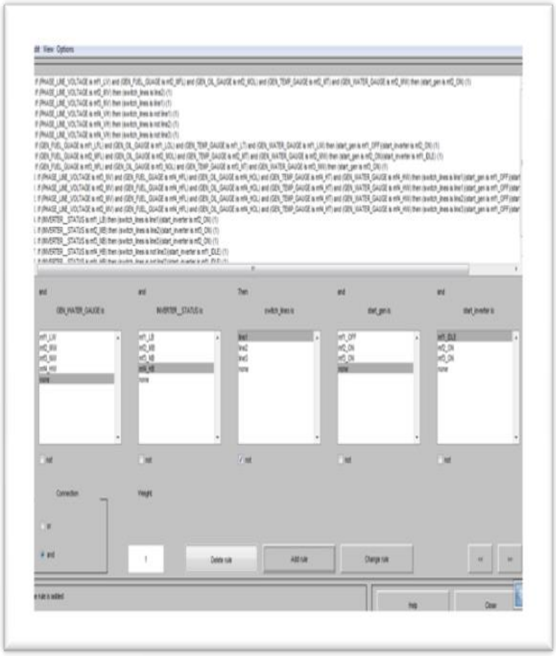

Figure 6.2: ICOFLS Rule base descriptions in Rule Editor

Figure 6.2 shows the rule base descriptions for the mamdani FIS for the ICOFLS. The idea behind fuzzy inference is to interpret the values in the input and based on some of rules in the rule editor, assigns values to the output vector. From figure 5.2, GEN_WATER_GAUGE and INVERTER_STATUS form the antecedent in this context while START_GEN and START_INVERTER are the derived consequents. For each rule, the inference engine looks up the membership values according to the condition of the rule. The resulting fuzzy set must be converted to a number that can be sent to the process as a control signal (crisp control signal). The fuzzification block converts each piece of input data to degrees of membership by a lookup in one or several membership functions. Thus, it matches the input data with the conditions of the rules to determine how well the condition of each rule matches that particular input instance.

Figure 6.3 shows the rule view plots. The Rule Viewer displays the roadmap of the whole fuzzy inference process. It is based on the fuzzy inference diagram described in the previous section. The Rule Viewer shows one calculation at a time and in great detail. In this sense, it presents a sort of micro view of the fuzzy inference system. Figure 6.3 shows a figure window with 19 plots nested in it. The nine plots across the top of the figure represent the antecedent and consequent of the entire rule base of the ICOFLS.

Each rule is a row of plots, and each column is a variable. The rule numbers are displayed on the left of each row. The columns of plots (6) show the membership functions referenced by the antecedent, or the if-part of each rule while the column of plots (the 3 blue plots) shows the membership functions referenced by the consequent, or the then-part of each rule. The Rule Viewer allows for interpretation of the entire fuzzy inference process at once and also shows how the shape of certain membership functions influences the overall result. Because it plots every part of every rule, it can become unwieldy for particularly large systems, but, for a relatively small number of inputs and outputs, it performs well (depending on how much screen space devoted to it) with up to 30 rules and as many as 6 or 7 variables.

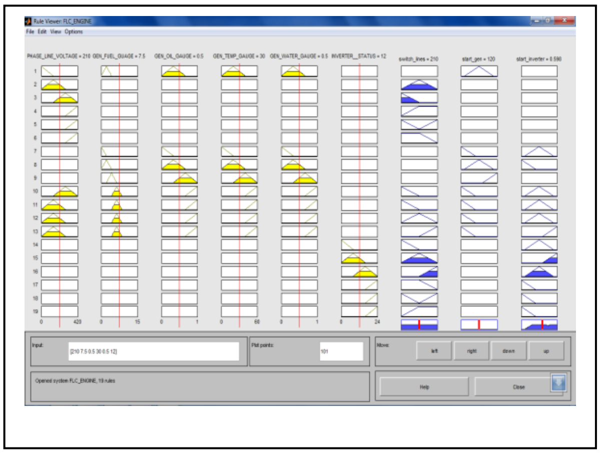

Figure 6.3: ICOFLS Rule viewer Results for the process and Control variables

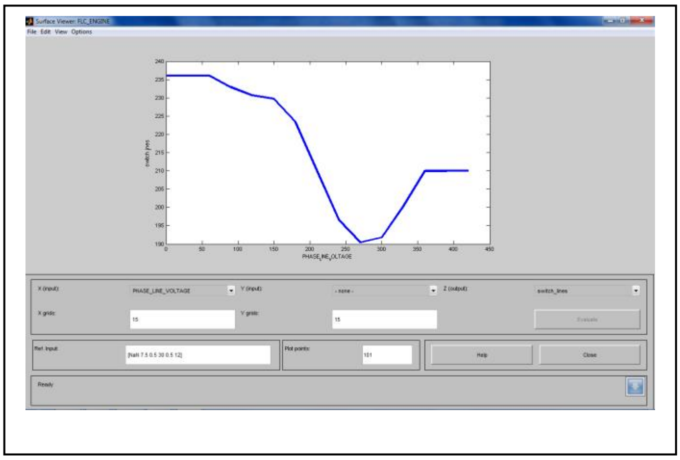

Figure 6.4: ICOFLS surface viewer plot for the phase voltage against switch lines control variable.

Figure 6.4 shows a surface view plot for the phase voltage against the switched lines. It allows the entire span of the output set based on the entire span of the input set. The phase line mapping [0 420] with the corresponding switched lines are tagged values; lines 1, line2 and line3 are tagged values of [0 240]. The phase selector is based on the input voltage state and its presence. The switch_line coordinate (Y) with a range of 0 to 240 fuzzy switching states detects the valid phase voltage membership states. From the plot, the target valid phase voltages are between 200 and 300 volts corresponding to the switch line states [0 205]. Outside these ranges, the ICOFLS bypasses the load switching. Hence, when all other conditions of the rule engine are not meet and either the phase lines satisfy the rule base, the switch_line routine manages the load satisfactorily.

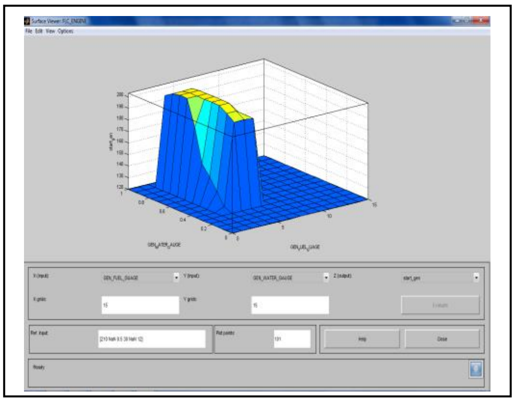

Figure 6.5: ICOFLS surface viewer plots for the process variables against start_gen control variable.

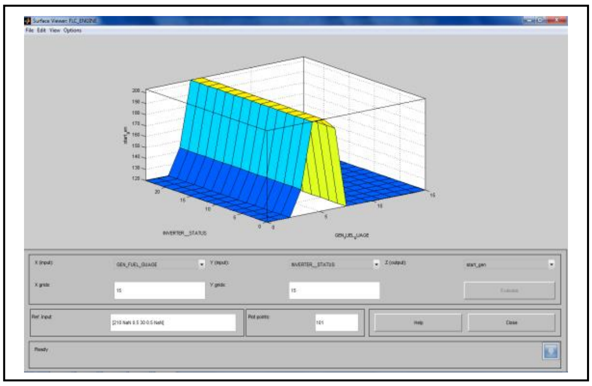

Figure 6.6: ICOFLS surface viewer results for the process fuel gauge, inverter status and Control variable start-gen.

From figure 6.5, the dependency relations for start_gen control process is between generator fuel gauge and water gauge assuming an outage in the phase lines. From the plot, surfaces in figure 5.5 show the functions approximated by the corresponding rule base. These results allow the start_gen routine to compare between the generator fuel gauge and water gauge and then powers up the terminal loads assuming a state failure in the phase lines. In this context, the start_gen stability state is at point 200 on the Y-axis of the surface diagram. At start_up phase, the inverter system starts its charging operations as specified in the algorithm in chapter 3.

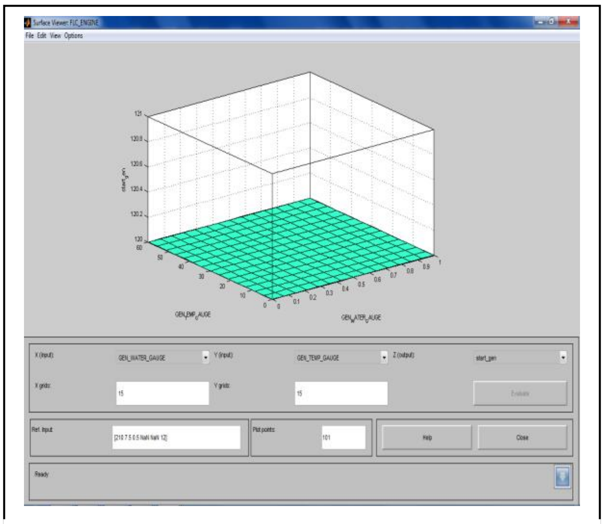

Figure 6.7: ICOFLS surface viewer plots for the process water gauge, temp gauge and Control variable start gen.

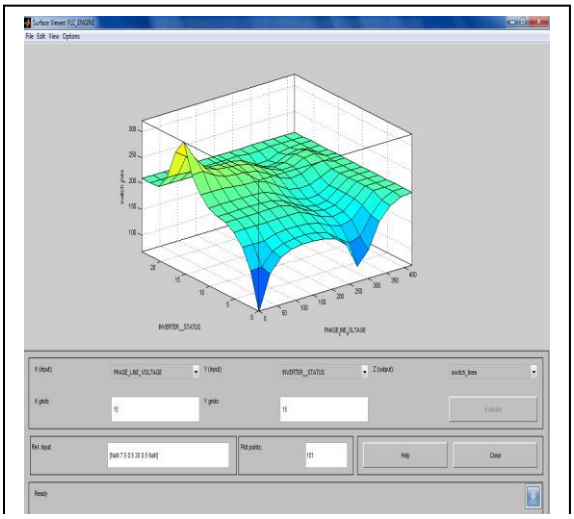

Figure 6.8: ICOFLS surface viewer plots for the process variable (Inverter Status, phase voltage) control variable (switch lines).

The plot in figure 6.7 is a ground state for the ICOFLS. The generator fuel gauge and temperature specifications in the surface diagram flags an error in the dashboard and as such remain in the idle state. A case of no phase voltage will demand load switching from the generator or the inverter on full charge, but an improper parameter level for generator will ground the system. As such this condition must be avoided to ensure continuous supply to the load system.

Figure 6.8 shows an optimized state for the loads. The presence of the phase voltage deactivates the generator system while switching the loads and the inverter system. The line states (membership values) of 0 to 300 volts are acceptable for our case and can power on the loads with no risk. The presence of a high voltage state [300-400] is abruptly ignored by the system. The rule and inference engine carries out this process control. As such, this regulation process can reliably safeguard domestic load systems or equipments. In this case also, the inverter system still charges its battery to maximum regulation charge. This is a desirable mode in the ICOFLS model.

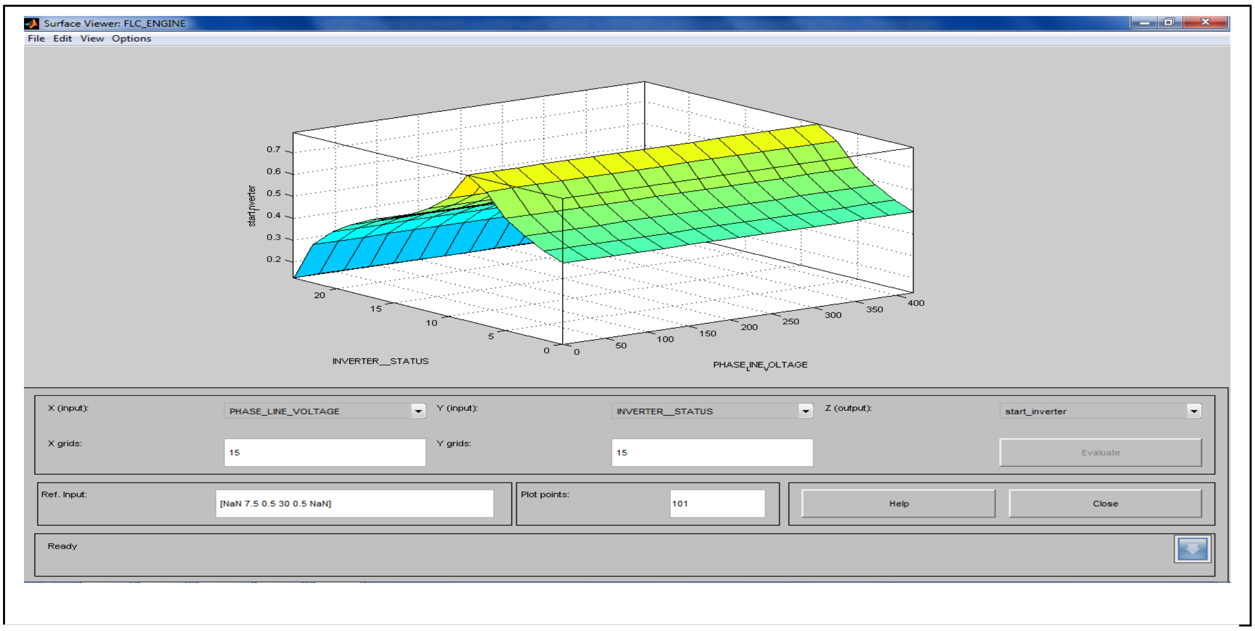

Figure 6.9: ICOFLS surface viewer plots for the process variable (Inverter Status, phase voltage) and control variable start inverter

sually, a sudden phase outage causes the start_inverter routine to fully power the loads from the inverter system. In this case, the phase voltage possibility ranges from 0 to 400 and 0 to 20 for the inverter system. The pattern explains a continuous supply to the loads at the instance of outage while neglecting the generator system and its parameters.

In all cases, the goal of the proposed model is to provide a system based on approximate reasoning that produces an estimate for the probability of switching success based on critical evaluation of the specified criteria in the rule base.

CONCLUSION

The automatic changeover switch has come a long way from manual switching to automatic switching using fuzzy logic algorithms. However, some of the already existing ones lack intelligence and users are looking forward to a smarter system. Unlike the manual and automatic changeover switch that have been in existence with their constraints ranging from the manpower to switch on and off, the time delay during which serious losses and even life is encountered, starting the generator without running the necessary checks and many other constraints. Hence this research embarked on developing a system that can switch and make efficient monitoring using fuzzy logic approach. This work has presented a hybrid fuzzy controller structure called the ICOFLS for load-power switching and control. It controls load switching using a rule based engine. The graphical plots show the reliability and its optimal performance with very low latency. The computational time of the conventional fuzzy controller can be reduced by 50 percent using the block based fuzzy controllers. From the different validation exercise, the intelligent units (attributes) of the ICOFLS such as collecting input signal from the sensors (Oil, fuel and water sensors), processing output, carrying out complex digital signal processing, displaying status of the parameters and the power source have been implemented in this work and ascertained fully working with a stability index of 95% considering the target chip for the implementation (ASIC FPGA fuzzy controller).

FUTURE WORK

In this work future research areas should include:

- An operational performance analysis of the ICOFLS model vis_a_vis fuzzy FPGA ASIC chip controllers.
- Priority analysis of load appliances and system stability under temporal and permanent over load conditions.
- Integrating radio frequency devices for tagging the phase lines for wireless load switching.

References

1. Okoyedike Ngozi Maureen, Prof A.N Nzeako, Okafor K.C, udeze C.C, "ICOFL : An Alternative Design Approach To An Intelligent Changeover System Based On Fuzzy Logic Controller For Domestic Load Management", International journal of advanced Research in Artifical intelligence,vol 1, No 5, 2012, pp16-21.

2. O.Cordon,F.Herrera,E.Herrera- Viedma,M.Lozano, " A Technical report on Genetic Algorithm and Fuzzy Logic in Control Processes", Department of computer science and Artificial intelligence, March,1995.

3. Vinod Kumar, R.R.Joshi: Hybrid Controller based Intelligent Speed Control of Induction Motor; Journal of Theoretical and Applied Information Technology ,2005.

4. B. K. Bose, "Expert systems, fuzzy logic, and neural network application in power electronics and motion control" , Proceeding of the IEEE, vol.82,Aug. 1994.

5. M.S. Ahmed, A.S. Mohammed and O.B. Agusiobo, " Development of a Single Phase Automatic Change-Over Switch" AU J.T. 10(1): Jul. 2006 68-74

6. Nwafor Chukwubuikem M., Mbonu Ekene S, Uzedhe Godwin, "A cost effective approach to implementing change over system" Academic Research International, Vol. 2, No. 2, March 2012

7. Nasri Sulaiman, Zeyad Assi Obaid, M. H. Marhaban and M. N. Hamidon, "FPGA-Based Fuzzy Logic: Design and Applications – a Review", IACSIT International Journal of Engineering and Technology Vol.1,No.5,December,2009.

8. Vinod Kumar, R.R.Joshi, Hybrid Controller based Intelligent Speed Control of Induction Motor, Journal of Theoretical and Applied Information Technology, 2005,pp 71-75.

9. Sheroz Khan, et al, "Design and Implementation of an Optimal Fuzzy Logic Controller Using Genetic Algorithm" Journal of Computer Science 4 (10): 799-806, 2008

10. Dr. I Santi Prabha1, K Durga Rao, D Siva Rama Krishna: "Fuzzy Logic Based Intelligent Controller Design for an Injection Mould Machine Process Control" International Journal of Advanced Engineering Sciences And Technologies, (IJAST) Vol No. 10, Issue No.1, 098 – 103.

11. Driankov D., Hellendoorn H. and Reinfrank M., *An Introduction to Fuzzy Control*, Springer-Verlag Berlin Heidelberg, 1996.

12. Yeh Z.M., A Systematic Method for Design of Multivariable Fuzzy Logic Control Systems, *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 6, 1999, pp. 741-752.
13. Vineet KUMAR, K.P.S. RANA and Vandna GUPTA, “Real-Time Performance Evaluation of a Fuzzy PI + Fuzzy PD Controller for Liquid-Level Process” INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS VOL. 13, NO. 2, JUNE 2008, 89-96.
14. Rahul Malhotra, Narinder Singh, Yaduvir Singh, “Fuzzy Logic Modelling, Simulation and Control: A Review; International Journal of Computer Science and technology,IJCST Vol. 1, Issue 2, December 2010.

15. B. Hutchings and M. Wirthlin, “Implementation approaches for reconfigurable logic applications,” in Proc. 5th Int. Workshop Field Programmable Logic and Applications, 1995, pp. 419–428.

16. http://en.wikipedia.org/wiki/Field-programmable_gate_array
17. A. M. Ibrahim, "Fuzzy Logic for Embedded Systems Applications", Elsevier Science, USA, 2004.

APPENDIX

RULE BASE FOR ICOFLS (FUZZY LOGIC RULES)

i. PROCESS VARIABLES

- *Voltage*
- *Water level*
- *Fuel level*

ii. CONTROL VARIABLES

- *Line 1, Line 2, line 3*
- *GRID*
- *Fuel tap*
- *Inverter*

iii. RULES FOR THE PROCESS VARIABLES

- **RULES FOR THE VOLTAGES AND THE THREE LINES**

i. UNDER VOLTAGE

IF voltage in LINE 1 is less than 160 and voltages in line 2 and 3 are within the range 160-230v

THEN connect to either line 2 or 3.

IF voltage in LINE 2 is less than 160 and voltage in lines 1 and 3 are within the range 160-230v

THEN connect to either line 1 or 3.

IF voltage in LINE 3 is less than 160 and voltage in lines 1 and 2 are within the range 160-230v

THEN connect to either line 1 or 2.

ii. NORMAL VOLTAGE

IF voltage in line 1 is within the range 160-230v THEN connect line 1 and disconnect lines 2 and 3.

IF voltage in line 1 is not within the range 160 – 230v and lines 2 or 3 are within the range

THEN Disconnect line 1 and connect to lines 2 or 3.

IF voltage in line 2 is within the range 160-230v THEN connect line 2 and disconnect lines 1 and 3.

IF voltage in line 2 is not within the range 160 – 230v and lines 1 or 3 are within the range

THEN disconnect line 2 and connect to lines 1 or 3.

IF voltage in line 3 is within the range 160-230v THEN connect line 3 and disconnect lines 1 and 2.

IF voltage in line 3 is not within the range 160-230v and lines 1 or 2 are within the range

THEN disconnect line 3 and connect to lines 1 or 2.

IF voltages from the three lines are within 160 – 230v THEN connect just one of the 3 lines and disconnect the other 2 lines and the generator.

iii. OVER VOLTAGE

IF voltage in line 1 is greater than 280v and voltage in line 2 and 3 are within 160-230v

THEN connect to either line 2 or 3

IF voltage in line 2 is greater than 280v and voltage in line 1 and 3 are within 160-230v

THEN connect to either line 1 or 3

IF voltage in line 3 is greater than 280v and voltage in line 1 and 2 are within 160-230v

THEN connect to either line 1 or 2

iv. VOLTAGE AND GENERATOR

v. NO VOLTAGE AND GENERATOR

IF there is no voltage in the three lines and Water and fuel level of the generator are ok

THEN start the generator.

IF there is no voltage in any of the three lines and fuel and water level are not ok THEN don't start the generator.

IF there is no voltage in any of the three lines and only fuel level is ok THEN don't start the generator.

IF there is no voltage in any of the three lines and only water level is ok THEN don't start the generator.

vi. UNDER VOLTAGE AND GENERATOR

IF there is under voltage in the three lines and Water and fuel level of the generator are ok

THEN start the generator.

IF there is under voltage in any of the three lines and fuel and water level are not ok THEN don't start the generator.

IF there is under voltage in any of the three lines and only fuel level is ok THEN don't start the generator.

IF there is under voltage in any of the three lines and only water level is ok THEN don't start the generator.

vii. OVER VOLTAGE AND GENERATOR

IF there is over voltage in the three lines and Water and fuel level of the generator are ok

THEN start the generator.

IF there is over voltage in any of the three lines and fuel and water level are not ok THEN don't start the generator.

IF there is over voltage in any of the three lines and only fuel level is ok THEN don't start the generator.

IF there is over voltage in any of the three lines and only water level is ok THEN don't start the generator.

IF the voltage from the water level detector is less than 4.202 THEN the water level is NOT OK

IF the voltage from the water level detector is greater than 4.202 THEN the water level is OK

IF height of the thread is equal to 15cm THEN the fuel level is NOT OK.

IF height of the thread is less than 15cm THEN the fuel level is OK.

If height of the thread is equal to zero THEN the fuel level is NOT OK

IF height (h_1) of the thread is equal to radius (15cm) THEN open the fuel tap

IF height (h_1) of the thread is equal to zero THEN close the fuel tap

IF height (h_1) of the thread is between 0 and 14.8cm THEN no change to the fuel tap

RULES FOR THE WATER LEVEL DETECTOR.

IF voltage from water level detector is less than 4.202v THEN open the water tap, sound the alarm and send to the LCD the current status.

IF voltage from water level detector is equal to 4.202v THEN no changes to the water tap and send to the LCD the current status.

IF voltage from water level detector is greater than 4.785v THEN close the water tap, sound the alarm and send to the LCD the current status.

IF voltage from the water level detector is greater than 4.202v but not up to 4.785v THEN keep the water tap open.

RULES FOR THE FUEL LEVEL DETECTOR

IF height (h_1) of the thread is equal to radius (15cm) THEN open the fuel tap

IF height (h_1) of the thread is equal to zero THEN close the fuel tap

IF height (h_1) of the thread is between 0 and 14.8cm THEN no change to the fuel tap

RULES FOR THE OUTPUT VARIABLES

IF line 1 is connected THEN disconnect lines 2 and 3

IF line 2 is connected THEN disconnect lines 1 and 3

IF line 3 is connected THEN disconnect lines 1 and 2

RULES FOR THE GENERATOR

IF water level and fuel level of the generator are OK and no voltage in the three lines THEN disconnect the three lines, start the generator and send the current power source to the LCD.

If water level and fuel level of the generator are NOT OK and no voltages in the three lines THEN do not start the generator.