CONGESTION DETECTION AND AVOIDANCE USING FUZZY SYSTEMS

MOHAMMED N. MUSTAFA

SEPTEMBER , 2012

Title of the Thesis : **Congestion Detection and Avoidance Using Fuzzy Systems**

Submitted by **MOHAMMED N.MUSTAFA**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof .Dr . Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Murat SARAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree Master of Science.

Prof. Dr. Mehmet R. Tolun
Supervisor

Examination Date : 3/9/2012

Examining Committee Members

Prof. Dr. Mehmet R. Tolun (TED Univ)

Asst. Prof. Dr. Reza HASSANPOUR (Çankaya Univ.)

Asst. Prof. Dr. fahd JARAD (Çankaya Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : **Mohammed, Mustafa**

Signature :

Date : 24/09/2012

ÖZET

TIKANIKLIK TESPİTİ VE ÖNLENMESİ

MOHAMMED NSAIF

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Danışman: Prof. Dr. Mehmet R. Tolun

2012, 86 Sayfa

Bu tezde, Tıkanıklık sorunu üzerinde çalışılmıştır. Çalışmada, tıkanıklığın nasıl tespit edileceği ve bunun nasıl önleneceği üzerinde durulmaktadır. Hizmet sağlayıcılar, ağ varlıklarının, reklam kuşağı ve Internet hizmeti ön tedarik hazırlığının rekabetçi doğasının yüksek maliyeti nedeniyle, ağlarının performansını optimize edilmesi ile ilgilenmektedirler. Bunlar, tıkanıklık popüler noktalarının azalmasına ve bütün ağ üzerinde kaynakların geliştirilmesine yol açarlar, ki bu da sırasıyla, artan oranda bir gelir birikimi sonucunu doğurmaktadır. Böylesi amaçlarda başarılı olma yollarından biri, kontrol edici projenin uygulanmasından ve erkenden tespit edilmesinden geçmektedir. Tıkanıklığı önlemek ve bundan korunmak üzere Bulanık Sistemleri kullanmaktayız, çünkü bu amacı geliştirmektedir. Bu Tezde, Bulanık Sistemler temeline dayalı olarak tıkanıklık sorunlarını tespit etmek ve kontrol edecek algoritmalar sunumu yapılmaktadır. Bu algoritmalara Bulanık Mantık Tıkanma Tespiti (FLCD) ve Bulanık Mantık Kontrolörü (FLC) adı verilmektedir. Bu algoritmalar konusundaki bu bilgiler, MATLAB Simulasyon programını kullanarak simule edilmektedir.

Amaçlanan algoritmalar iyi performans ve gecikmeyi azaltma konusunda iyi sonuçlar ve verimlilikte yükselme göstermiştir.

Anahtar Kelimeler: bulanık sistemler, tıkanıklık tanımı, tıkanıklık algılama ve kaçınma.

ABSTRACT

CONGESTION DETECTION AND AVOIDANCE

MOHAMMED, NSAIF

M.S. c. Department of Computer Engineering

Supervisor: Prof. Dr. Mehmet R. Tolun

September 2012, 86 pages

In this thesis, A congestion problem is studied, The study is talking about how to detect the congestion and how to avoid it. Service providers are interested in optimizing their networks performance because of the high cost of network assets and commercials, and competitive nature of Internet service provisioning,

This leads them to reduce congestion hotspots and to improve resources utilization across the network, which, in turn, results in an increased revenue collection.

One of the ways to achieve such goals is through applying the controller scheme and by early detect it, we use Fuzzy Systems to detect congestion and avoid it, Because it improve that purpose.

In this Thesis an algorithms to detect and control congestion problem based on Fuzzy Systems is presented, this algorithms are called Fuzzy Logic Congestion Detection (FLGD) and Fuzzy Logic Controller (FLC).

This information about these algorithms are simulated, using MATLAB Simulation program. The proposed algorithms showed good performance and good result in reduce delay and raise throughput.

Keywords: fuzzy systems, congestion definition, congestion detection and avoidance.

ACKNOWLEDGMENTS

The author wishes to express his deepest gratitude to his Prof. Dr. Mehmet R. Tolun for their guidance, advice, criticism, encouragements and insight throughout the research.

The author would also like to thank the **MATLAB** that I used its tool and toutrial to write my thesis.

The author would also like to thank his family Specifically Dr. tulin to help me to built this system.

AND

I' d like to thank my father and my mother and my brothers and sisters and my wife and children and my uncle and aunt and all people for helping me and support me very very much.

TABLE OF CONTENTS

SIMULATION RESULTS AND DISCUSSION

LIST OF TABELS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Background on Fuzzy Systems:

Since its introduction in 1965 by Lotfi Zadeh (1965) the fuzzy set theory has been Widely used in solving problems in various fields,and recently in education evaluation.Biswas (1995) presented two methods for students' answerscripts evaluation using fuzzy sets and a matching function :a fuzzy evaluation method and a generalized fuzzy evaluation method. Chen and Lee(1999) presented two methods for applying fuzzy sets to overcome the problem of rewarding two different fuzzy marks the same total score which could arise from Biswas' method.Echauz and Vachtsevanos (1995) proposed a fuzzy logic system for translating traditional scores into letter-grades. Law (1996) built a fuzzy structure model for education grading system with its algorithm to aggregate dofferent test scores in order to produce a single score for individual student.He also proposed a method to build the membership function (MFs) of several linguistic values with different weights.Wilson Karr and Freeman (1998) presented an automatic grading system based on fuzzy rules and genetic algorithms. Ma and Zhou (2000) proposed a fuzzy set approach to assess the outcomes of student-centered learning using the evaluation of their peers and lecturer. Wang and Chen (2008) presented a method for evaluating students' answerscripts using fuzzy numbers confidence of the evaluator . From the previous studies, it associated with degrees of can be found that fuzzy numbers, fuzzy sets,fuzzy rules, and fuzzy logic systems are used for various educational grading systems. Weon and Kim (2001) presented an evaluation strategy based on fuzzy MFs. They pointed out that the system for students' achievement evaluation should consider the three important factors of the question which the students.

answer: the difficulty, the importance, and the complexity. Weon and Kim used singleton functions to describe the factor of each question reflectting the effect of the

three factors individually, but not collectively.Bai and Chen (2008b) pointed out that the difficulty factor is a very subjective parameter and may cause an argument about fairness in evaluation.Bai and Chen (2008a) proposed a method to automatically construct the grade MFs of fuzzy rules for evaluating student's learning achievement. Bai and Chen (2008b) proposed a method for applying fuzzy MFs and fuzzy rules for the same purpose. To solve the subjectivity of the difficulty factor of Weon and Kim's method (2001), they obtained the difficulty as a function of accuracy of the student's answer script and time consumed to answer. However, their method still has the subjectivity problem, since the results in scores and ranks are heavily depend on the values of several weights which are determined by the subjective knowledge of domain experts. Here, we propose an evaluation method considering the importance, difficulty, and complexity of questions based on Mamdani's fuzzy inference (Mamdani, 1974) and center of gravity (COG) defuzzification which is an alternative to Bai and Chen's method (2008b). The transparency and objective nature of the fuzzy system makes it easy to understand and explain the result of evaluation, and thus to persuade the students.

1.2 What Are Fuzzy Systems

1.2.1 Fuzzy theory and systems:

Fuzzy sets are the basic concept supporting fuzzy theory. The main research field in fuzzy theory are fuzzy sets, fuzzy logic, and fuzzy measure. Fuzzy reasoning or approximate reasoning is an application of fuzzy logic to Knowledge processing. Fuzzy control is an application of fuzzy reasoning to control. Although most applications of fuzzy theory have been biased toward engineering, these application have recently reached other disciplines,such as medical diagnostics, psychology, educations ,economy, management, sociology, etc.

The number of fuzzy applications in the field of KANSEL- a synthetic concept of emotion,impression,intuition, and other human subjective factors – has especially increased in Japanese fuzzy society.

It is not within the scope of this chapter to provide an overview for every aspect of fuzzy theory . We will focus on a fuzzy controller as an example of a simple FS to see how the output of the FS is calculated by fuzzy rules and reasoning.

1.2.2 Aspects of fuzzy systems:

One feature of FSs is the ability to realize a complex nonlinear input- output relation as a synthesis of multiple simple input-output relations. The simple input-output relation is described in each rule. The boundary of the rule areas is not sharp but 'fuzzy.' It is like an expanding sponge soaking up water. The system output from one rule area to the next rule area gradually changes. The is the essential idea of FSs and the origin of the term'fuzzy.' Another feature of FSs is the ability to separate logic and fuzziness. Since conventional two- value logic –based systems cannot do this ,their rules are modified fuzzy rules when logic should be changed and modify membership functions which define fuzziness when fuzziness should be changed.

1.3 The Aim of This Thesis :

The aim of the project is to use fuzzy systems in the detection of overcrowding and control it, using the simulator, which is the program responsible for how to calculate and study and read and analyze the results to determine whether the changes we wish to make useful or not? we Have been used the simulation program (MATLAB VERGIN 7.9A, 7.13B) and that by identifying two measurements related to links, they are the average amount of load on each link in each second, measured by bit/s, and the amount of the exploitation of the link that represents the percentage of the load on the link to the width of the beam, for the purpose of knowing the overcrowded links and periods of overcrowding.

.

1.4 -The Contents Of The Thesis:

As well as the basic concepts of this chapter, the thesis contains:

1 – The second Chapter: will be presented the proposed algorithm and study all of its stages starting from the simulated network using simulated computer network MATLAB (7.9 A, 7.13 B), and after that the creating of data to be entered for the fuzzy systems, and will also clarify the details of the fuzzy systems and how to represent it using MATLAB and how to find their outputs.

2 – The third Chapter: will include viewing the results and discuss and test the performance of the proposed algorithm with the statement and charged tracks used in the algorithm.

3 - Chapter four: will include the conclusions.

CHAPTER II

CONGESTION DETECTION AND AVOIDANCE

2.1 Congestion Defining:

2.1.1 Queuing theory definition:

In queuing theory, traffic congestion is said to occur if the arrival rate into a system exceeds the service rate of the system at a point in time. This is typically expressed as formula 1 in figure (1). Note that traffic congestion is not just a binary proposition using this definition.

1: arrival rate>service rate

$$\lambda > \mu$$

2: congestion= arrival rate/e

$$P=\lambda/\mu$$

Figure 1 formula 1

Traffic congestion ,or traffic intensity, can be measured as the ratio of the arrival rate to service rate .Using this definition we could precisely account for the periods of time in which a given network resource is congested and formula 2 in figure (5). It is even characterize the intensity of congestion precisely the periods of time in which more network traffic has arrived then has been sent (i.e., all t for which p(t)>1).

If the arrival rate persistently exceeds the service rate of the systems, the queue of traffic will grow without bound- there will be no steady state behavior of the system.The system will always be congested with service times getting longer and longer. On the other hand, if the arrival rate exceeds the service rate only occasionally, then congestion will be a transient phenomenon.Queues will build,but will eventually clear in the system. Note that by this definition, if the input rate equals the output rate , the network is not considered congested. It says nothing about the steady state size of the network's queues. If during a previous period of congestion a backlog queue of traffic built up , and traffic then arrived and departed at precisely the same rate the backlog of packets would not grow but it would not shrink either.Admittedly such a perfectly balanced system is not likely to persist for long in the real world . But quickly draining the queues is considered to be a central requirement of "good" congestion control (since it makes the queues shorter to handle future transients when p(t)>1). This illustrates that there are other facets to congestion control than just keeping the average input rate below the average output rate.

2.1.2 Networking text book definition:

In contrast, a popular academic textbook on networking, defines the buildup of packets in a queue not as "congestion" but rather as "contention." "congestion," According to this textbook, is restricted to the situation in which a switch or router has so many packets queued for transmission that it runs out of buffer space and must start dropping packets if more arrive. A buffer must be filled to capacity for congestion to occur. See Figure (2).

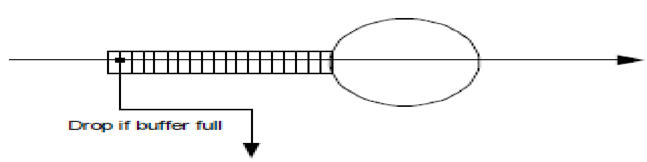

Figure 2 Congestion of a network router is said to occur if packets are dropped.

The queuing theory community does not make a distinction between "contention." According to their definition, as soon as a queue starts to build traffic congestion is occurring. "Contention" would imply "congestion" to queuing theorists.

These different defferent definition to the packet- dropped definition, congestion could be viewed as a binary phenomenon: nether a switch or router is dropping packets or it isn't. If someone asked for a more quantitative measure of congestion one might reply with number or the rate of dropped packets over some period of time.

The measure of trafic congestion according to the queuing theory definition, on the other hand , is a unit ratio of arrival rate over service rate. It is a measure of intensity.

2.2 Congestion Detection:

2.2.1 Fuzzy Logic Congestion Detection Algorithm:

2.2.1.1 Overview of The FLCD Algorithm:

Fuzzy logic is generalization of classical logic, in which there is a smooth transition between true and false. The basics of fuzzy logic are derived from the fuzzy set theory. In conventional (crisp) sets, members are always fully categorized and there is no ambiguity about membership while in fuzzy sets,the transition from membership to non-membership being gradual rather than abrupt.

A fuzzy set, $A \in X$, is characterized by a membership function, $\mu A(x)$, which associates each element in X with a real number in the interval [0,10]. $\mu A(x)$ is known as the grade of membership. Hence, the fuzzy set on the universe of discourse X is:

$$\mathbf{A = \{(x, \mu A(A)) \mid x \in x\}} \qquad \mathbf{(1)}$$

Fuzzy rules are the backbone of a fuzzy logic system. A simple fuzzy rule can be written as follows:

if x is HIGH then y is POSITIVE **(2)**

where HIGH and POSITIVE are linguistic values defined by fuzzy sets on the universe of discourse , X and Y, respectively. The if –part, if x is HIGH ,is known as the antecedent and the then – part, y is POSITIVE, is known as the consequent.

A set of linguistic rules used to map fuzzy inputs to outputs is known as the rule base. The functional componets of a fuzzy controller are shown in Figure (3).

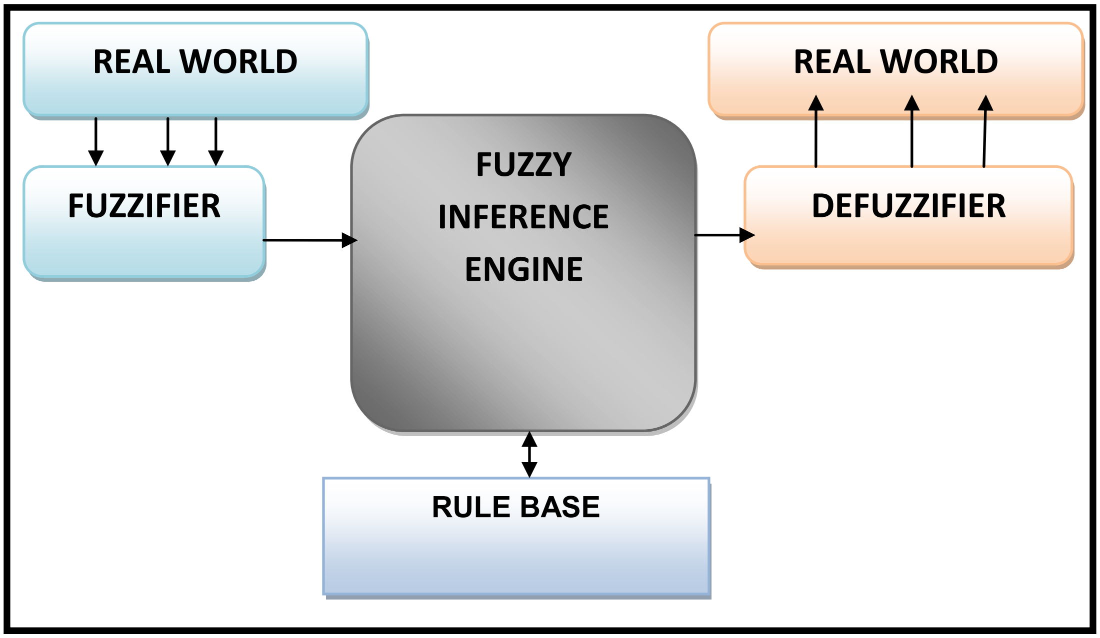

Figure 3 Fuzzy logic controller scheme.

The fuzzifier calculates suitable sets of degree of membership, called fuzzy sets, for the crisp(discrete) inputs.The inference engine evaluates output fuzzy sets from input sets using the predefined fuzzy rules contained in the rule base.The defuzzifier transforms the output fuzzy set into a crisp number to be useful in the real world.

The inference engine calculates the degree of activation for every rule in the rule base.

Fuzzy systems employ two two types of inference mechanism:

The mamdani approach and the Takagei-Sugeno approach. The FLCD algorithm employs the mamdami approach Therefore , an overview of this inference mechanism will now be presented. If the antecedent for rule j contains one variable, the rule's degree of activation is equal to the degree of membership of that single variable. If μ1j(x1) denotes the degree of membership of input x1 for rule j , then , μj, the degree of activation of rule j, is expressed as follows:

$$\mu j = \mu 1j\ (x1) \qquad (3)$$

If the antecedent for rule j contains more than one variable in the following form:

Rule j: if A1j and A2j and ….and Anj then bj. ... (4)

Where A^{K}_{j} is a fuzzy set with membership function,

$\mu^{k}_{j} : R \longrightarrow [0,1]$ **, = 1,........,m, k=1,.......n, bj ∈ R then** , in this case , the degree of activation for rule j is determined using the minimum

t- norm as follows:

$$\mu j = \mu^{1}_{j}(x_1) \otimes \mu^{2}_{j}(x_2) \otimes \ldots\ldots \otimes \mu^{n}_{j}(x_k) \qquad \ldots(5)$$

where $\mu^{n}j(x_k)$ is the degree of membership of input x_k . The degree of activation is inferred as the degree of membership of the output variable upon its fuzzy set, which is defined in the corresponding consequence.

The output of inferring m rules is the aggregation of the individual rule outputs. The implied output sets are combined to formulate a crisp output through a routine known as defuzzification. The widely applied defuzzification method is the Centre Of Gravity (COG) technique, which computes the weighted – average of the centre of gravity of each membership function. The COG of the system with m rules is as follows:

$$Y(x) = \frac{\sum_{J=1}^{m} b_j \mu_j}{\sum_{J=1}^{m} \mu_j} \qquad \ldots (6)$$

Where bj is the center of the membership function recommended by the consequent of rule j. The membership function for the fuzzy controller are initialized by the user, based on a priori knowledge.

The general framework of fuzzy systems is shown in figure (4).

2.2.1.2 Simulation of computer networks:

The Concept of simulation technology of computer networks, that used very much in communications researches and computer networks is based on the work model to see the behavior of real computer networks or networks that want to implement in the future on a computer screen. By using computer program provision for this purpose. Knowing this behavior is by calculating the interaction between the various entities of the network (nodes\routers data links,and packets, etc.) using mathematical equations or Soft Computing , and documenting observation resulting from the experience. The benefit of this that we can adjust all network features, components and everything related to it (parts and equipment), or change it or control. It is A low –cost process when measured against the cost of implementation of the network on the practice without prior study. The adjustment process may be done more than once until we reach the desired goal.

The emulator (or simulator network) is the program responsible for the account and how to read and study and analysis the results to determine whether the changes we wish to make a useful or not ? We use MATLAB (virgin 7.9 A , 7.13 B) to present the structure of the tow networks and run the result. By using this simulator we can represent different types of networks .we will use this procedure in both FLCD AND FLC

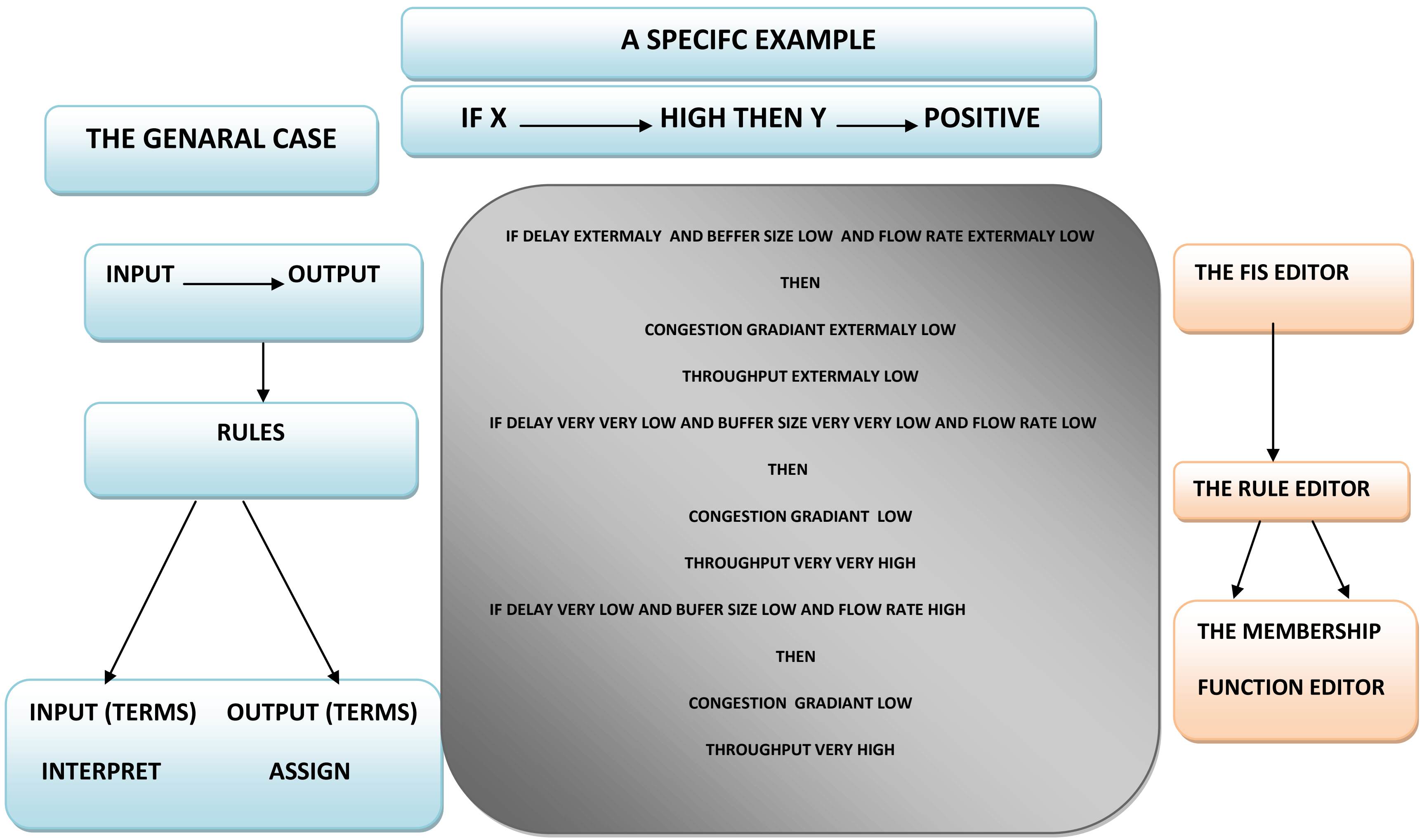

Figure 4 The general framework of fuzzy system

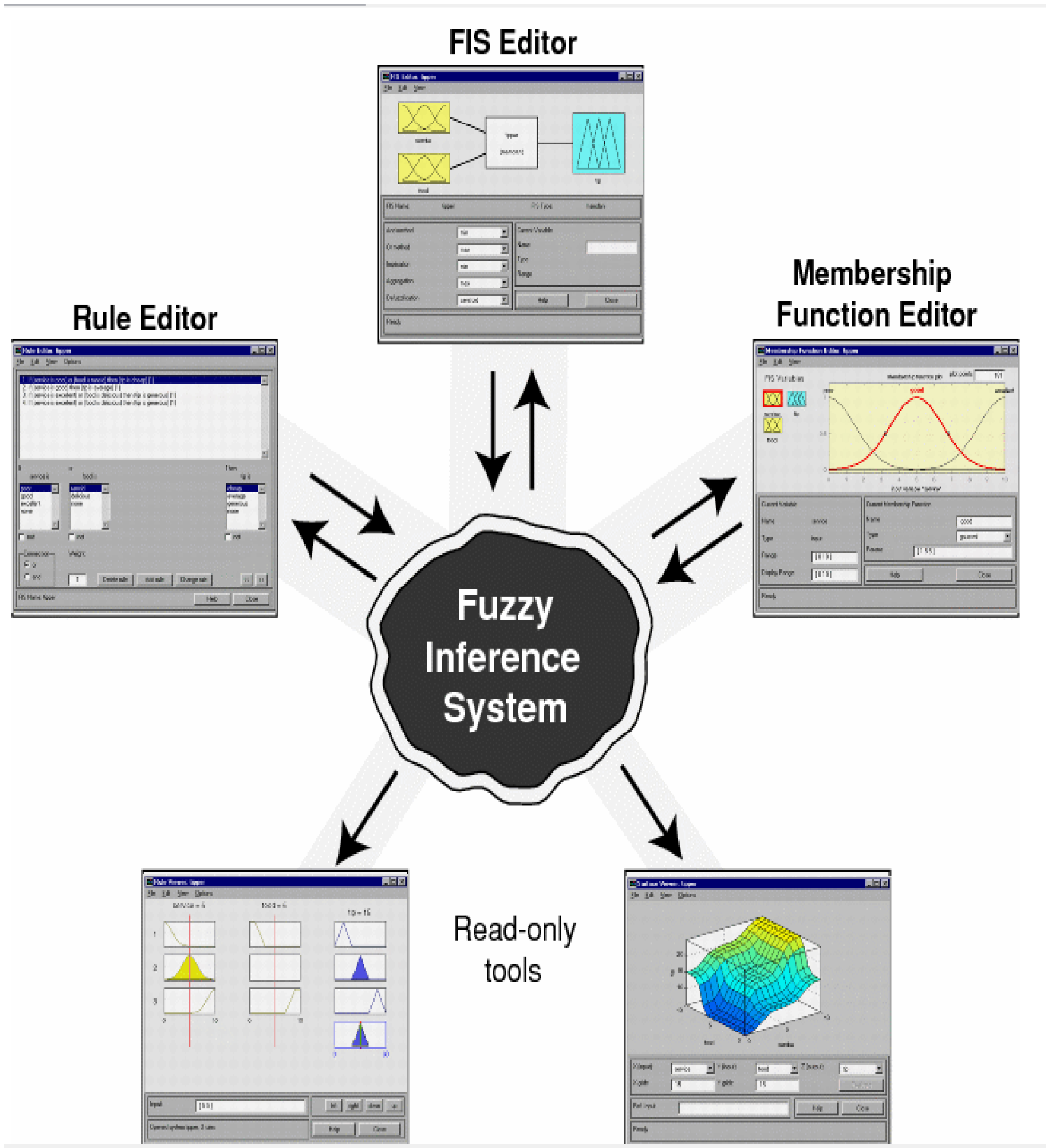

Figure 5 Fuzzy Logic Tolbox Grapgical User Interface (GUI) Tools:

2.2.1.3 FLCD Architecture:

The FLCD algorithm is composed of the fuzzy logic control unit (FLCU) ,the probability Adjuster (PA) and the CHOKE Activator (CA) . Figure (6) shows the FLCD architecture.

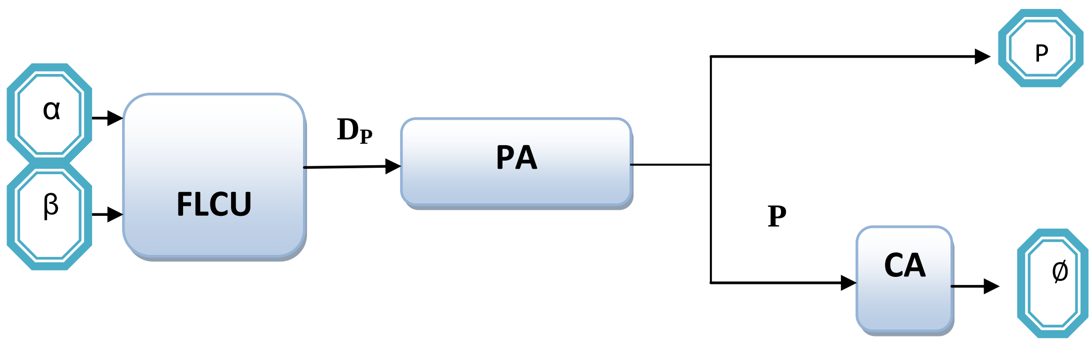

Figure 6 FLCD architecture

A single FIFO buffer in which all packets are treated equally is assumed. The queue status is sampled at a period τ of 0.0 02 seconds in order to obtain the queueoccupation size (backlog) q(t) and the traffic arrival rate r(t).

The backlog q(t) is translated into the backlog factor **α** which is the ratio of backlog with respect to the Buffer size BS as follows:

$$\alpha = q(t)\backslash BS \qquad .....(1)$$

The packet arrival rate is determined by counting the actual number of packets that arrive at the buffer (both those that are queued and those that are dropped) during sampling period τ , let n denote the number of packets that arrive at the buffer During period τ , ω1 denote the measuring weight and rm the maximum packet arrival rate. The weighted average packet arrival rate r(t) and the packet arrival factor β are determined as follows:

$$\overline{r(t)} = \omega 1 * r(t - \tau) + (1 - \omega 1) * n \qquad \text{.........(2)}$$

$$\beta = \begin{cases} \overline{r(t)} \backslash rm & \overline{r(t)} < rm \\ 1.0 & \overline{r(t)} \geq rm \end{cases}$$

The FLCU determines the change in packet marking\ dropping probability Δ**pb** by using the fuzzified values of parameters α and ***β*** **.** The set of linguistic rules that govern the inference process in the FLCU is shown in Table I.

IF ***α*** IS *LOW* AND ***β*** IS *LOW* THEN Δ*Pb* IS *NEGATIVE BIG*
IF *α* IS *LOW* AND *β* IS *MEDIUM* THEN Δ*Pb* IS *NEGATIVE SMALL*
IF *α* IS *LOW* AND *β* IS *HIGH* THEN Δ*Pb* IS *ZERO*
IF *α* IS *NORMAL* AND *β* IS *LOW* THEN Δ*Pb* IS *NEGATIVE SMALL*
IF *α* IS *NORMAL* AND *β* IS *MEDIUM* THEN Δ*Pb* IS *ZERO*
IF *α* IS *NORMAL* AND *β* IS *HIGH* THEN Δ*Pb* IS *POSATIVE SMALL*
IF *α* IS *HIGH* AND *β* IS *LOW* THEN Δ*Pb* IS *POSATIVE SMALL*
IF *α* IS *HIGH* AND *β* IS *MEDIUM* THEN Δ*Pb* IS *POSATIVE BIG*
IF *α* IS *HIGH* AND *β* IS *HIGH* THEN Δ*Pb* IS *POSATIVE BIG*

TABLE 1 The FLCD Rule Base

The PA computes the new packet marking probability pb as follows:

$$P_b\,(t) = p_b\,(t - \tau) + \Delta p_b\,(t) \qquad \text{........(3)}$$

Packets are either marked (if Explicit Congestion Notification is enabled) or dropped with a probability pb. Responsive flows react to these events by reducing their sending rates there by reducing congestion at the bottleneck link. In order to address the issue of fairness in light of non – responsive flows and network anomalies such as Denial of Service (DOS) attacks and routing loops which may dramatically flood the network as the responsive flows back off, we incorporate the CHOKe Activator (CA) which uses pb(t) to generate a fuzzy parameter Ø ∈ [0,1] . Let pthresh denote the CHOKe threshold then the fuzzy parameter is derived as follows:

$$\phi = \begin{cases} 0 & p_{thresh} > p_b \\ \left(\dfrac{p_b - p_{thresh}}{1 - p_{thresh}} \right)^2 & p_{thresh} \le p_b \end{cases} \quad \text{... (4)}$$

When pthresh > pb (low congestion), Ø is 0.0 During this period there is no CHOKe activity.

When pthresh ≤ pb (high congestion), the value of Ø increases rapidly such that more packets from non- responsive and TCP unfriendly flows are dropped at the bottlenck link. An arriving packet is picked probabilistically based on the value of Ø . This packet is compared with a randomly chosen packet from the buffer . If they have the same flow ID , they are both dropped. Otherwise the randomly chosen packet is kept in the buffer (in the same position as before) and the arriving packet is queued If the buffer is not full; otherwise it is dropped. Figure (7) shown the standard membership function that is used in the process of fuzzifying the parameters α and β.

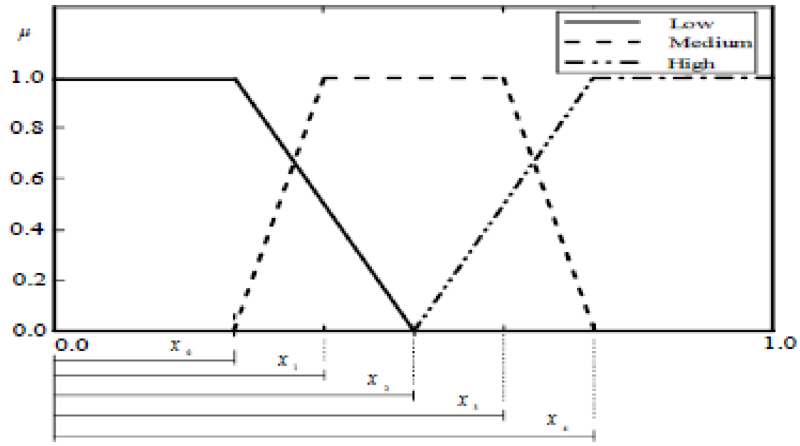

Figure 7 Standard Membership Function for backlog and packet arrival factors

Figure (8) shows the membership function used in the defuzzification process in order to generate the change in packet marking\ dropping probability Δpb.

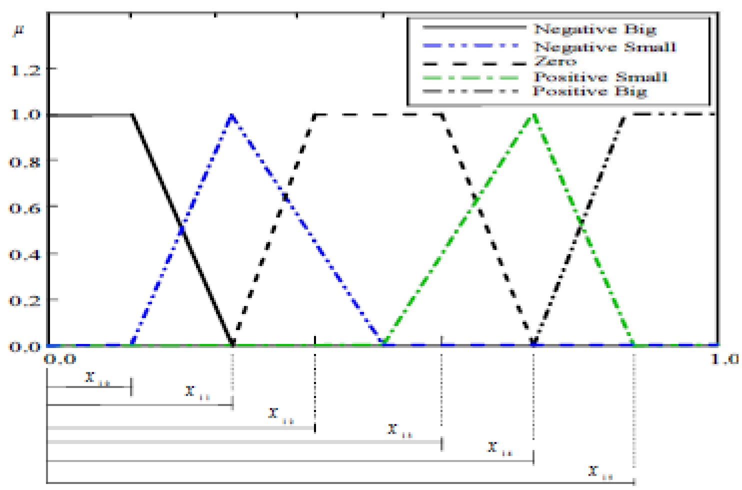

Figure 8 Membership Function for change in packet marking probability

The 18- dimensional parameter vector P that determines the membership function is expressed as follows :

$$\mathbf{P= [\ x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},x_{17}]......(5)}$$

The definition of these elements is presented as follows:-

1- X_0 , x_1 , x_2 , x_3 , x_4 , are parameters for the backlog factor (**α**) membership function (**MF1**) as shown in figure (7).

2- X_5 , x_6, x_7, x_8 , x_9 are parameters for the packet arrival (***β***) rate membership function (**MF2**) similar to Figure (8) because α and ***β*** use the same standard membership function.

3- X_{10} , x_{11} , x_{12} , x_{13} , x_{14} , x_{15} are parameters for the change in packet marking probability (∆ ***pb***) membership function (**MF3**) as depicted in Figure(7).

4- X_{16} , x_{17} denote the maximum negative and positive variations (∆ ***pneg and*** ∆ ***Ppos***) of the change in packet marking probability. The output from the defuzzification process which falls in the range [0,1.0] is scaled to [∆***Pneg***, ∆***Ppos***].

Parameters for individual membership functions are always sorted in ascending order.For instance, for MF1 the following:

$$x_0 < x_1 < x_2 < x_3 < x_4 \qquad (6)$$

Must always be true . The same applies to MF2 and MF3 .The parameters in equation (6) are modeled as single 18 –dimensional particle based on which a number of similar particles are randomly created and initialized within a decision variable space whose parameters are predefined. The Adaptive MOPSO (AMOPSO) algorithm , which is a special case of the particle swarm Optimization (PSO) algorithm, is then used to optimize these particles . Each particle is viewed as a potential solution. The concept of PSO is that each particle randomly searches the decision variable space by updating itself with its own memory and the social information gathered from other particles. This is done over a number of generations\ iterations. Unlike basic PSO which optimizes the particles based on a single objective function,MOPSO is tailored for multiple objective functions which are usually competing and noncommensurable. In this case ,the optimization process generates a pool of non – dominated solutions called the pareto optimal set . The optimization of the FLCD algorithm is based on four objective functions:

1) Maximizing link utilization;

2) Minimizing packet loss rates;

3) Minimizing link delay;

4) Minimization jitter.

The upper bound on packet arrival rate , rm, is set to 5 packets per sampling period τ of 0.002 seconds. Parameter $\omega 1$ is set to 0.9, in order to avoid drastic changes in the weighted average arrival rate of packets.

These setings are done, based on the premise that the other parameters of the algorithm will be realized by the optimization process.

After the optimization process , a fuzzy inference algorithm is used to draw the best compromise solution from the pareto optimal set as follows:

P = [0:01; 0:02; 0:03;0:04; 0:29; 0:95;0:96;0:97; 0:98; 0:99;0:01;0:02;0:03;0:34;0:61;0:64;-0:0005;0:0005] ....(7)

These parameters are used in configuring the membership functions of the practical FLCD algorithm.

2.3 Congestion Control And Avoidance:

The tremendous growth of wireless networks demands the need to meet different multimedia (such as voice , audio, video ,data, etc) applications available over the network. This application demand and allocation could lead to congestion if the network has to maintain such high resources for the quality of service requirements of the applications. Now , we will introduce a fuzzy based congestion control scheme to avoid congestion. The fuzzy input parameters such as delay , buffer size and flow rate for each packet is considered to produce a congestion gradient factor. Depending on the value of congestion gradient factor the flow rate is maintained same or reduced. Simulation results present better throughput and variation in congestion gradient factor. The results are compared with one of the congestion control policy of Qualnet wireless network simulator.

2.3.1 Fuzzy Controller For Congestion Avoidance:

A fuzzy controller operates on the concept of a fuzzy set, which allows imprecise or incomplete input information to be expressed and incorporated with the heuristic knowledge of the system in a quantitative, systematic fashion for the control purpose. The generic model for a fuzzy controller is depicted in Figure (9) .In essence , a fuzzy controller is constructed from the use of fuzzy sets and inference steps. In the fuzzification process, a real- number input (crisp input) is converted into linguistic values such as low or high characterized by fuzzy sets through the membership function. A single crisp input value can take on more than one linguistic value depending on how the membership functions are defined as will be described later. Then a set of rules called rule- base , which emulates the decision – making process of a human expert , is applied to the linguistic values of the inputs so as to infer the output fuzzy sets.These outputs are then defuzzified to the crisp output which represents the actual control signal for the process.Fuzzy logic systems are one of the main developments and successes of fuzzy sets and fuzzy logic. A FLS is a rule-base system that implements a nonlinear mapping between its inputs and outputs.

- Fuzzifier;
- Defuzzifier;
- Inference engine;
- Rule base.

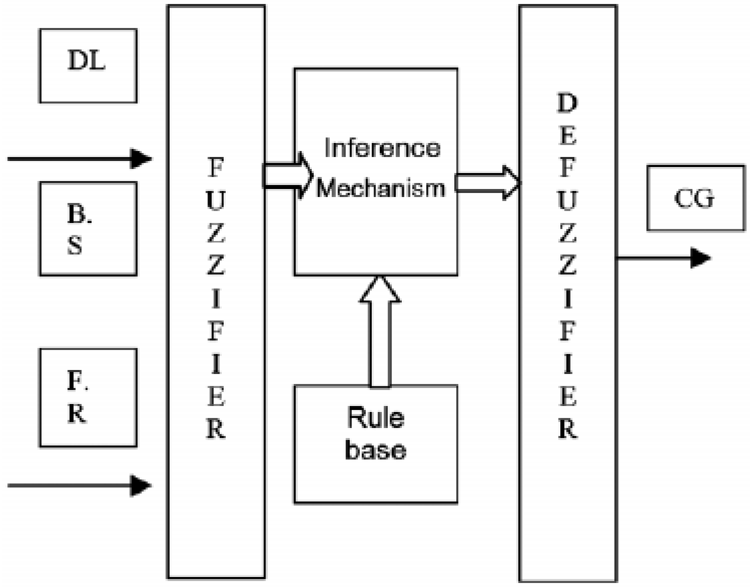

Figure 9 Fuzzy Controller

2.3.2 Fuzzification:

To translate crisp inputs for each input variable *i* into linguistic values, we define Ni linguistic values $\mathbf{A}^{(m)}$, m = 1; 2; ; ; ; ; Ni as well as their membership functions.

For delay , its linguistic values m= 1;2;3;4;5;6;7;8;9 are (i) Extremely Low (EL) , (ii) Very Very Low (VVL) , (iii) Very Low (VL), (iv) Low (L), (v) medium (m) (vi) high (vii) Very high (VH) (viii) Very Very high(VVH) and (xi) Extremely high (EH).

Besides a triangular shape , many other choices for the shape of the membership functions also exists, including trapezoidal, Gaussian , and etc. Note that a single value of un input (ranging from 0 to the buffer size) can take on more than one linguistic value. For instance, a input can be both medium and high but with different degrees of certainty indicated by outputs of the membership functions. Similarly the linguistic values for the other two inputs and output are defined.

2.3.3 Inference and Defuzzification:

After crisp inputs are mapped to the linguistic values through the membership functions in the fuzzification step, the inference rule is applied to determine the output by using the rule base. The rule- base is a set of rules that emulates the decision making process of the human expert controlling the system. The rule is written in the form:

IF premise THEN consequent\action

Where premise is a combination of input linguistic values and consequent is what action to be taken. Because there are nine linguistic values each for Delay, buffer size and flow rate, the number of rules is 729, out of which 40 rules have been selected for the proposed work. In our case, the rule- base is in a form called functional fuzzy system where each rule 1 is written down as:

RULE 1: IF delay is low and buffer size is low

AND : flow rate is low

THEN : Congestion gradient is low

Delay	*Buffer size*	*Flow rate*	*Congestion gradient*
EL	**EL**	**EL**	**EL**
VVL	**VVL**	**L**	**L**
VL	**VL**	**H**	**L**
EL	**L**	**VVL**	**L**
VVL	**L**	**L**	**VL**
VL	**M**	**VH**	**L**
EL	**H**	**VL**	**VL**
VVL	**VH**	**M**	**M**
VL	**VVH**	**VVH**	**H**
L	**EL**	**EL**	**V VL**

Table 2 Fuzzy Controller Rule Base

The fuzzy rule base includes 40 experimental rules, out of which some of the rules are shown in Table (2). The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However , the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

Perhaps the most popular defuzzification method is the centroid calculation, which returns the center of area under the curve. Once the congestion gradient factors are determined from the inference step, the defuzzification is performed to obtain final congestion gradient factor.

2.3.4 Algorithm:-

Algorithm 1: Congestion Controlling at the Base Station:

{Nomenclature = number of running applications, n= number of packets BW= maximum bandwidth of a cell BS = maximum buffer size of a base station, D=delay of the packets , FR= flow rate of the packet, CG = congestion gradient}

Begin

Step1 : Get the current values of delay flow rate and buffer size from the base station.

Step 2:Call Algorithm 2;

* compute congestion gradient factor *\

Step 3: IF congestion gradient factor below the lower threshold, maintain the previous flow rate ;

Step 4: IF congestion gradient factor is above the lower threshold, reduce the flow rate ;

Step5 : go to step2.

Step 6: stop

End

Algorithm 2: Computation of Congestion gradient Factor:

Begin

Step1 : For k=1 to n do

Begin

a-Intialize fuzzy controller with delay (D) , buffer size (BS) and flow rate (FR) to the packet 'k' ;

b- Find the membership values of Delay , buffer size and flow rate using triangular rule c- Compute CG by referring to rule base;

Step 2: Inform CG factor to congestion controller;

Step3 :Return;

Step4: stop.

End.

2.3.5 List Of Rule Base For Fuzzy Controller:

1) IF Delay is externly low

AND Buffer size is extermly low

AND Flow rate is extermly low

THEN Congestion gradient is extermly low

****

2) IF Delay is very very low

AND Buffer is size very very lowLow

AND Flow rate is

THEN Congestion gradient is low

****

3) IF Delay is very low

AND Buffer is size very low

AND Flow rate is high

THEN Congestion gradient is low

****

4) IF Delay is extermly low

AND Buffer size is low

AND Flow rate is very very low

THEN Congestion gradient is low

****

5) IF Delay is very very low

AND Buffer size is low

AND Flow rate is low

THEN Congestion gradient is very low

****

6) IF Delay is very low

AND Buffer size is medium

AND Flow rate is very high

THEN Congestion gradient is low

****

7) IF Delay is extermly low

AND Buffer size is high

AND Flow rate is very low

THEN Congestion gradient is very low

****

8) IF Delay is very very low

AND Buffer size is very high

AND Flow rate is medium

THEN Congestion gradient is medium

****

9) IF Delay is very low

AND Buffer size is very very high

AND Flow rate is very very high

THEN Congestion gradient is high

****

10) IF Delay is low

AND Buffer size is extermly low

AND Flow rate is extermly low

THEN Congestion gradient is very very low

****

11) IF Delay is very very low

AND Buffer size is extermly low

AND Flow rate is very low

THEN Congestion gradient is low

****

12) IF Delay is very low

AND Buffer size is very low

AND Flow rate is high

THEN Congestion gradient is low

****

13) IF Delay is extremely low

AND Buffer size is low

AND Flow rate is very low

THEN Congestion gradient is low

****

14) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate is low

THEN Congestion gradient is low

****

15) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate very very low

THEN Congestion gradient is low

****

16) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate very high

THEN Congestion gradient is very low

****

17) IF Delay is very very low

AND Buffer size is very high

AND Flow rate is very low

THEN Congestion gradient is medium

****

18) IF Delay is very very low

AND Buffer size is extermly low

AND Flow rate is very very high

THEN Congestion gradient is very low

****

19) IF Delay is very low

AND Buffer size is extremely low

AND Flow rate is extremely low

THEN Congestion gradient is low

****

20) IF Delay is low

AND Buffer size is very very low

AND Flow rate is low

THEN Congestion gradient is very very low

****

21) IF Delay is low

AND Buffer size is high

AND Flow rate is very low

THEN Congestion gradient is extremely low

****

22) IF Delay is low

AND Buffer size is very high

AND Flow rate is very very low

THEN Congestion gradient is very low

****

23) IF Delay is low

AND Buffer size is very very low

AND Flow rate is high

THEN Congestion gradient is extremely low

****

24) IF Delay is low

AND Buffer size is high

AND Flow rate is very high

THEN Congestion gradient is medium

****

25) IF Delay is extremely low

AND Buffer size is low

AND Flow rate is very very low

THEN Congestion gradient is low

****

26) IF Delay is very low

AND Buffer size is medium

AND Flow rate is very high

THEN Congestion gradient is low

****

27) IF Delay is low

AND Buffer size is very very high

AND Flow rate is high

THEN Congestion gradient is extremely low

****

28) IF Delay is very low

AND Buffer size is medium

AND Flow rate is very high

THEN Congestion gradient is low

****

29) IF Delay is extremely low

AND Buffer size is low

AND Flow rate is very very low

THEN Congestion gradient is low

****

30) IF Delay is very low

AND Buffer size is very low

AND Flow rate is extremely low

THEN Congestion gradient is low

****

31) IF Delay is very low

AND Buffer size is very low

AND Flow rate is very high

THEN Congestion gradient is medium

****

32) IF Delay is very low

AND Buffer size is very low

AND Flow rate is very low

THEN Congestion gradient is low

****

33) IF Delay is high

AND Buffer size is high

AND Flow rate is high

THEN Congestion gradient is very very high

****

34) IF Delay is very very high

AND Buffer size is very very high

AND Flow rate is very very high

THEN Congestion gradient is very very high

****

35) IF Delay is extremely low

AND Buffer size is extremely low

AND Flow rate is extremely low

AND Congestion gradient is extremely low

THEN Throughput is high

****

36) IF Delay is very very low

AND Buffer size is very very low

AND Flow rate is low

AND Congestion gradient is low

THEN Throughput very very is high

****

37) IF Delay is very low

AND Buffer size is very low

AND Flow rate is high

AND Congestion gradient is low

THEN Throughput is very high

****

38) IF Delay is extremely low

AND Buffer size is low

AND Flow rate is very very low

THEN throughput is high

****

39) IF Delay is very very low

AND Buffer size is low

AND Flow rate is low

AND Congestion gradient is very low

THEN Throughput is very very high

****

40) IF Delay is very low

AND Buffer size is medium

AND Flow rate is very high

AND Congestion gradient is low

THEN Throughput is very high

****

41) IF Delay is extremely low

AND Buffer size is high

AND Flow rate is very low

AND Congestion gradient is very low

THEN Throughput is high

****

42) IF Delay is very very low

AND Buffer size is very high

AND Flow rate is medium

AND Congestion gradient is medium

THEN Throughput is very very high

****

43) IF Delay is very low

AND Buffer size is very very high

AND Flow rate is very very high

AND Congestion gradient is high

THEN Throughput is very high

****

44) IF Delay is low

AND Buffer size is extremely low

AND Flow rate is extremely low

AND Congestion gradient is very very low

THEN Throughput is high

****

45) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate is very low

AND Congestion gradient is low

THEN Throughput is very very high

****

46) IF Delay is very low

AND Buffer size is very low

AND Flow rate is high

AND Congestion gradient is low

THEN Throughput is very high

****

47) IF Delay is extremely low

AND Buffer size is low

AND Flow rate is very low

AND Congestion gradient is low

THEN Throughput is high

****

48) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate is low

AND Congestion gradient is low

THEN Throughput is very very high

****

49) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate is very very low

AND Congestion gradient is low

THEN Throughput is very very high

****

50) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate is very high

AND Congestion gradient is very low

THEN Throughput is very very high

****

51) IF Delay is very very low

AND Buffer size is very high

AND Flow rate is very low

AND Congestion gradient is medium

THEN Throughput is very very high

****

52) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate is very very high

AND Congestion gradient is very low

THEN Throughput is very very high

****

53) IF Delay is very low

AND Buffer size is extremely low

AND Flow rate is extremely low

AND Congestion gradiant is low

THEN Throughput is very high

****

54) IF Delay is low

AND Buffer size is very very low

AND Flow rate is low

AND Congestion gradient is very very low

THEN Throughput is high

****

55) IF Delay is low

AND Buffer size is high

AND Flow rate is very low

AND Congestion gradient is extremely low

THEN Throughput is high

****

56) IF Delay is low

AND Buffer size is very high

AND Flow rate is very very low

AND Congestion gradient is very low

THEN Throughput is high

****

57) IF Delay is low

AND Buffer size is very very low

AND Flow rate is high

AND Congestion gradient is extremely low

THEN Throughput is high

****

58) IF Delay is low

AND Buffer size is high

AND Flow rate is very high

AND Congestion gradient is medium

THEN Throughput is high

****

59) IF Delay is extremely low

AND Buffer size is low

AND Flow rate is very very low

AND Congestion gradient is low

THEN Throughput is high

****

60) IF Delay is very low

AND Buffer size is medium

AND Flow rate is very high

AND Congestion gradient is low

THEN Throughput is very high

****

61) IF Delay is low

AND Buffer size is very very high

AND Flow rate is high

AND Congestion gradient is extremely low

THEN Throughput is high

****

62) IF Delay is very low

AND Buffer size is medium

AND Flow rate is very high

AND Congestion gradient is low

THEN Throughput is very high

****

63) IF Delay is extermly low

AND Buffer size is low

AND Flow rate is very very low

AND Congestion gradient is low

THEN Throughput is high

****

64) IF Delay is very low

AND Buffer size is very low

AND Flow rate is extremely low

AND Congestion gradient is low

THEN Throughput is very high

****

65) IF Delay is very low

AND Buffer size is very low

AND Flow rate is very high

AND congestion gradient is medium

THEN Throughput is very high

****

66) IF Delay is very low

AND Buffer size is very low

AND Flow rate is very low

AND Congestion gradient is low

THEN Throughput is very high

****

67) IF Delay is high

AND Buffer size is high

AND Flow rate is high

AND Congestion gradient is very very high

THEN Throughput is low

****

68) IF Delay is very very high

AND Buffer size is very very high

AND Flow rate is very very high

AND Congestion gradient is very very high

THEN Throughput is very very low

****

69) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate is extremely low

AND Congestion Gradient is low

THEN throughput is very very high

****

70) IF Delay is very very low

AND Buffer size is extremely low

AND Flow rate is very very high

AND Congestion gradient is very low

THEN Throughput is very very high

****

CHAPTER III

SIMULATION RESULTS AND DISCUSSION

3.1 Graphical User Interfaces (GUIs):

The continued growth in the volume of data and results and its complexity because the data sources complexity and algorithms, led to an urgent need to represent the data and the results obviously and display it by the graphical representation (GUI). Usually, the graphical representation, not only is the most effective way to communicate the results of the proposed study and data processing, but it is the amphidiarthrodial interaction listen and deal with all requests and changes and take the appropriate actions and reactions to show results, according to the requirements previously installed. With the increased financial capabilities for the computer hardware,MATLAB program has become one of the best software applications available, which have the potential to provide each of the computational capabilities to generate data and display it using various Schematic representation.

I have been used the MATLAB A(7.9),B(7.13) in the design of user interfaces In this

thesis, which includes the following steps:

1-The main interface: to choose the structure of the network that will be working on it as shown in Figure (10), and when choosing the required structure it will be move to the interface that represents it.

2-The first interface is fuzzy logic congestion detection, and it consist of two constant ($\boldsymbol{\alpha},\boldsymbol{\beta}$) , one max , fuzzy logic controller with rule viewer, and Δp_b display, and when choose it the fuzzy logıc congestıon detectıon screen wıll occure fıgure(11)

3. Then the FIS editor screen occurs ,fıgure(12).

4. Then the rule editor screen,occurs, fıgure(13).

5.Then the membership function of the three parameter (alpha, beta, change in probability) will occurs, figure (14, 1-2-3).

6.Then the rule viewer occur, figure(15).

7. the table of the result of the rule viewer is given in table (3).

8. After the rule viewer the surface viewer will occur, figure (16).

9. the second scheme is FUZZY LOGIC CONTROLLER which consist of three constant (delay, buffer size, flow rate) ,one max, one Dmax, Congestion gradient display and throughput display, when choose it the FLC SCREEN occurs, figure (17).

10. Then the FIS editor occurs, figure (18).

11. Then the rule editor occurs, figure (19).

12. Then the membership function of the five parameters (delay, buffer size, flow rate, congestion gradient, throughput) will occur, figure (20,1-2-3-4-5).

13. Then the rule viewer will occurs, Figure (21)

14. the table of the result of the FLC rule viewer is given in table (4) .

15. Finally the surface viewer of FLC occurs, figure (22).

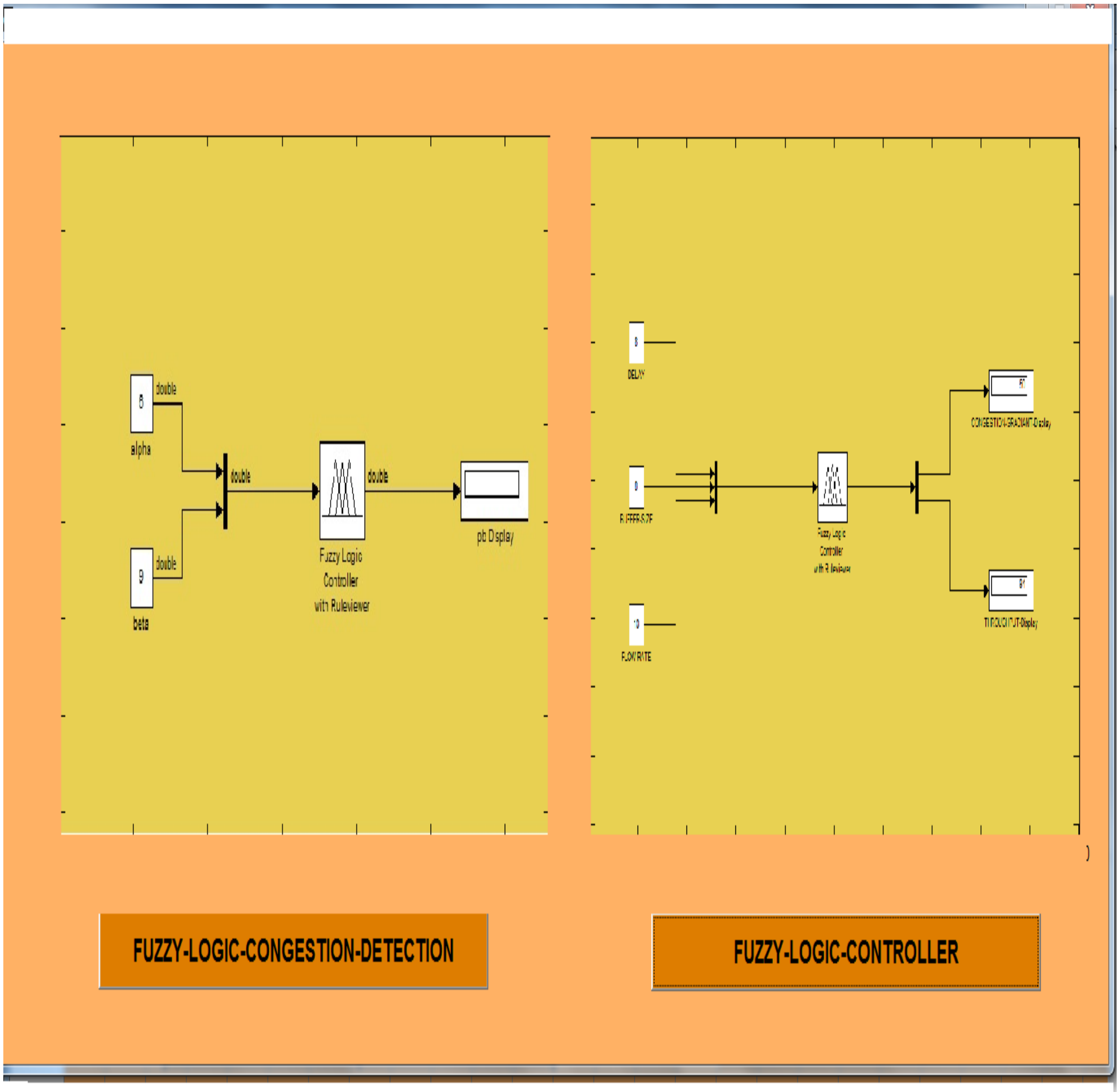

Figure 10 The Main Interface

Figure 11 Fuzzy Logic Congestion Detection Screen

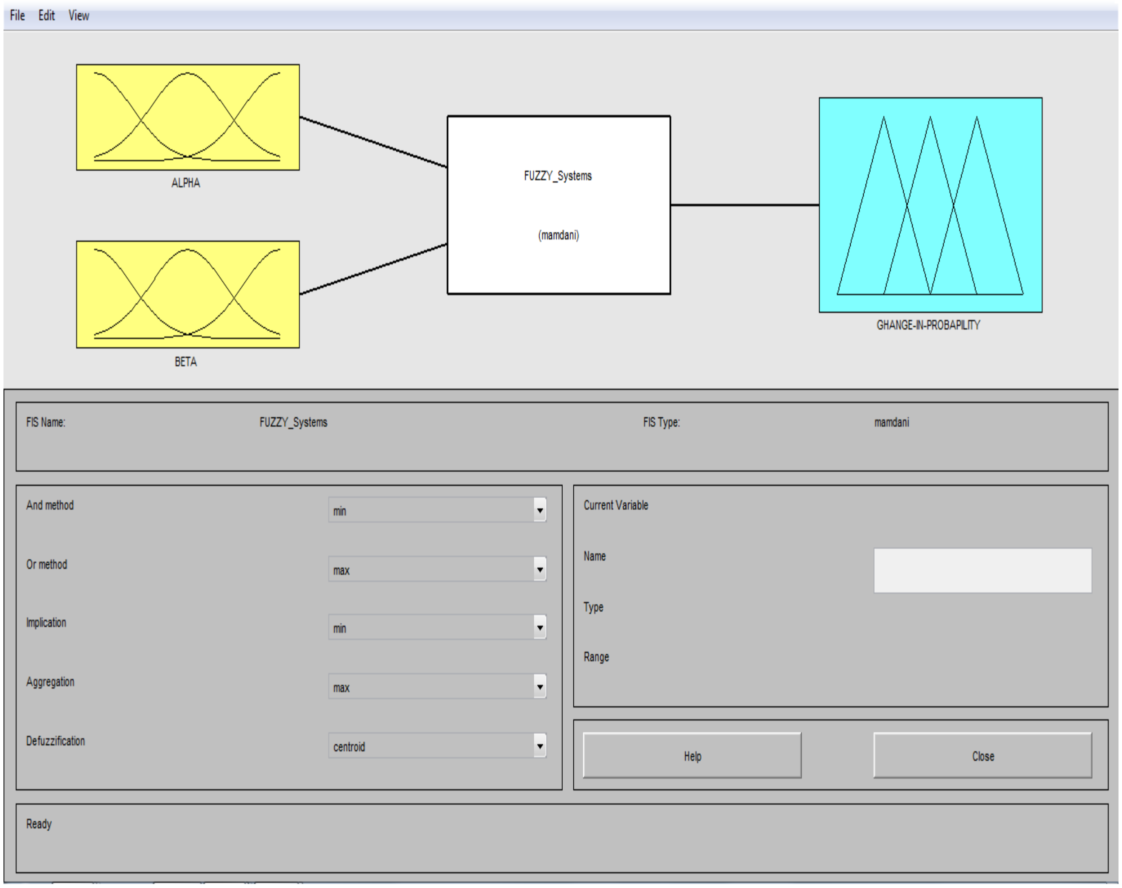

Figure 12 FLCD FIS Editor

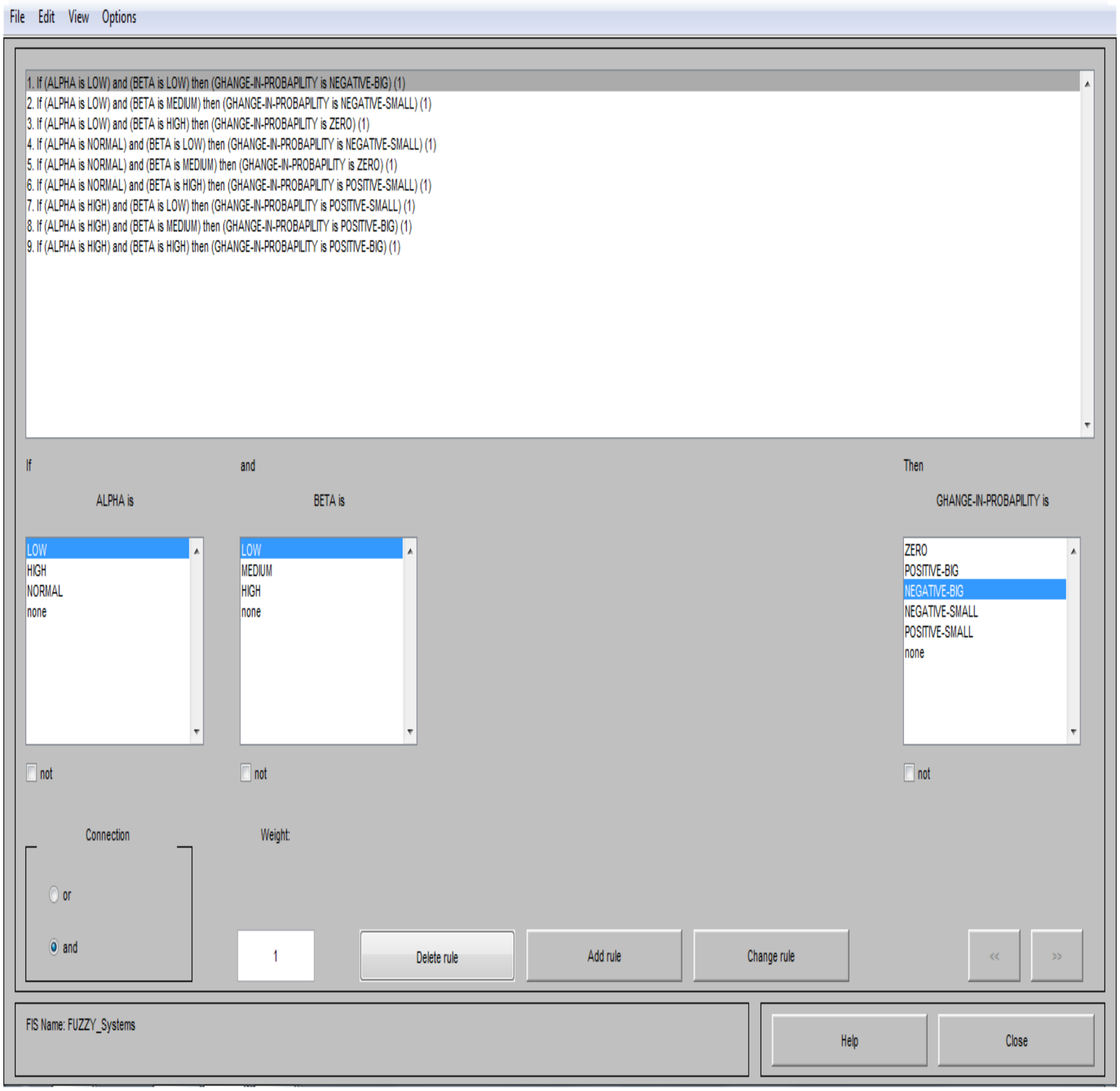

Figure 13 FLCD Rule Editor

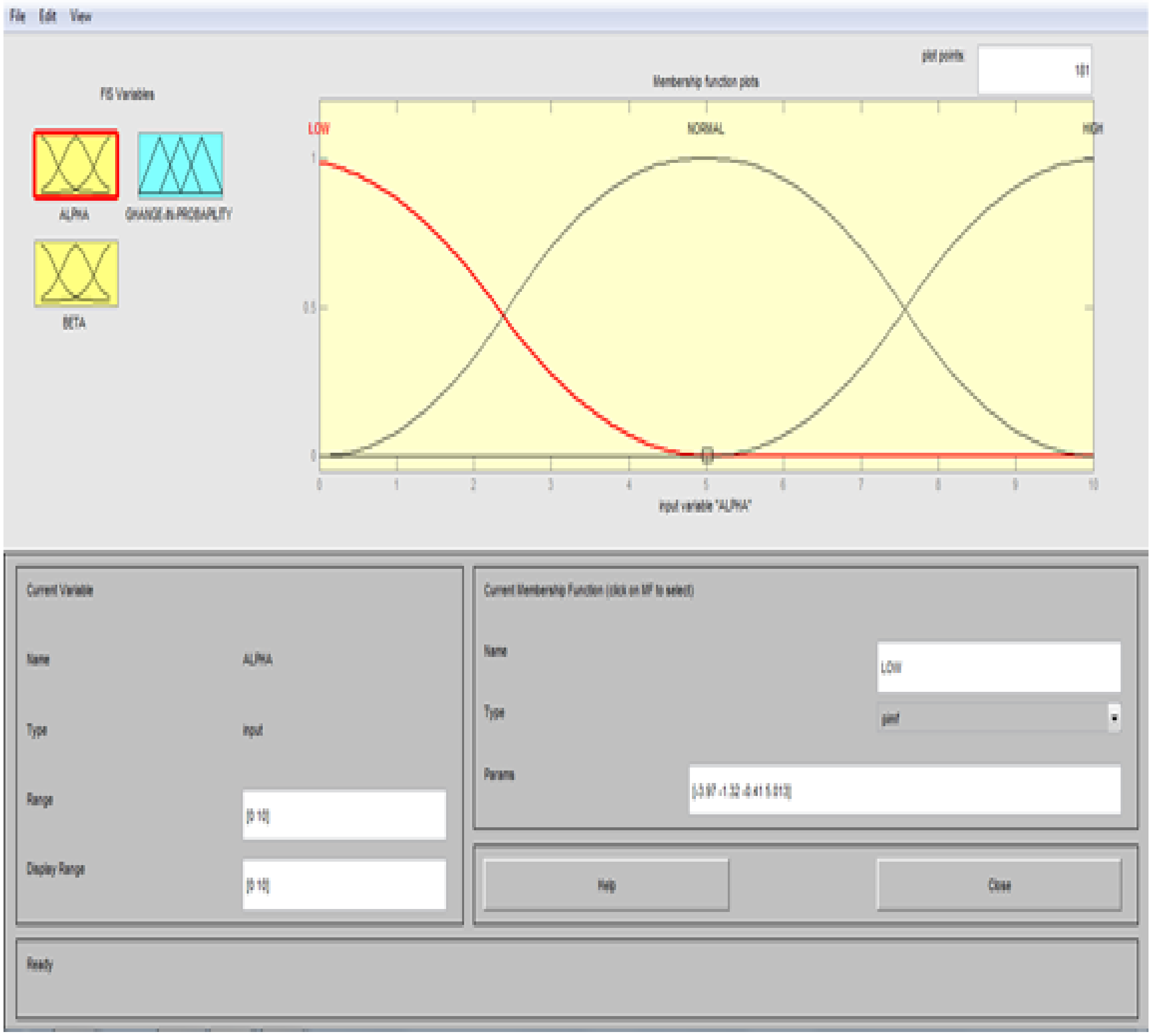

Figure 14-1 FLCD Membership Function of ALPHA

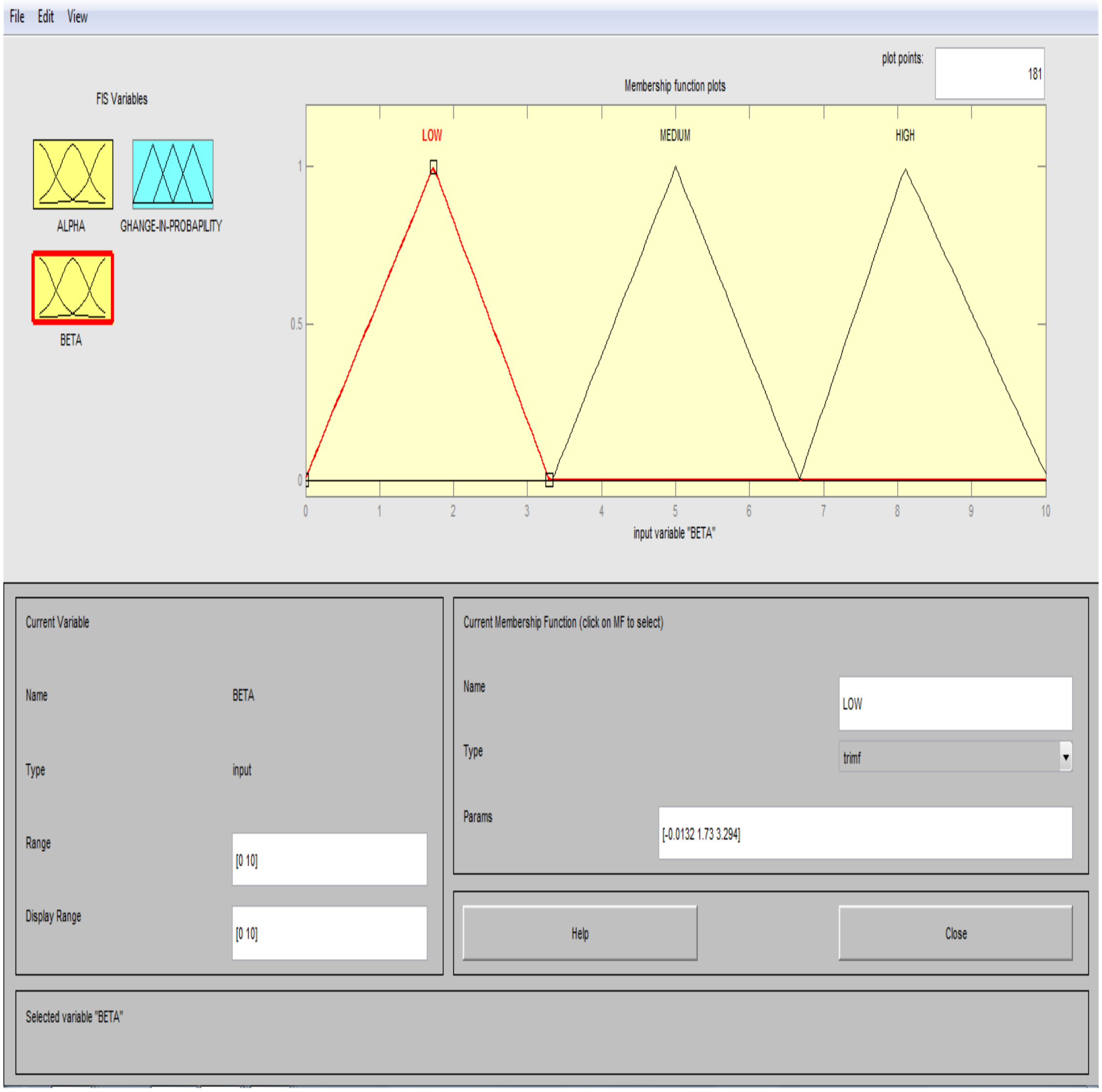

Figure 14-2 FLCD Membership Function of BETA

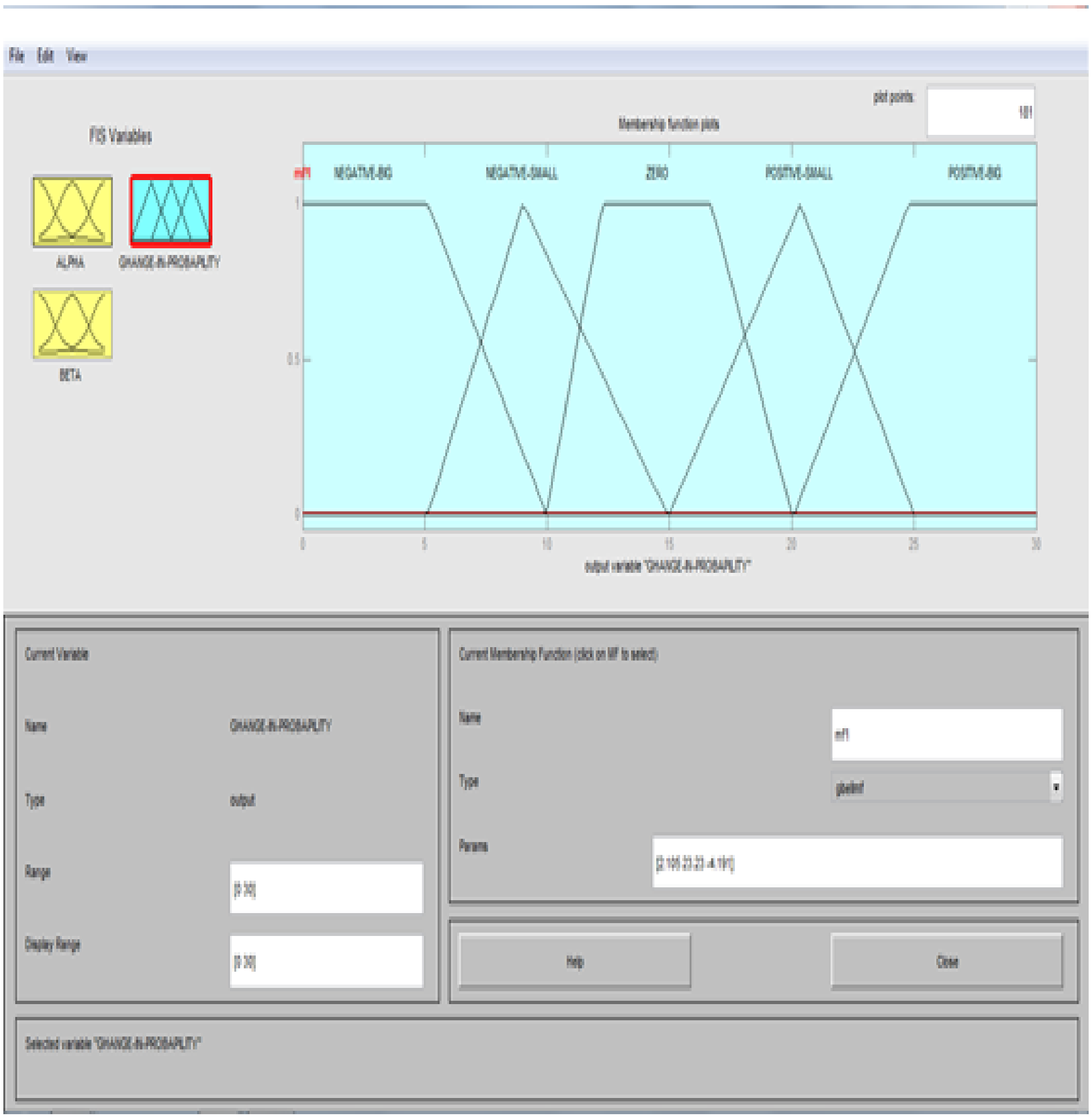

Figure 14-3FLCD Membership Function Change In Probability

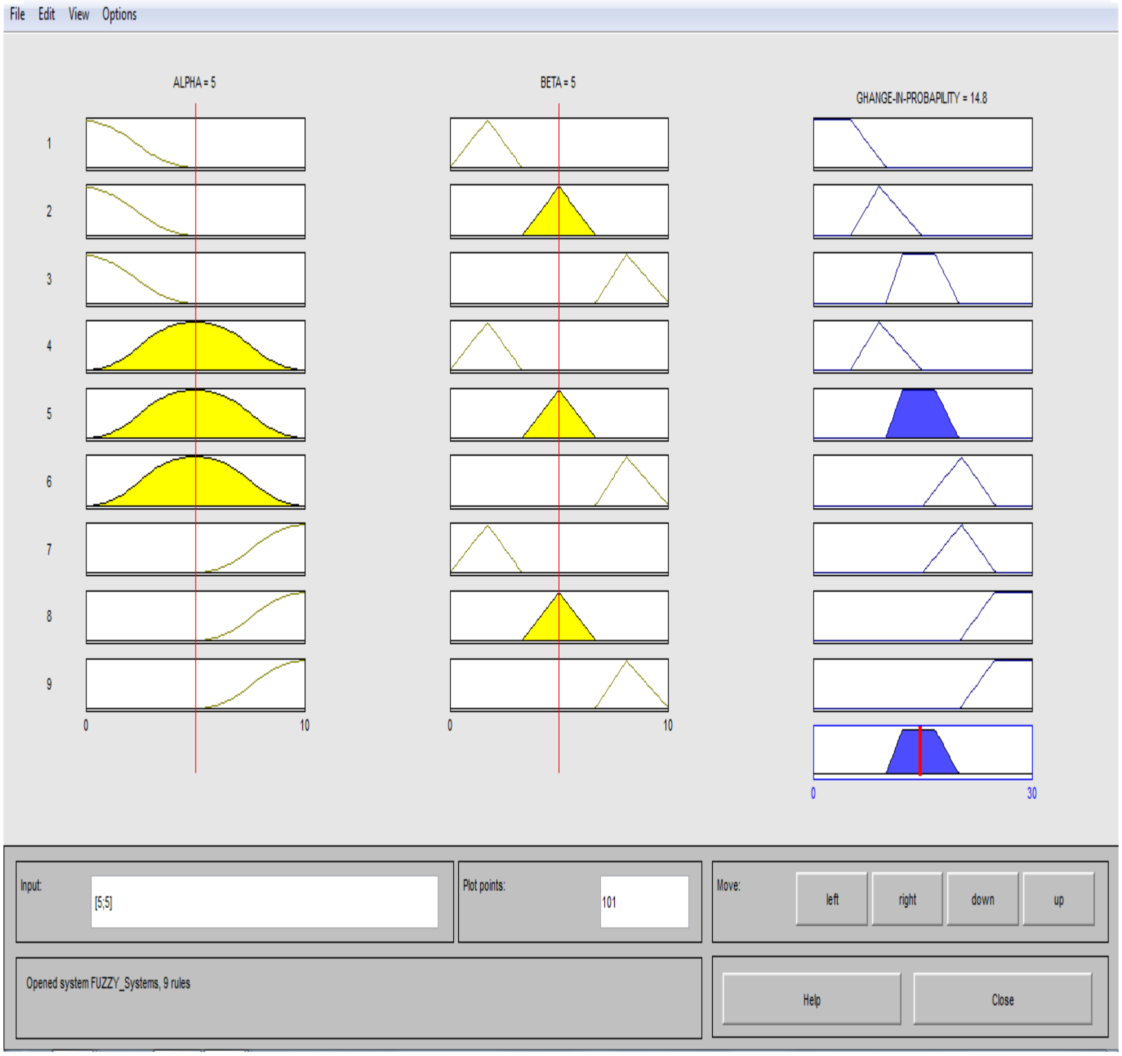

Figure 15 FLCD Rule Viewer

NO.	ALPHA	BETA	GHANGE IN PROPABILITY
1	0.728	1.52	4.17
2	0.28	0.88	4.36
3	0.616	0.843	4.59
4	0.802	0.993	4.62
5	1.21	0.768	5.35
6	0.429	0.0936	5.54
7	2.37	0.918	6.68
8	2.37	0.581	6.89
9	4.9	2.64	9.17
10	4.94	2.53	9.79
11	2.52	5.19	12.9
12	6.06	3.31	15.0
13	8.34	1.07	16.9
14	7.07	3.37	20.2
15	6.36	7.81	21.1
16	6.14	6.91	21.8
17	6.55	9.76	22.8

Table 3 Results Of FLCD Rule Viewer

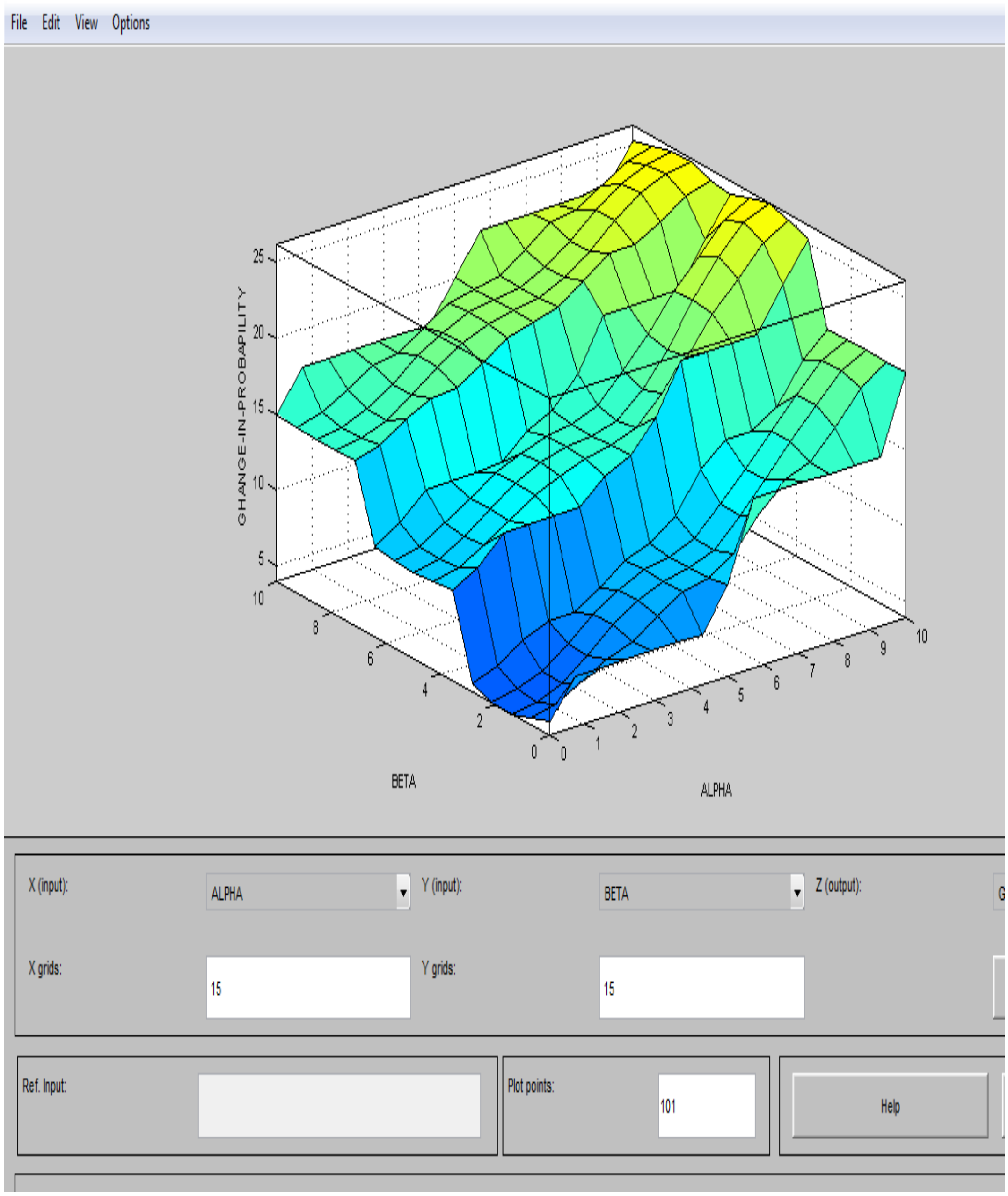

Figure 16 FLCD Surface Viewer

3.2 FLC Algorithm Simulation Results:

The congestion gradient factor is determined using fuzzy logic . The inputs to determine the congestion gradient factor are delay of the packets that are sent, the buffer size of the base station and the data rate . The algorithm is coded using MATLAB mfile and verified using the MATLAB fuzzy logic tool box with FIS editor.

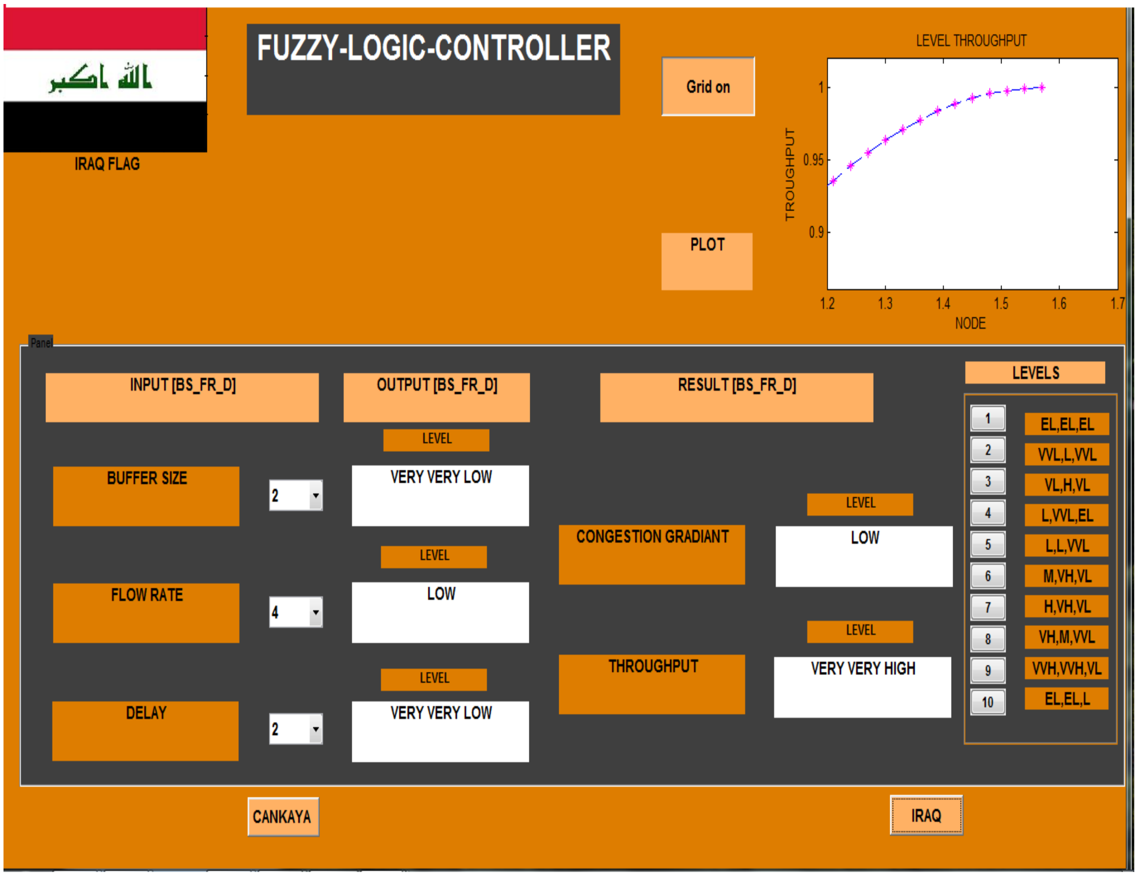

Figure 17 FLC Screen

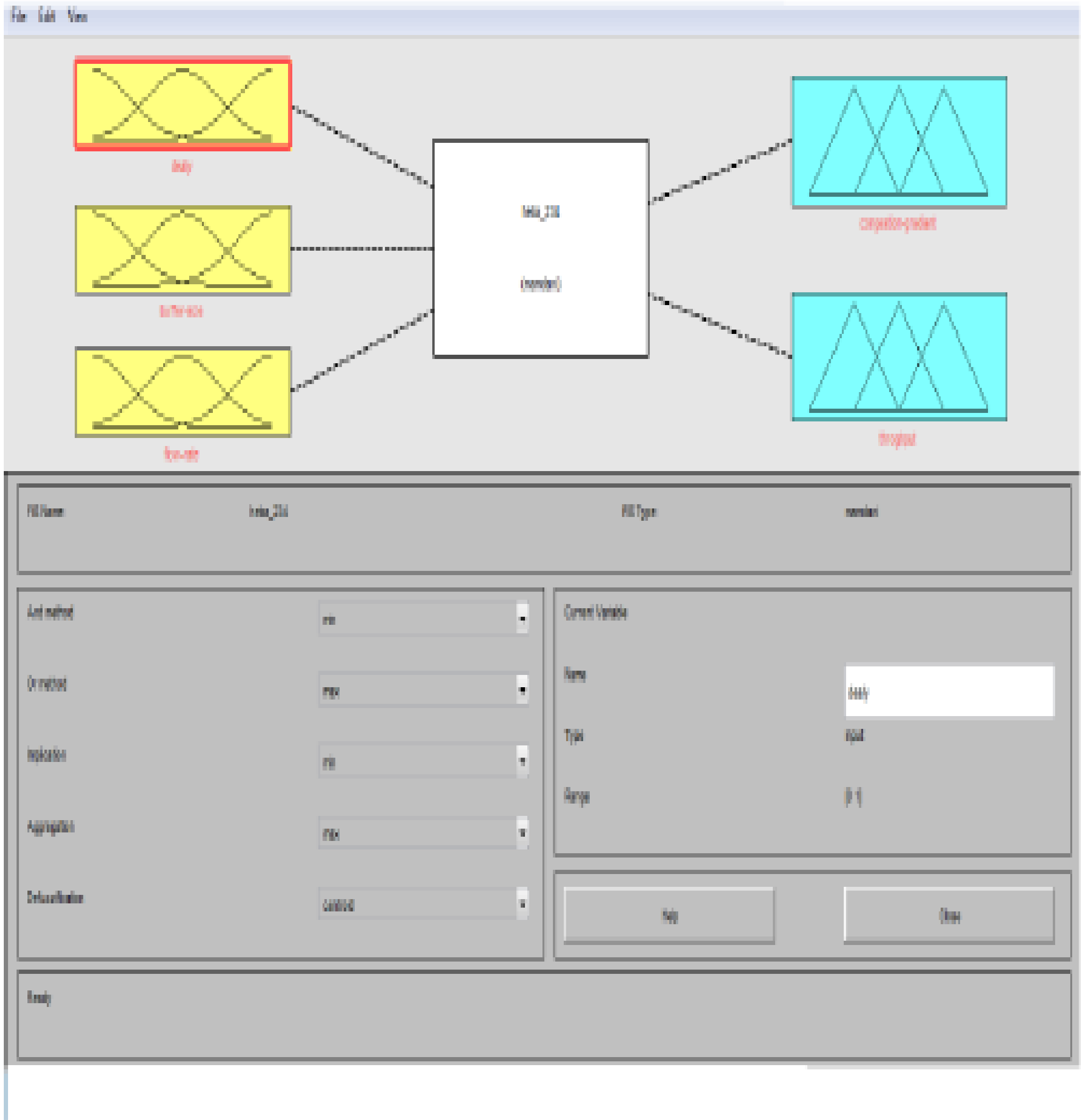

Figure 18 FLC FIS Editor

Figure 19 FLC Rule Viewer

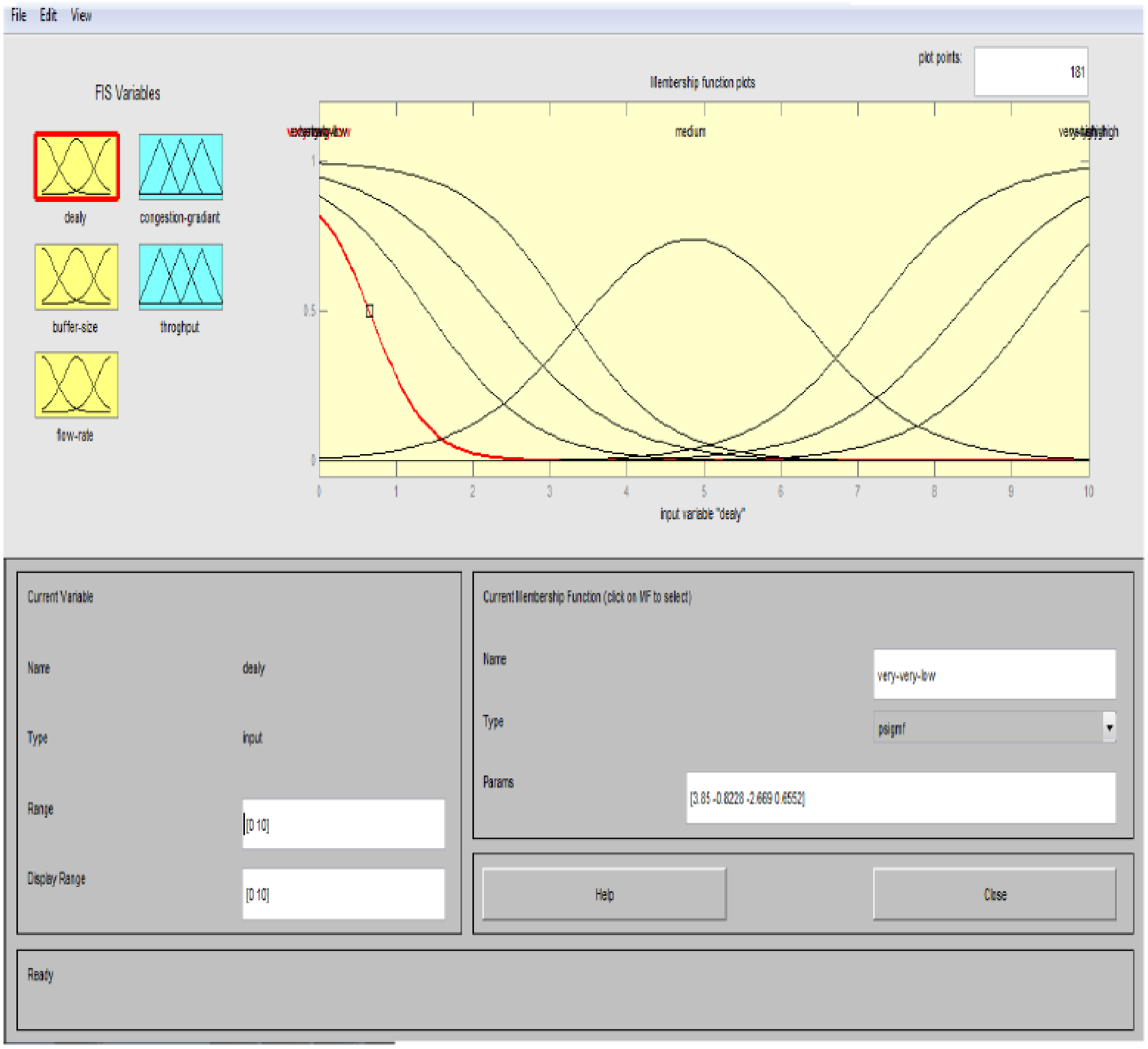

Figure 20.1 FLC Membership Function Of Delay

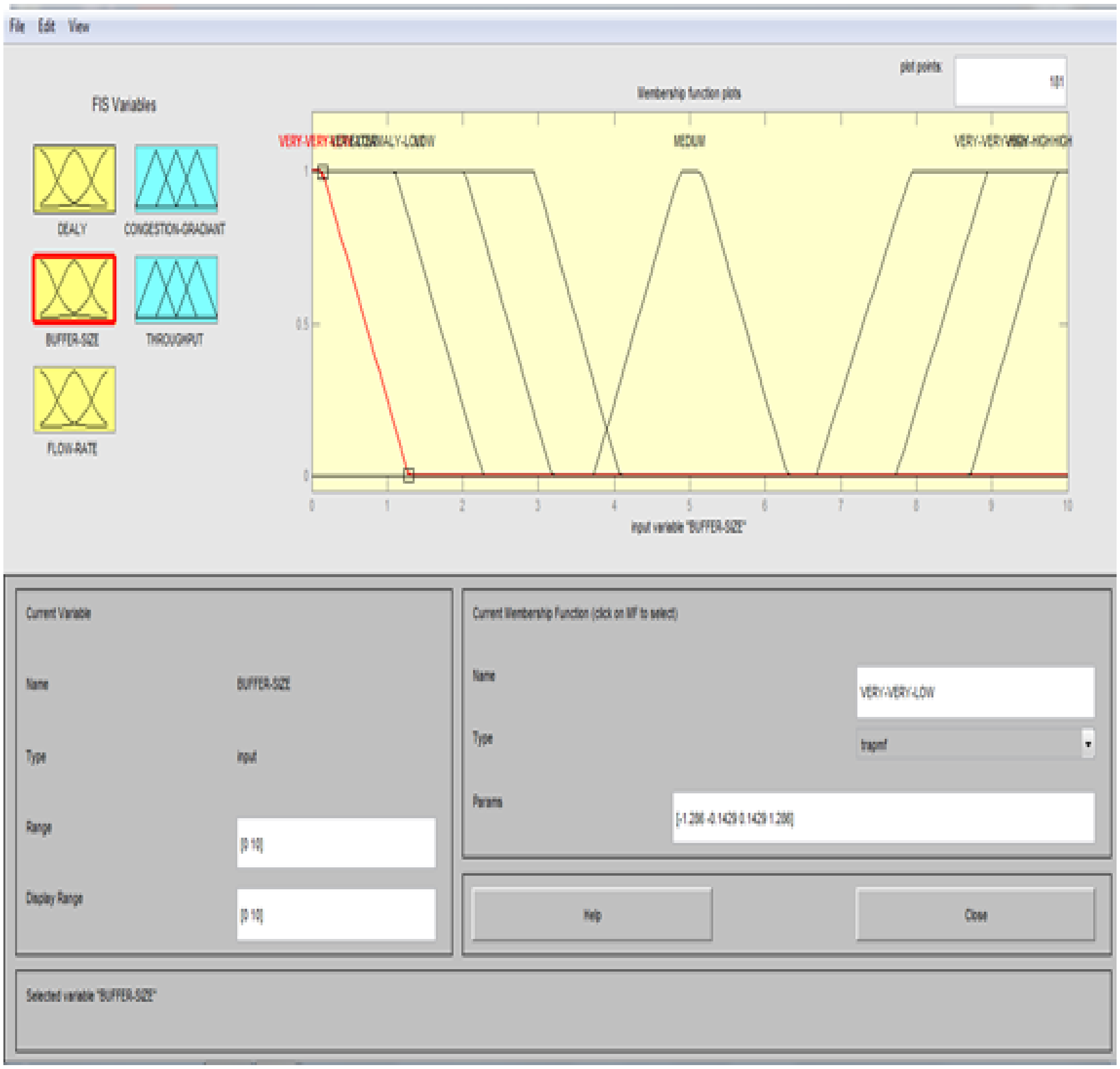

Figure 20-2 FLC Membership Function Of Buffer Size

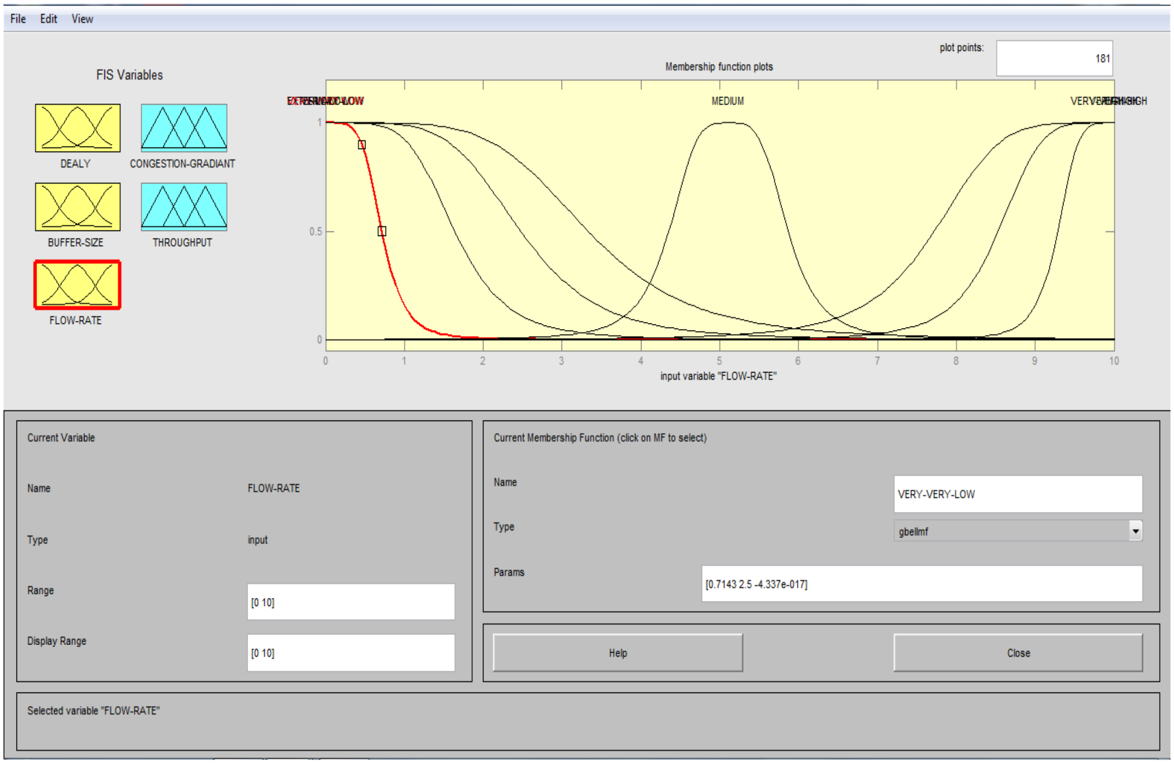

Figure20-3 FLC Membership Function Of Flow Rate

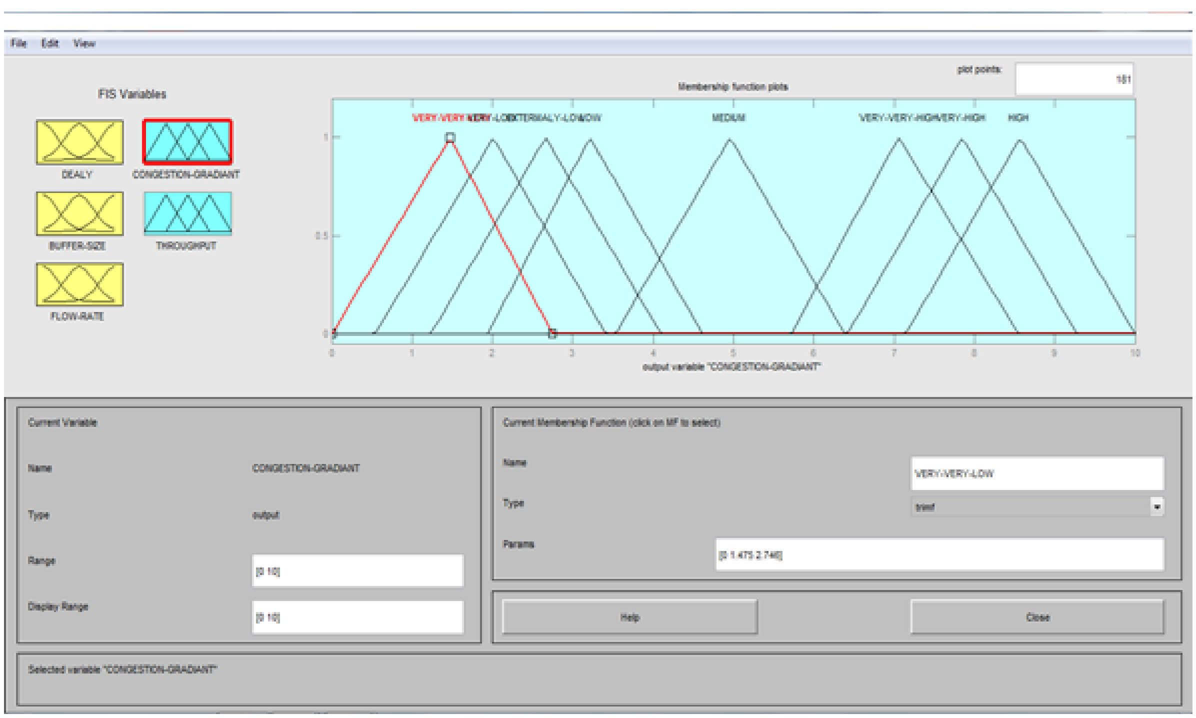

Figure20-4 FLC Membership Function Of Congestion Gradient

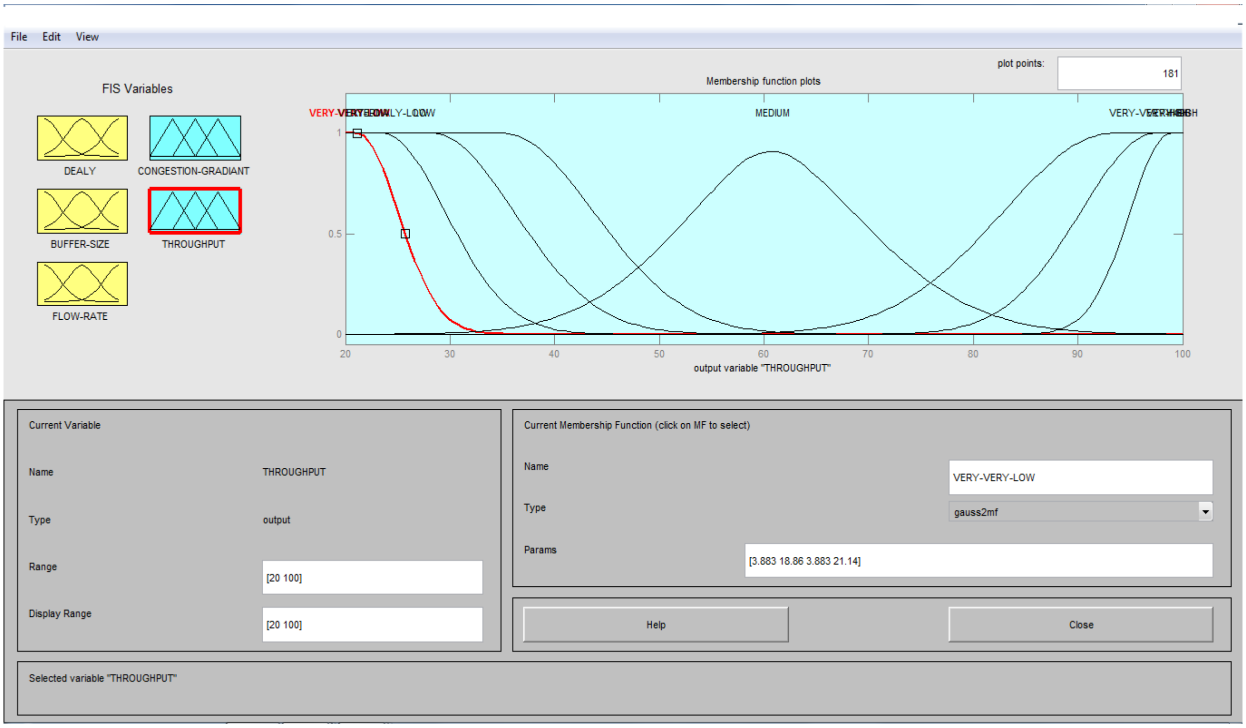

Figure20-5 FLC Membership Function Of Throughput

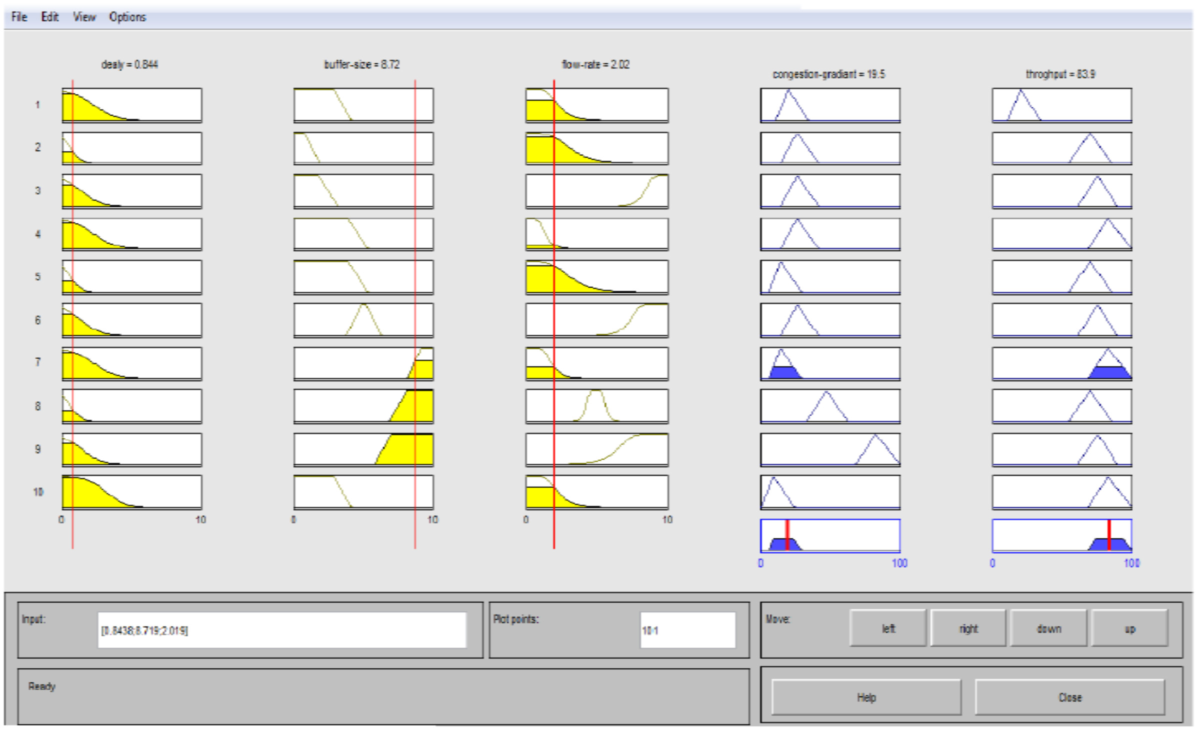

Figure 21 FLC Rule Viewer

NO	DELAY	BUFFER SIZE	FLOW RATE	CONGESTION GRADIANT	THROUGHPUT
1	1.97	2.78	2.39	4.87	49.8 %
2	2.97	2.66	5.0	4.9	52.6 %
3	0.844	1.22	1.71	4.87	55.4 %
4	0.469	0.969	0.776	4.34	59.1 %
5	0.344	0.531	0.528	4.25	59.8 %
6	7.22	6.34	8.73	5.0	60 %
7	0.281	0.406	0.342	4.23	60.1 %
8	0.156	0.219	0.28	4.22	60.5 %
9	0.313	0.969	2.27	4.2	64.9 %
10	3.84	2.28	1.46	6.11	69.2 %
11	0.406	2.78	1.83	3.85	70.4 %
12	0.531	0.781	4.63	3.29	74.3 %
13	2.19	0.531	8.54	3.23	80.5 %
14	3.91	5.22	6.19	3.3	84.7 %
17	1.78	9.28	8.54	8.51	90.6 %
18	0.0938	9.34	2.39	2.26	91.7 %

Table 4 Results Of FLC Rule Viewer

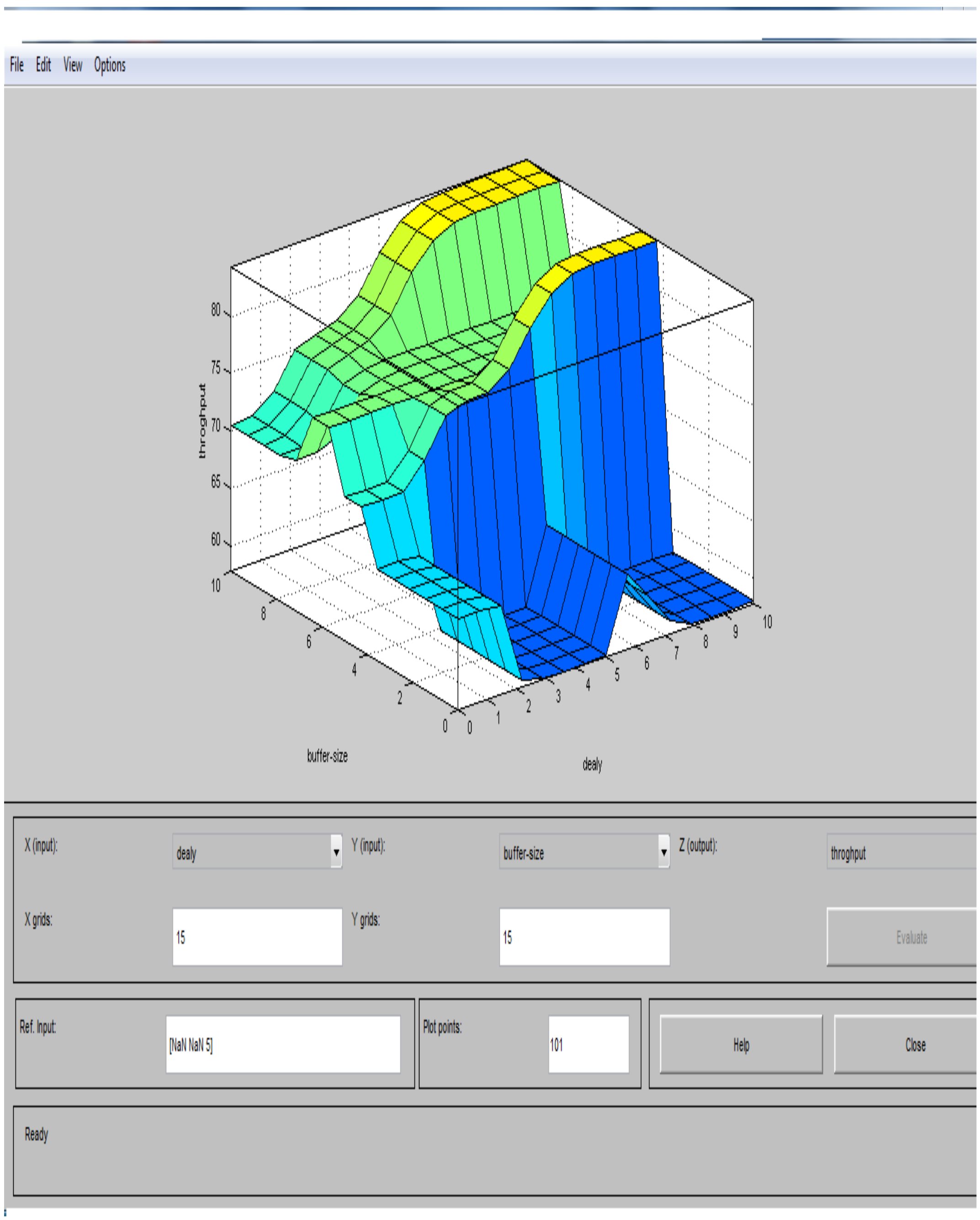

Figure 22 FLC Surface Viewer

CHAPTER IIII

CONCLUSION

This Thesis has proposed online self-organization structures for the FUZZY LOGIC CONGESTION DETECTION (FLCD) algorithm. These structures include an RTT based sampling mechanism and a self-learning and adaptation mechanism. The effectiveness of the proposed approach is proved, Performance results show that the proposed approach shows a robust performance, Apart from enhancing the robustness of the FLCD algorithm, these structures also reduce UDP traffic delay for short round trip propagation delays. They also help to reduce the FLCD algorithm's loss rate. It is also worth mentioning that the addition of the self-organization structures to the FLCD algorithm does not jeopardize other performance metrics, such as utilization, jitter and fairness.

And we proposed a FUZZY LOGIC CONTROLLER which improves the Quality of service parameters in cellular network. The fuzzy controller algorithm combines the input parameters such as buffer size, data rate and delay to find the congestion gradient factor. Unlike the normal sorting procedure for controlling congestion, the crisp congestion gradient factor is calculated by the fuzzy controller based on the above inputs which are derived from the network.

REFERANCES

[1] **AZAR, A**. (2010), Fuzzy systems , ISBN 978-953-7619-92-3. USA.

[2] **BOGDAN, Z.** (2005), Fuzzy Controller Design , ISBN 416. Zagreb,Croatia.

[3] **PASSINO, K.M., STEPHEN, Y.** (1998), Fuzzy Control, ISBN 0-201-18074-X.Addison Wealey Longman, Menlo Park, CA.

[4] **ATOUFI, B.** (2010), TMRF e-BOOK ,chapter 7,Bio-Inspired Algorithems for Fuzzy Rule-Based Systems , Vol .1, PP. 126-159.

[5] **IBRAHEM, I.** (2010), New Applications And Developments of Fuzzy Systems, pp. 87. Korea.

[6] **LAKHAMI, C.J.; MARTIN, N.M.** (1998), Fusion of Neural Networks,Fuzzy systems and Genetic Algorithms:Industrial Applications, ISBN 0849398045. School of electrical and informaion engineering/University of south Australia.

[7] **BAUER, S., CLARK, D.,** and **LEHR,W.** (2009), The evolution of internet congestion, pp. 34,Massachusetts Institute of Technology.

[8] **GROSS, D., HARRIS, C.M.** (1998), Fundamentals of Queueing Theory, Wiley Series in probability and statistics.

[9] **NYIRENDA, C.N., DAWOUD, D.S.** (2008), Self-Organization in a particle Swarm Optimized Fuzzy Logic Congestion Detection Mechanism for IP Networks,Vol.15,PP.589-604,Scientia Iranica.

[10] **FENGYUAN, R., YONG, R., XIUMING, S.** (2002), Design of a fuzzy comtroller for active queue management , PP. 874-883, Computer Communictions,25.

[11] **SMITHA, K., ANAND, H.U., MALLAPUR, J.D.** (2011), Fuzzy Based Congestion Control IN Wireless Network, Vol. 2 No.2 , PP. 469-473.VTU university,India.

[12] **QAZI, I.A., ZNATI, T.,** And **ANDREW, L.L.H.** (2009), Congestion Control Using Efficient Explicit Feedback , proceedings of the IEEE INFOCOM, Rio de Janeiro, Brazil, http://leecenter.caltech.edu/workshop09/papers/low7.pdf.

[13] **DOVROLIS, C., STILIADIS, D.,**and **RAMAMATHAN, P.** (2002), Proportional differentiated services:Delay differentiation and packet scheduling, IEEE /ACM Transaction on networking, 10(1) 12-26.

[14] **MALLI, M., QIANG, Ni., THIERRY, T.,** and **CHEDI, B.** (2004), Adaptive Fair Channel Allocation for QOS Enhancement in IEEE 802.11 Wireless Lans.

[15] **Cisco**, Weighted Random Early Detection on the Cisco 12000 Series Router, http://www.cisco.com/univercd/cc/td/doc/product/software/ios112/ ios11 2p/ gsr /wred_gs.htm: cisco Ltd., (2002).

[16] **ALLMAN, M.** et. al. (2002), Increasing TCP's initial Window, RFC 3390.

[17] **COELLO, C.A., PULIDO, G.T.** and **LECHUNGA, M.A**. (2004), Handling multiple objectives with particle swarm optimazation, 8(3), PP. 256-279, In proceedings of the 2002 Congress on Evolutionary Computation, In IEEE Transactions on Evolutionary Computation,8(3).

[18] **POMARES, H.** et. al. (2004), Online global learning in direct fuzzy controllers, IEEE Ttransactions on fuzzy systems, 12(2).

[19] **LOW, S.H.** et. al. (2002), Dynamics of TCP/AQM and ascalable control, in proc. IEEE INFOCOM.

[20] **ATHURALIYA, S.** et. al**.** (2001), REM: Active queue management, IEEE network magazine, 15 (3),pp. 48-53.

[21] **SKRIANC, I.** et. al. (2002), Direct fuzzy model reference adaptive control, International journal on Intelligent systems, pp. 943-963.

[22] **RAMESH, B., MANJNLA, D.** (2007), An Adaptive congestion control mechnism for streaming Multimedia in mobile Ad-hoc networks, No.6, pp. 290- 295.IJCSNS International Journal of Computer Science and Network Security,7.

[23] **HSU-FENG, H.** et. al**.** (2005), A new Multimedia pachet loss Classification Algorithm for congestion control over wired/wireless channels, pp . ii/1105-ii/1108.IEEE International Conference,2.

[24] **OYEBISI, T.O., OJESANMI, O.A**. (2008), Development of congestion control scheme for wireless mobile network, pp. 965-972,Journal of Theoretical and Applied Information Technology.

[25] **LEITH, D.** et. al**.** (2007), Delay-based AIMD congestion control, pp. 1-6.

[26] **LEI, H., JAY KUO, C.C.** (2004), Dynamic call Admission control with joint connection-level and packet-level Qos support in Wireless Multimedia Networks, IEEE Real-Time and Embeded Technology and Applications Symposium.

[27] **BOELAZE, M., ELNAGGAR, A.,** and **MUSLEH, M** (2004), A priority based call admision control protocol with call degradation for cellular networks, proceedings of the 1st international symposium on wireless communication system ISWCS port Louis,Mauritius 20-22.

[28] **YANFEI, F., CHUANG, F.L.** (2003), Design of a PID controller for active queue management.

[29] **RAMAKRISHNAN, K., FLOYD, S.** (2001), The addition of explicit congestion notification (ECN) to IP, RFC-3168.

APPENDIX: A

PSEUDO CODE OF FLCD AND FLC:

```
function varargout = FLC_FLCD(varargin)
% FLC_FLCD MATLAB code for FLC_FLCD.fig
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @FLC_FLCD_OpeningFcn, ...
                   'gui_OutputFcn',  @FLC_FLCD_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function FLCD_Callback(hObject, eventdata, handles)
[nombre dire]=uigetfile('*.jpg','IRAQ');
if nombre==0
    return
end
imagen= imread(fullfile(dire,nombre));
axes(handles.image)
image(imagen)
handles.img= imagen;
guidata(hObject,handles)
function FLC_Callback(hObject, eventdata, handles)
[nombre dire]=uigetfile('*.jpg','IRAQ');
if nombre==0
    return
end

imagen= imread(fullfile(dire,nombre));
axes(handles.image1)
image(imagen)
handles.img= imagen;

guidata(hObject,handles)
```

```
function varargout = MOHAMMED_NSAIF(varargin)
% MOHAMMED_NSAIF MATLAB code for MOHAMMED_NSAIF.fig
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...

                   'gui_Singleton',  gui_Singleton, ...

                   'gui_OpeningFcn', @MOHAMMED_NSAIF_OpeningFcn, ...
                   'gui_OutputFcn',  @MOHAMMED_NSAIF_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function popupmenu2_Callback(hObject, eventdata, handles)
a=get(handles.popupmenu2,'value');
if (a==1)
    set(handles.text2,'string','EXTERMLY LOW');
elseif (a==2)
    set(handles.text2,'string','VERY VERY LOW');
elseif (a==3)
    set(handles.text2,'string','VERY LOW');
elseif (a==4)
    set(handles.text2,'string','LOW');
elseif (a==5)
    set(handles.text2,'string','MEDIUM');
elseif (a==6)
    set(handles.text2,'string','HIGH');
elseif (a==7)
    set(handles.text2,'string','VERY HIGH');
elseif (a==8)
    set(handles.text2,'string','VERY VERY HIGH');
elseif (a==9)
    set(handles.text2,'string','EXTERMLY');
elseif (a==10)
    set(handles.text2,'string','EXTERMLY LOW');
elseif (a==11)
    set(handles.text2,'string','NOTHING');
end
```

```
function popupmenu3_Callback(hObject, eventdata, handles)
a=get(handles.popupmenu3,'value');
if (a==1)
   set(handles.text3,'string','EXTERMLY LOW');
elseif (a==2)
   set(handles.text3,'string','VERY VERY LOW');
elseif (a==3)
   set(handles.text3,'string','VERY LOW');
elseif (a==4)
   set(handles.text3,'string','LOW');
elseif (a==5)
   set(handles.text3,'string','MEDIUM');
elseif (a==6)
   set(handles.text3,'string','HIGH');
elseif (a==7)
   set(handles.text3,'string','VERY HIGH');
elseif (a==8)
   set(handles.text3,'string','VERY VERY HIGH');
elseif (a==9)
   set(handles.text3,'string','EXTERMLY LOW');
elseif (a==10)
   set(handles.text3,'string','VERY VERY HIGH');
elseif (a==11)
   set(handles.text3,'string','NOTHING');
end
function popupmenu4_Callback(hObject, eventdata, handles)
a=get(handles.popupmenu4,'value');
x=1:.03:pi/2;
if  (a==1)
   set(handles.text1,'string','EXTERMLY LOW');
   set(handles.text5,'string','EXTERMLY HIGH');
   set(handles.text9,'string','EXTERMLY LOW');
elseif (a==2)
   set(handles.text1,'string','VERY VERY LOW');
   set(handles.text5,'string','VERY VERY HIGH');
   set(handles.text9,'string','LOW');
   axes(handles.axes1)
   plot(x,sin(x),'-- *b','markeredgecolor','m','markerfacecolor',
[1 0 1])
   axis([1.2 1.7 0.86 1.02])
   title('LEVEL THROUGHPUT')
   xlabel('NODE')
   ylabel('TROUGHPUT')
   legend('TROUGHPUT','--*b','Location','northwest')
elseif (a==3)
   set(handles.text1,'string','VERY LOW');
   set(handles.text5,'string','VERY HIGH');
   set(handles.text9,'string','LOW');
```

```
    axes(handles.axes1)
    plot(x,sin(x),'-- sb','markeredgecolor','m','markerfacecolor',[1 1 0])
    axis([1.2 1.4 0.82 1.02])
    title('LEVEL THROUGHPUT')

    xlabel('NODE')
    ylabel('TROUGHPUT')
    legend('TROUGHPUT','--sb','Location','northwest')
elseif (a==4)
    set(handles.text1,'string','EXTERMALY LOW');
    set(handles.text5,'string','EXTERMALY HIGH');
    set(handles.text9,'string',' LOW');
    axes(handles.axes1)
    plot(x,sin(x),'-- *b','markeredgecolor','m','markerfacecolor',
[1 0 1])
    axis([1.2 1.4 0.82 1.02])
    title('LEVEL THROUGHPUT')
    xlabel('NODE')
    ylabel('TROUGHPUT')
    legend('TROUGHPUT','--sb','Location','northwest')
elseif (a==5)
    set(handles.text1,'string','VERY VERY LOW');
    set(handles.text5,'string','VERY VERY HIGH');
    set(handles.text9,'string','VERY LOW');


elseif (a==6)
    set(handles.text1,'string','VERY LOW');
    set(handles.text5,'string','VERY HIGH');
    set(handles.text9,'string','LOW');
    axes(handles.axes1)
    plot(x,sin(x),'-- sb','markeredgecolor','m','markerfacecolor',[0 0 0])
    axis([1.2 1.7 0.86 1.02])
    title('LEVEL THROUGHPUT')
    xlabel('NODE')
    ylabel('TROUGHPUT')
    legend('TROUGHPUT','--sb','Location','northwest')
elseif (a==7)
    set(handles.text1,'string','EXTERMALY LOW');
    set(handles.text5,'string','EXTERMALY HIGH');
    set(handles.text9,'string','VERY LOW');
    axes(handles.axes1)
    plot(x,sin(x),'-- sb','markeredgecolor','m','markerfacecolor',
[0 1 0])
    axis([1.2 1.3 0.82 1.02])
    title('LEVEL THROUGHPUT')
    xlabel('NODE')
    ylabel('TROUGHPUT')
```

```
    legend('TROUGHPUT','--sb','Location','northwest')
elseif (a==8)
    set(handles.text1,'string','VERY VERY LOW');
    set(handles.text5,'string','VERY VERY HIGH');
    set(handles.text9,'string','MEDIUM');

    axes(handles.axes1)
    plot(x,sin(x),'-- sb','markeredgecolor','m','markerfacecolor',
[1 0 0])
    axis([1.2 1.7 0.86 1.02])
    title('LEVEL THROUGHPUT')
    xlabel('NODE')
    ylabel('TROUGHPUT')
    legend('TROUGHPUT','--sb','Location','northwest')
elseif (a==9)
    set(handles.text1,'string','VERY LOW');
    set(handles.text5,'string','VERY HIGH');
    set(handles.text9,'string','HIGH');
    axes(handles.axes1)
    plot(x,sin(x),'-- sb','markeredgecolor','m','markerfacecolor',[0 0 1])
    axis([1.2 1.7 0.86 1.02])
    title('LEVEL THROUGHPUT')
    xlabel('NODE')
    ylabel('TROUGHPUT')
    legend('TROUGHPUT','--sb','Location','northwest')
elseif (a==10)
    set(handles.text1,'string','LOW');
    set(handles.text5,'string','HIGH');
    set(handles.text9,'string','VERY VERY LOW');
    axes(handles.axes1)
    plot(x,sin(x),'-- sb','markeredgecolor','m','markerfacecolor',[0 0 0])
    axis([1.2 1.7 0.80 1.02])
    title('LEVEL THROUGHPUT')
    xlabel('NODE')
    ylabel('TROUGHPUT')
    legend('TROUGHPUT','--sb','Location','northwest')
elseif (a==11)
    set(handles.text1,'string','NOTHING');
   set(handles.text5,'string','NOTHING');
    set(handles.text9,'string','NOTHING');
end
function G_Callback(hObject, eventdata, handles)
if get(handles.G,'value')==get(handles.G,'max')
    grid on
else
    grid off
end
```

```
axes(handles.axes1)
function iraq_Callback(hObject, eventdata, handles)
[nombre dire]=uigetfile('*.jpg','IRAQ');
if nombre==0
   return

end
imagen= imread(fullfile(dire,nombre));
axes(handles.axes2)
image(imagen)
handles.img= imagen;
guidata(hObject,handles)
function cankaya_Callback(hObject, eventdata, handles)
[nombre dire]=uigetfile('*.jpg','IRAQ');
if nombre==0
   return
end

imagen= imread(fullfile(dire,nombre));
axes(handles.axes3)
image(imagen)
handles.img= imagen;
guidata(hObject,handles)
function p2_Callback(hObject, eventdata, handles)
set(handles.text8,'string','EL,EL,EL')
function p3_Callback(hObject, eventdata, handles)
set(handles.text39,'string','VVL,L,VVL')
function p4_Callback(hObject, eventdata, handles)
set(handles.text11,'string','VL,H,VL')
function p5_Callback(hObject, eventdata, handles)
set(handles.text12,'string','L,VVL,EL')
function p6_Callback(hObject, eventdata, handles)
set(handles.text13,'string','L,L,VVL')
 function p7_Callback(hObject, eventdata, handles)
set(handles.text14,'string','M,VH,VL')
function p8_Callback(hObject, eventdata, handles)
set(handles.text15,'string','H,VH,VL')
function p9_Callback(hObject, eventdata, handles)
set(handles.text16,'string','VH,M,VVL')
function p10_Callback(hObject, eventdata, handles)
set(handles.text17,'string','VVH,VVH,VL')
 function p1_Callback(hObject, eventdata, handles)
set(handles.text18,'string','EL,EL,L')
function pushbutton25_Callback(hObject, eventdata, handles)
set(handles.text20,'string','L,VVL,EL')

function varargout = MOHAMMED_MOSTAFA(varargin)
% MOHAMMED_MOSTAFA MATLAB code for MOHAMMED_MOSTAFA.fig
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @MOHAMMED_MOSTAFA_OpeningFcn, ...
                   'gui_OutputFcn',  @MOHAMMED_MOSTAFA_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...

                   'gui_Callback',   []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
function popupmenu1_Callback(hObject, eventdata, handles)
a=get(handles.popupmenu1,'value');
if     (a==1)
    set(handles.text12,'string','LOW');
elseif (a==2)
    set(handles.text12,'string','NORMAL');
elseif (a==3)
    set(handles.text12,'string','HIGH');
end

function popupmenu2_Callback(hObject, eventdata, handles)
a=get(handles.popupmenu2,'value');
if (a==1)
    set(handles.text13,'string','LOW');
    set(handles.text15,'string','NEGATIVE BIG');
elseif (a==2)
    set(handles.text13,'string','MEDIUM');
    set(handles.text15,'string','NEGATIVE SMALL');
elseif (a==3)
    set(handles.text13,'string','HIGH');
    set(handles.text15,'string','ZER0');
elseif (a==4)
    set(handles.text13,'string','LOW');
    set(handles.text15,'string','NEGATIVE SMALL');
elseif (a==5)
    set(handles.text13,'string','MEDIUM');
    set(handles.text15,'string','ZERO');
elseif (a==6)
    set(handles.text13,'string','HIGH');
    set(handles.text15,'string','POSITIVE SMALL');
```

```
elseif (a==7)
    set(handles.text13,'string','LOW');
    set(handles.text15,'string','POSITIVE SMALL');
elseif (a==8)
    set(handles.text13,'string','MEDIUMH');
    set(handles.text15,'string','POSITIVE BIG');
elseif (a==9)

    set(handles.text13,'string','HIGH');
    set(handles.text15,'string','POSITIVE BIG');

end


function p2_Callback(hObject, eventdata, handles)
set(handles.text1,'string','LOW,LOW')
function p3_Callback(hObject, eventdata, handles)
set(handles.text2,'string','LOW,MEDIUM')
function p4_Callback(hObject, eventdata, handles)
set(handles.text3,'string','LOW,HIGH')
function p5_Callback(hObject, eventdata, handles)
set(handles.text4,'string','NORMAL,LOW')
function p6_Callback(hObject, eventdata, handles)
set(handles.text5,'string','NORMAL,MEDIUM')
function p9_Callback(hObject, eventdata, handles)
set(handles.text8,'string','HIGH,MEDIUM')
function p10_Callback(hObject, eventdata, handles)
set(handles.text9,'string','HIGH,HIGH')
function p7_Callback(hObject, eventdata, handles)
set(handles.text6,'string','NORMAL,HIGH')
function p8_Callback(hObject, eventdata, handles)
set(handles.text7,'string','HIGH,LOW')
function iraq_Callback(hObject, eventdata, handles)
[nombre dire]=uigetfile('*.jpg','IRAQ');
if nombre==0
    return
end
imagen= imread(fullfile(dire,nombre));
axes(handles.axes2)
image(imagen)
handles.img= imagen;
guidata(hObject,handles
```

APPENDIX: B

CURRICULUM VITAE

PERSONAL INFORMATION

Name,Surname :Mohammed ,Nsaif
Nationality:Iraq
Date and Place of Birth: 14 July1979 , Anbar
Marital Status:Married
Phone : +905353809821/+9647901251566
Email:mohammednsif@yahoo.com

EDUCATION

Degree	Institution	Year of Graduation
MS	Çankaya Univ. Computer Engineering	2011-2012
BS	Computer.Technology Univ Enginnering	2004-2005
High School	AL-High School Husain, Iraq	2001

WORK EXPERIENCE

Year	Place	Enrollment
2005	Baghdad	A coordinated for one of the food companies
2005	Baghdad/ Telecom Asiacel	Data Entry and Maintenance Engineering
2006-Present	Ministry of /Baghdad Higher Education	Maintenance Engineering

FOREIGN LANGUAGES

Advanced English, Arabic

HOBBIES

Driving, Wrestling, Shooting.