*IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250-3501*
*Vol.4, No5, October 2014*

# Execution Time and Failure Rate based model for Reliability Optimization in Distributed Systems

Sagar Gulati[1], K. Bhatia[2] and P. K. Yadav[3]

[1]*CSE Department, Technology Education & Research Institute, Kurukshetra, India.*
[1]*Department of Computer Science, Gurukul Kangri Viswavidyalaya, Haridwar, India.*
[3]*Central Building Research Institute, Roorkee, India.*

[*]**Corresponding author:** Email: sagarg40@gmail.com

***Abstract*— Distributed Systems operate upon the packets of data that rotates throughout the network; Reliability becomes a key parameter in such networks; reliability of information, reliability of link and reliability of nodes plays a vital role in overall performance of the distributed systems. In distributed systems, operations have to be performed timely, correctly and importantly in reliable manner. Finding reliable communication in Distributed System with constraints is shown to be NP-hard. Although some optimal algorithms have been employed in uni-processor systems, they fail when they are applied in multiprocessor and distributed systems. To alleviate these deficiencies, this paper simulates the reliability of a distributed system w.r.t. Execution Time and Failure Rate to enhance the reliability of the distributed systems. The problem has been simulated on Matlab 7.2 Mamdani Fuzzy Inference Engine using input variable as FR (Failure Rate) and ET (Execution Time) and Reliability as the output.**

***Keywords*-Execution time, Failure rate, Reliability, fuzzy, modeling.**

## I. INTRODUCTION

Distributed Systems is a mean to run multiple transactions simultaneously [1], [2]. Dependability, i.e. reliability and availability, is one of the biggest trends in distributed system. In the past, it has been considered acceptable for services to be unavailable because of failures. But with the developments and advancements, the requirement for high reliability and availability is continually increasing in every possible domain of science and technology. One solution for achieving fault-tolerance is to build software on top of fault- tolerant (replicated) hardware. This may indeed be a viable solution for some application classes, and has been successfully pursued by companies such as Tandem and Stratus. Economic factors have, however, motivated the search for cheaper simulation based fault-tolerance [3], [4].

Reliability in distributed systems has to be a major concern for all real time transactions. Various models have been produced to assess or predict reliability of large scale distributed applications, but reliability issues with these systems still exists. Ensuring distributed system's reliability in turns requires examining reliability of each individual component or factors involved in enterprise distributed applications before predicting or assessing reliability of the whole system, and Implementing transparent fault detection and fault recovery scheme to provide seamless interaction to end users. In this paper we have operated upon Matlab Fuzzy Inference Engine to understand the relation between execution time of task and probability of failure (Failure rate) w.r.t. reliability [5] [6].

Performance evaluation is an important task in the design of complex systems, especially of distributed systems. Distributed systems are widely used in control systems for automation, communication systems such as LAN and WAN, and computing systems. On a very abstract level, queuing networks [7], [8] or stochastic Petri nets [9], [10], [11], [12] are used to model the systems and to evaluate their performance. But, the behavior of real-time systems including communication delays, processing periods, and interactions between different processes via the communication channel are difficult to model on this level.

Execution time of task and reliability of communication are two of the most important factors that affect the overall performance of the system. The present research paper is an attempt to study the trade-off between execution time and reliability of a complex distributed system. The problem has been simulated on Mamdani Fuzzy Inference Engine using the tool Matlab version 7.2. In the proceeding section of the paper, fuzzy inference systems, proposed model, experimental results, Conclusion and future works are debated respectively.

## II. FUZZY INFERENCE SYSTEM (FIS)

There are number of different techniques that would work here and therefore a design choice must be made. Some of the techniques require a relatively accurate model of the system in order to develop a satisfactory system. On the other hand Fuzzy Logic does not require a model of the system. Fuzzy logic [13], [14] is a superset of conventional Boolean logic and extends it to deal with new aspects such as partial truth and uncertainty. Fuzzy inference is the process of formulating the mapping from a given input set to an output using fuzzy logic. The basic elements of fuzzy logic are linguistic variables, fuzzy sets, and fuzzy rules [15]. Therefore, with all of this in mind, a Fuzzy Logic expert system is introduced to study and understand the performance of Distributed system w.r.t. different parameters viz. reliability, failure rate, execution time etc. The main advantage of Fuzzy Logic is that it can be tuned and adapted if necessary, thus enhancing the degree of freedom of the system [16]. The schematic diagram of conceptual modeling using Fuzzy Logic is presented in Figure 1.

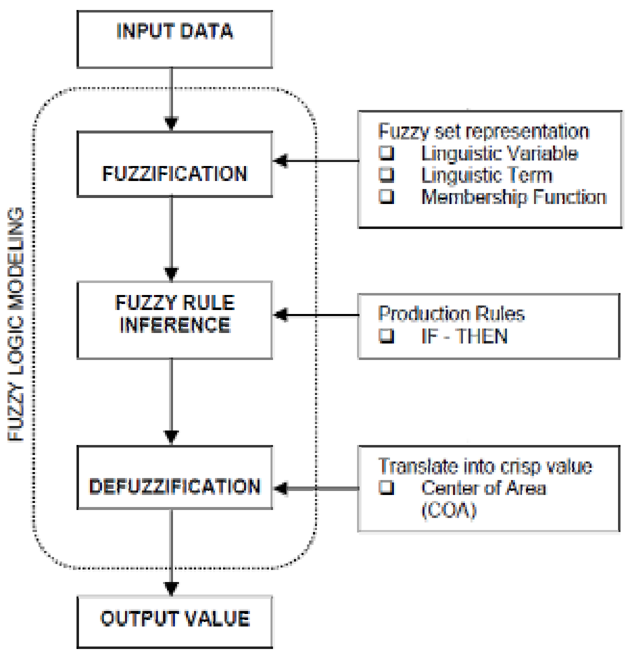


**Figure 1: Conceptual fuzzy logic modeling**

First step in the modeling is called fuzzification, which defines linguistic variables, term and membership functions. Originally the concept of linguistic variable was introduced by Zadeh [17]. The second step is called fuzzy rule inference whereby some sets of fuzzy logic operators and production rules are defined. The most common rule is called IF-THEN rule which can be used to formulate the conditional statements that comprise fuzzy logic. The final step in modeling is defuzzification process. The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. The aggregate of fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set. The detail discussion about the modeling steps is presented in the following sections.

There are two types of fuzzy inference models:

1. Mamdani [18],
2. TSK or Sugeno [19].

Interpreting an if-then rule involves two distinct parts: first evaluating the antecedent and then applying results to the consequent (known as implication) [20], [21]. In the case of two-valued or binary logic, if-then rules do not present much difficulty. If the premise is true, then the conclusion is true, whereas with fuzzy approach, if the antecedent is true to some degree of membership, then the consequent is also true to that same degree.

Mamdani-type [18] inference 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. It is possible, and in many cases much more efficient, to use a single spike as the output's membership function rather than a distributed fuzzy set. This is sometimes known as a singleton output membership function, and it can be thought of as a pre-defuzzified fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation required by the more general Mamdani method, which finds the centroid of a two-dimensional function. Rather than integrating across the two-dimensional function to find the centroid, Sugeno-type systems use weighted sum of a few data points. In general, Sugeno-type systems can be used to model any inference system in which the output membership functions are either linear or constant.

## III. THE PROPOSED MODEL

The block diagram of our inference system is presented in Figure 2.

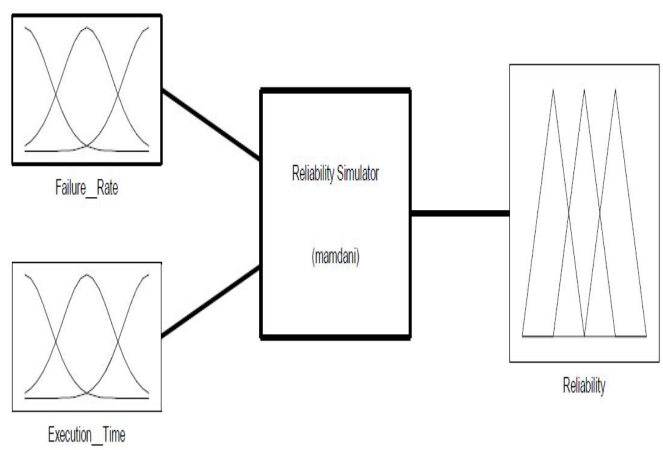


**Figure 2: Inference System Block Diagram**

For implementation MATLAB fuzzy logic toolbox was used. For prediction of reliability of the system by using Fuzzy inference system, Failure rate and execution time were used as input parameters and Reliability was used as output.

**Execution Time:** The execution time $et_{ij}$ Where $1 \leq i \leq m$, $1 \leq j \leq n$ of each task $t_i$ depends on the processor $p_j$ to which it is assigned and the work to be performed by each of tasks of that processor $p_j$.

$$ET = \sum_{j=1}^{n} et_{ij} x_{ij}, i = 1,2,...m \quad (1)$$

Where $x_{ij}$ is the $\begin{pmatrix} 1, & \text{if task } t_i \text{ is assigned to procerssor } p_j \\ 0, & \text{otherwise} \end{pmatrix}$

**Failure Rate:** The Failure rate [FR] that task ti shall not get executed per unit time interval on the processor pj is the probability FRij (1≤ i ≤ m and 1≤ j ≤ n), that task ti will not be successfully executed on processor pj, per unit time interval.

$$FR = \prod_{i=1}^{m} \left\{ \sum_{j=1}^{n} FR_{ij} x_{ij} \right\} \quad (2)$$

**Reliability:** The Reliability [R] of a task ti on the processor pj is the probability Rij (1≤ i ≤ m and 1≤ j ≤ n), that task ti will be successfully executed on processor pj, within specified conditions.

$$R = \prod_{i=1}^{m} \left\{ \sum_{j=1}^{n} R_{ij} x_{ij} \right\} \quad (3)$$

### Fuzzification

For fuzzification of input variable Failure Rate, five linguistic variables, namely, 'very low (VL)', 'low (L)', 'medium (M)', 'high (H)', and 'very high (VH)' have been used. Similarly for input variable Execution Time, another five linguistic variables i.e. 'very low (VL)', 'low (L)', 'medium (M)', 'high (H)', and 'very high (VH)' have been considered. For the both the input linguistic variable, the membership function chosen is 'triagular'. Triangular membership functions are most frequently used MFs' due to their simplicity. Similarly output variable 'Reliability' has been defined as a triangular-shaped membership function with membership degree between 'Very Low' between 'Very High'.

$$ET(i_1) = \begin{Bmatrix} i_1; 10 \leq i_1 \leq 100 \\ 0; otherwise \end{Bmatrix} \quad (4)$$

$$FR(i_1) = \begin{Bmatrix} i_2; 0 \leq i_2 \leq 1 \\ 0; otherwise \end{Bmatrix} \quad (5)$$

$$R(Ri) = \begin{Bmatrix} R_1; 0 \leq R_1 \leq 1 \\ 0; otherwise \end{Bmatrix} \quad (6)$$

### Membership Functions

Using MATLAB FUZZY Toolbox, prototype triangular fuzzy sets for the fuzzy variables, ET, FR and pi-shaped fuzzy set for variable Reliability are set up [22]. The membership values used for the FIS were obtained from above the above the functions 1, 2 and 3 are shown in the Figures 3a–3c.

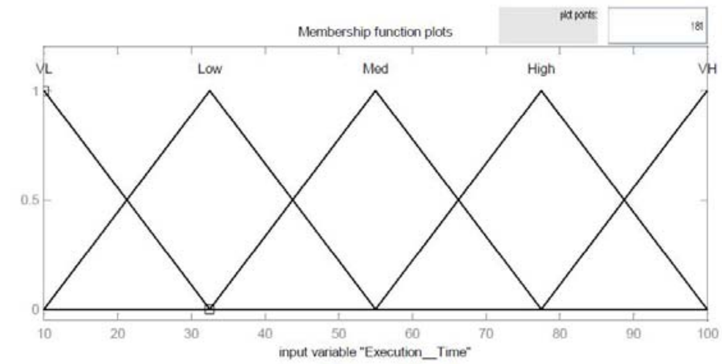


**Figure 3(a): Membership Functions Plots of ET**

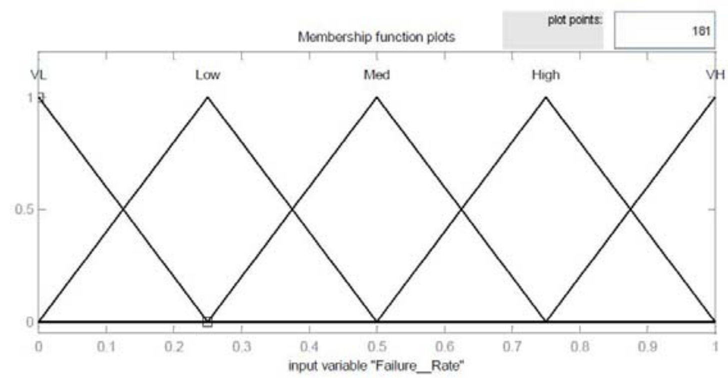


**Figure 3(b): Membership Functions Plots of FR**

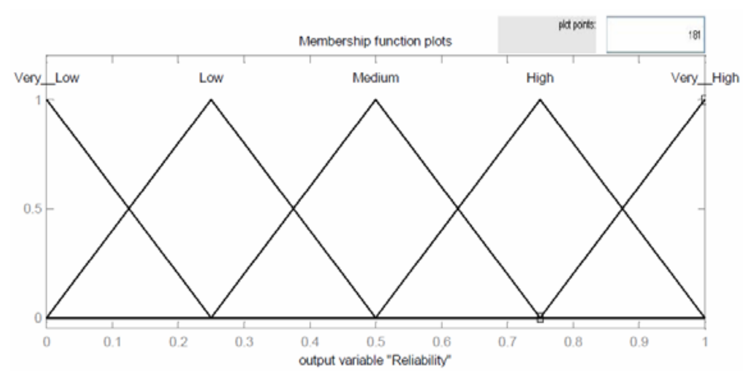


**Figure 3(c): Membership Functions Plots of Reliability**

These membership functions helped in converting numeric variables into linguistic terms. For example, the linguistic expressions and membership functions for ET obtained from the developed rules and above the formula are given as following.

$$\mu_{VL}(i_1)=\begin{cases}0;\ i_1<10\\ \dfrac{33-i_1}{23};\ 10\le i_1\le 33\\ 0;\ i_1>33\end{cases} \quad (7)$$

$$\mu_{VL}(i_1)=\left\{ 1/10+0.5/22.05+0/33 \right\} \quad (8)$$

$$\mu_{L}(i_1)=\begin{cases}\dfrac{i_1-10}{23};\ 10\le i_1\le 33\\ \dfrac{54-i_1}{21};\ 33\le i_1\le 54\\ 0;\ i_1>54\end{cases} \quad (9)$$

$$\mu_{L}(i_1)=\left\{ 0/10+0.5/23+1/33+0.5/44+0/54 \right\} \quad (10)$$

$$\mu_{M}(i_1)=\begin{cases}\dfrac{i_1-33}{21};\ 33\le i_1\le 54\\ \dfrac{77-i_1}{23};\ 54\le i_1\le 77\\ 0;\ i_1>77\end{cases} \quad (11)$$

$$\mu_{M}(i_1)=\left\{ 0/33+0.5/44+1/54+0.5/67+0/77 \right\} \quad (12)$$

$$\mu_{H}(i_1)=\begin{cases}\dfrac{i_1-54}{24};\ 54\le i_1\le 78\\ \dfrac{100-i_1}{22};\ 78\le i_1\le 100\\ 0;\ i_1>100\end{cases} \quad (13)$$

$$\mu_{H}(i_1)=\left\{ 0/54+0.5/67+1/78+0.5/89+0/100 \right\} \quad (14)$$

$$\mu_{VH}(i_1)=\begin{cases}0;\ i_1<78\\ \dfrac{100-i_1}{22};\ 78\le i_1\le 100\\ 1;\ i_1>100\end{cases} \quad (15)$$

$$\mu_{VH}(i_1)=\left\{ 0/78+0.5/89+1/100 \right\} \quad (16)$$

**Inference steps: determining conclusions**

Let P be a fuzzy relation from X to Y and Q be a fuzzy relation from Y to Z such that the membership degree is defined by P(x, y) and Q(y, z). Then, a third fuzzy relation R from X to Z can be produced by the composition of P and Q, which is denoted as PoQ. Fuzzy relation R is computed by the formula

R(x, z) = (PoQ)

(x,z) = max {min [P(x, y), Q(y, z)]}

For example, Let us consider ET=100 and FR= 0.37. For the above input values of ET and FR, Rule R14 and R15 shall be applied which notices that $\mu_{VH}(ET)=1$, $\mu_L(FR)=0.6$ and $\mu_M(FR)=0.6$. Inference mechanism for the above rules i.e. R14 and R15 is explained below.

**For Rule 14**

The premise (H), we have $\mu_M(14)=\min\{\mu_H(ET), \mu_M(FR)\} = \min\{1, 0.6\} = 0.6$, by using the minimum of the two membership values. So that we are 0.6 certain that this rule applies to the current situation. This rule indicates that if its premise is true then the action indicated by its consequent should be taken. For rule (14) the consequent is "Reliability is Medium". The membership function for the conclusion reached by rule (14), which is denoted as $\mu_{14}$, is given by $\mu_{14}$(Reliability) = min {0.6, $\mu_H$(*Reliability*)}. This membership function defines the implied fuzzy set for rule (14).

**For rule 15**

The premise (*M*), we have $\mu_M(15)=\min\{\mu_{VH}(ET), \mu_M(FR)\} = \min\{1,0.6\} = 0.6$, by using the minimum of the two membership values. The notation $\mu_{15}$ represents for rule (15) so that we are 0.6 certain that this rule applies to the current situation. This rule indicates that if its premise is true then the action indicated by its consequent should be taken. For rule (15) the main consequent is "Reliability is medium". The membership function for the conclusion reached by rule (15), which is denoted as $\mu_{15}$, is given by $\mu_{15}$(*Reliability*) = min {0.6, $\mu_M$(*Reliability*)}. This membership function defines the implied fuzzy set for rule (15).

It is noticed that we are certain that both rule (14) and (15) apply to the current situation. From Mamdani max–min inference, the membership function of system will be found as max ($\mu_{15}$, $\mu_{14}$) = 0.6 and then defuzzification operation is applied to the final component of the fuzzy controller produced by the inference mechanism and combines their effects to provide the "most certain" controller output. Then the output denoted by "*Reliability*" can be calculated that best represents the conclusions of the fuzzy controller that are represented with the implied fuzzy sets.

### Defuzzification

After the inference step, the overall result is a fuzzy value. This result should be defuzzified to obtain a final crisp output. Defuzzification is performed according to the membership function of the output variable. For instance, assume that we have the result in Figure 4 at the end of the inference. In this figure, the shaded areas all belong to the fuzzy result. The purpose is to obtain a crisp value, represented with a dot in the figure, from this fuzzy result.

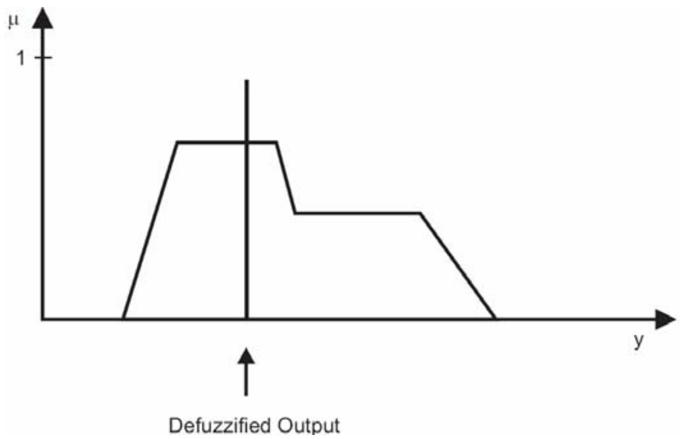


**Figure 4**: **Defuzzification step of fuzzy controller**

For defuzzification different techniques are available in the literature. The mostly-used technique is "center of gravity" (COG) defuzzification.

**Center of gravity defuzzification technique:**

The centroid defuzzification technique can be expressed as

$$\mathrm{Re}\,liability^{crisp} = \frac{\int \mu_i\ (\mathrm{x})\ \mathrm{x}\ \mathrm{dx}}{\int \mu_i\ (\mathrm{x})\ \ \mathrm{dx}} \qquad (17)$$

Where $\mathrm{Re}\,liability^{crisp}$ is the defuzzified output, $\mu_i(x)$ is the aggregated membership function and $x$ is the output variable. The only disadvantage of this method is that it is computationally difficult for complex membership functions. It can also be calculated as:

$$\mathrm{Re}\,liability^{crisp} = \frac{\sum x_i \int \mu_i\ (\mathrm{x})\ \mathrm{dx}}{\sum \int \mu_i\ (\mathrm{x})\ \ \mathrm{dx}} \qquad (18)$$

Where $x_i$ denotes the center of the membership function (i.e., where it reaches its peak) of the consequent rule i and $\int \mu_i\ (\mathrm{x})\ \mathrm{dx}$ denote the area under the membership function $\mu_i(x)$.

### Center of Gravity for Singletons

The output membership values are multiplied by their corresponding singleton values and then are divided by the sum of membership values.

$$\mathrm{Re}\,liability^{crisp} = \frac{\sum \mathrm{x_i}\ \mu_i\ (\mathrm{x})}{\sum \mu_i\ (\mathrm{x})} \qquad (19)$$

Where $x_i$ is the position of the singleton in i the universe, and $\mu_i$ is equal to the corresponding membership value. This method has a relatively good computational complexity.

For example: For the rule 15 and 14, from Fig. 4, we have $x_{15} = 0.5$ and $x_{14} = 0.75$

Using Eq. (17), We have

$$\mathrm{Re}\,liability^{crisp} = \left[\int_{0.25}^{0.5}\left(\frac{x-0.25}{0.25}\right)xdx+\int_{0.5}^{0.75}\left(\frac{0.75-x}{0.25}\right)xdx+\int_{0.5}^{0.75}\left(\frac{x-0.5}{0.25}\right)xdx+\int_{0.75}^{1}\left(\frac{1-x}{0.25}\right)xdx\right]$$
$$\div\left[\int_{0.25}^{0.5}\left(\frac{x-0.25}{0.25}\right)dx+\int_{0.5}^{0.75}\left(\frac{0.75-x}{0.25}\right)dx+\int_{0.5}^{0.75}\left(\frac{x-0.5}{0.25}\right)dx+\int_{0.75}^{1}\left(\frac{1-x}{0.25}\right)dx\right]$$
$$=0.875$$

and by using eq. (18), we have

$$\mathrm{Re}\,liability^{crisp} = \frac{0.75*0.4+1.0*0.4}{0.4+0.4} = 0.875$$

gives the same value as above.

and from the eq. (19), we have

$$\mathrm{Re}\,liability^{crisp} = \frac{0.75*0.6+1.0*0.6}{0.6+0.6} = 0.875$$

This just happens to be the same value as above.

## IV. RESULT AND DISCUSSION

With two inputs and one output the input-output mapping is a three dimensional surface. Using MATLAB, the fuzzy control surface is developed as shown in Fig. 5. This surface demonstrates relationship between execution time (ET) and failure rate (FR) on the input side, and controller output reliability on the output side. The control surface is the dynamic combination of ET and FR and display the range of all possible defuzzified values for reliability. In varying conditions of ET and FR values, the optimal value of Reliability is likely to change accordingly i.e. it depends on the inference system mechanism. The surface is the output plotted and displays the range of possible defuzzified values for all possible inputs. The two dimensional relation between Failure Rate and Reliability & Execution Time and Reliability respectively has been shown using Fig. 6 and 7.

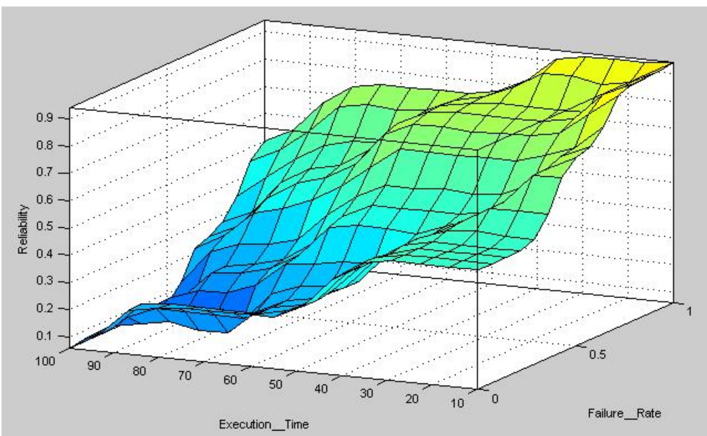


**Fig. 5 Decision Surface**

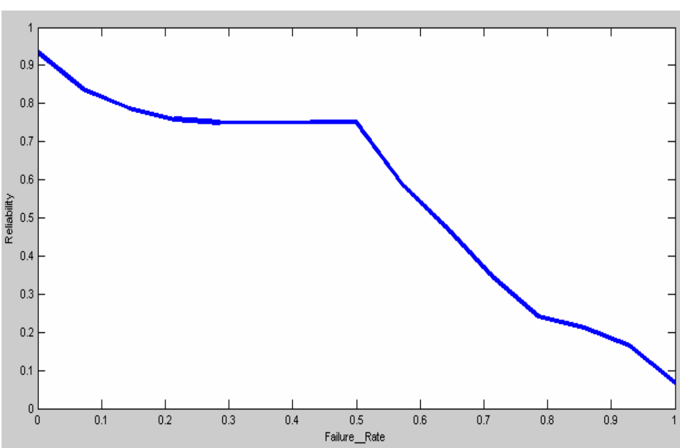


**Fig. 6 2D Relation between Failure Rate and Reliability**

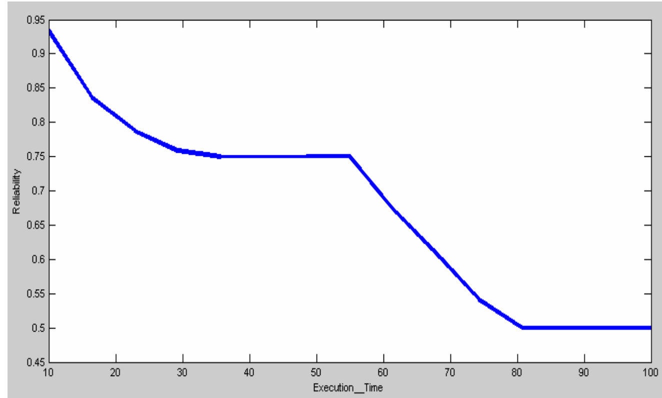


**Fig. 7 2D Relation between Execution Time and Reliability**

The plot results from the interpolation of rule base with twenty five rules. The plot is used to check the rules and the membership functions and to see if they are appropriate and whether modifications are necessary to improve the output. When a satisfactory system is achieved, the fuzzy program is converted to machine language and downloaded into a microprocessor controller.

To evaluate the performance of our proposed algorithm, we have compared the overall performance of our algorithm with [23]. It is found that the predicted reliability of proposed system is better than that of [23] as shown in Fig. 8.

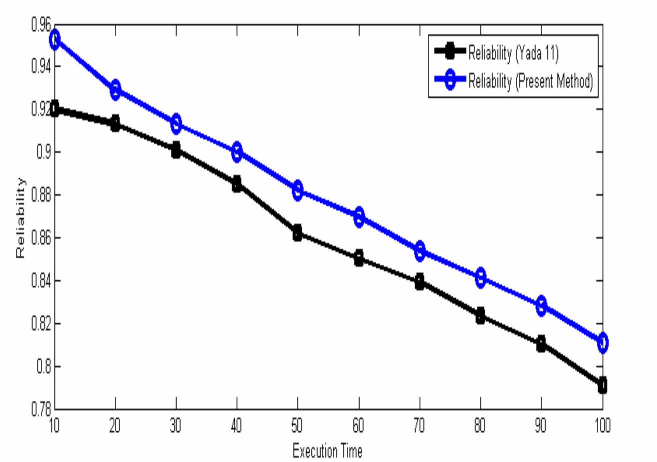


**Fig. 6.8 Comparison of Present Method with [23]**

**References:**


1. Yadav, P.K., Bhatia, K., and Sagar Gulati (2012), Reliability driven Soft Real-time Fuzzy Task Scheduling in Distributed Computing Environment, Advances in Intelligent and Soft Computing, Vol 130, pp. 207-214, Springer India, springerlink.com).
2. Gulati, Sagar, Yadav, P.K., Bhatia, K. (2012), A reliability model for the task scheduling in Distributed System based on Fuzzy Theory, CiiT International Journal of Networking and Communication Engineering, Vol. 4, No. 11, pp.684-688. (Impact Factor 0.569).

3. Guerraoui, R., A. Schiper (1996), Fault Tolerance by replication in distributed systems, Proceedings on Reliable Software Technologies, Springer.
4. M. F. Kaashoek and A. S. Tanenbaum (1991) Group Communication in the Amoeba Distributed Operating System. In IEEE 11th Intl. Conf. Distributed Computing Systems, pp. 222-230.
5. **AHMED, W. (2013), A SURVEY ON RELIABILITY IN DISTRIBUTED SYSTEMS, JOURNAL OF COMPUTER AND SYSTEM SCIENCES VOLUME 79 (8), PP. 1243–1255**
6. Z. Xuemei, P. Hoang (2000), An analysis of factors affecting software reliability, J. Syst. Softw. Vol. 50 (1) pp. 43–56.
7. Jain, R. (1991), The Art of Computer Systems Performance Analysis. Wiley & Sons, New York.
8. Walke, B., Spaniol, O. Hrsg. (1993), Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen. 7. ITG/GI-Fachtagung. Springer-Verlag.
9. Jensen, K., Rozenberg, G. (1991), High-level Petri-Nets. Springer Verlag.
10. Kelling, Ch. (1995), Simulationsverfahren fur zeiterweiterte Petri-Netze. Dissertation TU Berlin. Society for Computer Simulation Intemational, Ghent.
11. Klas, G.; Lepold, R. (1992) TOMSPIN, a Tool for Modeling with Stochastic Petri Nets. Proc. CompEuro 92, The Hague.
12. (1995), Proceedings of the 6th Intemational Workshop on Petri Nets and Performance Models. Durham 1995. IEEE Computer Society Press.
13. L. A. Zadeh (1980), "Fuzzy sets versus probability," Proc. IEEE, vol. 68, pp. 421-421.
14. L. A. Zadeh (1994), "Fuzzy logic, neural networks, and soft computing," Commun. ACM, vol. 37, pp. 77-84.
15. W. Pedrycz and F. Gomide (1998), An introduction to fuzzy sets: analysis and design: The MIT Press.
16. Rajagopalan, A., Washington, G., Rizzani, G. & Guezennec, Y. (2003) Development of fuzzy logic and neural network control and advanced emissions modeling for parallel hybrid vehicles, Center for Automotive Research, Intelligent Structures and Systems Laboratory, Ohio State University, U.S.A
17. Zadeh, L.A., (1965). Fuzzy Sets. Information and Control. Vol 8(3). 338-353
18. E. H. Mamdani (1974), "Application of fuzzy algorithms for the control of a dynamic plant," Proc. IEE, vol. 121, pp. 1585-1588.
19. T. Takagi and M.Sugeno (1985), "Fuzzy identification of systems and its applications to modeling and control," IEEE Trans. Syst., Man, Cybern., vol. 15, pp. 116-132.
20. H. Surmann and A. P. Ungering (1995), "Fuzzy rule-based systems on general-purpose processors," IEEE Micro, vol. 15, pp. 40-48.
21. G. Ascia and V. Catania (2001), "A general purpose processor oriented to fuzzy reasoning," in Proc. 10th IEEE International Conf. Fuzzy Systems, pp. 352-355.
22. K. Youngdal and L.-K. Hyung (1996), "An architecture of fuzzy logic controller with parallel defuzzification," in Proc. Biennial Conf. of the North American Fuzzy Information Processing Society, pp. 497-501.
23. Yadav, P., K., Singh, M., P., and Kuldeep Sharma, “Task Allocation Model for Reliability and Cost Optimization in Distributed Computing Systems”, International journal of Modeling Simulation and Scientific Computing, Vo. 2, No. 2, pp. 131-149, 2011.