[Blej*et al., 5(7): July, 2016]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

# TASK PARAMETERS MANAGING AND SYSTEM ACCURACY IN FUZZY REALTIME SCHEDULING

**Mohammed Blej and MostafaAzizi**
MATSI Lab., Mohammed first university, Oujda, Morocco

**DOI**: 10.5281/zenodo.56896

## ABSTRACT

The classical analysis of realtime systems tries to ensure that the instance of every task finishes before its absolute deadline (strict guarantee). The probabilistic approach tends to estimate the probability that it will happen. The deterministic timed behavior is an important parameter for analyzing the robustness of the system. Most of related works are mainly based on the determinism of time constraints. However, in most cases, these parameters are non precise. The vagueness of parameters suggests the use of fuzzy logic to decide in what order the requests should be executed to reduce the chance of a request being missed.The choice of task parameters and numbers of rules in fuzzy inference engine influences directly generated outputs. Our main contribution is proposing a fuzzy approach to perform realtime scheduling in which the scheduling parameters are treated as fuzzy variables. A comparison of the results of the use of each parameter as linguistic variable is also given.

**KEYWORDS**: fyzzy logic; real-time task; fuzzy inference engine; fuzzy real-time scheduling; task parameters.

## INTRODUCTION

Realtime systems are present and vital in very diverse fields such as avionics, process control, air traffic control systems, and mission critical computations [1]. In the literature, these systems have been defined as: “systems in which the correctness depends not only on the logical results of computation, but also on the time at which the results are delivered” [2]. Realtime activities are traditionally characterized by temporal constraints, called deadlines [3]. Each task is a stream of jobs; each one has a time of release, it is characterized by a computation time, and must finish before its absolute deadline. Realtime task is said to be hard, soft, or firm depending on the criticity of the consequence of a deadline miss.

For a hard task, no deadline miss accepted, since a single job of task finishes after its deadline could jeopardize the entire system [4]. A soft task tolerates jobs that finish after their deadlines, whereas a firm task can only tolerate some job failures. More precisely, a firm job must be finished before its deadline or does not execute at all. In other words, a soft job that misses its deadline can still do some useful work, while a firm job that misses its deadline is useless, though it does not jeopardize the system [5]. Because of their critical nature, hard realtime tasks must be guaranteed off-line; a hard task is accepted only if it is guaranteed that every job is executed before its deadline. For firm tasks, admission tests are usually online, a job is activated only if it is certain that it will end its execution before its deadline, otherwise it will be rejected.

A schedule is said to be feasible if all the tasks matched both their deadlines and any additional specified constraints [6]. Realtime scheduling can be classified in two categories: static [3] and dynamic [7]. A static realtime scheduling algorithm such as Rate Monotonic schedules all realtime tasks offline; this requires complete knowledge about tasks and system settings [8]. For dynamic scheduling, the feasibility of the algorithm is calculated online and tasks are invoked dynamically. These algorithms use dynamic parameters like deadline, latency and laxity [9,10,11].
However, the use of fuzzy logic to facilitate searching for a feasible schedule is motivated by several reasons.

First, in a dynamic hard realtime system, not all the characteristics of tasks (e.g., precedence constraints, resource allocations …) are known a priori. For example, for aperiodic tasks , the arrival time for the next task is unknown . In particular, there is a wide uncertainty in hard realtime environment which will worsen scheduling problems (e.g. arbitrary arrival time, system load, and uncertain computation time). Futhermore, in overload case, we must slow down the execution of tasks while ensuring that the most important tasks are run first, thus allowing a ratio of flexibility in the scheduler under adverse conditions to determine which tasks are run and which are not [12].
The remaining of this paper is organized as follows. Section II describes scheduling algorithms, classic and fuzzy, and model tasks. Section III discusses the fuzzy inference engine. In section IV, we present the concept of Fuzzy Inference Engine with the difference between Mamdani and Sugeno types. Section V provides some concluding remarks.

## SCHEDULING MODEL

### Classical case

The scheduling in real time systems based on the use of the CPU time and other resources to execute all the tasks in question in order to meet the time constraints [2]. However, scheduling in real-time systems is more important than in classic systems [13,14]. The real-time tasks must be performed correctly while respecting the deadline [15].
There are brief descriptions of the main scheduling algorithms:
FCFS algorithm (First-Come-First-Served) selects the task with the earliest arrival time [16].The release time of periodic tasks in the system will be considered, but this algorithm makes no efforts to consider task deadline, so it is not feasible for hard real time task.
Rate Monotonic (RM) algorithm assigns priority according to period; A task with a shorter period has a higher priority [3].This algorithm is an optimal static-priority scheduling [17].
Earliest Deadline First (EDF) algorithm always chooses the task with the earliest deadline [18], a job with the earliest deadline is executed. Since it cannot consider priority and therefore cannot analyze it. This algorithm is optimal dynamic priority scheduling.
Least Laxity First (LLF scheduling algorithm assigns higher priority to a task with the least laxity, and has been proved to be optimal for uniprocessor systems [2].

### Fuzzylogic

After at least two decades since its elaboration, fuzzy logic was finally accepted as the basis for an emerging technology, ranging from consumer products, to industrial process control, to automotive applications [19]. It is so closer to human brain thinking compared to conventional logical systems. Fuzzy logic is a multi-valued logic. It deals with approximation rather than exactness. In contrast to classical sets (A classical set takes true or false values), fuzzy logic variables (also known as linguistic variables) can have a truth value that ranges into an floating interval between 0 (completely true) and 1 (completely false) [20]. The linguistic variable degree may be determined with a specific method [21].
Nowadays, fuzzy logic is largely used in real world issues, and it is also a research interest of a great number of researchers. It has been shown to be a strong methodology of design and analysis in control theory, enabling the implementation of advanced knowledge-based control methods for complex dynamic systems like those rising applications for systems and artificial biology [20]. Hiwarkar et al gave a large list of use cases of fuzzy logic in [21]. K.A. Verma et al discussed Type 1 fuzzy systems and the origin of type 2 fuzzy logic systems [22] and its application in the field of engineering, finance and medicine. Xia Feng et al designed a schedule for control of embedded systems based on fuzzy logic [23]. A priority scheduler has been developed for mobile by C. Gomathy et al in [24].

### Fuzzy algorithms

The main conventional algorithms are based on binary logic; the new versions of these algorithms are introduced based on fuzzy logic. This logic is used in different conventional algorithms in the case of the highest-priority task, giving birth to fuzzy versions of these algorithms.
In the literature, most of the work is about soft realtime systems. In what follows, we quote the most famous results:
M. Sabeghi et al introduced a fuzzy algorithm for scheduling periodic tasks on multiprocessor soft realtime systems [25]. P. Vijayakumar et al presented a fuzzy EDF algorithm for soft realtime systems [26], they used laxity and deadline as fuzzy parameters. Fuzzy inference rules include 15 fuzzy inference rules. The miss deadline ratio of the tasks will be reduced compared to the traditional EDF algorithm.V. Salmani et al proposed a new fuzzy-based algorithm for scheduling realtime tasks on uniform parallel machines in [27]. It is shown that the proposed approach

ensured a performance close to that of EDF in non-overloaded conditions and it has supremacy over EDF in overloaded situations in many aspects.
In [28], Hamzeh et al proposed a new fuzzy scheduler. They use laxity, CPU time and deadline as a fuzzy parameter. This scheduler has low complexity due to the simplicity of fuzzy inference engine.
Sheo Das et al have described the use of fuzzy logic to multiprocessor realtime scheduling [29]. They have showed that using deadline as a fuzzy parameter in multiprocessor realtime scheduling is more promising than laxity, and that this model is efficient when the system has heterogeneous tasks with different constraints.
For embedded systems, T. Springer et al [30] presented a new scheduling approach for realtime tasks in an embedded system. The results are a demonstrated reduction in deadline misses for all tasks during periods of overload as compared to traditional fixed priority based scheduling mechanisms.
In the case of hard realtime systems, publications on fuzzy logic are scarce. Very early John Yen et al introduced a Designing of a Fuzzy Scheduler for Hard Realtime Systems [31]. They worked on the scheduling problem as a search problem, using a set of fuzzy rules to guide the search for a feasible schedule, and the scheduler is triggered by a newly arrival task.
In conclusion, most papers dealt with soft realtime systems, but not enough for hard realtime systems. Fuzzy versions of different conventional algorithms are given and showed that they are more promising than conventional ones. The choice of fuzzy parameters and the comparison of different results are studied in special cases, and research over the generalization is an open work.

## FUZZY INFERENCE ENGINE

Fuzzy Inference Systems are conceptually very simple. They consist of an input, a processing, and an output stage [32]. The input stage receives inputs like deadline, execution time, laxity and so on, and maps these to appropriate membership functions and truth values. In the processing stage, each specific rule is invoked and the corresponding result is generated. Then results are combined so that it will be given as an input to the output stage. In output stage, the combined result is converted back into a specific value [33]. The membership function of a fuzzy set is a generalization of the indicator function in classical sets. In fuzzy logic, it represents the degree of truth as an extension of valuation. It can be expressed in the form of a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. It can also have many forms (triangular, trapezoidal and Bell curves) [33].
The processing stage also called inference engine is based on a set of logical rules in the form of IF-THEN statements.
An example of fuzzy IF-THEN rules is: **IF Speed is "Low" AND Race is "Dry" THEN Braking is "Soft",** Where the IF part is called the "antecedent" and the THEN part is called the "consequent". The terms Speed, Race and Braking are linguistic variables, and Low, Dry and Soft are linguistic terms. Each linguistic term corresponds to a value of the membership function. Typically, Fuzzy Inference Systems have dozens of rules [32]. The inference engine processes the inputs and generates outputs based on the rules already defined. There are five steps in the fuzzy inference:

- Fuzzify inputs,
- Apply the fuzzy operator,
- Apply the implication method,
- Aggregate all outputs,
- Defuzzify outputs.

Below is a brief overview of these five steps. The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. Fuzzification of the input amounts to either a table lookup or a function evaluation. After fuzzifying the inputs, we know then the degree to which each part of the antecedent has been fullfilled for each rule. So if the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number will then be applied to the output function. The number of inputs to the fuzzy operator is two or more membership values from fuzzified input variables. Whereas the output is a single truth value.
The output fuzzy set has been modified by the implication function to the degree specified by the antecedent. Since decisions are based on the testing of all of the rules in the Fuzzy Inference Subsystem (FIS), the results from each rule must be combined in order to generate the final decision. Aggregation is the process by which the fuzzy sets

ISSN: 2277-9655
Impact Factor: 4.116

that represent the outputs of each rule are combined into a single fuzzy set. It occurs only once for each output variable, just prior to the fifth and final step, defuzzification. The input for the defuzzification process is an aggregate output fuzzy set, and the output is a single number.

At run-time, based on the parameter of tasks, the fuzzy scheduler selects the highest priority task that is ready for execution. Several parameters determine the priority of tasks: task deadline, task criticality, task execution time, laxity. The task deadline is the time before the task should be completed. The task criticality relates to the consequences of missing a deadline. The worst case execution time of task is his execution time. Laxity is the time that separates the task deadline and the worst case execution time of task. These parameters constitute the linguistics variables and then fuzzified. Fuzzy rules are then applied to the linguistic variables to compute the service value. The linguistic values for the chosen parameters are defined. The fuzzification is applied to the conclusion; the way in which this happens depends on the inference model.

There are two common inference methods [33]: Mamdani's fuzzy inference method proposed in 1975 by EbrahimMamdani [34] and Takagi-Sugeno-Kang, method of fuzzy inference introduced in 1985 [35]. Mamdani's efforts are based on the work of Lotfi A. Zadeh on fuzzy algorithms for complex systems and decision processes [36].

The main difference between Mamdani-type FIS and Sugeno-type FIS resides in the way the crisp output is generated from the fuzzy inputs. While Mamdani-type FIS uses the technique of defuzzification of a fuzzy output, Sugeno-type FIS uses weighted average to compute the crisp output [37], so the Sugeno's output membership functions are either linear or constant but Mamdani's inference expects the output membership functions to be fuzzy sets. Furthermore, Sugeno method has better processing time since the weighted average replace the time consuming defuzzification process [38].

## THE PROPOSED MODEL

In the proposed model, the input stage consists of two linguistic variables. The first one is an priority, each task arrives in the system with its priority.This priority is assigned to the task from the outsideworld and it's static.The other input variable can be one of other task parameters.This input can be tasks interval, laxity, wait time, or so on, for different scheduling algorithms.Each parameter may cause the system to react in a different way, sincechanging the input variables the corresponding membership functions may be changed accordingly. We will compare the impact of the change of the second parameter, while keeping the same task priority.

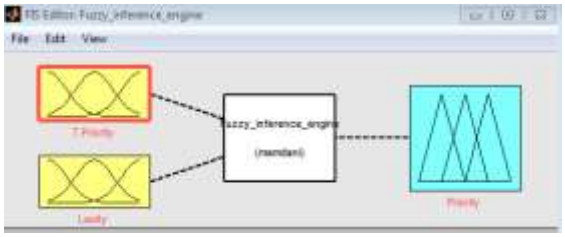


*Fig.1:The proposed Fuzzy Inference Engine*

**Laxity&Priority**

Laxity of a task at any time is defined as remaining time to deadlineminus the amount of remaining execution. We choose this parameter as the secondlinguistic variable.A task's priority shows the importance of the task, the notion of laxity is used in the proposed approach to facilitate the computation. The inputs of these parameters are justified. In our example, we choose task priority and laxity as inputs parameters.We considered 4 triangular membership functions for task's priority. "High", "Normal", "Low" and "Very low" are these membership functions.Laxity membership function considered 3 and also triangular. "Critical", "Medium" and "sufficient" are the name of these functions. Fuzzy rules try to combine these parameters as they are connected in real worlds. Some of these rules are mentioned here:

- If (T.Priority is "High") and (Laxity is "Critical"), then (Priority is "Very high").
- If (T.Priority is "High") and (Laxity is "Medium"), then (Priority is "high").
- If (T.Priority is "High") and (Laxity is "sufficient"), then (Priority is "Medium").

ISSN: 2277-9655
Impact Factor: 4.116

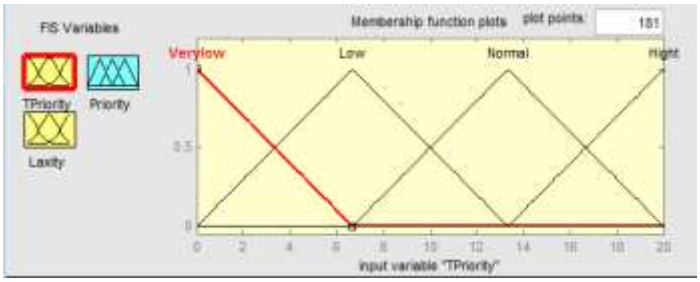


*Fig.2: Input Task Priority Membership Function*

**Deadline&Priority**

To see the impact of changing the parameters of the tasks in the input stage on the final output priority, we choose the task deadline as second parameters in the input system. It is very important to remember that assigning priority to tasks according to their deadlines is a simple yet successful strategy for uniprocessor real-time scheduling.EDF (Earliest Deadline First) and DM (DeadlineMonotonic) are shown as optimal dynamic- and static-priority policies for preemptive uniprocessor scheduling, although they are not in the case multiprocessor.We considered 3 triangular membership functions for task's deadline. Low, medium and sufficient are these membership functions.

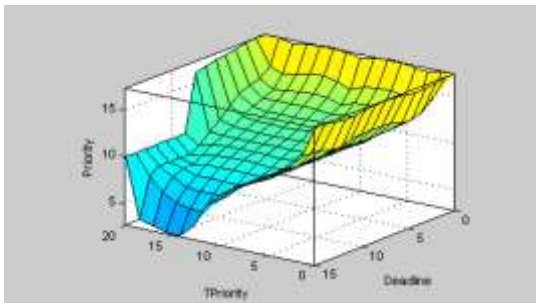


*Fig.3:The decision surface Corresponding to Inference rules*

**Results analysis**

To compare the external priority which is the priority assigned to the task from the outside world and a new priority based on fuzzy logic, we use this external priority as an input linguistic variable and we use separately two parameters, namely laxity and deadline, as secondlinguistic variable. Fuzzy rules compute the crisp value using centroid Defuzzification method of Mamdani inference, and an output priority of task is calculed. The output of the system is priority that determines which is used as a parameter for making a decision.Third step, we use both parameters at the same time in an inputstage then we look and compare different results.The following figure shows the different results. The priority calculed in the third case is the least best. This can be explained by the fact that in this case the number of rules is very large, since we have three input linguistic variables. When we have more rules and member functions, it will directly affect the overall system accuracy, even though performance of the system can be made better when number of rules are reduce. For the other two cases, we have the same number of rules for the two linguistics variables, namely deadline and laxity. As it is depicted in the figure 4, the simulation results show that the output priority based on deadline is much better than the output priority based on laxity, knowing that the initial priority is the same for the two cases.

ISSN: 2277-9655
Impact Factor: 4.116

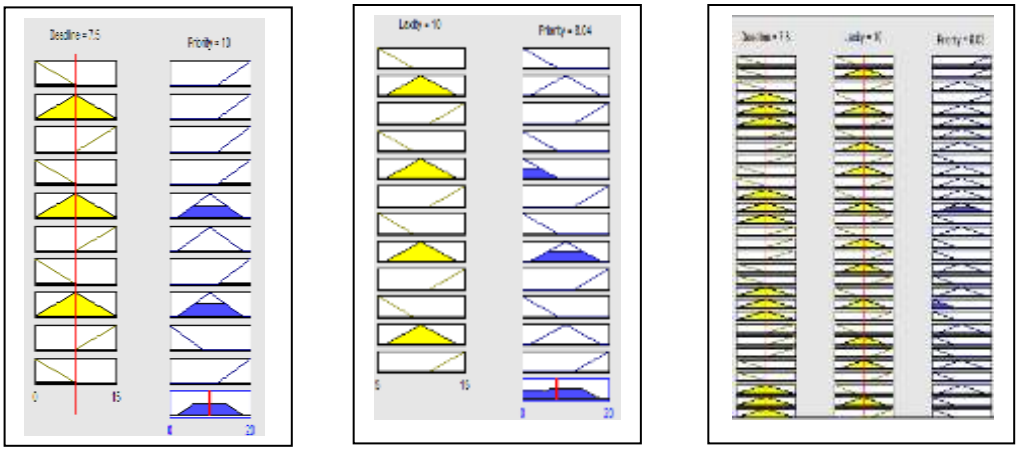

*Fig. 4: Computation of New Priority for the three cases*

## CONCLUSION

For scheduling, fuzzy logic is used in the different conventional algorithms with the choice of the highest-priority task, giving thus birth to fuzzy versions of these algorithms. This paper presents a comparison of fuzzy priorities based on deadline and priorities based on laxity. As results, we conclude that using deadline as a fuzzy parameter is more promising than laxity. The choice of parameters and numbers of rules in fuzzy inference engine influences directly generated outputs. Being based on Mamdanior Sugeno, Fuzzy Inference Systems (FIS) are still on-going research areas.

As a future work, we plan to redo this work but based this time on Sugeno-type FIS and to compare the results of the two models.

## REFERENCES

[1] Sha, L. a. Real-Time Scheduling Theory and Ada, IEEE Computer, Vol. 23, No. 4, pp. 53-62.1990.
[2] Ramamnitham K., s. J. Scheduling algorithms and operating systems. Proceedings of IEEE, vol 82, No1, pp 55-67.1994.
[3] Layland, C. L. Scheduling alghorithms for multiprogramming systems.Journal of the ACM, 20(1), 1973.
[4] J. Zhu, T. G. Scheduling in hard real-time applications. IEEE Softw, vol 12, pp. 54-63, 1994.
[5] Haibin, W. L. Research on a soft real-time scheduling algorithm based on hybrid adaptive control architecture. Proc. American Control Conf, Lisbon, Portugal, pp. 4022-4027 Vol.5, 2003.
[6] T. F. Abdelzaher and K. G. Shin."Comment on a pre-runtime scheduling algorithm for hard realtime systems". IEEE Trans Software Engineering, Vol. 23, pp. 599-600, 1997.
[7] K. Ramamritham and J. A. Stankovic. "Dynamic task scheduling in hard real-time distributed systems," EEE Softw, vol. 1, pp. 65-75, 1984.
[8] P. A. Laplante. "The certainty of uncertainty in real-time systems," IEEE Instrum. Meas. Mag, vol. 7, pp. 44-50, 2004.
[9] Kreuzinger, A.S "Real-time scheduling on multithreaded". Proc. 7th Intl. Conf. Real-Time Computing Systems, Cheju Island, South Korea, pp. 155-159, 2000.
[10] Z. Deng, J. W, "Dynamic scheduling of hard realtime applications in open system environment".Tech. Rep. University of Illinois at Urbana-Champaign, 1996.
[11] G. Buttazzo and J. A. Stankovic. "RED: robust earliest deadline scheduling". Proc. 3rd Intl. Workshop Responsive Computing, Lincoln, NH, pp. 100-111, 1993.
[12] John Yen, J. L. Designing a Fuzzy Scheduler for Hard Real-Time Systems. Department of Computer Science Texas A&M University, College Station, TX 77843, 1993.
[13] F. Gruian. "Energy-centric scheduling for real-time systems". Department of Computer Science. Ph.D dissertation: Lund University, p. 164, 2002.

[14]W. Lifeng and Y. Haibin. "Research on a soft real-time scheduling algorithm based on hybrid adaptive control architecture", in Proc American Control Conf, Lisbon, Portugal, pp. 4022-4027 vol.5, 2003.

[15] M. Silly-Chetto, "Dynamic acceptance of aperiodic tasks with periodic tasks under resource sharing constraints". IEEE Proc. Software, vol. 146, pp. 120-127. (Apr 1999).

[16] A. S. Tanenbaum. Distributed operating systems: Prentice Hall, 1994.

[17] Yoshifumi Manabe, S. A. A Feasibility Decision Algorithm for Rate Monotonic Scheduling of Periodic Real-Time Tasks . NTT Basic Research Laboratories. Atsugi-shi, Kanagawa 243-01 Japan, 1995.

[18] N. D. Thai, "Real-time scheduling in distributed systems". Proc.Intl. Conf. Parallel Computing in Electrical Engineering, Warsaw, Poland,, pp. 165- 170, 2002.

[19] J. Yen and R. Langari. ( 2004). FuzzyLogic. Pearson Education.

[20] RajaniKumari, V. K. (September 2013). Design and Implementation of Modified Fuzzy based CPU Scheduling Algorithm. International Journal of Computer Applications (0975 – 8887), Volume 77 – No.17.

[21] Hiwarkar, T. A. (2013). "New Applications of Soft Computing, Artificial Intelligence, Fuzzy Logic & Genetic Algorithm in Bioinformatics".

[22] Varma, K. A. (2013). Applications of type-2 fuzzy logic in power systems: A literature survey. Environment and Electrical Engineering (EEEIC),12th International Conference on. IEEE, 2013.

[23] Xia, F. e. (2005). "Fuzzy logic based feedback scheduler for embedded control systems". Advances in Intelligent Computing. Springer Berlin Heidelberg, 453-462.

[24] Gomathy, C. a. (2004). "An efficient fuzzy based priority scheduler for mobile and hoc networks and performance analysis for various mobility models". Wireless Communications and Networking Conference, WCNC. 2004 IEEE. Vol. 2. IEEE, 2004.

[25] MojtabaSabeghi, a. M. ( 2006). A Fuzzy Algorithm for Real-Time Scheduling of Soft Periodic Tasks. IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.2A.

[26] P.Vijayakumar, P. (2010). Fuzzy EDF Algorithm for Soft Real Time Systems. International Journal of Computer Communication and Information System Vol.2. No1. ISSN: 0976–1349.

[27] V.Salmani, R. N. (2007). A Fuzzy-based Multi-criteria Scheduler for Uniform Multiprocessor Real-time Systems. 10th International Conference on Information Technology, 0-7695-3068-0/ 2007 IEEE.

[28] M. Hamzeh, S. M. (2007). Soft Real-Time Fuzzy Task Scheduling for Multiprocessor Systems. International Journal of Intelligent Technology, Vo 2 No 4 2007 ISSN 1305-6417.

[29] Sheo Das, P. G. (2012). A Fuzzy Approach Scheduling on More Than One Processor System in Real Time Environment. International Journal of Scientific Research Engineering & Technology, Volume 1 Issue 5 pp 289-293.

[30] Tom Springer, S. P. (2015). Fuzzy Logic Based Adaptive Hierarchical Scheduling for Periodic Real-Time Tasks. Springer, EWiLi'15, October 8th, 2015, Amsterdam, The Netherlands.

[31] John Yen, J. L. (1993). Designing a Fuzzy Scheduler for Hard Real-Time Systems. Department of Computer Science Texas A&M University, College Station, TX 77843.

[32] H.Deldari, M. a. ( 2006). A Fuzzy Algorithm for Scheduling Periodic Tasks on Multiprocessor. IJCSN International Journal of Computer Science and Network Security, VOL.6 No.3A.

[33] Wang Lie-Xin. (1996). A course in fuzzy systems and control, Prentice Hall, Paperback.

[34] Mamdani E.H., A. S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller, International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13.

[35] Sugeno, M. (1985). Industrial applications of fuzzy control, Elsevier Science Inc, New York, NY.

[36] Zadeh, L. (Jan.1973). Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 3, Vol. 3, No. 1, pp. 28-44,

[37] A.Hamam and N. D. Georganas. (2008.). A comparison of Mamdani and Sugeno fuzzy inference systems… IEEE International Workshop on Haptic Audio Visual Environments and their Applications,Ottawa, Canada , pp. 18-19

[38] Mohammed Blej and MostafaAzizi, Survey On Fuzzy Logic in Real-time System, International Journal of Advanced Computer Technology (IJACT), ISSN: 2319-7900, Feb, 2016.