*International Journal of Applied Engineering and Technology ISSN: 2277-212X (Online)*
*An Open Access, Online International Journal Available at http://www.cibtech.org/jet.htm*
*2014 Vol. 4 (2) April-June, pp.42-47/Sharma et al.*
***Research Article***

# PERFORMANCE ANALYSIS OF C-LOOK DISK SCHEDULING ALGORITHM FOR REAL-TIME SYSTEM CONSIDERING ROTATIONAL DELAY IN FUZZY

**Sachin Sharma[1], *Silky Miglani[1] and T.P Singh[3]**
*[1]CSE, Yamuna Institute of Engg. & Technology, Gadholi, Yamunanagar, Haryana*
*[3]Department of Mathematics & Computing, Yamuna Institute of Engg. & Tech., Gadholi, Yamunanagar, Haryana*
**Author for Correspondence*

## ABSTRACT
Most of the disk scheduling algorithms available in literature is either studied in deterministic or stochastic environment but practically, it has been observed that the rotational delay of a disk varies to some extent due to certain time latencies. This paper is designed with an objective to measure the performance of C-LOOK algorithm in uncertain environment for calculating disk access time with more accuracy from practical point of view and to handle the starvation problem associated with disk scheduling. Our findings are more accurate and closer to Real-Time world.

***Keywords:*** *C-LOOK, disk Scheduling, Rotational Delay, seek time, Fuzzy Environment*

## INTRODUCTION
When the objects are lined-up according to certain criteria, the term is coined as sequencing but when a time constraint is added with this sequencing then it is named as scheduling. The scheduling in the sense of computer system can be used interchangeably with CPU or I/O devices of the system. The computer system consists of hardware and software and the secondary storage devices are one of the most important components of hardware. The hard disk is the major storage component of system and there arises a need to schedule this disk so as to fulfil the requests made by CPU for accessing the memory either by read or write operations.

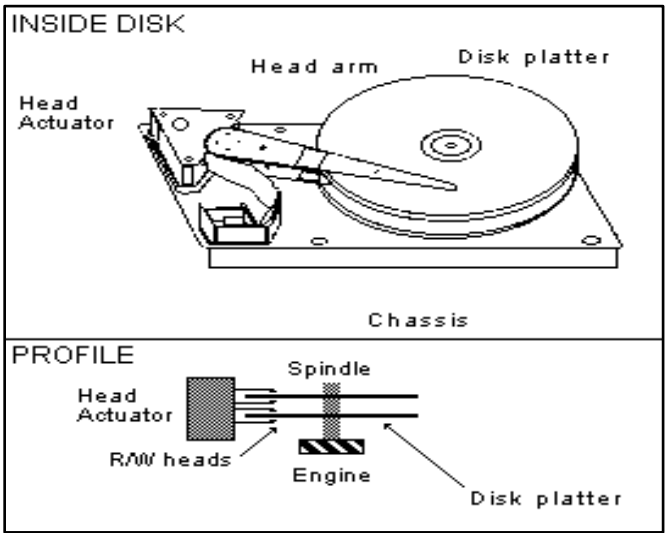



**Figure 1: Inside the Disk**

The hard disk consists of tracks and sectors that are combined to form the platters which are used for storing the data as shown in Figure 1. The Read/Write head is an important part that is used to locate the data to be read or written from and on to the disk as per the request made by CPU. So, the disk is to be

scheduled for determining the data on a particular track or sector. The time taken by the read/write head to locate the particular track is the **access time** while that taken to locate the particular sector is known as the **seek time**. The amount of data that can be transferred by the disk per second is the **data transfer rate** and the time taken by the disk until the right sector comes below the head is called the **Rotational Delay**.

There are many disk scheduling algorithms already available namely FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK etc in order to minimize the time taken by the head to locate the sector. We have considered C-LOOK disk scheduling algorithm to schedule the tracks or sectors. C-LOOK is expanded as Circular-LOOK disk scheduling algorithm that takes the **head** in either outer or inner area of disk depending upon track requested which is closer to its initial position. The head after serving the last request jumps to the other end of the track requested and continue in the same manner. Hence, the total seek time, rotational delay and transfer time are then calculated to determine the access time of a disk.

***Literature Survey***

A lot of work has been done by different researchers in the field of disk scheduling taking different parameters. But most of the work is confined to deterministic or stochastic environment. Javed and Khan [2000] in their study simulated four disk scheduling algorithms. The work was further extended by Z. Dimitrijevic *et al.*, [2005], Tsai *et al*., [2008], Muqaddas [2009] by taking different parameters. The work of Muqaddas *et al.*, [2009] deserves credit in the sense that they made S-LOOK in a priority disk scheduling in offline and online environments. Suri and Mittal [2011] designed a simulator for optimising the performance of disk scheduling algorithm. Recently, Supriya [2014] explored a new approach to disk scheduling using fuzzy logic.

We have extended the work of earlier authors by considering some of the parameters such as Rotational Delay in uncertain environment and to handle the starvation problem associated with disk scheduling. An attempt has been made to measure the performance of C-LOOK scheduling algorithm for calculating disk access time by taking fuzzy values of rotational delay so that the final result comes out to be more précised and accurate. The Seek Time has been minimised through C-LOOK algorithm in efficient manner and hence value of disk access time is minimised which increases the overall speed of the system and gets the quick response.

***Disk Scheduling***

Disk Scheduling is mainly the abstraction provided by the operating system to the user when user need not worry about the number of head, cylinder and sectors. When I/O request is made in real-time situation, an operating system issues a system call. If a disk is available, the request is serviced immediately. If it is busy, new request for service will be placed in the queue of pending requests. After completing one request, the pending requests are then serviced next and the process goes on till all the pending requests are serviced. The requests for the disk is mainly carried out to locate the desired track or sector which is implemented through various disk scheduling algorithms and C-LOOK is one of that.

C-LOOK is considered to be one of the efficient algorithms that minimises the response time and maximises the throughput of the system. According to this algorithm, the head moves inwards servicing the requests until there are no more requests in that direction. Then it jumps to the outermost outstanding requests and serves the other requests in the outermost area in the same manner. This step is then repeated over and over.

***Need for Fuzzy Disk Scheduling***

The system has been considered in fuzzy due to the imprecise value of rotational delay of a disk. In real-time systems, the disk suffers with certain delays and latencies such as controller overhead, spindle delay, delays due to scratches on the hard disk, shut down and wakeup delay that inadvertently affects the rotational delay of the disk. As rotational time of the disk is the time taken by the platter to complete its one revolution but the rotational latency occurs when the head moving to the track and then waiting for the right part of platter to pass under it before reading the data which further affects the disk access time.

The disk of our system possesses the data transfer rate of 5400 rpm (revolutions per minute) so the time taken by the platter to complete its one revolution in seconds is:

$$\frac{60000}{5400} = 11.11msec$$

On average, the platter has to complete 180 degree of rotation before it can perform each I/O. Hence, to complete 180 degree of rotation, the time required will be:

$$\frac{11.11}{2} = 5.5msec$$

Thus the maximum rotational latency that may occur according to the maximum data transfer rate of a disk is 11.11msec while the minimum of it is 5.5 msec. But as already discussed above this latency is affected due to certain other delays so this rotational latency moves towards uncertainty and hence this latency turns to be fuzzy.

Considering the triangular fuzzy numbers <a, b, c> in different environments. A fuzzy number possesses uncertain values but uses the linguistic variables such as low, medium, high considering the different range of values.

The membership function of triangular fuzzy number is given by:

$$f(x;a,b,c) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases}$$

For the defuzzification of triangular fuzzy number, Yagar's gave Average High Ranking formula as:
P(A)= (3b+(c-a))/3

The rotational delay (R) can be fuzzified into three values as follows:
**< a , b, c> = < 5.5, 8.3, 11.11>**
On applying Yager's defuzzifying formula, we get the Rotational delay=9.85msec

## MATERIALS AND METHODS

### *Solution Methodology*

The disk serves the I/O request for different tracks. These track requests are generated randomly on simulating the C-LOOK disk scheduling algorithm in C. The Initial Head Position is taken to be equal to 50.The Total Seek Time for the 20 track requests for each file of 20MB is calculated through the following formula:

Total seek time = $\sum_{i=1}^{n} Seek\ Time\ (i)$ **.....** (i)

The Data Transfer Rate of our disk is calculated through:

$$\text{Data Transfer Rate} = \frac{\left(\frac{RPM}{60}\right)*\left(\frac{Sectors}{Track}\right)*\left(\frac{Bytes}{Sector}\right)*8}{10^6} \quad \text{..... (ii)}$$

The Data Transfer Time of our disk is calculated through:

$$\text{Transfer Time} = \frac{Total\ \ amount\ \ of\ Data\ \ to\ be\ transferred}{Data\ \ Transfer\ \ \ Rate} \quad \text{..... (iii)}$$

The Rotational Delay of a disk is calculated through:
Rotational Delay = Yager' Formula for defuzzification = 9.85
Finally, the Disk Access Time is calculated through:
Disk Access Time= Seek Time + Rotational Delay + Transfer Time..... (iv)

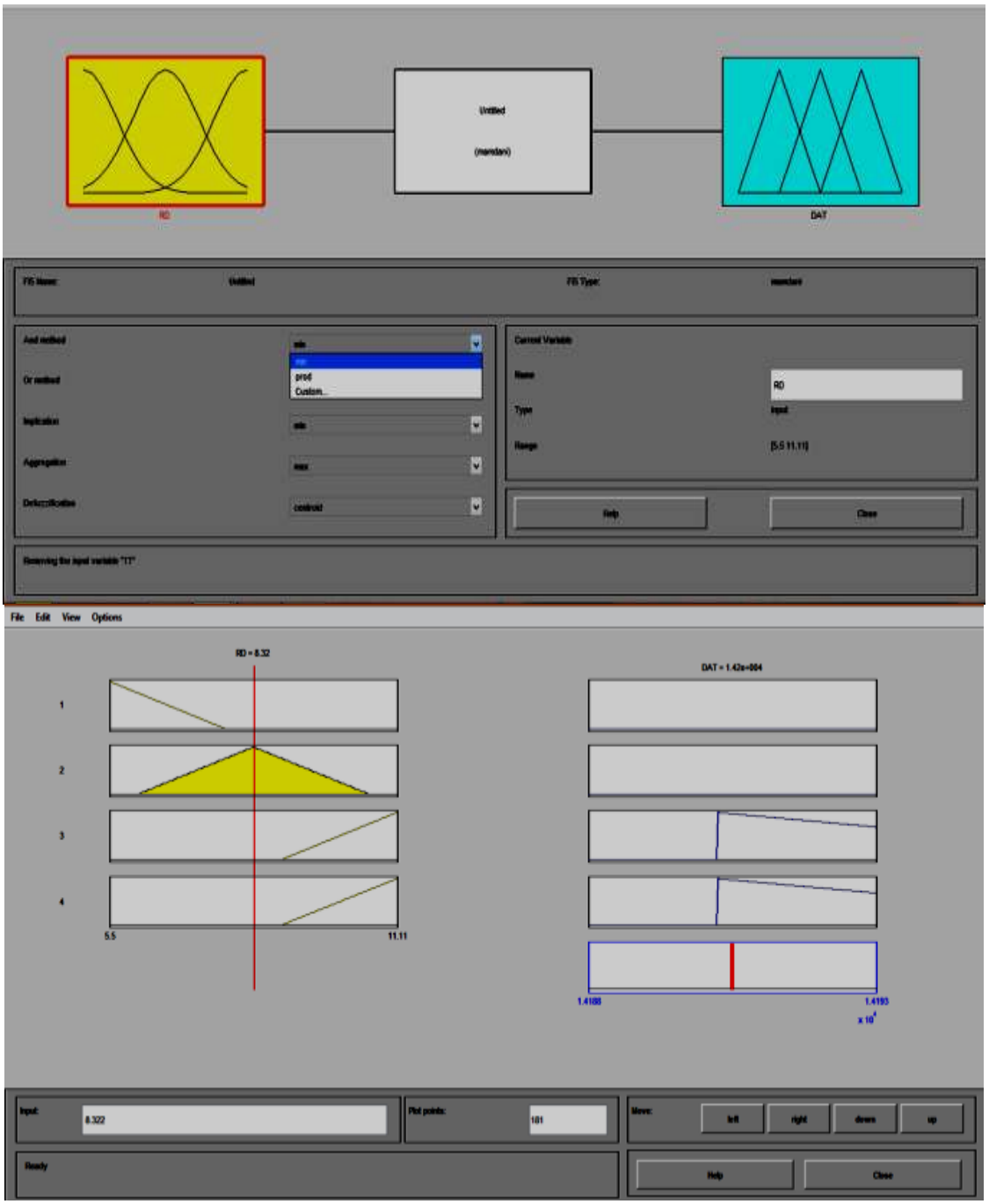
## RESULTS AND DISCUSSSION

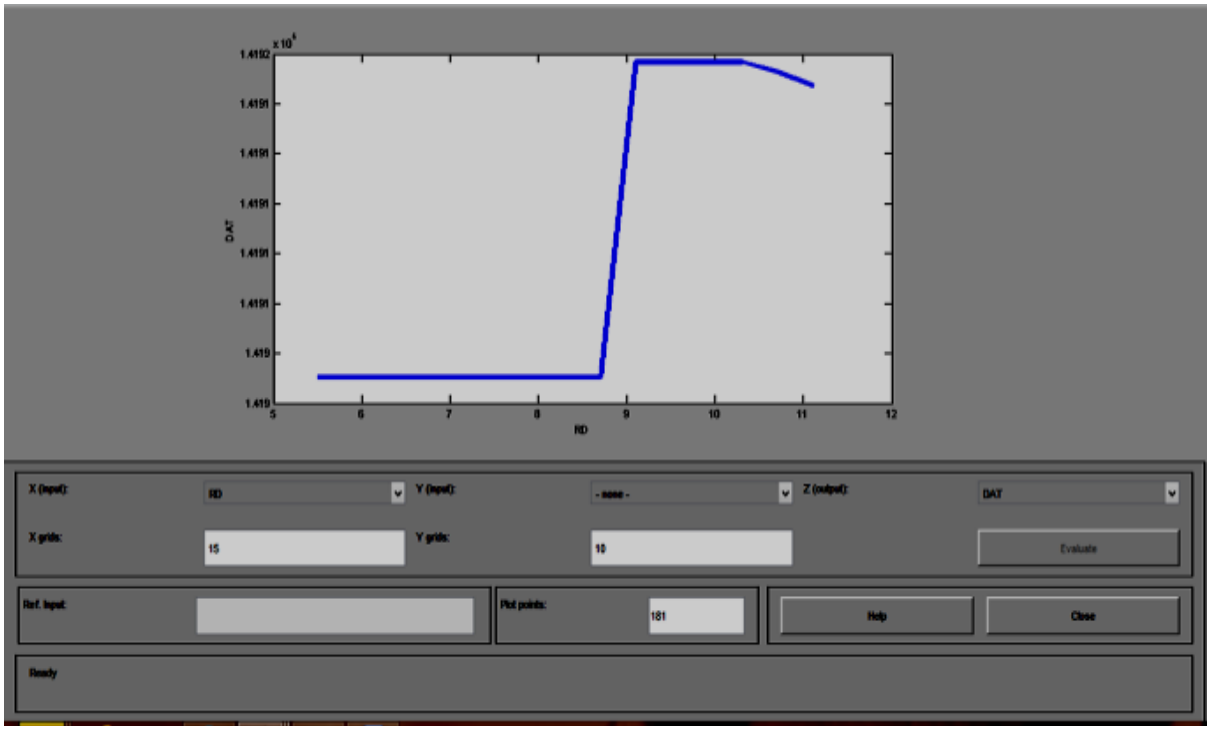
After applying the above formulae, our system reaches the following results:

**a.** Seek Time = 411 msec
**b.** Transfer Time = 13771 msec
**c.** Rotational Delay= 9.85 msec

**Table 1: Rotional Delay in Fuzzy**

| ***Rotational delay (msec)*** | | | ***Defuzzified Value Of Rotational Delay(Msec)*** |
|---|---|---|---|
| ***< a*** | ***b*** | ***c >*** | |
| **5.5** | **8.3** | **11.11** | **9.85** |






**d.** Disk Access Time = 411+13771+9.85 =14201 msec = **14.201 sec**

On applying C-LOOK algorithm, we find that the disk access time is 14.201 sec when rotational delay is 9.85 msec. Generally, average rotational delay for our system has been calculated 5.5 msec that gives disk access time =14.187 sec. The difference between both the values of access time comes out to be nearly 4 msec which cannot be neglected in the computer system.

Hence, our results are more accurate and significant.

***Conclusion***

We applied C-LOOK efficient algorithm .The rotational delay has been considered in fuzzy which is practically more significant. This delay affects the disk access time so to handle the inexact information; we design a fuzzy inference disk system using fuzzy arithmetic. This improved the accuracy in access time which affects the overall performance of the system. The pattern of request after applying C-LOOK algorithm shows that the read/write **head** does not stick to one area of disk.20 track requests with 20 MB data each are given as input to this algorithm that finally results with more accurate disk access time.

The main advantage of the study is that all the requests are fairly serviced such that none of the related factors causes starvation.

**REFERENCES**

**Dimitrijevic Z, Rangaswami R and Chang EY (2005)**. Support for Pre-emptive Disk Scheduling. *IEEE Transactions on Computers* **54**(10).

**Javed MY and Khan IU (2000).** Simulation and Perform- ance Comparison of Four Disk Scheduling Algorithms. *Proceedings of TENCON* 2000, Kuala Lumpur 10-15.

**Muqaddas H Abdulsalam and Salman A (2009)**. S-LOOK: A Preemptive Disk Scheduling Algorithm for Offline and Online Environments, CSIT, Lviv, Ukraine, 15-17th Oct 2009, 1-4.

**Raheja Supriya (2014)**. A New Approach to Disk Scheduling Using Fuzzy Logic. *Journal of Computer and Communications* **2** 1-5.

**Sachin & Silky Singh TP (2014).** Performance Measure Of CPU Scheduling For Real-Time Operating System In Fuzzy Environment. *Yamuna Journal of Technology Business Research* **4** (1-2) 54.1-54.9

**Suri PK and Mittal S (2011)**. Sim_Dsc: Simulator for Opti- mizing the Performance of Disk Scheduling Algorithms. *Global Journal of Computer Science and Technology* **11**(18) 1-6.
**Tsai C, Huang T, Chu E, Wei C and Tsai Y (2008)**. An Efficient Real-Time Disk-Scheduling Framework with Adaptive Quality Guarantee. *IEEE Transactions on Computers* **57**(5).
**Yager RR (1981).** A procedure for ordering fuzzy subset of the unit interval. *Information Science* **24** 143-161.
**Zadeh LA (1965)**. Fuzzy sets and systems. In: *System Theory* (Brooklyn N.Y.) 29-37.