3 SIMULATION ENVIRONMENT FOR RESEARCH WORK

This section describes all parameters that have been considered while evaluating the proposed algorithm. During the investigation, observations are gathered with different periodic task sets. Algorithms have been observed regarding Success Ratio (SR) and Effective CPU utilization (ECU). Empirical study has been executed with a hefty Dataset consist of more than 6800 task sets, and a set contains different 1 to 9 tasks where CPU load is dynamic for each task set and differs from 0.5 to 5. Algorithms have been executed on a 500-time line for each task set to authenticate the accuracy of all algorithms.

3.1 The Performance Parameter

The performance of all algorithms has been tested with two primary parameters call SR (Success Ratio) and ECU (Effective CPU Utilization) in this research. These parameters have been described as follows.

1) Success Ratio (SR) - In Soft Real-Time System, meeting the deadline is of utmost significance and crucial, and therefore, more concerned about the result, whether the task is meeting the deadline or not. Based on that, the most reliable metric is the Success Ratio (SR) and is defined as the ratio of a set of tasks that meets their deadline and the total number of tasks in the system. Success Ration determined with the following equation 1 [30],

$$SR = \frac{Number\ of\ Task\ successfully\ scheduled}{Total\ Number\ of\ Task\ arrived} \qquad (1)$$

2) Effective CPU Utilization (ECU) - It is potentially significant to know how effectively the scheduling algorithm exploits the tasks, peculiarly during heavy load conditions. Therefore, it also considered other performance metrics such as Effective CPU utilization (ECU). Effective CPU Utilization is defined as how much CPU time has been utilizing for the tasks which can meet their deadline. ECU determined with the following equation 2 [21],

$$ECU \ = \sum_{i \in R} \frac{V_i}{T} \qquad (2)$$

where,

- V represents task value and,
    - Task Value = time required to complete the task if the task meets its deadline.
    - Task Value = 0 if the task does not meet the deadline.
- R is a set of tasks, which is scheduled profitably, i.e., executed within its deadline.
- T is the total time of scheduling.

An online scheduling algorithm has a competitive factor C_f if and only if the value of the schedule of any finite sequence of tasks produced by the algorithm is at least C_f times the value of the schedule of the tasks produced by an optimal clairvoyant algorithm [5]. Since the maximum value obtained by a clairvoyant scheduling algorithm is a hard problem, thesis has instead used a rather simplistic upper bound on this maximum value, which is obtained by summing up the value of all tasks [42]. Therefore, thesis have considered the importance of ECU for the clairvoyant scheduling algorithm is 100%.

3.2 The Dataset

The Real-Time System has three types of tasks: Periodic, Aperiodic, and Sporadic (As explained in Section 1.3). All algorithms in this research have been evaluated with a periodic task set. This research has considered an extensive data set of periodic tasks. İt has found the 6800 tasks set, which vary in CPU load and the number of tasks within the task set. CPU load ranges from 0.5 to 5.0, and the number of task sets varies from 1 to 9. CPU load of task set is referred to total processor utilization factor ($\mathrm{U_p}$) and represent the fraction of processor time used by the periodic task set and calculated based on equation 3.

$$\mathrm{U_p} = \sum_{i=1}^{n} \frac{\mathrm{C_i}}{\mathrm{T_i}} \qquad (3)$$

Where $\mathrm{C_i}$ is execution time required by each task in task set and $\mathrm{T_i}$ is the occurrence period of each task in the task set. It has considered the underload scenario, overload scenario, and highly overload scenario. This 6800-task set contains 28600 tasks, and it has been tested on the 500-time unit to validate the correctness of the algorithm. To confirm the above task set, it has been published on the website [43][44]. Table 3.1 represents task set detail, and several different tasks have been considered for simulation in the proposed thesis.

Table 3.1 - Dataset detail for periodic Task Set

Load	**Number of Task in each Task Set**									**Task Set Load wise**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
0.50	50	50	50	50	00	00	00	00	00	2200
0.55	50	50	50	50	00	00	00	00	00	
0.60	50	50	50	50	00	00	00	00	00	
0.65	00	50	50	50	50	00	00	00	00	
0.70	00	50	50	50	50	00	00	00	00	
0.75	00	50	50	50	50	00	00	00	00	
0.80	00	50	50	50	50	00	00	00	00	
0.85	00	50	50	50	50	00	00	00	00	
0.90	00	50	50	50	50	00	00	00	00	
0.95	00	50	50	50	50	00	00	00	00	
1.00	00	50	50	50	50	00	00	00	00	
1.05	00	50	50	50	50	00	00	00	00	2000
1.10	00	50	50	50	50	00	00	00	00	
1.15	00	50	50	50	50	00	00	00	00	
1.20	00	50	50	50	50	00	00	00	00	
1.25	00	50	50	50	50	00	00	00	00	
1.30	00	50	50	50	50	00	00	00	00	
1.35	00	50	50	50	50	00	00	00	00	
1.40	00	50	50	50	50	00	00	00	00	
1.45	00	50	50	50	50	00	00	00	00	
1.50	00	50	50	50	50	00	00	00	00	
1.60	00	00	50	50	50	50	00	00	00	2600
1.70	00	00	50	50	50	50	00	00	00	
1.80	00	00	50	50	50	50	00	00	00	
1.90	00	00	50	50	50	50	00	00	00	
2.00	00	00	50	50	50	50	00	00	00	
2.25	00	00	00	50	50	50	50	00	00	
2.50	00	00	00	50	50	50	50	00	00	
2.75	00	00	00	50	50	50	50	00	00	
3.00	00	00	00	50	50	50	50	00	00	
3.50	00	00	00	00	50	50	50	50	00	
4.00	00	00	00	00	50	50	50	50	00	
4.50	00	00	00	00	00	50	50	50	50	
5.00	00	00	00	00	00	50	50	50	50	
Total Task Set	150	1050	1300	1500	1450	650	400	200	100	6800
Total Task	150	2100	3900	6000	7250	3900	2800	1600	900	28600

The entire Dataset is divided into three major categories as follow

1) if $\mathrm{U_p} \leq 1.0$, it considered as underload scenario,
2) if $1.0 < \mathrm{U_p} \leq 1.5$, it considered an overload scenario,
3) if $1.5 < \mathrm{U_p} \leq 5.0$, it considered a highly overload scenario,

In this research, all algorithms have been evaluated for the above three scenarios. For underload scenario, where $\mathrm{U_p} \leq 1.0$, total 2200 task set has been considered for evaluation of these algorithms. Each task set contains different number of tasks which varies between 1 to 5. For overload scenario, where $1.0 < \mathrm{U_p} \leq 1.5$, total 2000 task set has been considered for evaluation of these algorithms. Each task set contains different number of tasks which varies between 2 to 5. For highly overload scenario, where $1.5 < \mathrm{U_p} \leq 5.0$, total 2600 task set has been considered for evaluation of these algorithms. Each task set contains different number of tasks which varies between 4 to 9.

3.3 The Simulator

The entire simulator for all algorithms has been developed in C programming language, and the compiler is GNU GCC. The simulator has been executed on hardware configuration – Core i5 processor with 8 GB of RAM. Simulation has been carried out on a 64-bit Windows 10 Enterprise operating system. The Real-Time System has three types of tasks like Periodic task, Aperiodic task, and Sporadic task. All algorithm has been evaluated with a periodic task set. This simulator is using an extensive data set of periodic tasks.