# A PRACTICAL COMPARISON BETWEEN HARD, FIRM, AND SOFT REAL-TIME APPLICATIONS

**Abdusamea I. A. Omer**
Department of Computer Engineering and Information Technology, Faculty of Engineering, Sabratha University, Sabratha, Libya
Email: abdusamea@gmail.com

**Ibtihal Al-Ajeeli Ali Al-Mrabet**
Department of Computer Science, Faculty of Science, University of Azzawia, Azzawia, Libya
Email: booto199924@gmail.com

**Haneen Nouri Al-Khunjari Al-Juaidi**
Department of Computer Science, Faculty of Science, University of Azzawia, Azzawia, Libya
Email: Nouri152@gmail.com

## Abstract

This paper presents a comparative simulation study of real-time systems, focusing on Hard, Firm, and Soft Real-Time models. The system is implemented using C++ to simulate task execution under a fixed deadline constraint. The study evaluates system behavior based on general performance aspects such as deadline adherence, task handling, and execution behavior. The results demonstrate clear differences among the three models. The Hard Real-Time system enforces strict timing constraints, ensuring high reliability but limiting flexibility in task execution. The Firm Real-Time system provides a balanced approach by discarding late tasks while maintaining system continuity. The Soft Real-Time system prioritizes task completion over strict timing accuracy, resulting in higher flexibility. The study is based on fundamental concepts of real-time systems, particularly scheduling theory and deadline-based execution models. The main contribution is a simple simulation framework that highlights the trade-offs between timing strictness, flexibility, and system performance.

**Keywords:** Real-time systems, Hard real-time, Firm real-time, Soft real-time, Deadline scheduling.

## 1. Introduction

Real-time systems are computing systems in which the correctness of operations depends not only on the logical result but also on the time at which the result is produced [1]. These systems are widely used in embedded systems, industrial control, robotics, and other time-sensitive applications [2], [3].

In real-time system theory, computing systems are generally classified into three categories: Hard, Firm, and Soft Real-Time systems. Hard Real-Time systems require strict adherence to deadlines, where missing a deadline may lead to system failure. Firm Real-Time systems allow occasional deadline violations; however, late results are considered useless. In

contrast, Soft Real-Time systems are more flexible, where timing constraints are relaxed and missing deadlines does not lead to system failure [4], [5].

Research in real-time scheduling focuses on improving system efficiency, optimizing task execution, and managing workloads under timing constraints. These studies aim to enhance the predictability and reliability of time-critical systems across different application domains.

The objective of this work is to simulate and compare the behavior of Hard, Firm, and Soft Real-Time systems under identical conditions using a C++ implementation. A fixed deadline is used to ensure consistent evaluation across all models.

The significance of this study lies in demonstrating the trade-offs between strict timing constraints, system flexibility, and overall system performance in real-time environments.

### 1.1 Related Work

Gaukler et al. [1] examined control systems under the Weakly-Hard Real-Time model, focusing on how occasional deadline misses influence system stability and convergence behavior. The findings suggest that a bounded number of timing violations can be tolerated without significantly affecting overall system stability in practical control scenarios.

Hertneck et al. [2] investigated the stability of nonlinear systems operating under Weakly-Hard Real-Time constraints. The work analyzed the impact of missed deadlines and execution delays on system performance, providing analytical conditions under which stability can still be maintained despite timing violations.

Akesson et al. [6] provided an overview of how real-time systems are applied in industrial domains such as automotive, aerospace, and healthcare. The paper highlighted the practical use of both Hard and Soft Real-Time models and discussed the challenges associated with strict timing requirements in real-world systems.

Salamun et al. [4] introduced and formalized the Weakly-Hard Real-Time model, where a limited number of deadline misses is permitted. The survey discussed the relationship between Hard, Firm, and Soft Real-Time systems and emphasized that strict deterministic constraints may not always be suitable for modern and adaptive applications.

Seidel et al. [7] evaluated the performance of control systems operating under Weakly-Hard constraints. The study focused on system behavior under timing violations and showed that acceptable performance levels can still be achieved when deadline misses are controlled and bounded.

Ortiz Enguix et al. [8] analyzed schedulability in multicore real-time systems using fixed-priority scheduling. The work examined how resource contention affects timing predictability and deadline satisfaction in complex computing environments.

Sun et al. [9] compared Fixed-Priority and Earliest Deadline First (EDF) scheduling approaches, focusing on cache behavior and execution efficiency. The study demonstrated that scheduling decisions have a direct impact on timing predictability and overall system performance.

He et al. [10] addressed scheduling strategies for parallel real-time tasks with improved resource utilization. The research aimed to enhance system efficiency while preserving timing guarantees in real-time execution environments.

Niu and Musselwhite [5] proposed a scheduling approach for (m,k)-Firm Real-Time systems that balances reliability, energy constraints, and deadline satisfaction. The study showed how controlled task degradation can be used to maintain acceptable service quality under limited resources.

Yoo and Choi [3] evaluated the real-time performance of the RISC-V architecture, focusing on latency, determinism, and execution predictability. The results indicated that RISC-V can effectively support time-critical applications with predictable behavior.

Zou et al. [11] reviewed scheduling techniques in heterogeneous computing environments such as CPU, GPU, and FPGA systems. The survey compared Hard and Soft Real-Time requirements and highlighted how hardware acceleration influences real-time performance.

Baruah et al. [12] explored the use of machine learning techniques to support schedulability analysis in real-time systems. While improvements in prediction accuracy were observed, the study also noted limitations when applied to safety-critical applications.

Another study compared several embedded platforms used in real-time acquisition systems, evaluating latency, determinism, and cycle time across different architectures. The results showed that FPGA-based systems provide the highest determinism, RTOS-based systems offer a balanced trade-off between performance and flexibility, while Linux-based systems tend to introduce higher timing variability [13].

Seidel et al. [14] proposed a framework for designing control systems under Weakly-Hard Real-Time constraints. The work analyzed the impact of missed deadlines on system stability and highlighted the need for more flexible timing models compared to traditional Hard Real-Time approaches.

Although a considerable amount of research has been conducted on Hard, Firm, Soft, and Weakly-Hard Real-Time systems, most existing studies remain focused on theoretical modeling, isolated scheduling strategies, or system-specific analyses. There is still a clear lack of a unified experimental comparison that evaluates Hard, Firm, and Soft Real-Time systems under identical workloads and timing constraints within a single consistent environment. In particular, the practical behavioral differences between these models, such as task acceptance, deadline handling, and system continuity under identical conditions, are not sufficiently addressed in the literature. This highlights a gap in existing research, which this study aims to address by providing a unified simulation-based comparison that clearly demonstrates the operational differences between these real-time models under the same execution scenario.

## 2. Methodology

### 2.1 System Overview

A simulation system was developed to analyze and compare the behavior of Hard, Firm, and Soft Real-Time systems under a fixed deadline constraint of 2000 ms. The main objective

of this simulation is to investigate how different real-time scheduling policies affect system reliability, timing behavior, and task completion efficiency under identical workload conditions.

The system is designed with a simple architecture consisting of three main components: the Task Generator, which is responsible for producing tasks with variable execution times; the Simulation Engine, which executes tasks according to different real-time scheduling policies; and the Output Logger, which records execution results and performance statistics for further analysis.

### 2.2 System Architecture of the Real-Time Simulation Model

Figure 1 shows the architecture of the simulation model and the interaction between its main components.

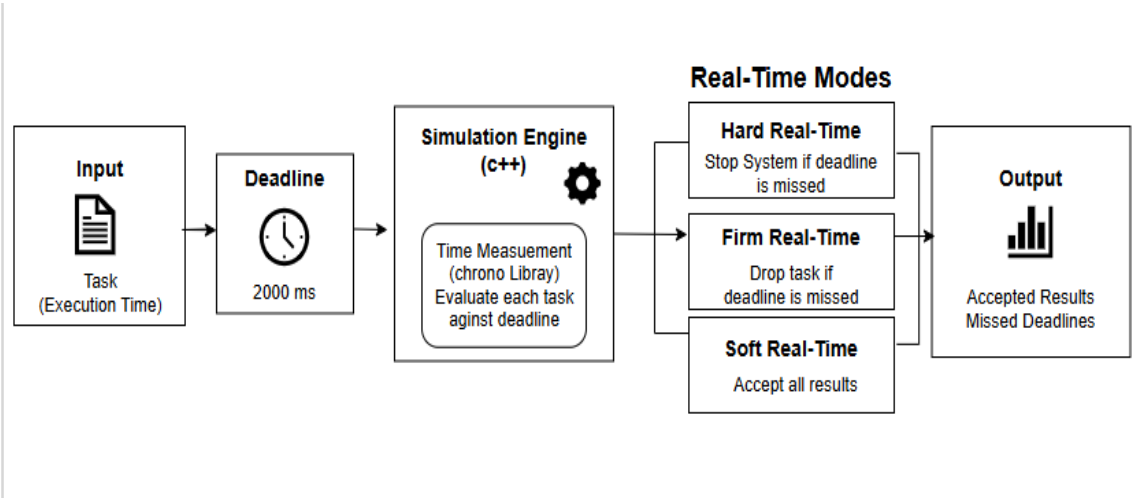



**Figure 1. System architecture of the real-time simulation model**

### 2.3 Execution Flow

Figure 2 illustrates the workflow followed for processing each task.

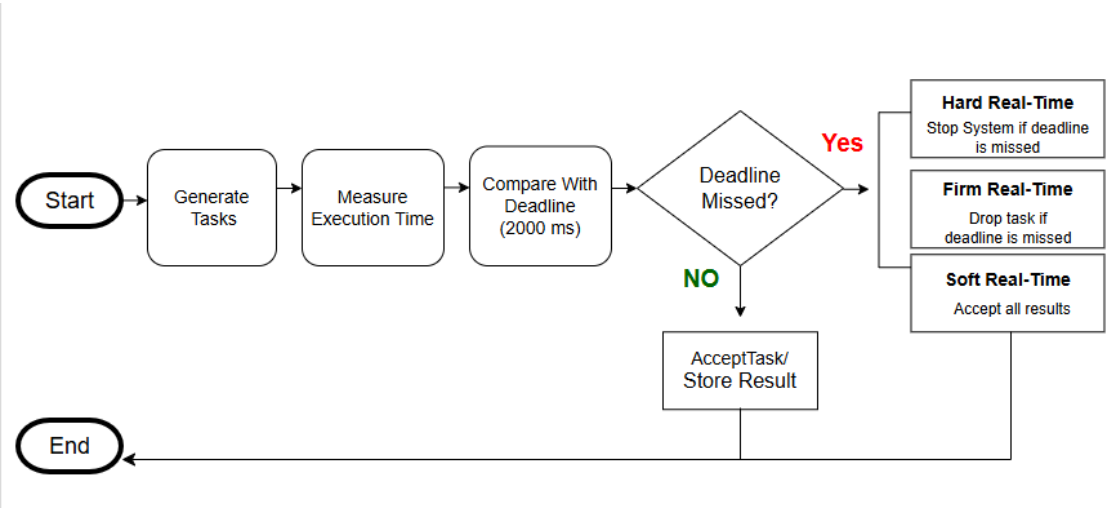



**Figure 2. Execution workflow of task processing**

### 2.4 Real-Time System Behavior

The behavior of the system varies depending on the selected real-time model. In a Hard Real-Time system, strict deadline constraints are enforced, where any task that misses its deadline is immediately rejected, and in extreme cases, system execution may be halted. In a Firm Real-Time system, tasks that miss their deadlines are discarded; however, the system continues execution normally without interruption. In contrast, a Soft Real-Time system executes all tasks regardless of deadline violations, prioritizing system continuity over strict timing accuracy.

### 2.5 Evaluation Metrics

To ensure a consistent and quantitative evaluation of system performance, several key metrics are defined. Deadline Miss Rate measures the percentage of tasks that fail to meet their deadlines, while Task Acceptance Rate represents the proportion of successfully executed tasks. Average Execution Time is used to compute the mean execution time of all processed tasks. Additionally, System Efficiency reflects the overall performance of the system based on both timing compliance and task completion ratio.

These metrics are defined at this stage of the study, while their actual computed values are presented in the Results and Discussion section after system execution.

### 2.6 Comparison Strategy

All three real-time systems are evaluated under identical workload conditions to ensure a fair comparison. The only variable factor in the experiment is the scheduling policy applied by each real-time model, while all other parameters remain constant. This controlled setup allows the study to isolate the impact of timing constraints on system behavior and performance.

### 2.7 Implementation

The system is implemented using the C++ programming language due to its efficiency in simulating real-time systems and providing precise performance measurement. The <chrono> library is used for high-precision time measurement, while <thread> is utilized to simulate task execution delays. Simple data structures are also used to manage the task set.

The system follows a unified execution flow where task generation, execution, and timing evaluation are handled within a single process. Each task is assigned a random execution time to simulate realistic workload conditions. The actual execution time is measured and compared against a fixed deadline (2000 ms).

Based on this comparison, the selected real-time policy (Hard, Firm, or Soft Real-Time) determines the system's decision: accept the task, reject it, or continue execution despite the deadline violation. All results are recorded for later analysis. The complete procedure is summarized in Algorithm 1.

**Algorithm 1:** Real-Time Task Scheduling Simulation

**Input:** Task set T, Deadline D = 2000 ms, System mode M ∈ {Hard, Firm, Soft}

**Output:** Execution results, Performance metrics

```
Generate task set T with random execution times
for each task t in T do
    Start execution timer
    Execute task t
    Stop timer and compute execution time ET
    if ET > D then
        if M = Hard then
            Reject task and stop execution
        else if M = Firm then
            Discard task and continue
        else if M = Soft then
            Accept task
    else
        Accept task
    Record result
end for
Generate final report
```

## 3. Results and Discussion

A comparative experiment was conducted between three real-time systems (Hard, Firm, and Soft Real-Time) using the same deadline value (2000 ms). The results are presented below along with their corresponding analysis.

### 3.1 Hard Real-Time System

**Table 1. Hard Real-Time performance**

| Total Runs | Deadline Misses | Accepted Tasks | Average Time (ms) |
|---|---|---|---|
| 3 | 1 | 2 | 1800.00 |

The Hard Real-Time system shows the lowest number of deadline misses, reflecting strict timing enforcement. However, this strictness leads to a reduced number of accepted tasks, as any deadline violation results in rejection.

### 3.2 Firm Real-Time System

**Table 2. Firm Real-Time performance**

| Total Runs | Deadline Misses | Accepted Tasks | Average Time (ms) |
|---|---|---|---|
| 8 | 3 | 5 | 1950.12 |

This system provides a balance between performance and timing constraints, as late tasks are discarded while execution continues. Therefore, it is more flexible than Hard Real-Time but less strict in timing accuracy.

### 3.3 Soft Real-Time System

**Table 3. Soft Real-Time performance**

| Total Runs | Deadline Misses | Accepted Tasks | Average Time (ms) |
|---|---|---|---|
| 8 | 3 | 8 | 1950.00 |

The Soft Real-Time system achieves the highest task acceptance rate because it does not reject tasks even when deadlines are missed. This makes it the most flexible system but the least strict in timing compliance.

### 3.4 Performance Comparison of Real-Time Systems

Figure 3 compares the number of deadline misses and accepted tasks for the three real-time systems, based on the results in Tables 1 to 3.

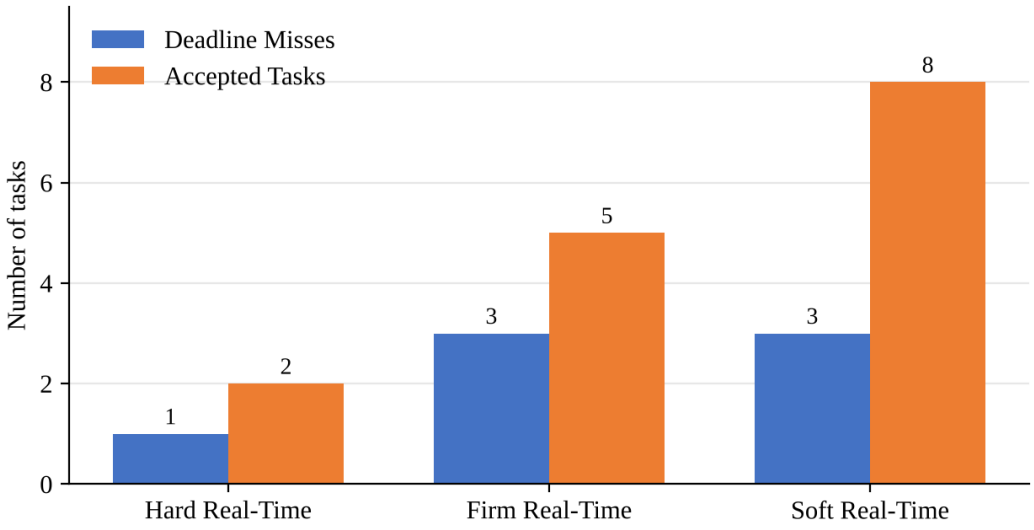



**Figure 3. Performance comparison of real-time systems**

## 4. Conclusion

This study presented a comparative simulation of real-time systems, focusing on Hard, Firm, and Soft Real-Time models implemented using C++. The simulation was designed to analyze system behavior under a fixed deadline constraint and identical workload conditions.

The results of the study highlight the fundamental trade-offs between timing strictness and system flexibility. The Hard Real-Time system ensures strict adherence to deadlines, making it suitable for critical applications where timing accuracy is essential. However, this strictness reduces flexibility in task execution. The Firm Real-Time system provides a balanced approach by discarding late tasks while maintaining overall system operation. In

contrast, the Soft Real-Time system offers the highest level of flexibility by prioritizing task completion over strict timing constraints.

Overall, the study demonstrates that no single real-time model is universally optimal. The choice of system depends on application requirements, where reliability, flexibility, and performance must be carefully balanced.

Future work may extend this simulation by incorporating dynamic task scheduling, priority-based algorithms, or real-world workload traces to achieve more realistic system behavior analysis.

## References

[1] Gaukler, M., Rheinfels, T., Ulbrich, P. and Roppenecker, G., 2020, Convergence rate abstractions for weakly-hard real-time control, *arXiv preprint*, arXiv:1912.09871. doi: 10.48550/arXiv.1912.09871.

[2] Hertneck, M., Linsenmayer, S. and Allgöwer, F., 2020, Stability analysis for nonlinear weakly hard real-time control systems, *IFAC-PapersOnLine*, 53(2), pp. 2594–2599. doi: 10.1016/j.ifacol.2020.12.307.

[3] Yoo, T. and Choi, B. W., 2024, Real-time performance benchmarking of RISC-V architecture: Implementation and verification on an EtherCAT-based robotic control system, *Electronics*, 13(4), p. 733. Available: https://www.mdpi.com/2079-9292/13/4/733.

[4] Salamun, K., Pavić, I., Džapo, H. and Čuljak, I., 2023, Weakly hard real-time model for control systems: A survey, *Sensors*, 23(10), p. 4652. doi: 10.3390/s23104652.

[5] Niu, L. and Musselwhite, J., 2024, Reliability-aware scheduling for (m,k)-firm real-time embedded systems under hard energy budget constraint, *Journal of Systems Architecture*, 154, p. 103185. Available: https://www.sciencedirect.com/science/article/pii/S138376212400122X.

[6] Akesson, B., Nasri, M., Nelissen, G., Altmeyer, S. and Davis, R. I., 2022, A comprehensive survey of industry practice in real-time systems, *Real-Time Systems*, 58(3), pp. 358–398. doi: 10.1007/s11241-021-09376-1.

[7] Seidel, M., Lang, S. and Allgöwer, F., 2024, On $\ell_2$-performance of weakly-hard real-time control systems, *European Journal of Control*, 80, p. 101056. doi: 10.1016/j.ejcon.2024.101056.

[8] Ortiz Enguix, L., Guasque Ortega, A., Balbastre Betoret, P., Simó Ten, J. E. and Crespo Lorente, A., 2024, Schedulability analysis in fixed-priority real-time multicore systems with contention.

[9] Sun, B., Kloda, T., Garcia, S. A., Gracioli, G. and Caccamo, M., 2024, Minimizing cache usage with fixed-priority and earliest deadline first scheduling, *Real-Time Systems*, 60(4), pp. 625–664. doi: 10.1007/s11241-024-09423-7.

[10] He, Q., Sun, Y., Jiang, X., Lv, M., Lee, J. and Guan, N., 2024, Real-time scheduling for parallel tasks with resource reclamation, *Real-Time Systems*, 60(2), pp. 291–327. doi: 10.1007/s11241-024-09421-9.

[11] Zou, A. et al., 2025, A survey of real-time scheduling on accelerator-based heterogeneous architecture for time critical applications, *arXiv preprint*, arXiv:2505.11970. doi: 10.48550/arXiv.2505.11970.

[12] Baruah, S., Ekberg, P. and Sudvarg, M., 2025, Learning-assisted schedulability analysis: Opportunities and limitations, *Real-Time Systems*, 61(3–4), pp. 332–358. doi: 10.1007/s11241-025-09450-y.

[13] Karthik, R. P., Govindasamy, M., Muthusamy, S. and Routray, A., 2026, Spectrally optimised YOLOv10s-SeqOpt framework for real-time UAV-based early detection of avocado foliar diseases in Indian orchards, *Scientific Reports*, 16(1), p. 17129. doi: 10.1038/s41598-026-50025-z.

[14] Seidel, M., Maggio, M. and Allgöwer, F. 2026. "A controller synthesis framework for weakly-hard control systems". *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. doi: 10.1109/RTAS68450.2026.00011.