e-ISSN: 2548-060X

International Journal of Energy Applications and Technologies

journal homepage: https://dergipark.org.tr/en/pub/ijeat

Review Article

An overview of fault tolerance techniques and energy consumption for real time embedded system

Nedjoua Louchene *, Riadh Hocine, Malika Bachir

Computer Science Department, University of Batna2, Batna, Algeria

ARTICLE INFO

* Corresponding author
louchene.katia@gmail.com

Received March 28, 2023
Accepted December 15, 2025

Published by Editorial Board
Members of IJEAT

doi: 10.31593/ijeat.1245738

ABSTRACT

The progress of technology has led to an increase in processor performance, but this has also made them more vulnerable to faults These faults can be transient or permanent. A transient fault happens for a short period time and then disappears without caused physical damage to the processor. On the contrary, a permanent fault disables a processor permanently, it impacts on system reliability and increases the power consumption. Due to the emergence of various fault-tolerance techniques maintaining time constraints has become more difficult. Redundancy is one strategy employed for this (spatial and temporal redundancy). Since embedded systems require the satisfaction of several constraints such as energy which is considered as a limited resource (battery size and recharge interval) that must be conserved as much as possible, and while fault tolerance increases its consumption, dynamic power and energy management must be taken into account to ensure the dependability of these systems. There have been numerous proposed task distribution/scheduling policies: on the one hand techniques for ensure timing and reliability and on the other techniques for enforce reducing power/energy consumption while also satisfying timeliness and reliability simultaneously This paper gives an overview of research projects that use fault-tolerance methods while taking into account timing, reliability, power/energy in critical real-time embedded systems' design.

Keywords: Real-time embedded systems, Scheduling algorithms, Fault tolerance, Energy minimization

1. Introduction

Embedded real-time systems are increasingly appearing in all areas of daily life, from the field of consumer applications (consumer electronics, automotive...) to the fields of critical applications (space, nuclear, health...). These systems are designed to perform complex and especially critical tasks and are subject to very strict constraints in terms of time and resources. In critical embedded real-time systems, a system failure can have catastrophic consequences (loss of time, money, or worse, loss of human lives), Faults are inevitable and can occur at any time, so these systems must be dependable. Dependability of a computing system is the ability to deliver service that can justifiably be trusted [1]. Reliability is a crucial attribute of the dependability of a system, it is a measure that allows to evaluate it quantitatively. One of the most means widely used in the literature to ensure dependability is fault tolerance which aims to preserve the expected service despite the occurrence of faults [2]. Redundancy is one of the methods used to achieve it. Two redundancy families are used, spatial and temporal redundancy. In the first family, we distinguish between active, passive and hybrid redundancy, the second family includes the technique of re-execution and the technique of Check pointing. Since embedded systems require the satisfaction of several constraints such as energy which is considered as a limited resource (battery size and recharge interval) that must be conserved as much as possible, although fault tolerance is necessary to increase system reliability, it often results in increased power consumption, it is essential to take into account dynamic power and energy management to ensure the dependability while minimizing energy usage of these systems. Many energy management techniques have been developed to minimize energy consumption in real-time systems, among others we find the Standby technique and the DVS (dynamic voltage scaling) technique. The latter is one of the most popular and widely deployed schemes. The objective of our work is to study more particularly the real-time scheduling problem and model it as an optimization problem to determine the best compromise between software and hardware components by satisfying multiple constraints (time, energy) which is more tolerant to faults in critical embedded real-time systems. Simultaneous optimization of several criteria adds a considerable difficulty to this scheduling problem. These criteria that are sought to be optimized are often contradictory, the optimization of one of the criteria leading to the degradation of the other. An optimization algorithm makes it possible to generate static scheduling and three objectives must be optimized: maximizing the overall reliability of the system and minimizing the overall execution time of all tasks and reducing energy consumption. Most of the real time distribution/scheduling algorithms developed in the literature to tolerate hardware faults in processors and communication media are based on the software redundancy technique Processor faults can be largely classified as transient or permanent. Many different replication methods were explored to make tradeoffs between fault tolerance and system resource usage, In the following we present some works that are divided into two categories, those that optimize two criteria and the others that optimize three criteria.

2.Bi-Criteria Timing-Aware Fault-Tolerance Techniques

The following proposed scheduling policies are limited to meeting timing and reliability constraints:

Assayad et al [3] have proposed an approach and that is a list scheduling heuristic that establishes priorities between the operations that need to be planned and determines which subset of processors they should be scheduled on, based on a bi-criteria compromise function called Bcf. In order to increase reliability, it employs active replication of operations. An algorithm parameter can be modified Bcf uses a parameter [0, 90◦] and the algorithm re-executed if the system reliability or schedule length requirements are not fulfilled. this process is repeated Until both objectives are satisfied.

Alain Girault et al [4] have proposed heuristic based on active redundancy that can tolerate a set number of failed communication links and fixed number of arbitrary processors. Fault tolerance is achieved offline in two steps: the first one is based on a formalism for transforming a specification of a non-redundant graph into a specification with redundant software components to tolerate the faults of processors and communication links The second phase consists of spatially and temporally allocating the software components of this components of this new redundant graph on the hardware architecture using a real-time distribution and real-time scheduling heuristics.

The focus of the study in [5] was hardware faults, particularly in the area of communication. He proposed a heuristic based on both active and passive backup copies and used linear programming to formulate the optimization problem of the fault-tolerant scheduling data with two types of backup copies in order to reduce scheduling length. Comparing the simulations to algorithms with just one type of backup copy reveals a considerable improvement. Results have shown that in all benchmarks, the FTA-RD heuristic that uses both replicated and deallocated backup copies outperforms the heuristic that uses only replicated and deallocated backup copies.

Authors of the study have observed that heuristic can reduce the scheduling length by 6.19% on an average when compared to only replication heuristic(Fig.1), and 19.29% on average when compared to only dealocation heuristic (Fig.2). Additionally, they observe that, in comparison to the optimal solution produced by the formulation of linear programming, their algorithm loses 8.41% on average in the scheduling length (Fig.3). The results are presented in the following graphs:

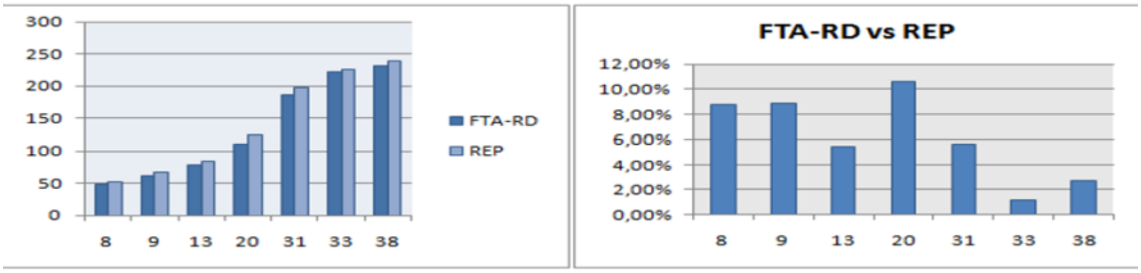

Fig. 1. Scheduling length of FTA-RD heuristic and only-replication heuristic

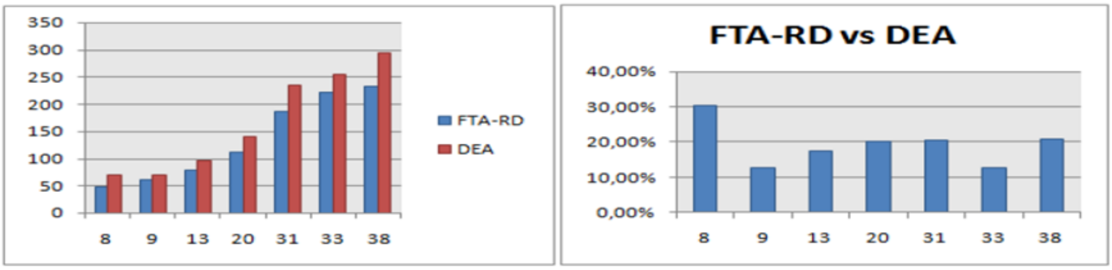

Fig. 2. Scheduling length of FTA-RD heuristic and only-deallocation heuristic

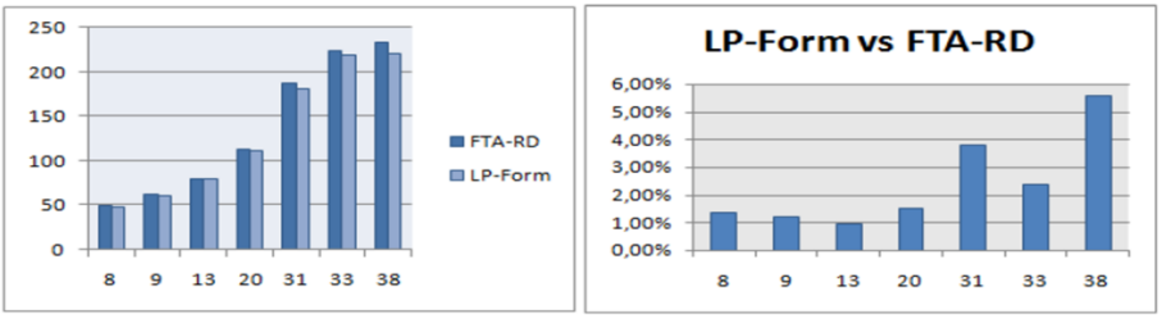

Fig. 3. Scheduling length of FTA-RD heuristic and LP-formulation solution

Bachir and Kalla [6] have proposed tow fault-tolerant scheduling heuristics to tolerate permanent faults of a single processor. The first heuristic based on passive replication with error detect technique Watchdogs and cost function called scheduling pressure. The second is based on hybrid of active and passive redundancy and cost function called scheduling pressure this heuristic benefits the both advantages of passive and active replication. It has a small execution delay even when there are no errors and a small recovery delay following failures. The Fig 4 has presented the schedule length L as a function of N, Where N: tasks number.

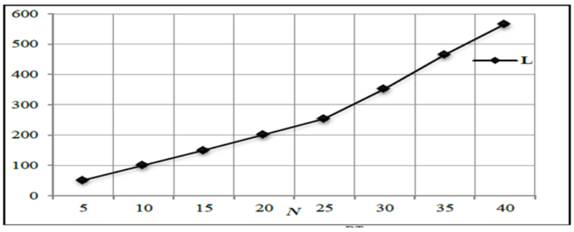

Fig. 4. Effect of N on AAA-FAULTDT for p=5 and CCR=2

and Fig5 has presented the schedule length L as a function of P, Where P: Processorsnumber.

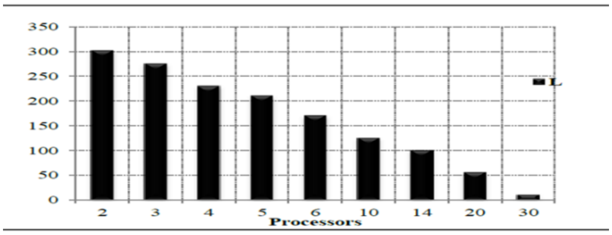

Fig 5. Effect of number of processors on AAA-FAULTDT for N= 40 and CCR=1

And Fig 6 has presented the schedule length L as a function of CCR. Where CCR: ratio of the average communication time and the average execution time.

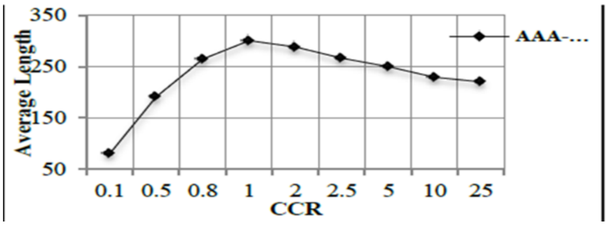

Fig. 6. Effect of CCR on AAA-FAULTDT for P = 6 and N =50

Kadda and Kalla [7] have proposed fault-tolerant scheduling heuristics to tolerate transient faults based on check- pointing with rollback technique and strategy of active replication that uses replicas collaboration to guarantee the task deadline even in the case of faults. The study found that this approach's average timing overhead is lower than check pointing strategy in the presence of many transient faults. Additionally, the proposed strategy performs better in terms of feasibility rate when there are multiple transient faults.

The study in [8] has proposed a fault tolerant scheduling heuristics based on passive and active redundancy of dependent tasks and also to tolerate permanent faults of a single processor, it based on the replication of each task in both primary and secondary copies. First, two copies are scheduled simultaneously (active replication), and when one copy completes its execution, it sends a signal to the other copy to stop running (passive replication), This method reduces the distribution/scheduling duration by a significant amount, especially since it eliminates the need for an error detection mechanism. the following diagrams describe the methodology:

doi: 10.31593/ijeat.1245738

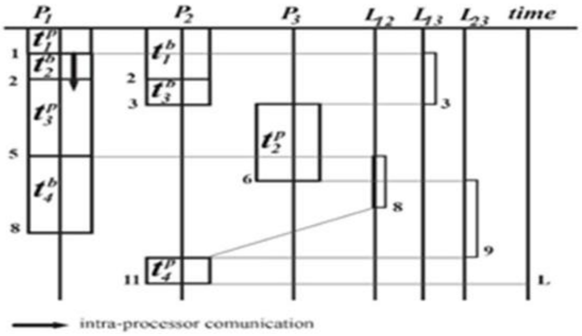

Fig. 7. Active replication of tasks

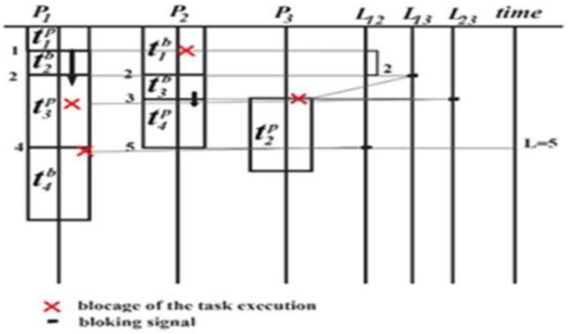

Fig. 8. Passive replication of tasks

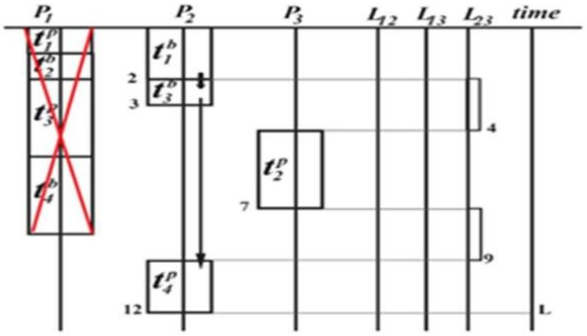

Fig. 9. Distribution/Scheduling after failure of P1

3. Tri-criteria timing-aware fault-tolerance techniques with power/energy-aware

Assayad et al [9] have proposed a scheduling heuristic to minimize the schedule length, the global system failure rate and the power consumption of the generated schedule. Their proposed method maximizes reliability through active replication and employs DVS to reduce power consumption. they obtained the Pareto solutions in the space (length, GSFR, power), they pre-define a virtual grid in the objective plane (GSFR, power) and they solve different single-objective problems constrained to each cell of the grid, by using a scheduling heuristic minimizing the schedule length. The study in [10] has proposed fault-tolerant scheduling heuristics to tolerate multiple transient faults based on check pointing technique and active replication strategy for maximizes reliability and Dynamic Voltage Frequency Scaling for energy minimization. The proposed method performs better than previous relevant approaches in terms of energy savings, according to simulation findings, while preserving the system’s required fault tolerance.

Authors of the study evaluated the performance of their methodology by comparing it with the following heuristics: DVFS CH and EXH FTS. And calculated the energy saving according to the number of tasks, the number of faults and checkpoint overhead they obtained the energy saving present in the graphs below:

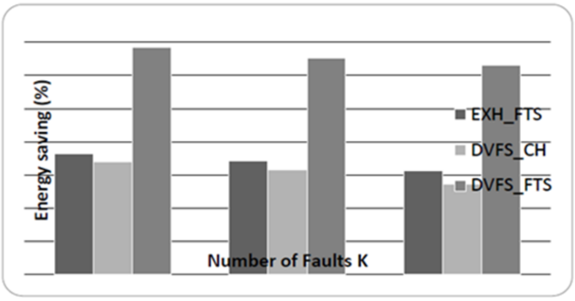

Fig. 10. The impact of number of faults on energy saving

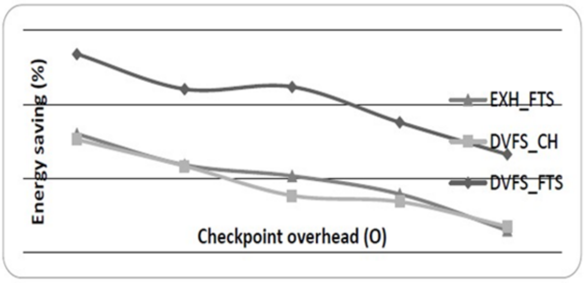

Fig. 11. impact of application size on energy saving considering k=3 faults

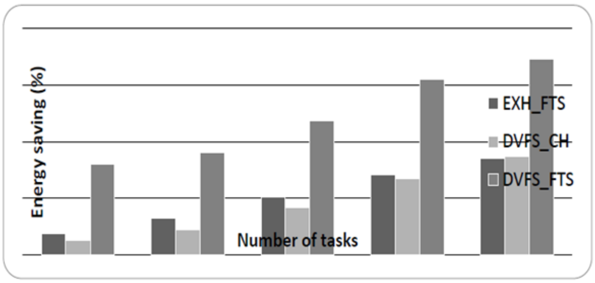

Fig. 12. The impact of checkpoint overhead on energy saving considering k=3 faults

Salim Kalla et al [11] have proposed an approach used degraded modes and dynamic voltage–frequency scaling to lower the power consumption the battery energy level has periodically verified and when an anomaly is detected a system switch between degraded modes according to a jitter. The authors of the study conducted a comparison between their approach and the static fault tolerance technique (SFTT) introduced in a previous work by Dima, Girault, and Sorel (2004), which also relies on degraded modes. The study's findings indicate that the approach proposed by the authors can achieve a 50% reduction in energy consumption

doi: 10.31593/ijeat.1245738

compared to the static fault tolerance technique, and a 40% reduction compared to the dynamic approach.

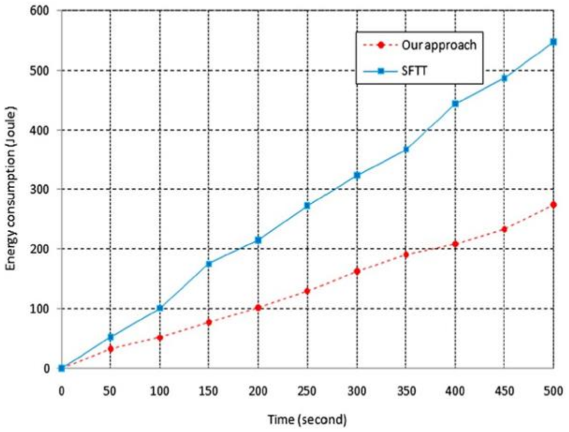

Fig. 13. A comparison between their approach and SFTT

A fault-tolerant DVFS algorithm has been developed by Djosic and Jevtic [12] for the real-time application of independent tasks. This technique combines re- execution recovery for fault tolerance with DVFS to optimize energy usage, although their application is limited to single processor systems. The simulation results have demonstrated that the suggested technique saves a significant amount of energy when compared to the solution without taking energy into account. Another benefit of the proposed method is that it produced almost ideal results in a reasonable amount of calculation time.

Yeganeh-Khaksar et al. [13] have presented a unique mapping and scheduling strategy in order to meet the chip-level power constraint while attaining the necessary reliability target, in multi-core embedded systems, propose a technique for mapping and scheduling periodic soft real- time tasks. Reliability-Aware Lowest Utilization Mapping, Maximum-Power-Aware EDF Scheduling, and Reliability and Peak Power-Aware Dynamic Voltage Frequency Scaling make up the three components of the proposed method. An ARM processor with core-level DVFS capability is used to evaluate the effectiveness of the suggested method. There are six different frequency/voltage levels from [0.85Volt, 1GHz] to [1.1Volt, 2GHz] to lower peak power consumption.

The study in [14] has proposed two energy-aware fault-tolerance scheduling algorithms based on a primary- backup approach called “Fault-tolerant Energy Efficient task scheduling with Delayed and Over loaded backups(FEED-O)”, and “FEEDO with Dynamic-deferring(FEED-OD)”. Primary tasks will be executed on a DVS-enabled processor, while the backup copies are scheduled on the auxiliary processor with DPM for reducing energy consumption. For lowering the energy consumption of scheduling periodic task-sets with rate monotonic scheduling, it has proposed overloading backup jobs and examined it. Indeed, a scheduling technique known as backup-overloading is used to assure that none of the backup jobs will be executed out on an auxiliary processor during runtime. If overloading of backup jobs on an auxiliary processor is done, then by delaying the start time of backup copies, it may further improve energy savings.

Tavana et al. [15] have proposed a scheme that simultaneously addresses reliability and energy consumption. The proposed scheme, by using both hardware redundancy (standby- sparing) and time redundancy (re-execution) in some cases, can tolerate many transient failures. They employed two strategies to reduce energy use: DVS (Dynamic Voltage Scaling) for the primary processor and DPM (Dynamic Power Management) for the spare unit.

4. Conclusion

In this paper, we have reviewed some closely related work in the literature and provided a review for different significant task mapping/scheduling rules for real-time embedded systems that use fault-tolerance solutions to meet reliability requirements and reduce energy usage. Firstly, in order to prevent missing timing limits, the scheduling process should take into account the additional timing overhead that fault-tolerance approaches introduce. For this reason, we concentrate on existing research real-time scheduling with timing-aware fault-tolerance Secondly, the power consumption of the cores will increase as a result of fault tolerance techniques, Therefore, we have focused on the related works, which consider timing-aware fault-tolerance, power/energy aware.

Conflict of interest

There is no conflict of interest.

References

[1] A. Avizienis, J.-c. Laprie, and B. Ran- dell, “Fundamental concepts of depend- ability,” 09 2001.

[2] M.BACHIR, *Ordonnancement Tolérant Aux Fautes Pour Les Système Distribués Temps Réel Embarqués (Doctoral dissertation, Université de Batna 2*),2019.

[3] I. Assayad, A. Girault, and H. Kalla, “A bi-criteria scheduling heuristic for distributed embedded systems under reliability and real-time constraints,” in *International Conference on Dependable Systems and Networks*, 2004. IEEE, 2004, pp. 347–356.

[4] A. Girault, H. Kalla, and Y. Sorel, “An active replication scheme that tolerates failures in distributed embedded real-time systems,” in *IFIP Working Conference on Distributed and Parallel Embedded Systems*. Springer, 2004, pp.83–92.

[5] C. Arar and M. S. Khireddine, “Anefficient fault-tolerant multi- bus data scheduling algorithm based on replication and deallocation,” *Cybernetics and*

Information Technologies, vol. 16, no. 2, pp. 69–84, 2016.

[6] B. Malika and H. Kalla, “A fault tolerant scheduling heuristics for distributed real time embedded systems,” *Cybernetics andInformation Technologies*, vol. 18, pp. 48–61, 092018

[7] B. Kada and H. Kalla, “A fault- tolerant scheduling algorithm based on checkpointing and redundancy for distributed real-time systems,” *Int.J. Distrib. Syst. Technol.*, vol. 10, no. 3, p. 58–75, jul 2019. [Online]. Available: https://doi.org/10.4018/IJDST.2019070104

[8] B. Malika, R. Hocine, N. Louchene, and H. Kalla, “A fault tolerant scheduling heuristics for distributed real time embedded system,” *in 2021 International Conference on Engineering and Emerging Technologies (ICEET)*, 2021, pp. 1– 6.

[9] I. Assayad, A. Girault, and H. Kalla, “Scheduling of real-time embedded systems under reliability and power constraints,” *in 2012 IEEE International Conference on Complex Systems (ICCS).* IEEE, 2012, pp.1–6.

[10] B. Kada and H. Kalla, “An efficient fault-tolerant scheduling approach with energy minimization for hard real-time embedded systems,” in *International Workshop on Distributed Computing for Emerging Smart Networks*. Springer, 2019, pp. 102–117.

[11] S. Kalla, R. Hocine, H. Kalla, and A. Chouki, “Graceful degradation for reducing jitter of battery life in fault-tolerant embedded systems,” *Inter- national Journal of Systems Science*, vol. 49, no. 11, pp. 2353–2361,2018.

[12] S. Djosic and M. Jevtic, “Dynamic voltage and frequency scaling algorithm for fault-tolerant real-time systems,” *Micro-electronics Re- liability*, vol. 53, no. 7, pp. 1036–1042, 2013.

[13] A. Yeganeh-Khaksar, M. Ansari, and A. Ejlali, “Remap: Reliability management of peak-power-aware real-time embedded systems through task replication,” *IEEE Transactions on Emerging Topics in Computing*,2020.

[14] S. Bansal, R. K. Bansal, and K. Arora, “Energy efficient backup overloading schemes for fault tolerant scheduling of real-time tasks,” *Journal of Systems Architecture*, vol. 113, p. 101901, 2021.

[15] M. K. Tavana, N. Teimouri, M. Abdollahi, and M. Goudarzi, “Simultaneous hardware and time redundancy with on-line task scheduling for low energy highly reliable standby-sparing system.” *ACM Trans.Embed. Comput. Syst.*, vol. 13, no. 4, pp. 86–1, 2014