2026 Design, Automation & Test in Europe Conference (DATE 2026)

Design and Optimization of Solar-Powered Embedded Systems with Uppaal Stratego

Ismael Samaye and Abdoulaye Gamatié
LIRMM, Univ. Montpellier and CNRS
Montpellier, France
firstname.lastname@lirmm.fr

***Abstract*—Energy intermittency in solar-powered embedded systems threatens Quality of Service (QoS) and system autonomy. In this study, we address the design of these systems with a formal co-design approach that provides verifiable guarantees, a critical advantage over traditional heuristic or predictive methods that often fail under unpredictable conditions. We use timed automata-based modeling within Uppaal Stratego to minimize grid reliance and battery capacity in a typical system, under QoS guarantees. Our methodology demonstrates that synthesized control strategies can reduce grid reliance by 58-72%, while an optimized task scheduling heuristic can decrease required battery capacity by up to 13% compared to the baseline. Our approach provides a formal basis for comparing these techniques to inform system design.**

***Index Terms*—Solar powered-embedded systems, energy minimization, Uppaal Stratego, formal modeling, resource allocation**

I. INTRODUCTION

Solar-powered embedded systems are increasingly prevalent in domains such as the Internet of Things (IoT), urban monitoring, and edge computing. These systems operate in environments where energy availability is inherently intermittent, which imposes stringent constraints on both reliability and Quality of Service (QoS). As highlighted by [1], reducing energy consumption is a critical factor for extending the operational autonomy of embedded platforms. Nevertheless, the inherent variability of weather conditions complicates the prediction of available energy resources, thereby introducing uncertainty into system design and operation.

Traditional energy optimization approaches primarily rely on heuristics or predictive models [2], which lack formal guarantees and often fail under unpredictable weather conditions. For instance, [3] proposed techniques for batteryless systems; however, these solutions are highly dependent on hardware architecture and exhibit limited adaptability to dynamic environments. Similarly, methods based on reactive policies [4] improve task continuity but do not ensure deadline compliance. Therefore, new approaches are needed to provide formal guarantees while addressing energy variability.

This work combines timed automata modeling with statistical model checking (SMC) and control strategy synthesis, using the Uppaal Stratego tool-set [5]. This approach aims to guarantee QoS under fluctuating energy conditions while simultaneously optimizing energy consumption. It not only provides a systematic way to handle resource variability but also establishes a foundation for designing resilient and energy-efficient solar-powered embedded systems. Finally, it aims to bridge the gap between theoretical guarantees and practical deployment in energy-constrained environments.

II. PROPOSED APPROACH

We model a solar-powered embedded system based on a modular architecture composed of multiple interconnected computing stations. Each station is equipped with a compute and energy storage equipment, e.g., a microcontroller and a battery. The whole system is powered by a solar panel and employs a *Maximum Power Point Tracking* (MPPT) algorithm to maximize energy production. The embedded system model is composed of the following three components:

- **Energy generator:** This component produces solar energy according to weather profiles, e.g., sunny *versus* cloudy conditions. It is typically implemented by a solar panel associated with a suitable MPPT module. The associated solar panel model generates harvested energy according to various weather profile parameters.
- **Computing station:** It executes the tasks realized by a system, according to different operating modes: *Low, Middle* and *High*. It comprises a local battery module to store the energy harvested by **Energy generator**. Additionally, a station is connected to the utility grid to enable a secondary energy source when solar energy is not available. In *Grid* mode, the station operates on grid power and adopts a similar energy consumption rate as the *Low* mode. to reduce the energy share from the grid.
- **Global controller:** This module orchestrates the energy and compute resource allocation to tasks, through workload migration and energy transfer between different stations to maintain QoS. These features are inspired by an edge data center paradigm such as [6].

We use timed automata to describe the above system model similarly to [7]. We then use the formal analysis capabilities of Uppaal Stratego, namely SMC and control strategy synthesis, to study system design and behavioral configurations that minimize grid energy consumption while satisfying a QoS constraint. The latter is defined as the system's ability to execute all tasks within a specific deadline time. In addition, we minimize the overall battery capacity of the system, thereby reducing its resource cost.

III. EVALUATION AND RESULTS

We conduct experiments on a system composed of nine interconnected stations, each equipped with a System-on-Chip

978-3-9826741-1-7/DATE26/

(SoC) dissipating a maximum power of 10W [8]. The system is powered by a 2 m^2 solar panel with a conversion efficiency of 19% and a MPPT. Each station is equipped with one battery of 200 Wh. Considered weather profiles include *sunny, rainy, cloudy, foggy, and erratic* conditions, derived from Open-Meteo datasets [9]. They offer a diversity of solar irradiation scenarios that enables the evaluation of the system design relevance under consideration.

A. Baseline design evaluation w.r.t. QoS constraint

Let us consider the baseline system design defined by the above parameters. We first evaluate the feasibility of completing all tasks within one day by the system, regardless of the solar irradiation conditions. We use a specific SMC query in Uppaal Stratego to compute the success probability of execution the corresponding workload within a 1440-minute horizon, i.e., a full day, as follows:

```
Pr[<=1440](<> forall (i:id_t) Task[i] == 0)
```

The obtained probability ranges from 95% to 100%. We further validate this trend through 1,000 randomized simulations, ensuring statistical robustness.

Next, we compute (i) the probability that all stations complete their allocated tasks without relying on the electrical grid, and (ii) the average grid energy consumption over 1,000 randomized 24-hour simulations. We assess the capacity of the system to operate autonomously w.r.t. grid energy. For this purpose, we use the following Uppaal queries:

```
Pr<=1440 Task[i] == 0 && forall (i:id_t) station(i).b
    == 0)
E[<=1440; 1000] (max : sum(i:id_t) station(i).b)
```

The former query evaluates the probability of successfully completing all tasks without grid energy, within a range from 40% to 65%. This indicates that the grid remains essential for the baseline design. The latter query computes the average consumed grid energy. Figure 1 shows that a sunny day requires only 14.02 Wh from the grid, whereas other solar irradiation profiles demand 5–8 times more energy.

B. Design optimization for more grid energy-autonomy

We explore a few techniques for increasing the system autonomy regarding the grid energy through workload scheduling and battery storage resizing.

First, we consider the control strategy synthesis available in Uppaal Stratego. The objective is to generate execution scenarios in which the stations execute the tasks while minimizing the use of grid energy. Applying such control strategies enforces the system behavior in a way that reduces grid energy consumption. Figure 1 shows that the expected energy reduction ranges from 58% to 72% (curve in green color), depending on the solar profiles. The strategies synthesized by Uppaal Stratego aim to decrease energy transfer transactions compared with the baseline scenario. As a result, task migrations become more frequent. These migrations are prioritized because they generally appear less energy-expensive than energy transfers, which incur energy losses.

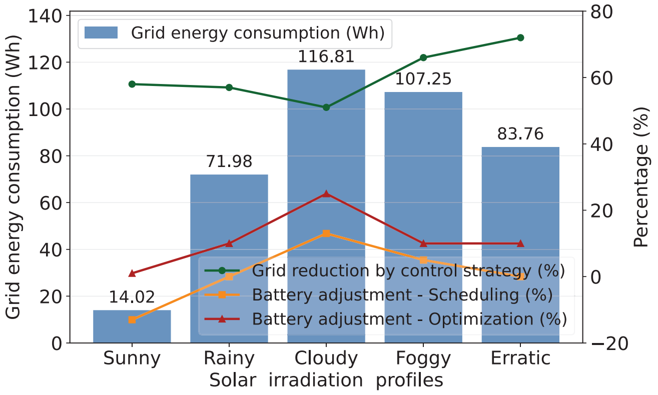

Fig. 1: Solar profile impact on grid energy and battery sizing.

Next, based on an analysis of energy transfer dynamics, we evaluate an alternative technique that resizes battery capacity using a *binary search method*, combined with the previous control strategy. This technique relies on increasing the battery capacity to store more solar energy and reduce dependence on the grid by compensating for fluctuations in solar energy production. Our results show that this technique increases the baseline battery capacity by 1% to 25% (see Figure 1, red color), depending on solar irradiation profiles, and under the previous control strategy. In particular, irregular irradiation profiles require larger storage margins to ensure service continuity.

The final technique postpones task execution to day-periods with solar energy production peaks to execute the most demanding workloads, thereby limiting costly energy transfers between stations. This can reduce the required battery storage as harvested energy is directly consumed by the tasks scheduled on purpose. Applying this scheduling technique decreases battery size from –13% to +13% compared with the baseline battery capacity (see Figure 1, orange color curve). While task shifting provides a better battery sizing than binary search under control strategy, it can degrade QoS, as deferring tasks to match favorable solar periods may violate deadline constraints. This is a trade-off to consider carefully for ensuring the operational viability of the system under stringent QoS constraints.

IV. CONCLUSION

We presented a formal design approach that evaluates some design issues for solar-powered embedded systems, using statistical model checking and control strategy synthesis. Control strategies synthesized with Uppaal Stratego provide strong QoS guarantees but do not enable the minimization of battery storage as with a simple heuristic scheduling technique. However, the latter can yield weaker QoS guarantees. Future work will investigate how to combine the capabilities of both techniques to achieve optimal designs.

ACKNOWLEDGMENTS

This work has received the financial support from Région Occitanie under the grant Cossues-EnR.

REFERENCES

[1] I. Sofianidis, V. Konstantakos, and S. Nikolaidis, “Reducing energy consumption in embedded systems applications,” *Technologies*, vol. 13, no. 2, 2025. [Online]. Available: https://www.mdpi.com/2227-7080/13/2/82

[2] M. Gerndt, M. Ispir, I. Nunez, and S. Benedict, “Energy-aware duty cycle management for solar-powered iot devices,” *Sensors*, vol. 25, no. 14, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/25/14/4500

[3] S. Tabrizchi, S. Angizi, and A. Roohi, “Diac: Design exploration of intermittent-aware computing realizing batteryless systems,” in *2024 Design, Automation &. Test in Europe Conference &. Exhibition (DATE)*, 2024, pp. 1–6.

[4] S. Tabrizchi, N. Taheri, J. Feng, N. Sehatbakhsh, D. Z. Pan, and A. Roohi, “Src: Sustainable reactive computing for battery-free edge intelligence,” in *2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC)*, 2024, pp. 93–98.

[5] A. David, P. G. Jensen, K. G. Larsen, M. Mikučionis, and J. H. Taankvist, “Uppaal stratego,” in *Tools and Algorithms for the Construction and Analysis of Systems*, C. Baier and C. Tinelli, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 206–211.

[6] M. D. M. da Silva, A. Gamatié, G. Sassatelli, M. Poss, and M. Robert, “Optimization of data and energy migrations in mini data centers for carbon-neutral computing,” *IEEE Transactions on Sustainable Computing*, vol. 8, no. 1, pp. 68–81, 2023.

[7] A. Gamatié, G. Sassatelli, and M. Mikučionis, “Modeling and analysis for energy-driven computing using statistical model-checking,” in *2021 Design, Automation &. Test in Europe Conference &. Exhibition (DATE)*, 2021, pp. 980–985.

[8] kontron S&T Group, “Dd3714-v/r mstx industrial series,” https://www.bicker.de/files/downloads/datenblatt/d3714-x.pdf, 2025, consulté le 11 septembre 2025.

[9] P. Zippenfenig, “Open-meteo.com weather api,” 2023, doi:10.5281/zenodo.7970649, Creative Commons Attribution 4.0 International. [Online]. Available: https://open-meteo.com