# Addressing the Execution Control Problem in Mixed-Criticality Systems

Dávid Juhász
*TU Wien, Vienna, Austria*
*Imsys AB, Stockholm, Sweden*
*david.juhasz@tuwien.ac.at*

Axel Jantsch
*TU Wien, Vienna, Austria*
*axel.jantsch@tuwien.ac.at*

## 1. Introduction

Real-Time (RT) scheduling has a long history with well-established results [1]. The conservative approach for ensuring reliability of critical RT applications accepts under-utilized and over-sized systems for performance guarantees. However, economic considerations favor resource sharing to increase hardware utilization and decrease costs. Mixed-Criticality (MC) Systems (MCSs) — a term coined by Vestal [2] — are actively researched [3] as a way of providing performance guarantees for and safe resource sharing among tasks of different importance.

Providing RT guarantees for MCSs requires tight integration of power management and task allocation i.e., mapping and scheduling

We formulate integrated task allocation and power management as a *dynamic* control problem of optimization theory [4], as it continuously optimizes for given objectives, and call it *Execution Control Problem (ECP)*. Based on this formalism we define ECP for MCSs with two novel characteristics:

- performance requirements of MCSs are defined as Quality of Service (QoS) constraints based on the min-plus calculus;
- RT constraints of tasks are unified under the proposed QoS constraint formulation.

Simulation-based preliminary evaluation shows that ECP allows: (1) to optimize energy efficiency on-demand by keeping tasks with higher QoS requirements maintaining high throughput, while the performance of tasks with lower QoS requirements may decrease; and (2) to adjust energy consumption and throughput by tuning QoS requirements of tasks.

## 2. MC Models

Task models of MCSs are typically based on the notation of sporadic task systems [3, Section 2]. Each task is defined by its period (minimal inter-arrival time), deadline, Worst-Case Execution Time (WCET), and Criticality Level (CL). Most, in some cases indeed all, of the parameters have values depending on the CL. Tasks generate a potentially unbounded sequence of jobs to execute. The MCS starts running at its lowest CL and executes jobs for all tasks. When any of the executed jobs violates its WCET with respect to the current CL of the system, current CL is raised to the next level and jobs whose generating task has a lower CL are ignored.

While the academically investigated MCS models allowed the research community to establish key properties of MCSs, those models do not address key requirements in industrial practice. Abandoning tasks and never returning to a low CL state is a major issue raised by systems engineers. Some approaches mitigate the issue by reconfiguring the system [3, Section 6]. Further, the compliance of MCS models to safety-related standards is important. [5] exemplifies the delicate connection between CLs and safety assurance levels (Safety Integrity Level (SIL) in IEC 61508, Automotive SIL (ASIL) in ISO 26262, or Development Assurance Level (DAL) in DO 178C), and debates the practical applicability of the MCS models. The common ground of concerns is that safety standards require separation among different assurance levels, which the researched MCSs do not provide among CLs [3], [5], [6].

In contrast to other work, we use a sporadic task model without CL-dependent values. We define MCSs based on ”continuous” QoS requirements rather than discrete CLs. The model provides separation of tasks with respect to a minimal level of required service.

## 3. Preliminary Results

### 3.1. Platform and Task Models

We model the platform as a set of Processing Elements (PEs) with the effects of shared memory and the NoC being implicit. While this abstraction limits the accuracy, it does not, in principle, change the approach taken. Each PE supports Dynamic Voltage and Frequency Scaling (DVFS) with an individual set of Voltage-Frequency Pairs (vf-pairs). We model power dissipation following the classic power model: the total power dissipation consists of dynamic and static parts.

We follow the sporadic task model. A task is defined by its minimal inter-arrival time, relative deadline, PE-dependent WCET, and PE-dependent power dissipation. Each task generates a potentially infinite sequence of jobs.

We consider a state-based execution model with known system dynamics and stochastic components.

### 3.2. Execution Control Problem (ECP)

We define ECP as a multi-objective stochastic model predictive control problem.

The optimization objective incorporates two performance characteristics: (1) energy consumption of the system and (2) the number of deadline misses indicating throughput of the system. The relative importance of the objectives may be adjusted dynamically by an optimization weight.

The optimized control needs to meet two kinds of constraints: (1) jobs complete within the minimal inter-arrival time of their generating tasks, and (2) the number of deadline misses for each task are bounded by QoS constraints.

We define QoS requirements as event-rate curves of deadline misses, which allows covering a spectrum of weakly hard RT constraints [7]. The bounds may be tuned dynamically: deadline miss constraints are defined by the convolution in min-plus algebra as in network calculus [8].

### 3.3. Solving ECP

Computing the convolution for all possible time intervals during the system's lifetime is costly. Hence, we introduce a window-based approximation for the constraints. The window length is a design parameter of ECP. A longer window increases accuracy as in approximating the infinite constraint better, while a shorter window lowers computational complexity at the cost of losing accuracy.

ECP with the window-based QoS constraints is a regulator problem with discrete stochastic processes [4, Section 7]. Following the literature we develop a heuristic Execution Controller (EC) for preliminary evaluation of ECP.

### 3.4. Simulation Results

We performed simulations with the chronSIM simulator of the industrial INCHRON Tool-Suite [9]. User-defined schedulers in chronSIM enable us to implement our EC.

Simulations were performed with randomly generated synthetic task systems, whose timing characteristics are representative for a real world automotive application [10]. We generated and simulated task systems with utilization levels between $20\%$ and $80\%$.

Consider a system with depleting battery and increasing importance of energy consumption as an example. Energy can be saved by reducing throughput as the optimization weight is tuned towards energy efficiency, see Fig. 1.

## 4. Conclusion

We propose ECP that defines MCSs with tasks subject to QoS requirements covering a unified spectrum of RT constraints. ECP is an MC model, which allows dynamic flexibility in fine tuning QoS requirements and specifies separation with respect to those requirements.We believe that ECP constitutes a promising ground for research in MCSs and plan to continue this work in several directions.

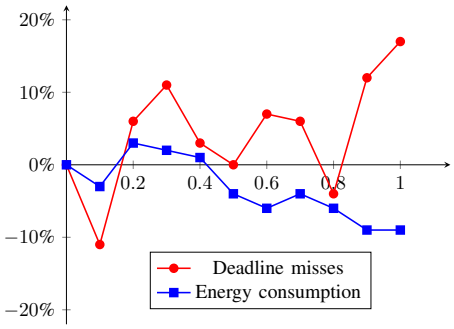


**Figure 1: The horizontal axis swipes over the possible values of the relative weight of optimizing for energy efficiency over deadline misses ($0 \rightarrow$minimize lost deadlines; $1 \rightarrow$ minimize energy consumption). The vertical axis shows values relative to $0$ weight. Each data point is the average of values from $8$ simulation runs.**

## Acknowledgments

This research was partially funded by the European Unions Horizon 2020 Framework Programme for Research and Innovation under grant agreement no 674875 (oCPS Marie Curie Network).

## References

[1] L. Sha, T. Abdelzaher, K.-E. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A. K. Mok, "Real Time Scheduling Theory: A Historical Perspective," *Real-Time Syst.*, vol. 28, no. 2-3, pp. 101–155, nov 2004.

[2] S. Vestal, "Preemptive Scheduling of Multi-criticality Systems with Varying Degrees of Execution Time Assurance," in *28th IEEE Int. Real-Time Syst. Symp. (RTSS 2007)*. IEEE, 2007, pp. 239–243.

[3] A. Burns and R. I. Davis, "Mixed criticality systems - a review," 2018. [Online]. Available: https://www-users.cs.york.ac.uk/burns/review.pdf

[4] D. A. Pierre, *Optimization Theory with Applications*. Dover Publications, 1986.

[5] R. Ernst and M. Di Natale, "Mixed Criticality Systems-A History of Misconceptions?" *IEEE Des. Test*, vol. 33, no. 5, pp. 65–74, 2016.

[6] A. Esper, G. Nelissen, V. Nélis, and E. Tovar, "How realistic is the mixed-criticality real-time system model?" in *Proc. 23rd Int. Conf. Real Time Networks Syst. - RTNS '15*. New York, New York, USA: ACM Press, 2015, pp. 139–148.

[7] G. Bernat, A. Burns, and A. Liamosi, "Weakly Hard Real-Time Systems," *IEEE Transactions on Computers*, vol. 50, no. 4, pp. 308–321, 2001.

[8] J.-Y. Le Boudec and P. Thiran, *Network Calculus*, version ap ed., ser. Lecture Notes in Computer Science, J.-Y. Le Boudec and P. Thiran, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, vol. 2050.

[9] S. Anssi, K. Albers, M. Dörfel, and S. Gerard, "chronVAL/chronSIM: A Tool Suite for Timing Verification of Automotive Applications," in *Embed. Real Time Softw. Syst. 2012*, 2012.

[10] S. Kramer, D. Ziegenbein, and A. Hamann, "Real World Automotive Benchmarks for Free," in *6th Int. Work. Anal. Tools Methodol. Embed. Real-time Syst.*, 2015.