# Open Problems on Non-preemptive Scheduling of Mixed-Criticality Real-Time Systems

Mitra Nasri and Gerhard Fohler
Chair of Real-time Systems, Technische Universität Kaiserslautern, Germany
{nasri,fohler}@eit.uni-kl.de

## I. INTRODUCTION

In many industrial systems, design complexity is partly reduced by the use of *non-preemptive* message passing or execution. Doing so, context switch overheads are diminished and resource sharing mechanisms are simplified. However, in a mixed-criticality system, long blocking time caused by non-preemptive execution of a task which has lower priority than some other high criticality (HC) tasks might threaten schedulability of the HC tasks in their HC mode. In single criticality systems, it has been proven that non-preemptive scheduling is an NP-Hard problem for periodic [1] or harmonic tasks [2], where each period is an integer multiple of shorter periods. In this work, we want to extend the existing solutions for single criticality systems to two criticality systems to solve the scheduling problem of the Vestal task model for mixed criticality systems [3].

Recently, an online non-preemptive non-work-conserving scheduling algorithm called *Precautious-RM* has been introduced in [4]. This algorithm prioritizes the jobs according to rate monotonic, however, it schedules a job only if its execution will not cause a deadline miss for the next instance of the task with the smallest period. Otherwise, it schedules an idle interval until the next release of the highest priority task. It has $O(n)$ computational complexity and it guarantees the schedulability of harmonic tasks in three cases: a) period ratio is 2, b) period ratio is greater than 2, and c) tasks have arbitrary period ratio, yet there are enough *vacant intervals*. A vacant interval is defined as the slack of two consecutive instances of the task with the smallest period. Period ratio is defined as the ratio of two consecutive periods, i.e., $T_i/T_{i-1}$ if the periods are sorted in ascending order. In the third case, suggested conditions are only sufficient since the general problem is NP-Hard, i.e., system is schedulable by Precautious-RM if each task has its guaranteed slot before its deadline.

## II. OPEN PROBLEMS

At this stage, we limit the solution to uni-processor systems and harmonic task sets. We assume tasks are indexed according to their period so that $T_1 \leq T_2 \leq \ldots \leq T_n$ where $T_i$ is the period of task $\tau_i$ and $n$ is the number of tasks. We consider a dual criticality task set with HC and low criticality (LC) tasks which are independent from each other. To guarantee the schedulility of the tasks, we use Precautious-RM as the underlying scheduling algorithm. In this section, we investigate schedulability challenges in two cases; a) all tasks are non-preemptive, b) HC tasks are non-preemptive while LC tasks can be preempted.

### *A. Schedulability Challenges for Fully Non-preemptive Task Sets with Two Criticality Levels*

Although Precautious-RM guarantees the schedulability of harmonic tasks in some cases, it cannot be easily applied in mixed-criticality systems because it only schedules a low priority task, i.e., a task with larger period, if this task will not cause a deadline miss for the next instant of the task with the smallest period in the task set, i.e., $\tau_1$. As a result, the following two challenges must be solved to be able to use Precautious-RM with mixed-criticality task sets:

- Precautious-RM does not distinguish between different execution times of a task, hence, it may schedule an HC task which is not yet switched to its HC mode without paying attention to possible mode changes during the execution of this task. As a result, it may cause a long blocking time for other HC tasks because of a late mode change. If those HC tasks have earlier deadlines than the finish time of the executing HC task, their schedulability will be endangered. We call it the *long blocking problem*.
- Before scheduling a low priority task, Precautious-RM checks if this task causes a deadline miss for the next instance of the task with the smallest period. If it does, this task will not be scheduled, instead, an idle interval will be scheduled until the next release of the task with the smallest period. However, in mixed-criticality systems, the task with the smallest period might be a low criticality task which can be ignored after switching the mode to HC. As a result, the schedulability condition must be revised according to the occurrence of a mode change during the execution of a task.

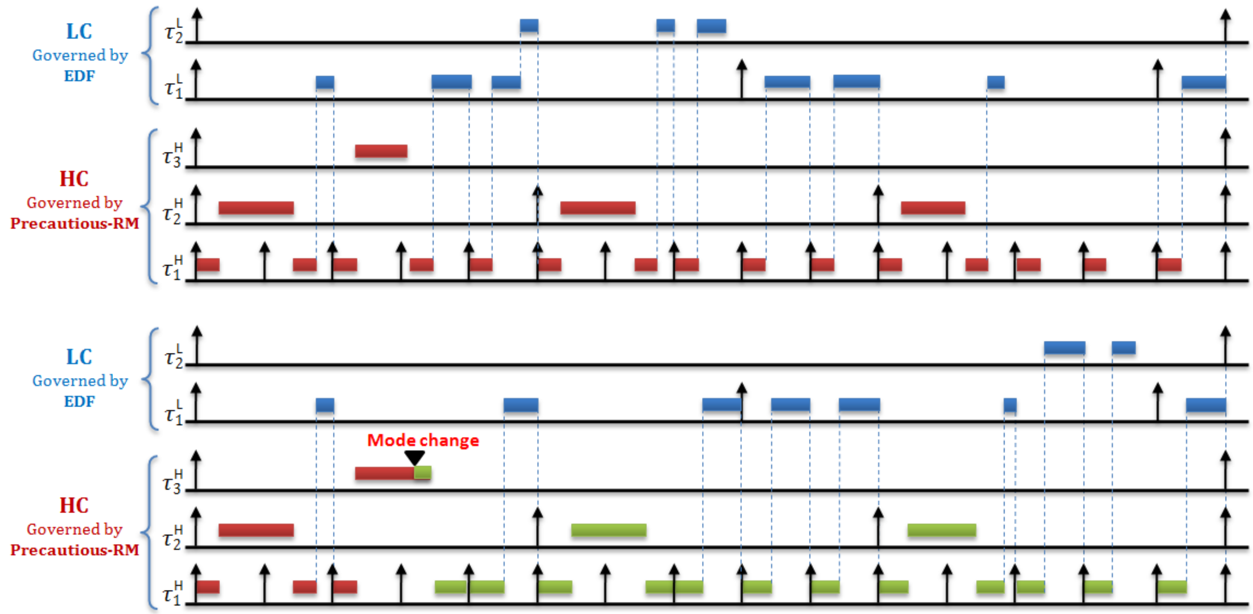


Fig. 1. An example for a hierarchical scheduling to support non-preemptive execution of HC tasks and preemptive execution of LC tasks

### *B. Schedulability Challenges for Non-preemptive HC Tasks Scheduled Together with Preemptive LC Tasks*

If the system is non-preemptive, schedulability conditions become conservative because of the fact that in a feasible non-preemptive schedule, the maximum execution time of each task cannot be larger than two times the slack of the task with the smallest period [2], [4]. For the low criticality tasks, the precision in the WCET is not as important as HC tasks. Moreover, in some cases, they have larger execution time than HC tasks and cannot fit into the mentioned limited slack. For example, to keep an unmanned aerial vehicle in the air, usually control tasks with small execution times and periods are used, while, for example, the navigation task has larger execution time and period [5]. In other words, there is usually a difference between the time granularity of the required responses from LC and HC tasks.

To support mixed execution of preemptive LC tasks with non-preemptive HC tasks, we can use a hierarchical scheduling approach with two levels. The first level is governed by Precautious-RM and is responsible for HC tasks while the second level handles LC tasks using EDF within the idle intervals produced by Precautious-RM. This hierarchy separates interference between LC and HC tasks. Fig. 1 shows an example of this hierarchy. In this example there are 3 high criticality tasks which are scheduled by Precautious-RM. Assume that the challenges in Sect. II-A are solved and we can come up with a feasible approach to schedule non-preemptive HC tasks. At the times when Precautious-RM schedules an idle interval, we can use EDF to schedule low critical tasks. In this example, period of the low criticality tasks is an integer multiple of period of $\tau_1^H$ which is the first HC task with the smallest period.

The open problem in this setup is how to guarantee schedulability of some of the low priority tasks at design-time. As shown in Fig. 1, LC tasks can be executed within the structured schedule which is produced by Precautious-RM. It is called *structured*, because in the worst-case, each low priority HC task is scheduled between two consecutive instances of $\tau_1^H$.

The remaining challenge is that if HC tasks switch to HC mode, like the example in the bottom of Fig. 1, the slots which are available for LC tasks will change. The consequence of such changes must be considered in the schedulability condition of low critical tasks.

## REFERENCES

[1] K. Jeffay, D. F. Stanat, and C. U. Martel, "On non-preemptive scheduling of period and sporadic tasks," in *IEEE Real-Time Systems Symposium (RTSS)*, 1991, pp. 129–139.

[2] Y. Cai and M. C. Kong, "Nonpreemptive scheduling of periodic tasks in uni- and multiprocessor systems," *Algorithmica*, vol. 15, no. 6, pp. 572–599, 1996.

[3] S. Vestal, "Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance," in *IEEE Real-Time Systems Symposium (RTSS)*, 2007, pp. 239–243.

[4] M. Nasri and M. Kargahi, "Precautious-RM: a predictable non-preemptive scheduling algorithm for harmonic tasks," *Real-Time Systems*, vol. 50, no. 4, pp. 548–584, 2014.

[5] J.-H. Kim, S. Sukkarieh, and S. Wishart, "Real-time navigation, guidance, and control of a uav using low-cost sensors," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, S. Yuta, H. Asama, E. Prassler, T. Tsubouchi, and S. Thrun, Eds. Springer Berlin Heidelberg, 2006, vol. 24, pp. 299–309.