Performance comparison of partitioned and global approaches for weakly hard real-time systems

Habibah Ismail[1], Dayang N A Jawawi[2], Ismail Ahmedy[3] and Mohd Adham Isa[4]

[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA
[2,4]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia
[3]Department of Computer System and Technology, Faculty of Computer Science and Technology, University of Malaya

E-mail: habibahisma@gmail.com, dayang@utm.my, ismailahmedy@um.edu.my, mohdadham@utm.my

Abstract. One way to minimizing resource requirements is through the careful management and allocation, in example, scheduling. Research on weakly hard real-time scheduling on multiprocessor has been extremely limited; most prior research on real-time scheduling on weakly hard real-time has been confined to uniprocessors. The need for multiprocessor is due to issues that impose limits on the performance of a single processor. As real-time application systems increasingly come to be implemented upon multiprocessor environments, thus, this study applies multiprocessor scheduling approach for verification of weakly hard real-time tasks and to guaranteeing the timing requirements of the tasks. In fact, within the multiprocessor, the task allocation and migration problems seem even harder than in uniprocessor case; thus, in order to cater that problem, the sufficient and efficient scheduling algorithm supported by accurate schedulability analysis technique is present to provide weakly hard real-time guarantees. The proposed approach involves the two existing multiprocessor real-time scheduling algorithms combining with the hyperperiod analysis and deadline models; weakly hard constraints and μ-pattern. In this paper, the Matlab simulation tool is used in order to validate the result of analysis. From the experimental and performance evaluation results, it proved that the proposed approach is satisfied the tasks deadlines with less number of misses.

1. Introduction

Real-time systems are characterized by stringent timing constraints (generally expressed in the form of a deadline). For a real-time system, there is a need for other mechanisms (real-time scheduling) to handle and satisfy time constraints. Selecting appropriate approaches for scheduling activities is one of the most important considerations in the design of a real-time system. In order to do the schedulability analysis and design for real-time systems, a task execution time and a task period must be obtain because both information are needed in related scheduling parameters especially for periodic tasks.

Schedulability analysis is used to verify whether every task in a given task set can meet a deadline when scheduled according to the adopted scheduling algorithm. Much research has been conducted on schedulability analysis although it has tended to be focused on uniprocessor system. In multiprocessor, schedulability analysis is more challenges because it consists of more than one processor in the

[1] habibahisma@gmail.com

IOP Conf. Series: Materials Science and Engineering **884** (2020) 012050 doi:10.1088/1757-899X/884/1/012050

system. Thus, the present research applies multiprocessor scheduling algorithm approaches for verification of weakly hard real-time tasks and to guaranteeing the timing requirements of the tasks. There are two types of multiprocessor scheduling, called partitioned and global [1]. Timing requirements or constraints are being defined in terms of deadlines for the activities. Computations occurring in a real-time system that have timing constraints are called real-time tasks. The classification of real-time systems or tasks divided into two categories, based on the “seriousness” of deadline misses – hard real-time tasks and soft real-time tasks [2]. Nowadays, most real-time application consists of a mix of hard and soft real-time tasks, and it called weakly hard real-time system. In a weakly hard real-time system, the number of deadlines that may be missed can be specified. This makes a weakly hard real-time system stronger than a soft real-time system. It is also a concept based on hard real-time systems, weakened to include a precisely bounded and predictable distribution of lost deadlines.

The goal of this paper is to propose approaches for scheduling when not all such restrictions can be satisfied and to determine the loss to timeliness that relaxing the restriction entails. Generally, automatic control system (robotics) have been analysed only in the context of hard real-time systems, and as such can require that the workload be restricted. Thus, this study attempted to bridge this gap by analysing some known algorithms in the context of guaranteeing bounded deadline misses, designing a combination between algorithm and technique specifically for weakly hard real-time systems and analysing it.

This paper is organized as follows. In section 2, the proposed partitioned scheduling approach is given. In section 3, the proposed global scheduling approach is presented. Finally, our work of this paper is summarized and discussed as conclusions in section 4 including the remaining ongoing work.

2. The proposed partitioned scheduling approach

The combination of the partitioned scheduling algorithm with weakly hard models’ flowchart is illustrated in Figure 1.

A schedulability tests or analysis is a condition in order to know whether a task set meets its deadlines or not. Usually, schedulability tests for partitioned multiprocessor task systems (combining well-known uniprocessor scheduling techniques with solutions for the bin-packing problem). The processor is assigned with each task in order to assign a local (for a processor) with static priority. *MobileRobot* has the highest priority because the tasks are numbered by decreasing priority order in each processor. Consider seven tasks of AMR case study in Table 1, where these tasks are derived from [3] needed to be scheduled using 2 processors with R-BOUND-MP-NFRNS. Furthermore, the algorithm is responsible to sort the task in ascending order of periods. For this analysis, processor 1 is the current one with the tasks has been assigned in order. Table 1 shows on which processor of each task is assigned.

Table 1. Task parameters of the task set.

Task	Period (T_i)	Deadline (D_i)	C_i	U_i	P_i	h_i	a_i
MobileRobot	50	50	1	0.02	1	50	1
motorctrl_left	50	50	20	0.4	1	50	1
motorctrl_right	50	50	20	0.4	1	50	1
Subsumption	80	80	1	0.0125	1	400	5
Avoid	90	90	17	0.19	2	450	5
Cruise	90	90	1	0.01	2	450	5
manrobotintf	95	95	16	0.17	1	950	10

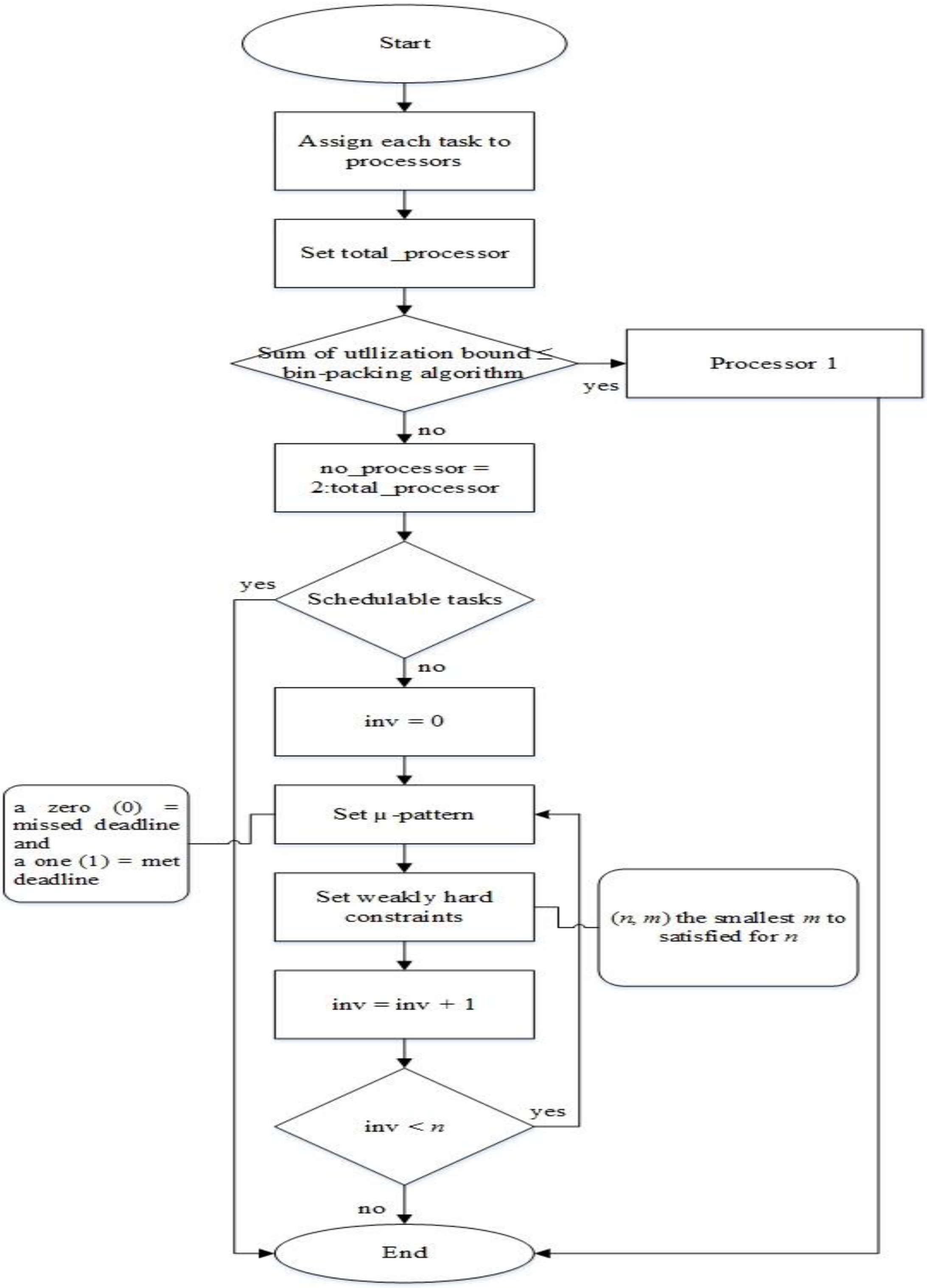

Figure 1. Flowchart of the proposed partitioned scheduling approach.

Even though task *manrobotintf* missed its deadline initially, however by using hyperperiod analysis, the number of deadlines missed for that task can be specified. The Figure 2 shows the utilization bound at each invocation within the hyperperiod at priority level 7. The task is invoked α_7 = 10 times within the hyperperiod at level 7.

The following Table 2 shows the exact distribution of the missed and met deadlines for invocations 1 to 10 of task *manrobotintf*.

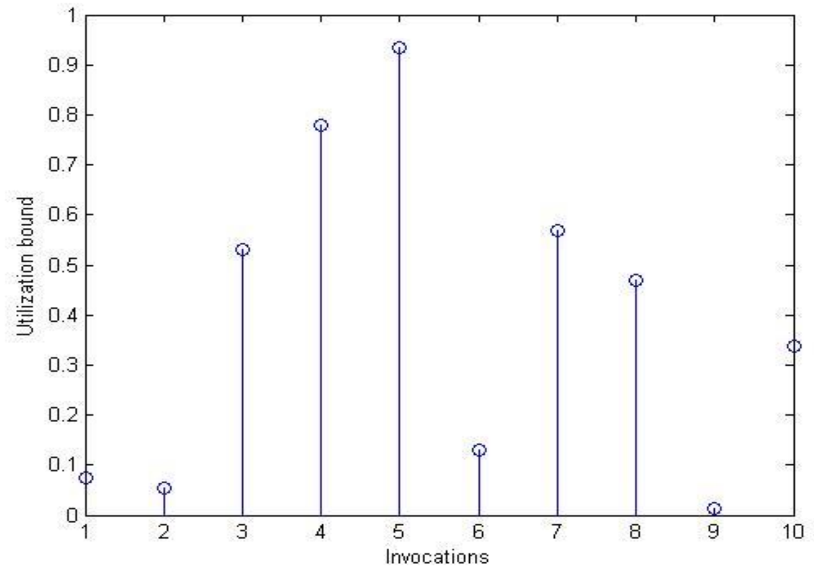

Figure 2. Invocation of *manrobotintf* in the hyperperiod.

Table 2. The exact distribution of the AMR task.

Task *manrobotintf*	
Invocations	**μ-pattern**
1 – 10:	11000 10010

As can been seen task *manrobotintf* missed its deadline at its third, fourth, fifth, seventh, eighth and tenth invocations. Using weakly hard constraint, the number of deadlines met and missed for the task can be precisely specified. So, task *manrobotintf*'s μ-pattern would be 11000 10010. A 0 represents a deadline missed and a 1 a deadline met. It can miss it at most 6 times during its hyperperiod, (6 times every 10 invocations). Thus, the weakly hard constraint for task *manrobotintf* is defined as $\begin{pmatrix} 2 \\ 5 \end{pmatrix}$ constraint.

3. The proposed global scheduling approach

The combination of the global scheduling algorithm with weakly hard models' flowchart is illustrated in Figure 3.

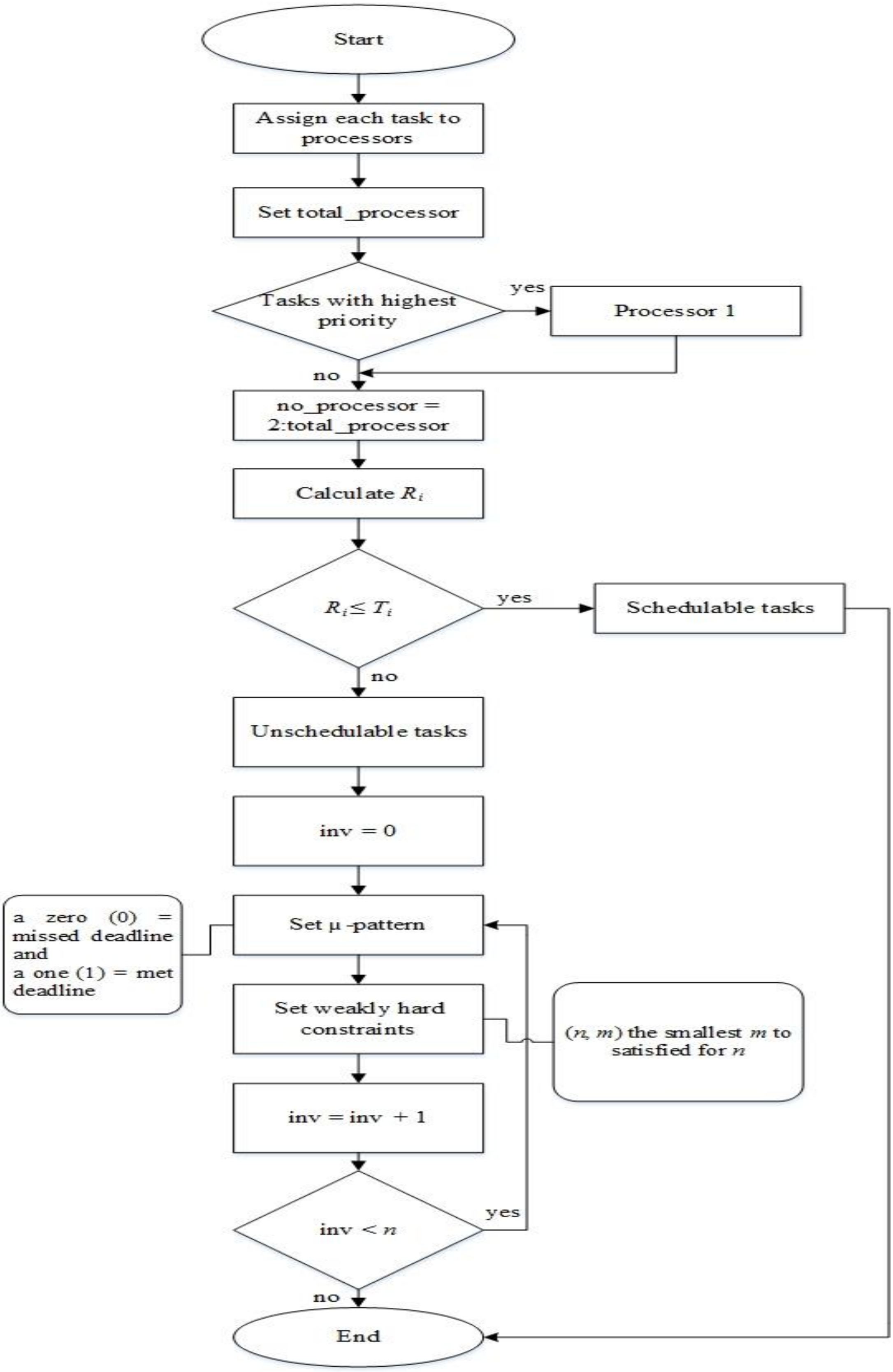

Figure 3. Flowchart of the proposed global scheduling approach.

An upper bound of the response time of a task τ_i in a multiprocessor system scheduled with fixed priority can be derived by the fixed-point iteration on the value R_i, starting with $R_i = C_i$. We determine whether a set of tasks allocated to a processor is schedulable given a particular fixed priority assignment by checking whether the worst-case response time for each task is less than or equal to its relative deadline. *MobileRobot* has the highest priority among the seven tasks, thus it selected as the first task to assign to Processor 1. From the definition of interference, it is clear that no task can contribute to the interference on a higher priority task. Consider the seven tasks in Table 3, where these tasks needed to be scheduled using 2 processors with response time analysis technique.

Table 3. Task parameters of the task set.

Task	Period (T_i)	Deadline (D_i)	C_i	R_i	P_i	h_i	a_i
MobileRobot	50	50	1	1	1	50	1
motorctrl_left	50	50	20	11	1	50	1
motorctrl_right	50	50	20	42	1	50	1
Subsumption	80	80	1	51	1	50	5
Avoid	90	90	17	52	2	450	5
Cruise	90	90	1	31	2	450	5
manrobotintf	95	95	16	12	2	8550	90

Even though task *Subsumption* missed its deadline in the worst case, but by using hyperperiod analysis, the number of deadlines missed for that task can be specified. The Figure 4 shows the worst-case response time at each invocation within the hyperperiod at priority level 4. Each vertical line represents the worst-case response time of the task. The deadline of the task is D_4 = 50. The hyperperiod h_4 = 50. This means that the task is invoked α_4 = 1 times within its hyperperiod at level 4. Response time analysis under the rate monotonic algorithm was used as a schedulability tests. Although there have one task missed its deadline in the worst case, however, it is weakly hard schedulable. The following Table 4 shows the exact distribution of the missed and met deadlines for invocations 1 to 52 of task *Subsumption*. By using weakly hard constraints, the number of deadlines met and missed for that task can be precisely specified. It can miss 20 times every 60 invocations during its hyperperiod.

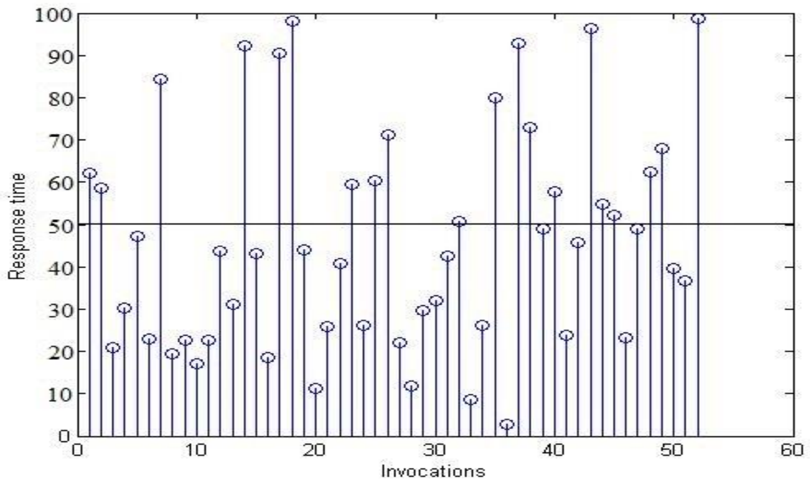

Figure 4. Invocation of *Subsumption* in the hyperperiod.

Table 4. The exact distribution of the AMR task.

Task *Subsumption*	
Invocations	**µ-pattern**
1 – 52:	0011 1101 1111 1011 0011 1101 0011 1110 1101 0010 1100 0110 0110

Checking *Subsumption*'s weakly hard constraint shows that it is satisfied, despite the miss and can conclude that the system is weakly hard schedulable. Thus, the weakly hard constraint for task *Subsumption* is defined as $\binom{2}{4}$ constraint.

4. Conclusion

As conclusion, there are two performances that can be highlighted in this paper. Firstly, from the result of experimental performance, task *Subsumption* missed its deadline using multiprocessor response time analysis while task *manrobotintf* missed its deadline when it is scheduled by using R-BOUND-MP-NFRNS. Secondly, based on the result as depicted in Figure 4, AMR case study on global scheduling has better performance of schedulability tests compared with partitioned scheduling as illustrated in Figure 2. The comparison is in terms of achieves a higher success/meet ratio when there is some deadlines missed for the task. This is due to the fact that the number of invocations for the task of AMR when using global scheduling approach is lesser than partitioned scheduling approach. Another reason is when using a schedulability test based on response time analysis, it provides a slightly higher success ratio than utilization bound test. The applicability of the proposed approach for AMR task sets, can be concluded that eventhough some of the tasks would still strongly hard, however some have weakly hard constraints, wherein the satisfaction of delay tasks can be guaranteed by specify on the number of deadlines missed during a period of time precisely. It proved that the automatic control system (robotics) are not that hard because some degree of missed deadlines can be tolerated by the algorithm.

In this paper, the approaches addressed partitioned used the utilization-based test and global scheduling used the response time analysis. Forward, the idea is to design a global static priority scheduling with a utilization bound. Meanwhile, a common solution for partitioning scheduling approach is to use a bin-packing algorithm, instead, use a schedulability test based on response time analysis.

Acknowledgments

The authors would like to express profound gratitude to the Universiti Teknologi Malaysia (UTM) and University of Malaya for their financial support under research grant RF006D-2018 and not forgotten, the Software Engineering Research Group (SERG) for their help.

IOP Conf. Series: Materials Science and Engineering **884** (2020) 012050 doi:10.1088/1757-899X/884/1/012050

References

[1] Carpenter J, Funk S, Holman P and Anderson J 2004 A categorization of real-time multiprocessor scheduling problems and algorithms *Handbook on Scheduling Algorithms, Methods and Models* (Chapman and Hall: CRC) pp 30.1 – 30.19

[2] Klein M H 1993 *A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time System* (Boston: Kluwer Academic Publisher)

[3] Jawawi D N A, Deris S and Mamat R 2006 Enhancement of PECOS embedded real-time component model for autonomous mobile robot application *Proc. Int. Conf. on Computer Systems and Applications (Dubai)* pp 882 – 889

[4] Afshar S, Behnam M, Bril R J and Nolte T 2015 Resource sharing in a hybrid partitioned/global scheduling framework for multiprocessors *IEEE 20th Conf. on Emerging Technologies Factory Automation*

[5] Lee J 2016 New response time analysis for global EDF on a multiprocessor platform *Journal of Systems Architecture* **65** pp 59 – 63

[6] Rincón C A C, Zou X and Cheng A M K 2017 Real-time multiprocessor scheduling algorithm based on information theory principles *IEEE Embed. Syst. Letters* **9(4)** pp 93–96

[7] Sun Y and Natale M D 2017 Weakly hard schedulability analysis for fixed priority scheduling of periodic real-time tasks *ACM Trans. on Embed. Comp. Syst.* **16(5s)** pp 171:1–171:19

[8] Zhou Q, Li G, Li J and Deng C 2018 Execution-efficient response time analysis on global multiprocessor platforms *Trans. on Parallel and Distributed Syst.* **29(12)** pp 2785–2797

[9] Mayank J and Mondal A 2019 An integer linear programming framework for energy optimization of non-preemptive real-time tasks on multiprocessors *Journal of Low Power Electronics* **15(2)** pp 162–167