

Introduction to Embedded Systems – WS 2022/23

Sample Solution to Exercise 4: Scheduling Periodic and Mixed Task Sets

Task 1: Earliest Deadline First (EDF) and Total Bandwidth Server (TBS)

Consider the following set of periodic tasks:

	τ_1	τ_2	τ_3
C_i	1	1	2
T_i	3	5	13

A Total Bandwidth Server (TBS) executes along with the periodic tasks above.

1. What can be the maximum value of U_s such that the whole set (i.e., periodic tasks and the TBS) is schedulable with EDF?
2. Now assume $U_s = 0.25$. Construct the EDF schedule (in Figure 1) in the case in which three aperiodic requests $J_4(r_4 = 0, C_4 = 2)$, $J_5(r_5 = 15, C_5 = 1)$ and $J_6(r_6 = 10, C_6 = 1)$ are served by TBS. Assume that the arrival time of the first instance/job of each periodic task is 0.

Solution to Task 1:

1. Maximum utilization of the Total Bandwidth Server:
$$U_{s,max} = 1 - U_p = 1 - (1/3 + 1/5 + 2/13) = 61/195 \approx 0.3128$$
2. First, we need to order the tasks by increasing release time r_i : J_4, J_6, J_5 . Then, we calculate the deadlines with $d_i = \max(r_i, d_{k-1}) + \frac{C_k}{U_s}$, where d_{k-1} denotes the previously calculated deadline ($k - 1$ means the predecessor in the ordering according to the release time):
$$d_4 = \max(r_4, d_0) + 2/0.25 = 0 + 8 = 8$$
$$d_6 = \max(r_6, d_4) + 1/0.25 = 10 + 4 = 14$$
$$d_5 = \max(r_5, d_6) + 1/0.25 = 15 + 4 = 19$$

For the resulting EDF schedule see Figure 1.

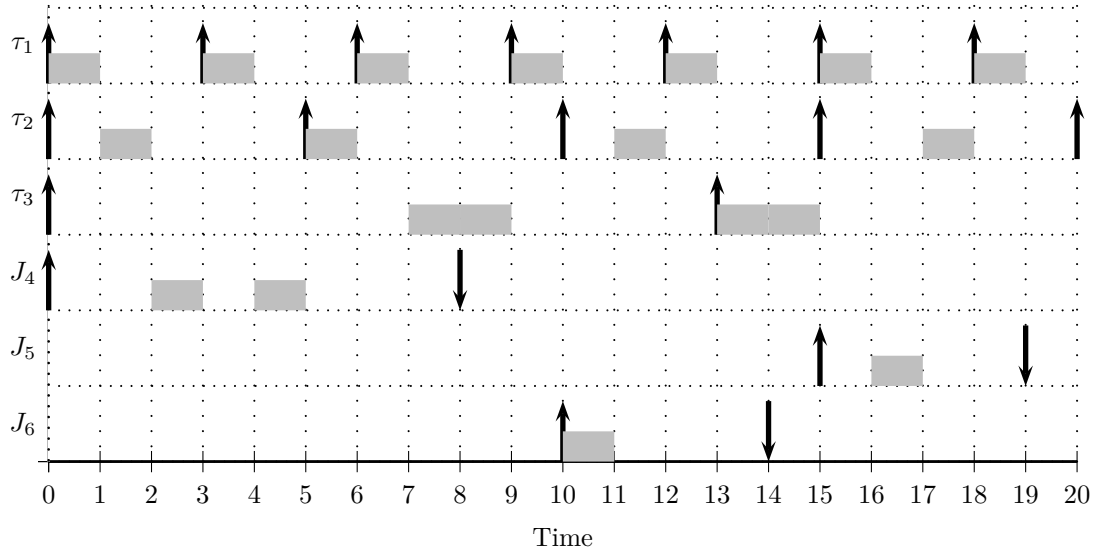


Figure 1: EDF schedule solution for Task 1

Task 2: Schedulability Test for Fixed Priorities – Rate Monotonic (RM)

	τ_1	τ_2	τ_3
C_i	1	3	2
T_i	3	8	9

1. Test if the given task-set is schedulable under RM, using the sufficient test.
2. Test if the given task-set is schedulable under RM, using the necessary test.
3. Assume that the first job of each task arrives at time 0. Construct the schedule for the interval $[0, 20]$ and illustrate it graphically. In case they exist, identify deadline misses.

Solution to Task 2:

The priorities of the tasks are assigned statically, before the actual execution of the task set. Rate Monotonic scheduling scheme assigns higher priority to tasks with smaller periods. It is preemptive (tasks are preempted by the higher priority tasks). It is an optimal scheduling algorithm amongst fixed-priority algorithms; if a task set cannot be scheduled with RM, it cannot be scheduled by any fixed-priority algorithm.

1. The sufficient schedulability test is given by:

$$U = \sum_{i=1}^n \frac{C_i}{D_i} \leq n(2^{1/n} - 1)$$

The term U is said to be the processor utilization factor (the fraction of the processor time spent on executing task set). n is the number of tasks.

In our case: $1/3 + 3/8 + 2/9 = 0.93 \not\leq 3(2^{1/3} - 1) = 0.78$ failed!

The above condition is *not* necessary, hence we don't know whether the task set is schedulable with RM or not. We can do a somewhat more involved sufficient and *necessary* condition test, as follows.

2. We have to guarantee that all the tasks can be scheduled, in any possible instance. In particular, if a task can be scheduled in its critical instances, then the schedulability guarantee condition holds (a

critical instance of a task occurs whenever the task is released simultaneously with all higher priority tasks). In the following, we perform the iterative schedulability algorithm on slide 6-54 (for the details of the algorithm, please have a look at page 99, Buttazzo's book).

The tasks are first ordered by their priorities: τ_1 , τ_2 and τ_3 . (In this case the tasks are already ordered.)

τ_3 :

$$\begin{aligned} R_3^0 &= C_3 = 2 & I_3^0 &= \lceil \frac{2}{3} \rceil 1 + \lceil \frac{2}{8} \rceil 3 = 1 + 3 = 4 & 4 + 2 &\neq 2 \\ R_3^1 &= 4 + 2 = 6 & I_3^1 &= \lceil \frac{6}{3} \rceil 1 + \lceil \frac{6}{8} \rceil 3 = 2 + 3 = 5 & 5 + 2 &\neq 6 \\ R_3^2 &= 5 + 2 = 7 & I_3^2 &= \lceil \frac{7}{3} \rceil 1 + \lceil \frac{7}{8} \rceil 3 = 3 + 3 = 6 & 6 + 2 &\neq 7 \\ R_3^3 &= 6 + 2 = 8 & I_3^3 &= \lceil \frac{8}{3} \rceil 1 + \lceil \frac{8}{8} \rceil 3 = 3 + 3 = 6 & 6 + 2 &= 8 \dots \text{OK} \\ & & & & & (\text{since } R_3 = 8 \leq T_3 = 9) \end{aligned}$$

τ_2 :

$$\begin{aligned} R_2^0 &= C_2 = 3 & I_2^0 &= \lceil \frac{3}{3} \rceil 1 = 1 & 1 + 3 &\neq 3 \\ R_2^1 &= 1 + 3 = 4 & I_2^1 &= \lceil \frac{4}{3} \rceil 1 = 2 & 2 + 3 &\neq 4 \\ R_2^2 &= 2 + 3 = 5 & I_2^2 &= \lceil \frac{5}{3} \rceil 1 = 2 & 2 + 3 &= 5 \dots \text{OK} (\text{since } R_2 = 5 \leq T_2 = 8) \end{aligned}$$

τ_1 :

$$R_1^0 = C_1 = 1 \quad I_1^0 = 0 \quad 0 + 1 = 1 \dots \text{OK} (\text{since } R_1 = 1 \leq T_1 = 3)$$

The necessary and sufficient test succeeds. This means that the task set is schedulable with RM.

3. The schedule is represented graphically in Figure 2. (There are no deadline misses.)

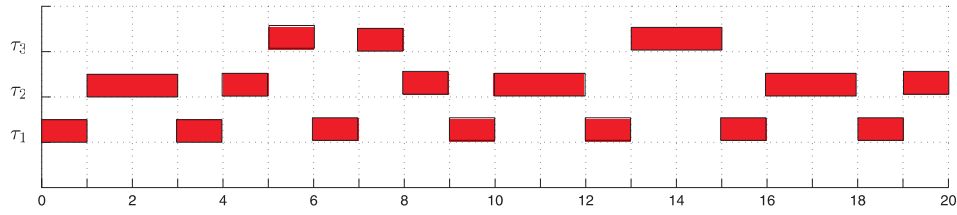


Figure 2: RM schedule of Task 2.

Task 3: Scheduling with Polling Server

	τ_1	τ_2	τ_3
C_i	2	2	2
D_i	6	8	16
T_i	6	8	16

In addition to the above periodic tasks, we have an aperiodic job J_a with computation time $C_a = 1$, and relative deadline D_a . The scheduling policy is RM. The aperiodic job is scheduled through a Polling Server (PS).

1. Let the period and computing time of the polling server be $T_s = 25$ and $C_s = 1$, respectively. Compute the aperiodic guarantee available to J_a , i.e., compute the minimum relative deadline of J_a which is guaranteed not to be missed.
2. Using the sufficient test of RM, test if the polling server of (a) is schedulable along with the periodic task-set?
3. [optional] Determine integer parameters (C_s , T_s) of the polling server such that (1) the relative deadline guaranteed to J_a is minimised, and (2) the RM schedule satisfies the sufficient schedulability test.
4. [optional] For the optimal setting of (c) devise a necessary schedulability test with the relative deadline of the aperiodic task $D_a = 32$.

Solution to Task 3:

Recall the way the polling server works. If at the start of a new period of the polling server, there are no pending aperiodic jobs, then no aperiodic jobs are executed for the whole period. Thus, the worst-case arrival time of an aperiodic job is just after the start of the period of the polling server. Then, the aperiodic guarantee is given by

$$(1 + \lceil \frac{C_a}{C_s} \rceil) T_s \leq D_a$$

1. By using the aperiodic guarantee and substitute the given parameters, we have $D_a = 50$.
The above computation of D_a holds only if the RM schedule meets all deadlines.
2. The sufficient schedulability test for RM (including the Polling Server introduced to handle the firm aperiodic request J_a) is given by:

$$\sum_{i=1}^n \frac{C_i}{T_i} + \frac{C_s}{T_s} \leq (n+1)(2^{1/(n+1)} - 1)$$

Substituting the parameters of the task-set, we arrive at $0.748 \leq 0.76$. This is true, and hence the RM schedule meets all deadlines.

3. Let C_s and T_s be unknown. Then the sufficient RM schedulability test implies that

$$\frac{C_s}{T_s} \leq (n+1)(2^{1/(n+1)} - 1) - \sum_{i=1}^n \frac{C_i}{T_i} = 0.7568 - 0.7083 = 0.0485$$

Since, we use only integer values for C_s and T_s , we can set (C_s, T_s) as (1, 21) or (2, 42) or (3, 63) and so on.

For the optimal configuration of the server, we have $T_s < 50$ (as we managed to get a guarantee of 50 with the previous setting). Upon inspection, (1, 21) is the optimal solution with the minimum relative deadline of 42.

4. The sufficient schedulability test guarantees $D_a = 42$. We use the necessary test of RM to obtain a smaller D_a . Note that the polling server is the lowest priority task.

$$\begin{array}{lll} R_a^0 = C_a = 1 & I_a^0 = \lceil \frac{1}{6} \rceil 2 + \lceil \frac{1}{8} \rceil 2 + \lceil \frac{1}{16} \rceil 2 = 6 & 6 + 1 \neq 1 \\ R_a^1 = 7 & I_a^1 = \lceil \frac{7}{6} \rceil 2 + \lceil \frac{7}{8} \rceil 2 + \lceil \frac{7}{16} \rceil 2 = 8 & 8 + 1 \neq 7 \\ R_a^2 = 9 & I_a^2 = \lceil \frac{9}{6} \rceil 2 + \lceil \frac{9}{8} \rceil 2 + \lceil \frac{9}{16} \rceil 2 = 10 & 10 + 1 \neq 9 \\ R_a^3 = 11 & I_a^3 = \lceil \frac{11}{6} \rceil 2 + \lceil \frac{11}{8} \rceil 2 + \lceil \frac{11}{16} \rceil 2 = 10 & 10 + 1 = 11 \end{array}$$

The above analysis shows from the time of beginning of execution of J_a it needs 11 time units to finish. But, as already mentioned, J_a can arrive just after the start of the polling server period. Thus, we have $D_a = 11 + T_s = 11 + 21 = 32$. This is indeed much smaller than the earlier value of 42.

Task 4: Periodic Scheduling with Fixed Priorities – DM

Given the following set of periodic tasks:

	τ_1	τ_2	τ_3
C_i	1	2	3
D_i	5	4	8
T_i	5	6	10

1. Check the schedulability of the task set using the Deadline Monotonic (DM) policy.
2. Construct the schedule graphically. Let the phase $\Phi_i = 0 \forall i$. In case they exist, identify deadline misses.

Solution to Task 4:

DM (Deadline Monotonic) scheduling scheme assigns priorities based on the relative deadlines of the periodic tasks. Higher priority corresponds to a task having an earlier deadline.

1. One first schedulability test is (sufficient, not necessary): $1/5 + 2/4 + 3/8 = 1.075 \not\leq 3(2^{1/3} - 1) = 0.78$ failed!

The other test to do follows the lines of the solution of the previous exercise: the tasks are ordered with respect to their priorities τ_2, τ_1, τ_3 .

τ_3 :

$$\begin{aligned} R_3^0 &= C_3 = 3 & I_3^0 &= \lceil \frac{3}{5} \rceil 1 + \lceil \frac{3}{6} \rceil 2 = 1 + 2 = 3 & 3 + 3 &\neq 3 \\ R_3^1 &= 3 + 3 = 6 & I_3^1 &= \lceil \frac{6}{5} \rceil 1 + \lceil \frac{6}{6} \rceil 2 = 2 + 2 = 4 & 4 + 3 &\neq 6 \\ R_3^2 &= 4 + 3 = 7 & I_3^2 &= \lceil \frac{7}{5} \rceil 1 + \lceil \frac{7}{6} \rceil 2 = 2 + 4 = 6 & 6 + 3 &\neq 7 \\ R_3^3 &= 6 + 3 = 9 & I_3^3 &= \lceil \frac{9}{5} \rceil 1 + \lceil \frac{9}{6} \rceil 2 = 2 + 4 = 6 & 6 + 3 = 9 \dots 9 &\not\leq D_3 = 8 \text{ failed!} \end{aligned}$$

The schedule is not feasible.

2. The graphical representation of the failed scheduling is presented in Figure 3:

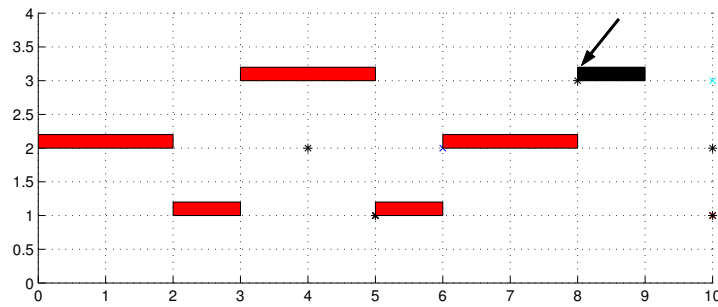


Figure 3: DM schedule (not feasible).

Task 5: Mixed Tasks – Polling Server

Two periodic tasks are given, with execution times and periods given in the following table (deadlines equal periods). The phase of the periodic tasks is assumed to be $\Phi_i = 0 \forall i$. The given set of tasks should be scheduled with the Rate Monotonic scheduling scheme.

	τ_1	τ_2
C_i	1	2
T_i	5	8

Construct a schedule graphically for following aperiodic requests (a Polling Server with integer parameters should be introduced). The CPU utilization has to be maximized.

	J_1	J_2	J_3
a_i	2	7	9
C_i	3	2	1

Solution to Task 5:

For the task set to be schedulable with RM (and the introduction of the Polling Server), one should have the following condition satisfied:

$$\frac{C_s}{T_s} + \left(\frac{1}{5} + \frac{2}{8}\right) \leq 3(2^{1/3} - 1)$$

$$\frac{C_s}{T_s} \leq 0.3298$$

With parameters chosen, for example: $C_s = 1$ and $T_s = 4$, the schedule, depicted in Figure 4 is feasible.

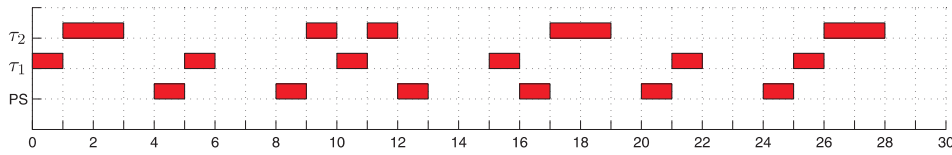


Figure 4: RM schedule - Polling Server with $C_s = 1$, $T_s = 4$

We could have also chosen different (integer-valued) server parameters, say $C_s = 2$ and $T_s = 7$. This solution of the RM schedule is depicted in Figure 5.

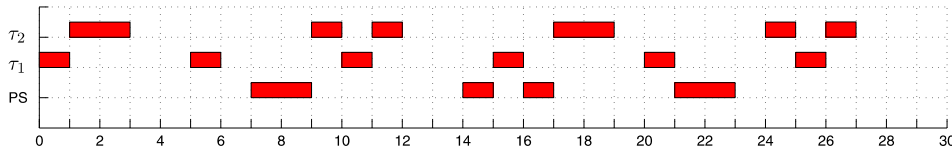


Figure 5: RM schedule - Polling Server with $C_s = 2$, $T_s = 7$

The actual CPU utilization (by the time the aperiodic requests have been served) is greater in the second case. $U_2 = 17/23 = 0.739 > 0.68 = 17/25 = U_1$.

Actually, one could assign the server parameters in such a way that the task set is still schedulable (we can show that this is the case by using the - somewhat more involved - test presented in the solution to the Task 2). For this task set, the parameters of the server may be even $C_s = 2$ and $T_s = 4$. (This is, wrt other periodic tasks in the system, optimal for serving the aperiodic requests, as the server will become the highest priority task in the system).

Task 6: Mixed Tasks – Total Bandwidth Server

We have to design a system that schedules periodic tasks with EDF and employs a total bandwidth server to serve aperiodic requests. We know of one sporadic aperiodic request with computation time $C_a = 2$ and a relative deadline $D_a = 7$. What is the maximum processor utilization available for periodic tasks if we want to guarantee that this aperiodic task completes within its deadline?

Solution to Task 6:

For a schedule to be feasible, the sufficient and necessary condition to hold is $U_p + U_s \leq 1$. Therefore we have

$$U_p \leq 1 - U_s = 1 - \frac{C_a}{D_a} = 1 - \frac{2}{7} = \frac{5}{7}$$

The maximum processor utilization available for periodic tasks is $5/7 = 0.714$.

Task 7: Periodic Scheduling

A processor is supposed to execute the following set of tasks described by their execution times C , relative deadlines D and periods T .

	τ_1	τ_2	τ_3
C_i	2	2	4
D_i	5	4	8
T_i	6	8	12

1. Execute the sufficient schedulability test under DM and calculate the result. What statement regarding schedulability can be made based on your result?
2. Execute the sufficient and necessary schedulability test under DM and calculate the result. What statement regarding schedulability can be made based on your result?
3. If there is a feasible schedule for the given task set, construct it graphically. Let the phase $\Phi_i = 0 \forall i$.

Solution to Task 7:

1. A sufficient schedulability test for a given task set (and with respect to the fixed-priorities scheduling schemes, DM in this case, since the deadlines are smaller than the periods) is given by:

$$\sum_{i=1}^n \frac{C_i}{D_i} \leq n(2^{1/n} - 1)$$

In our case, we have: $2/5 + 2/4 + 4/8 = 1.4 \not\leq 0.78 = 3(2^{1/3} - 1)$ failed!.

The above test was sufficient but not necessary. Hence the tasks might be schedulable by DM.

2. The sufficient and necessary test for DM can be done by determining the value of the longest response time $R_i = C_i + I_i$ of a periodic task τ_i at its critical instances.

(DM) The tasks are ordered with respect to their priorities (earlier deadline - higher priority): τ_2 , τ_1 and τ_3 .

Let's first check for the lowest priority task τ_3 :

$$R_3^0 = C_3 = 4 \quad I_3^0 = \lceil \frac{4}{6} \rceil 2 + \lceil \frac{4}{8} \rceil 2 = 2 + 2 = 4 \quad 4 + 4 \neq 4$$

$$R_3^1 = 4 + 4 = 8 \quad I_3^1 = \lceil \frac{8}{6} \rceil 2 + \lceil \frac{8}{8} \rceil 2 = 4 + 2 = 6 \quad 6 + 4 \neq 8 \text{ failed!}$$

Since $R_3^2 = 4 + 6 = 10 \not\leq D_3 = 8$, the tasks are *not* schedulable with DM!

3. Despite the fact that the schedulability tests for DM have failed, there is a feasible schedule under EDF. It is shown in Figure 6.

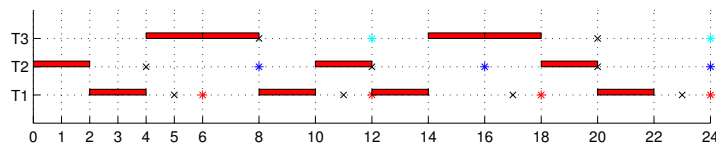


Figure 6: EDF schedule

A necessary and sufficient test for EDF is to use the *demand bound* analysis, which is not covered in the course. Interested reader may look at the following reference

S. Baruah, A. Mok, L. Rosier. *Preemptive Scheduling Hard-Real-Time Sporadic Tasks on One Processor*. Proceedings of the Real-Time Systems Symposium, 182-190, 1990.