VT16 – EL2450 – Assignment II

Alexandros Filotheou
871108-5590
alefil@kth.se

Roberto Sanchez-Rey
840616-9139
rosr@kth.se

1 Question 1

Rate Monotonic is an scheduling method that assigns fixed priorities to tasks, proportional to its activation frequency. That means that for any given tasks J_a, J_b with periods $T_a < T_b$, J_a is assigned a higher priority than J_b.

2 Question 2

A set of periodic tasks $\{J_i\}$ is schedulable with Rate Monotonic scheduling if

$$U = \sum_i \frac{C_i}{T_i} \leq n(2^{1/n} - 1)$$

In the case where $T_1 = 20, T_2 = 29, T_3 = 35$ ms and $C_i = 6$ ms, $i = \{1, 2, 3\}$, $U = 0.678$ and $n(2^{1/n} - 1) = 0.78$. Hence tasks J_1, J_2, J_3 are schedulable with RM.

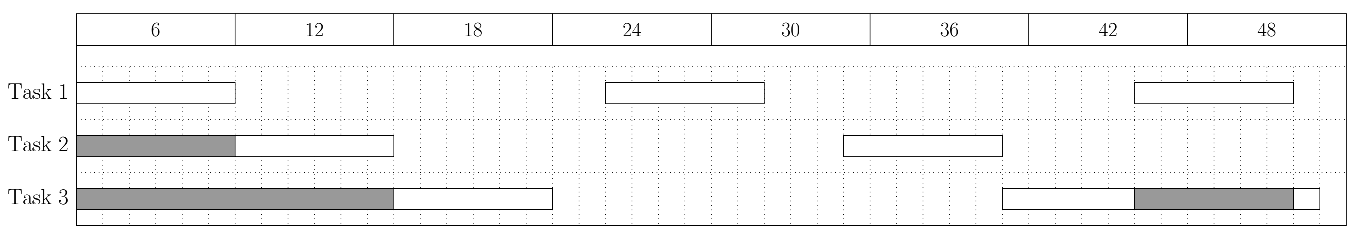

Figure 1: A portion of the RM schedule σ for tasks J_1, J_2, J_3. Shaded areas denote the waiting time.

3 Question 3

All penduli are stable. We observe that the higher the natural frequency of a pentulum, the quicker the response is in its rise time, although with magnified overshoot. This makes sense since the higher the natural frequency of a pendulum, the lower its length and the more difficult it is to stabilize, hence the control must be swift. Figure 2 shows the angular displacement of each pendulum.

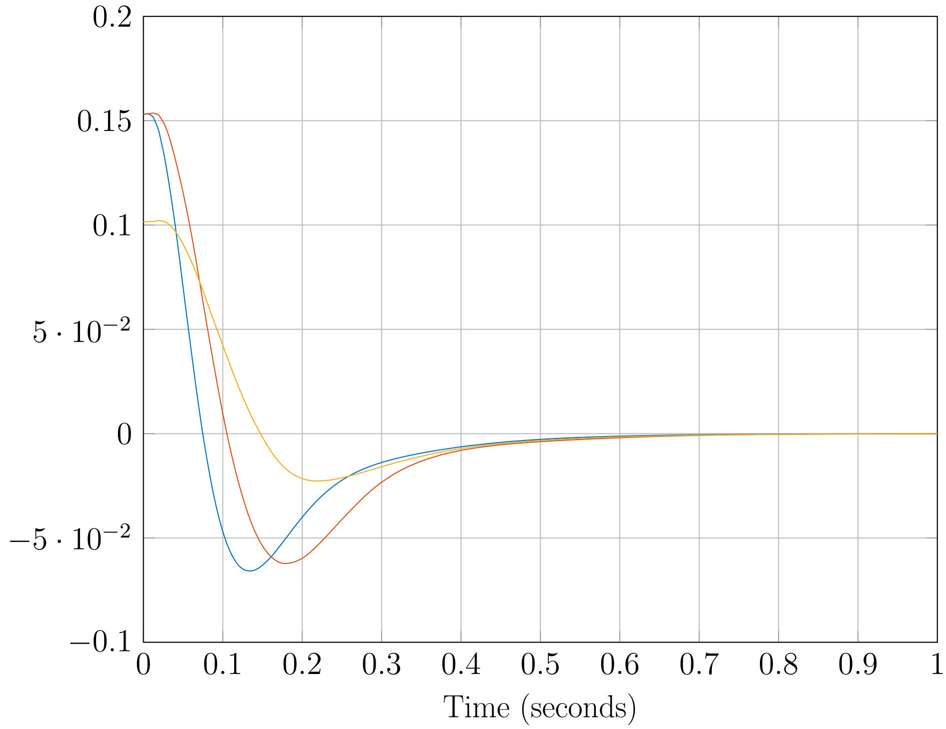

Figure 2: The angular displacement of each pendulum as a function of time. `Blue`: P_1, `Red`: P_2, `Orange`: P_3

4 Question 4

Figure 3 shows the schedule calculated for each pendulum over a timespan of $lcm(20, 29, 35) =$ 4060 ms: exactly one period of the schedule. As per the response to question 2, figure 4 illustrates that the schedule is indeed feasible by plotting the overall usage of the CPU over the aforementioned timespan.

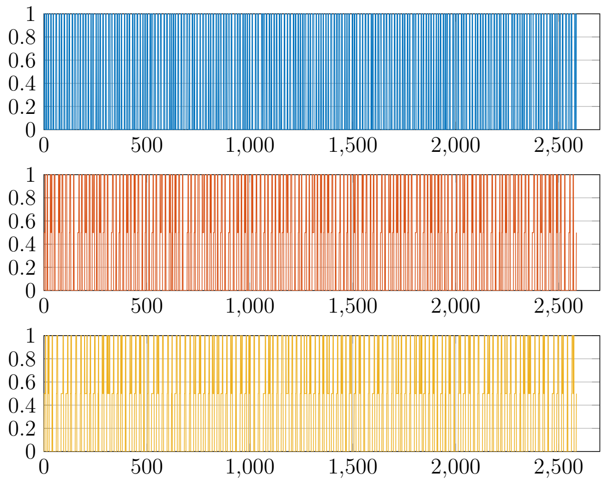

Figure 3: The calculated schedule for the three penduli. `Blue`: P_1, `Red`: P_2, `Orange`: P_3. $C_i = 6$ ms.

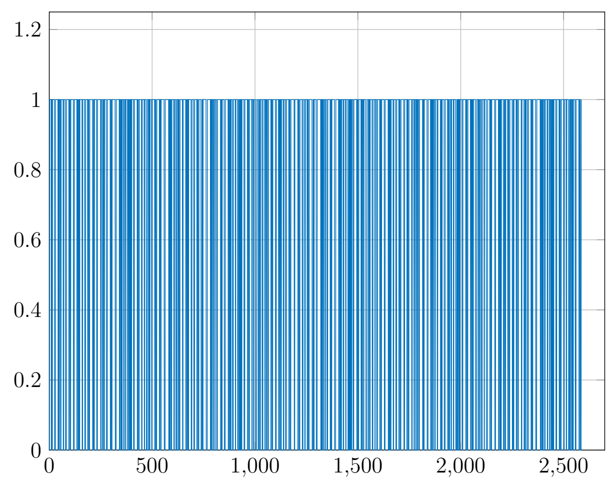

Figure 4: The overall processing usage. Notice that it is at most at 100%. $C_i = 6$ ms.

5 Question 5

In the case where $T_1 = 20, T_2 = 29, T_3 = 35$ ms and $C_i = 10$ ms, $i = \{1, 2, 3\}$, $U = 1.131 > 1$. Hence tasks J_1, J_2, J_3 are not schedulable in any scheduling scheme.

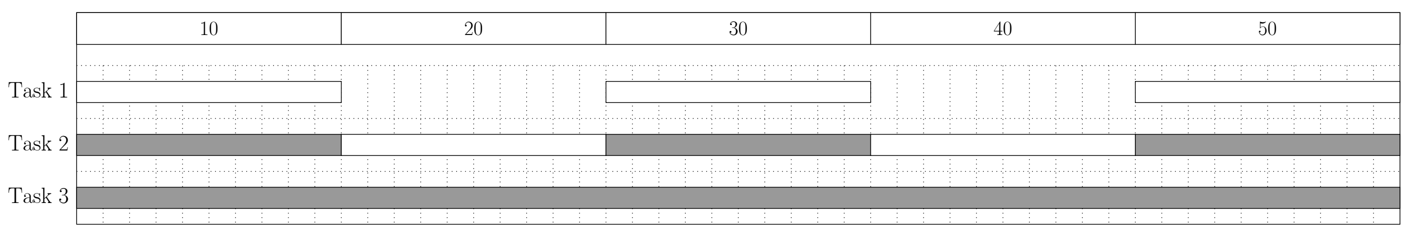

Figure 5: A portion of the RM schedule σ for tasks J_1, J_2, J_3 for $C_i = 10$ ms. Shaded areas denote the waiting time. Notice that J_3 misses its deadlines consecutively, indicative of the inability of schedulability.

All penduli are still stable. However, due to the increased execution time, the control input is not as swift as before, hence the increased magnitude of the overshoot.

Figure 6 shows the angular displacement of each pendulum.

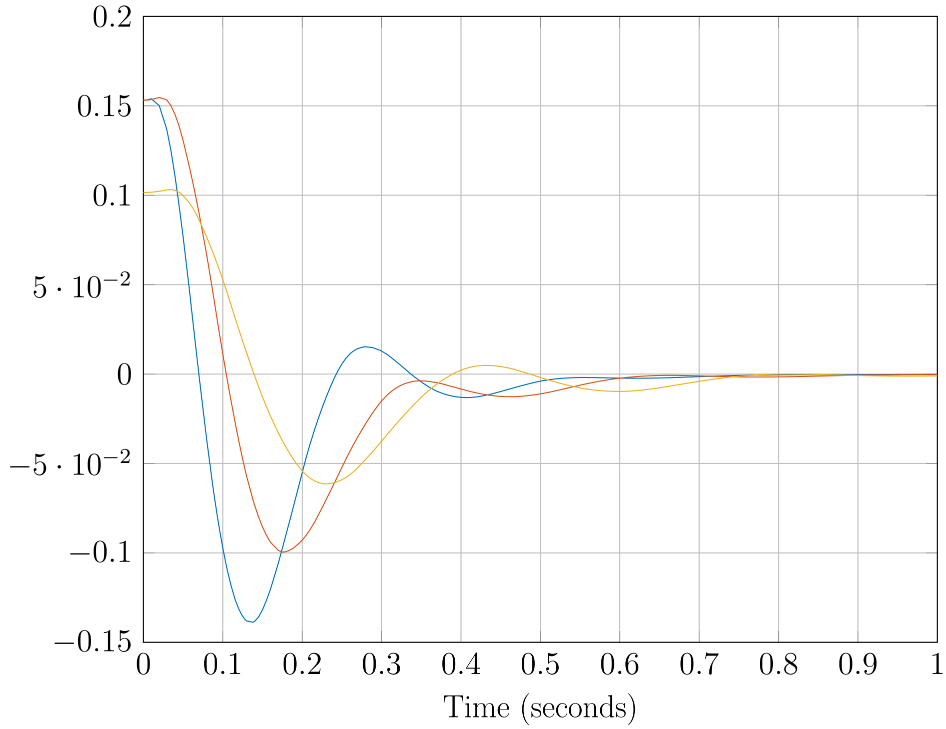

Figure 6: The angular displacement of each pendulum as function of time. `Blue`: P_1, `Red`: P_2, `Orange`: P_3. $C_i = 10$ ms.

Figure 7 shows the schedule calculated for each pendulum with all jobs having execution time $C_i = 10$ ms. Figure 8 illustrates that the schedule is not feasible by ploting the overall usage of the CPU over the the length of a schedule period, which is at all times 100%.

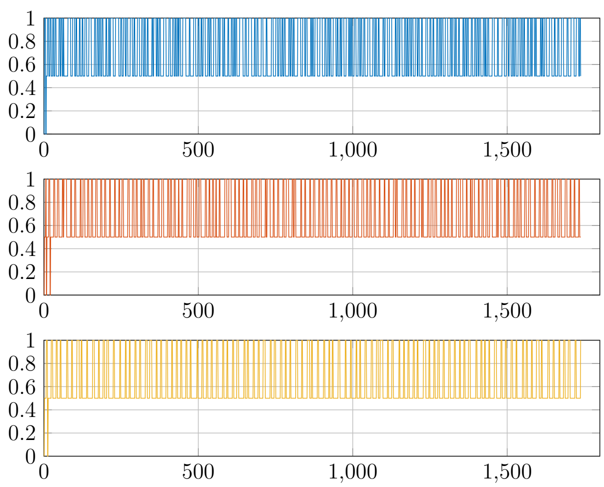

Figure 7: The calculated schedule for the three penduli. `Blue`: P_1, `Red`: P_2, `Orange`: P_3. $C_i = 10$ ms.

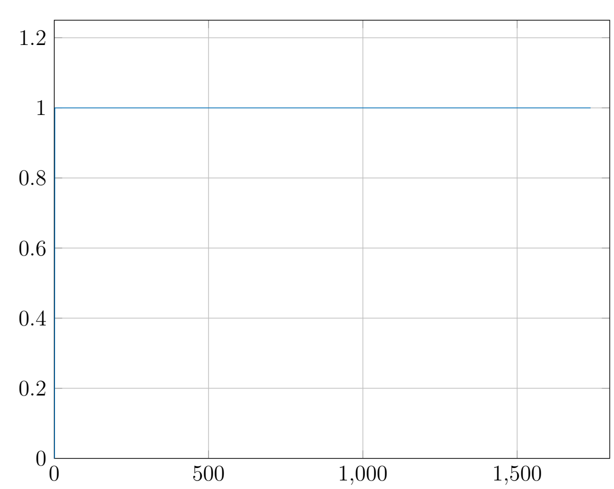

Figure 8: The overall processing usage. Notice that it always at 100%. $C_i = 10$ ms.