VT16 – EL2450 – Assignment II

Alexandros Filotheou
871108-5590
alefil@kth.se

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

1 Part One: Scheduling

Throughout this part, the pendulum with length $l_1 = 0.1$ m will be referred to as P_1, $l_2 = 0.2$ m as P_2 and $l_3 = 0.3$ m as P_3.

1.1 Rate Monotonic scheduling

1.1.1 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.

1.1.2 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. A portion of the calculated schedule can be found in the appendix in figure 33.

1.1.3 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 1 shows the angular displacement of each pendulum.

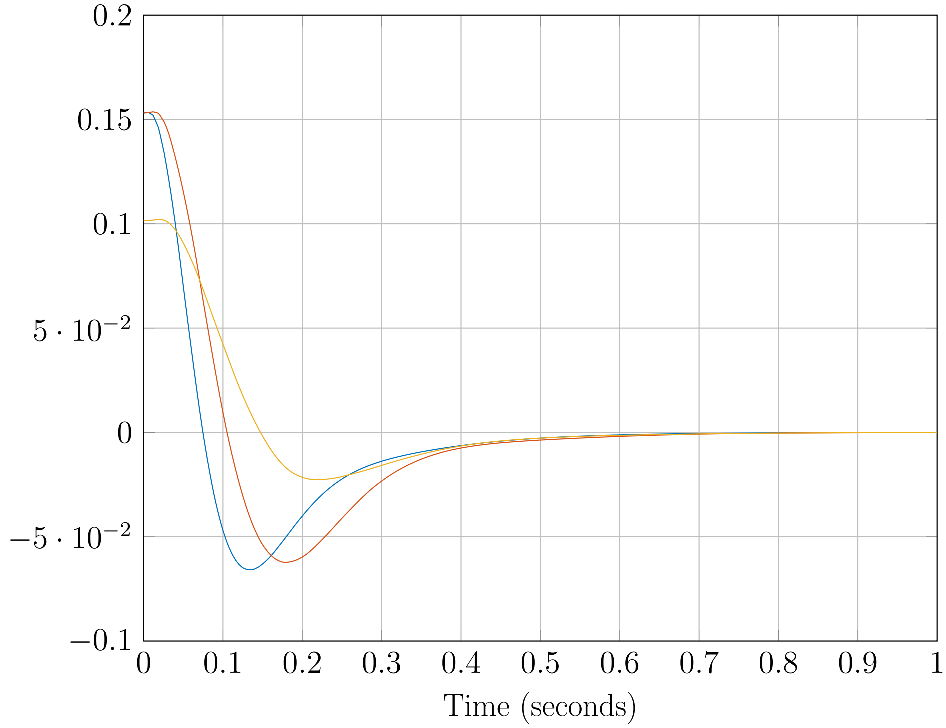

Figure 1: The angular displacement of each pendulum as a function of time. `Blue`: P_1, `Red`: P_2, `Orange`: P_3

1.1.4 Question 4

Figure 2 shows the actual execution schedule 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 3 illustrates that the schedule is indeed feasible by plotting the overall usage of the CPU over the aforementioned timespan.

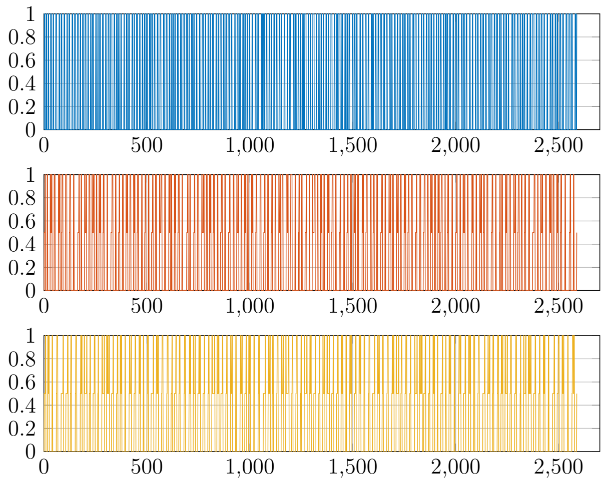

Figure 2: The calculated schedule for the three penduli. `Blue`: P_1, `Red`: P_2, `Orange`: P_3. $C_i = 6$ ms.

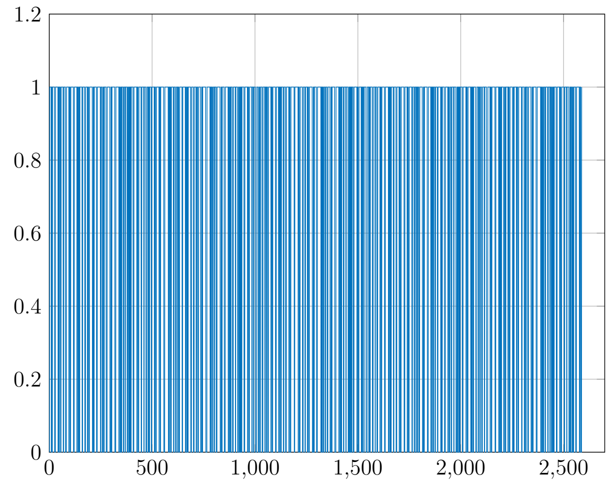

Figure 3: The overall processing usage. Notice that it is at most at 100%. $C_i = 6$ ms.

1.1.5 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 under any scheduling scheme. A portion of the calculated schedule can be found in the appendix in figure 34.

Pendulum P_3 is left to its own devices and tends slowly but surely towards instability. This makes sense since no control signal pertaining to P_3 is assigned execution time in the processor. Figure 5 shows the angular displacement of pendulum P_3 as a function of time.

For penduli P_1 and P_2, however, due to the increased execution time, delays are introduced, and the control input is not as swift as before, hence the increased magnitude of the overshoot and the larger rise and settling times. Figure 4 shows the angular displacement of each stable pendulum as a function of time.

Figures 6 and 7 show the angular displacement of penduli P_1 and P_2 respectively, as a function of time, for the two different cases of execution time.

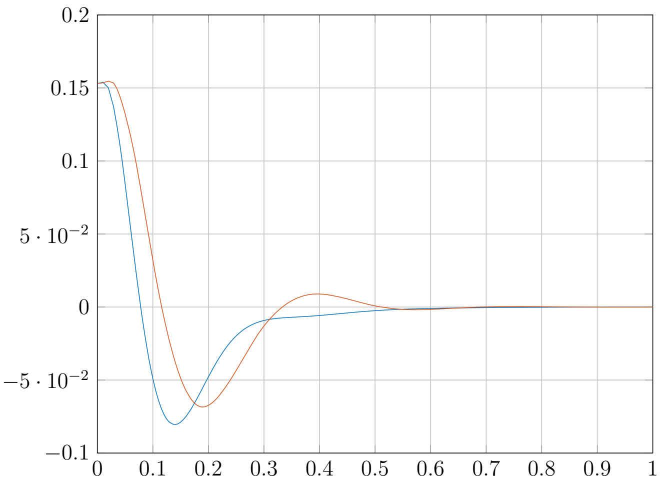

Figure 4: The angular displacement of the stable penduli P_1 and P_2 as a function of time. `Blue`: P_1, `Red`: P_2. $C_i = 10$ ms.

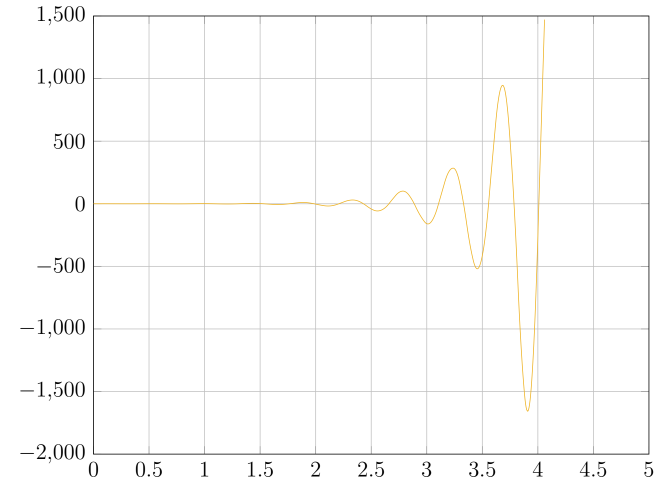

Figure 5: The angular displacement of the unstable pendulum P_3 as a function of time. $C_i = 10$ ms.

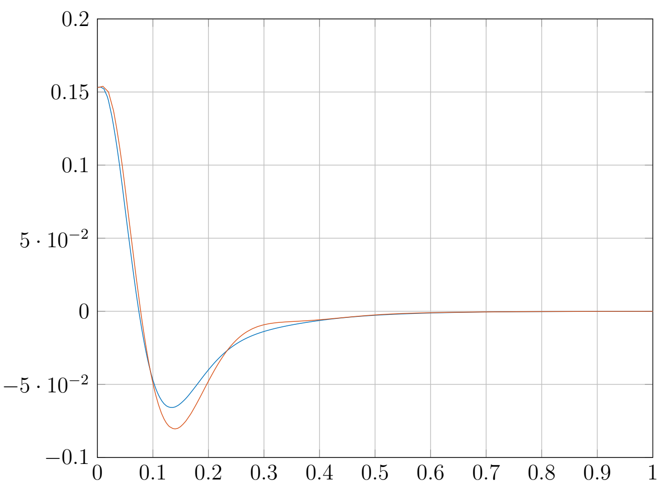

Figure 6: The angular displacement of pendulum P_1 as a function of time. `Blue`: $C_1 = 6$ ms, `Red`: $C_1 = 10$ ms

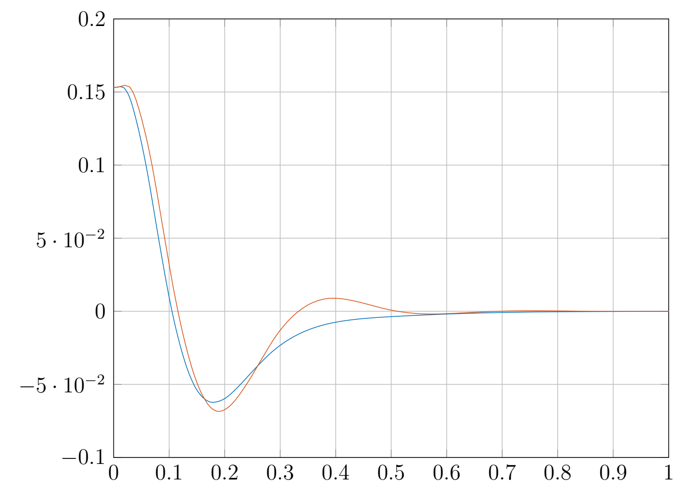

Figure 7: The angular displacement of pendulum P_2 as a function of time. `Blue`: $C_2 = 6$ ms, `Red`: $C_2 = 10$ ms

Figure 8 shows the schedule calculated for each pendulum with all jobs having execution time $C_i = 10$ ms. Figure 9 illustrates that the schedule is not feasible by ploting the overall usage of the CPU over the length of a schedule period, which is at all times 100%, indicative of the excessive processing load demanded.

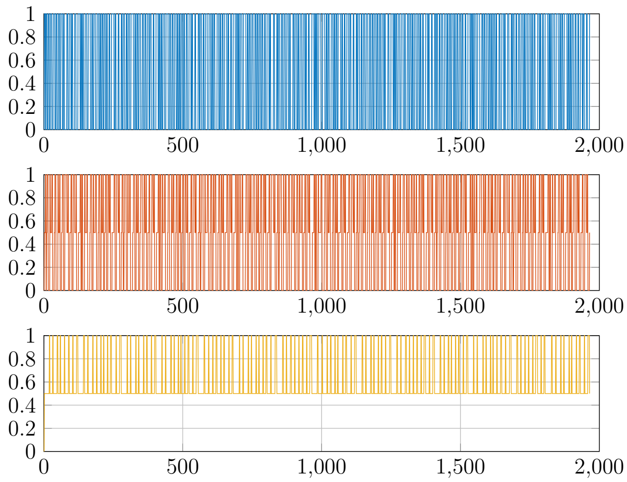

Figure 8: The calculated schedule for the three penduli. `Blue`: P_1, `Red`: P_2, `Orange`: P_3. $C_i = 10$ ms.

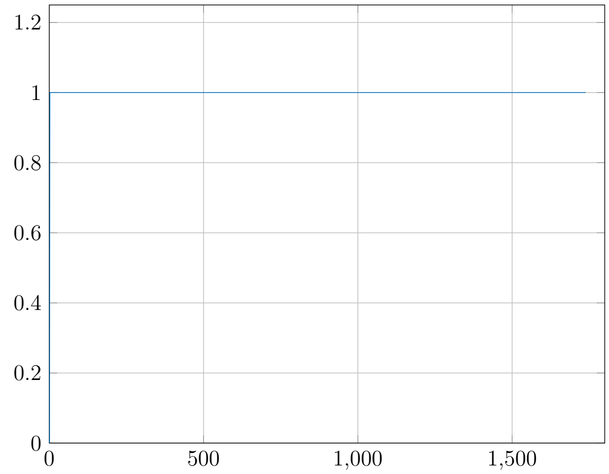

Figure 9: The overall processing usage. Notice that it is always at 100%. $C_i = 10$ ms.

1.2 Earliest Deadline First scheduling

1.2.1 Question 1

In contrast to Rate Monitoring scheduling, Earliest Deadline First scheduling assigns dynamic priorities to tasks, introducing a degree of flexibility. The task whose deadline is closest to the current timestep is given the highest priority and is executed for one time unit.

EDF is more lenient than RM, and this can be seen in the condition that identifies it:

$$U \leq 1 \Leftrightarrow \sigma \text{ is feasible}$$

whereas in RM

$$U \leq n(2^{1/n} - 1) \ (< 1 \text{ for } n \geq 2) \Rightarrow \sigma \text{ is feasible}$$

This means that if $U \leq 1$ for a certain collection of tasks $\{J_i\}$, there is always a feasible schedule for $\{J_i\}$, where the processor is utilized to the fullest it can be, whereas the former certainty does not hold with RM. However, if certain tasks are indeed more important than others and there are extra requirements per their execution, it is possible that EDF introduces delays between their release and start times due to the indiscrimination it shows to absolute task priorities.

1.2.2 Question 2

As stated above, a set of periodic tasks $\{J_i\}$ is schedulable under Earliest Deadline First scheduling if and only if

$$U = \sum_i \frac{C_i}{T_i} \leq 1$$

In the case where $T_1 = 20, T_2 = 29, T_3 = 35$ ms and $C_i = 6$ ms, $i = \{1, 2, 3\}$, $U = 0.678 \leq 1$. Hence tasks J_1, J_2, J_3 are schedulable with EDF. A portion of the calculated schedule can be found in the appendix in figure 35.

1.2.3 Question 3

All penduli are stable for execution time $C_i = 6$ ms. 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 10 shows the angular displacement of each pendulum.

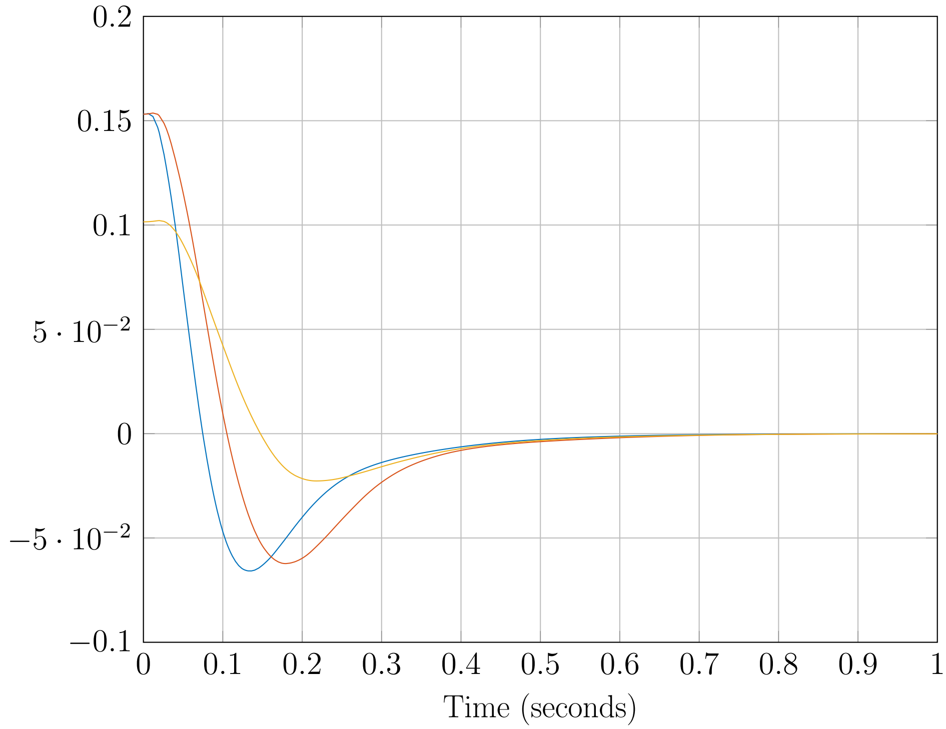

Figure 10: The angular displacement of each pendulum as a function of time. `Blue`: P_1, `Red`: P_2, `Orange`: P_3

1.2.4 Question 4

Figure 11 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 12 illustrates that the schedule is indeed feasible by plotting the overall usage of the CPU over the aforementioned timespan.

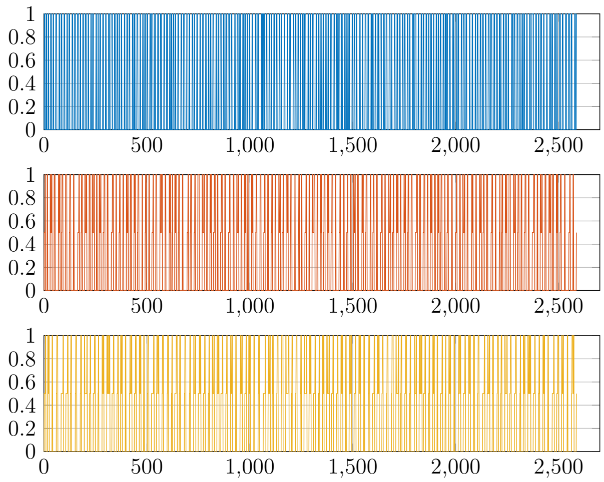

Figure 11: The calculated schedule for the three penduli. `Blue`: P_1, `Red`: P_2, `Orange`: P_3. $C_i = 6$ ms.

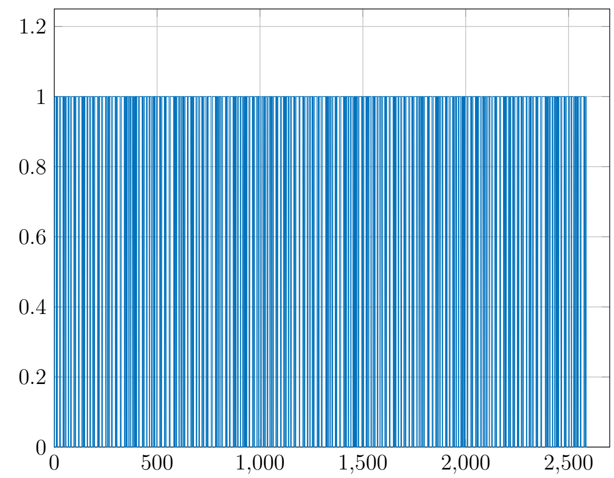

Figure 12: The overall processing usage. Notice that it is at most at 100%. $C_i = 6$ ms.

1.2.5 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 under any scheduling scheme. A portion of the calculated schedule can be found in the appendix in figure 35.

Here we note that the lack of feasibility of a schedule is connected to the notion and fact of deadlines not being met. This, however, does not mean that there can be no control over a system. All penduli are stable as opposed to the case of RM scheduling with the same execution time. This is because all control tasks can be preemped, hence every one of them is allowed execution in the processor, which results to all of them being executed. Hence, despite the fact that deadlines are missed, it is possible to control all three penduli, although it may be with worse control performance than in the case where $C_i = 6$ ms.

Due to the missing of deadlines, and, ultimately, due to the increased execution time, the control input is not as swift as before, hence the increased magnitude of the overshoot and the larger rise and settling times. Figure 13 shows the angular displacement of each pendulum as a function of time. Figures 14, 15 and 16 show the angular displacement of penduli P_1, P_2 and P_3 respectively, as a function of time, for the two different cases of execution time.

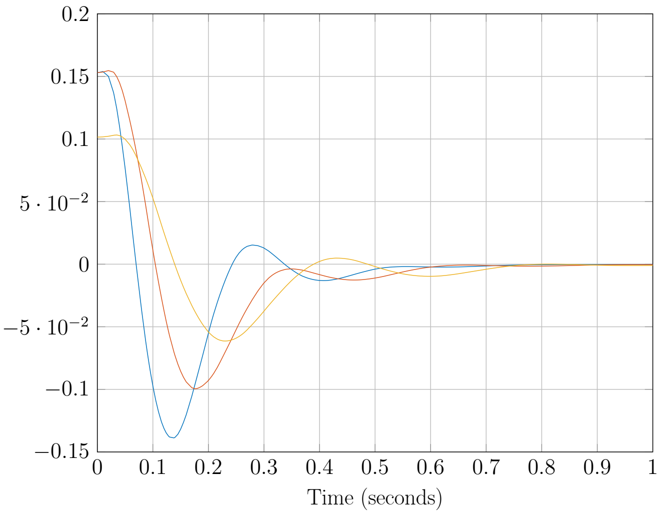

Figure 13: The angular displacement of each pendulum as a function of time. `Blue`: P_1, `Red`: P_2, `Orange`: P_3. $C_i = 10$ ms.

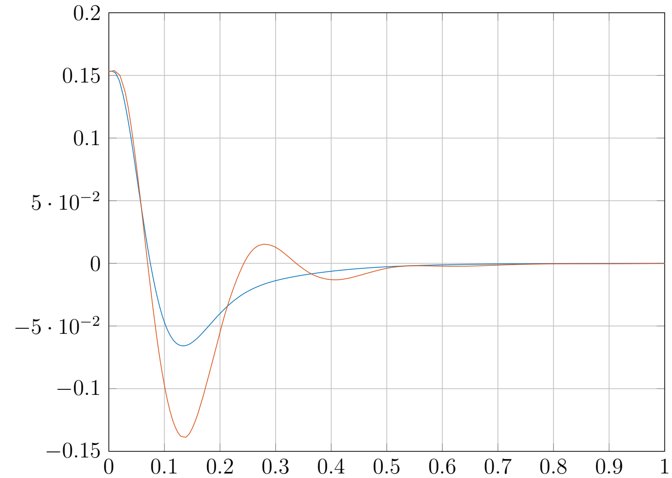

Figure 14: The angular displacement of pendulum P_1 as a function of time. `Blue`: $C_1 = 6$ ms, `Red`: $C_1 = 10$ ms

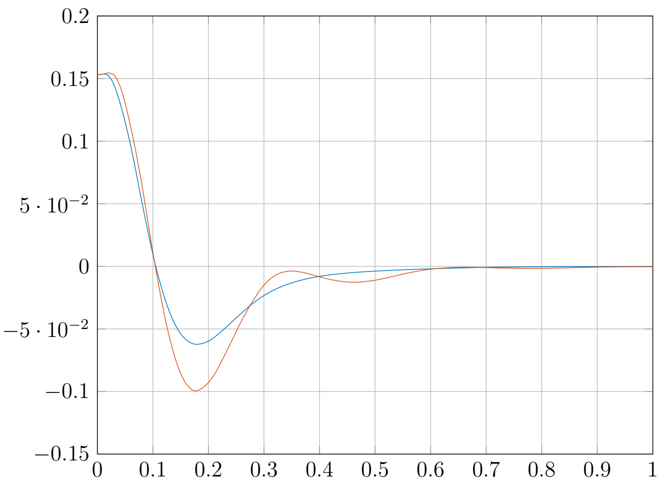

Figure 15: The angular displacement of pendulum P_2 as a function of time. `Blue`: $C_2 = 6$ ms, `Red`: $C_2 = 10$ ms

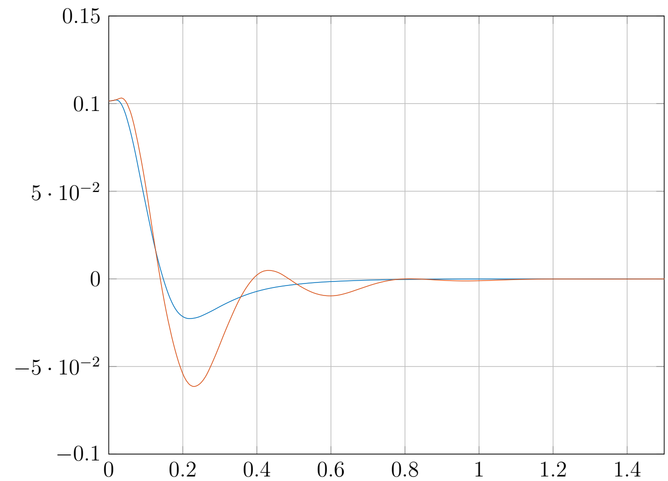

Figure 16: The angular displacement of pendulum P_3 as a function of time. `Blue`: $C_3 = 6$ ms, `Red`: $C_3 = 10$ ms

Figure 17 shows the schedule calculated for each pendulum with all tasks having execution time $C_i = 10$ ms. Figure 18 illustrates that the schedule is not feasible by ploting the overall usage of the CPU over the length of a schedule period, which is at all times 100%, indicative of the excessive processing load demanded.

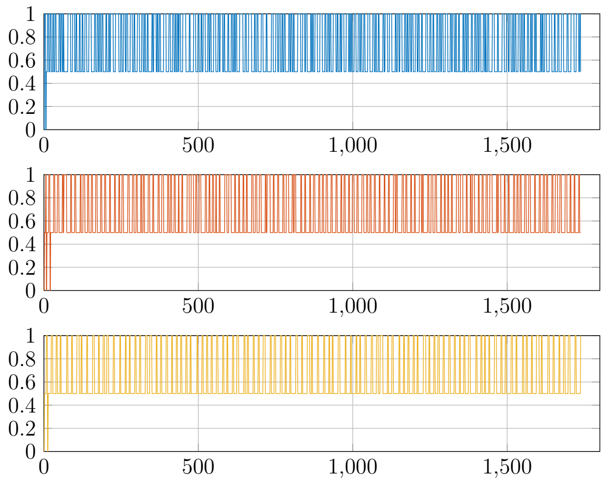

Figure 17: The calculated schedule for the three penduli. `Blue`: P_1, `Red`: P_2, `Orange`: P_3. $C_i = 10$ ms.

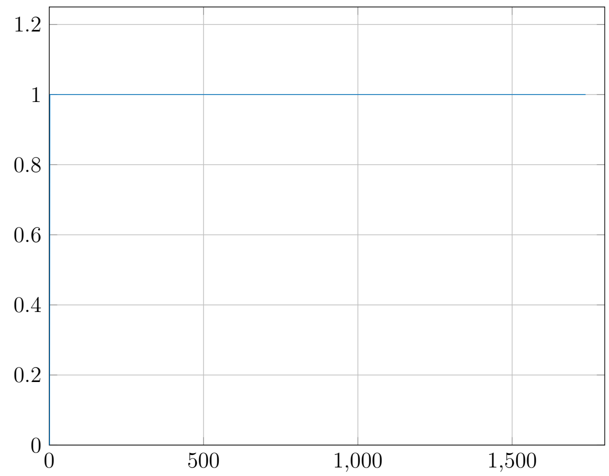

Figure 18: The overall processing usage. Notice that it is always at 100%. $C_i = 10$ ms.

1.2.6 Question 6

Figures 19, 21, 23, 25, and 27 feature the comparison of the course of the angular displacement of every stable pendulum, between the two considered scheduling policies. Figures 20, 22, 24, 26 and 28 graphically illustrate the evolution of the difference in angular displacement between the two considered scheduling policies for the two considered execution times for all stable penduli.

With execution time $C_i = 6$ ms the behaviour of all three penduli is nearly identical with respect to the two different scheduling policies. Scheduling under EDF makes the system have a higher overshoot but its peak difference from that of RM's is by an order of magnitude of 10^{-3} degrees. Hence their difference is negligible.

With execution time $C_i = 10$ ms there are two things to consider. First that pendulum P_3 is not controllable under RM but it is under EDF. Hence, if the purpose at hand is to control all three penduli, EDF is conclusively the only choice there is. On the other hand, though, EDF makes the system behave oscillatory in a degree larger than RM, with higher overshoot, rise and settling times.

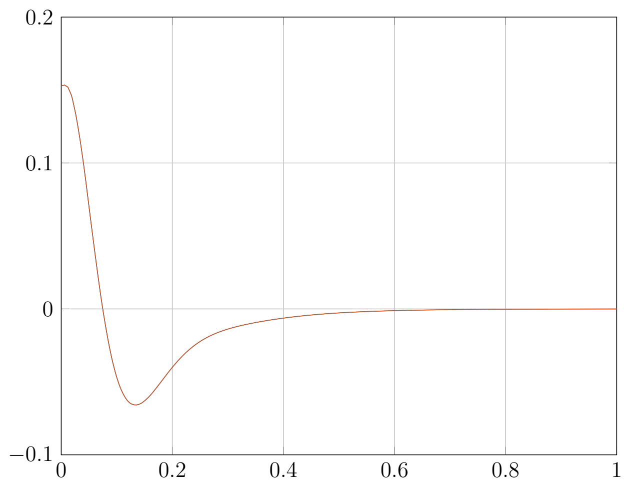

Figure 19: Evolution of the angular displacement of pendulum P_1 for $C_1 = 6$ ms. `Blue`: RM scheduling, `Red`: EDF scheduling

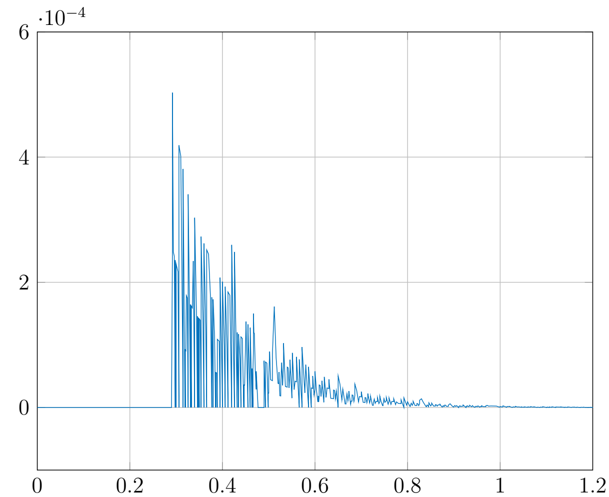

Figure 20: The evolution of the difference in angular displacement between RM and EDF of pendulum P_1 for execution time $C_1 = 6$ ms.

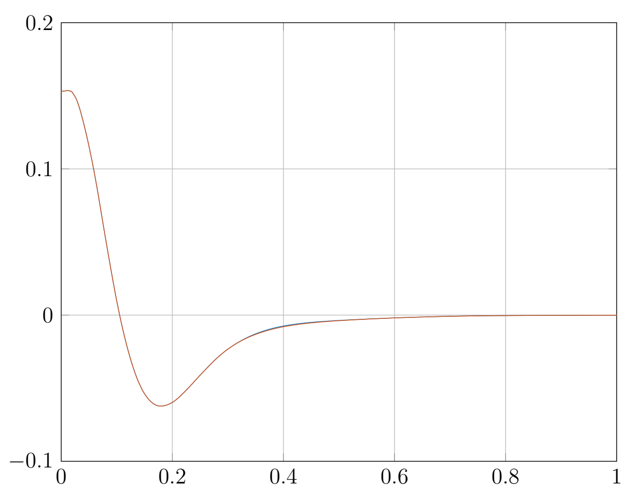

Figure 21: Evolution of the angular displacement of pendulum P_2. $C_2 = 6$ ms. `Blue`: RM scheduling, `Red`: EDF scheduling

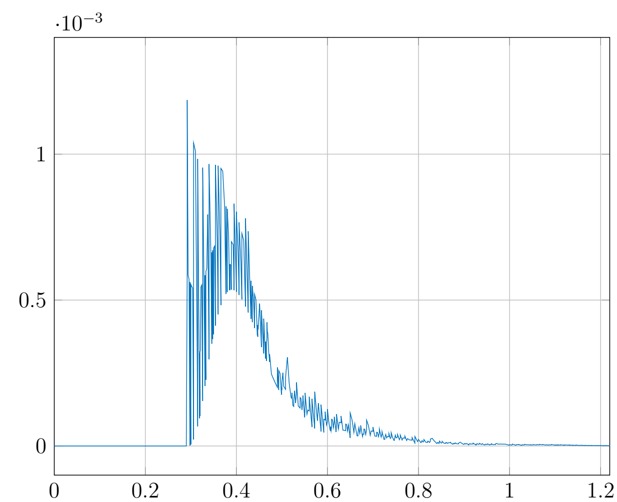

Figure 22: The evolution of the difference in angular displacement between RM and EDF of pendulum P_2 for execution time $C_2 = 6$ ms.

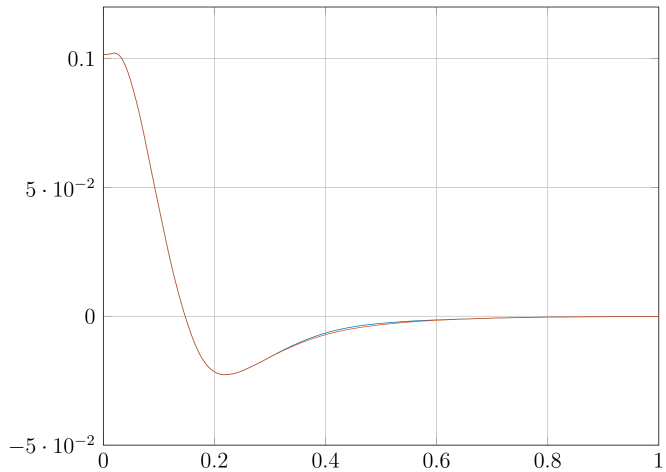

Figure 23: Evolution of the angular displacement of pendulum P_3. $C_3 = 6$ ms. `Blue`: RM scheduling, `Red`: EDF scheduling

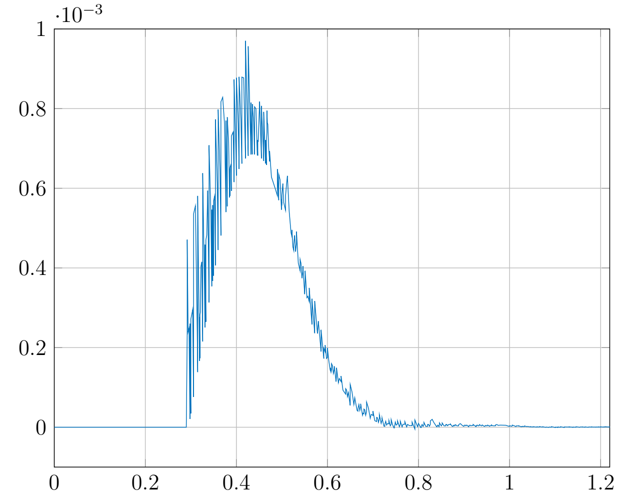

Figure 24: The evolution of the difference in angular displacement between RM and EDF of pendulum P_3 for execution time $C_3 = 6$ ms.

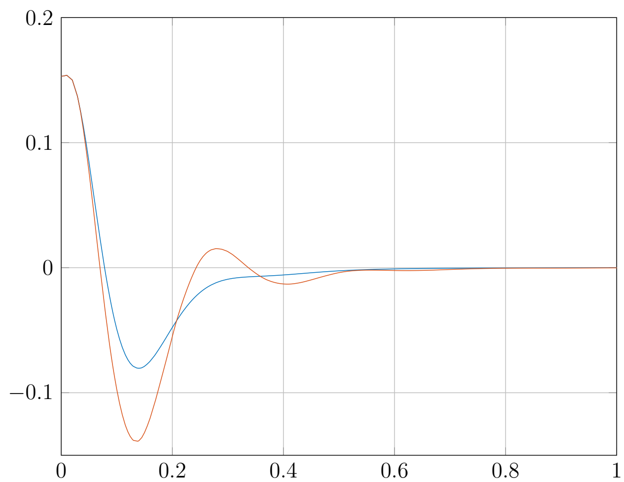

Figure 25: Evolution of the angular displacement of pendulum P_1 for $C_1 = 10$ ms. `Blue`: RM scheduling, `Red`: EDF scheduling

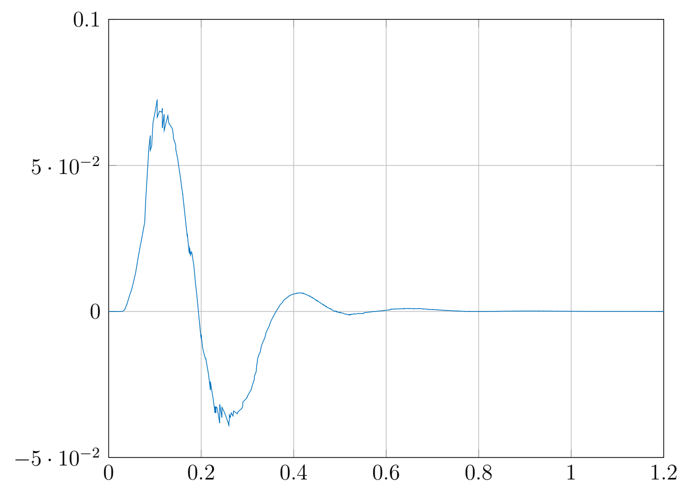

Figure 26: The evolution of the difference in angular displacement between RM and EDF of pendulum P_1 for execution time $C_1 = 10$ ms.

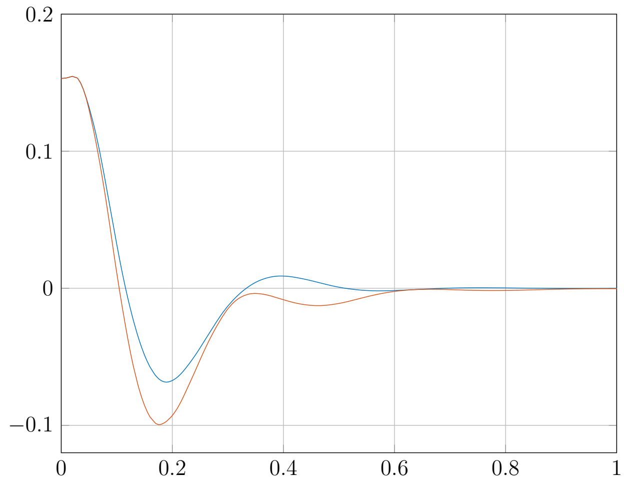

Figure 27: Evolution of the angular displacement of pendulum P_2. $C_2 = 10$ ms. `Blue`: RM scheduling, `Red`: EDF scheduling

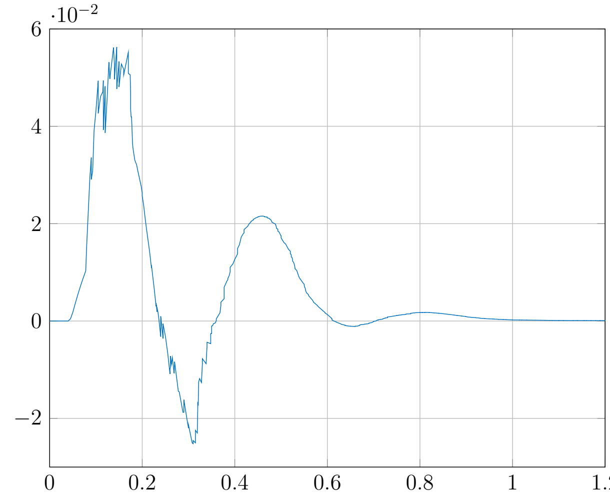

Figure 28: The evolution of the difference in angular displacement between RM and EDF of pendulum P_2 for execution time $C_2 = 10$ ms.

2 Part Two: Networked Control Systems

2.1 Question 1

From the continuous plant

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

and the fact that $A = 0$ we can derive the equation for the discretized state, where $t \in [kh, kh + h)$:

$$\begin{aligned} x(t) &= e^{A(t-kh)}x(kh) + \int_{kh}^{t} e^{A(t-s)}Bu(s)ds \\ &= x(kh) + \int_{kh}^{t} B(-Kx(kh))ds \\ &= x(kh) - (t-kh)BKx(kh) \\ &= (I - (t-kh)BK)x(kh) \end{aligned}$$

and

$$y(t) = Cx(t) = C\big(I - (t-kh)BK\big)x(kh) \tag{1}$$

2.2 Question 2

In order for the system to be stable the eigenvalues of $(I - (t - kh)BK)$ should lie inside the unit circle. Assuming $K = [K_1\ K_2]$:

$$\begin{aligned} I - (t-kh)BK &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - (t-kh)\begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} K_1 & K_2 \end{bmatrix} \\ &= \begin{bmatrix} 1 - K_1(t-kh) & -K_2(t-kh) \\ -K_1(t-kh) & 1 - K_2(t-kh) \end{bmatrix} \end{aligned}$$

The characteristic polynomial of this matrix is

$$\begin{aligned} |\lambda I - (I - (t-kh)BK)| &= (\lambda - 1)^2 + (\lambda - 1)(K_1 + K_2)(t-kh) \\ &= \lambda^2 + \lambda\big((t-kh)(K_1 + K_2) - 2\big) + 1 - (K_1 + K_2)(t-kh) \end{aligned}$$

By Jury's criterion the system is stable if the following inequalities hold:

$$\left.\begin{aligned} 1 - (K_1 + K_2)(t-kh) &< 1 \\ 1 - (t-kh)((K_1 + K_2)) &> (t-kh)(K_1 + K_2) - 2 - 1 \\ 1 - (t-kh)((K_1 + K_2)) &> -(t-kh)(K_1 + K_2) + 2 - 1 \end{aligned}\right\}$$

For $t = kh + \tau$

$$\left.\begin{aligned} (K_1 + K_2)\tau > 0 \\ 2(K_1 + K_2)\tau < 4 \end{aligned}\right\}$$

or

$$0 < (K_1 + K_2)\tau < 2$$

which means that

$$0 < hKB\frac{\tau}{h} < 2$$

$$0 < \frac{\tau}{h} < 2(hKB)^{-1} \tag{2}$$

2.3 Question 3

The minimum value for the communication delay that causes the system to be unstable is approximately $\tau = 0.04$ sec. Figures 29 and 30 show the response of the system for $\tau = 0.039$ and $\tau = 0.04$ sec respectively.

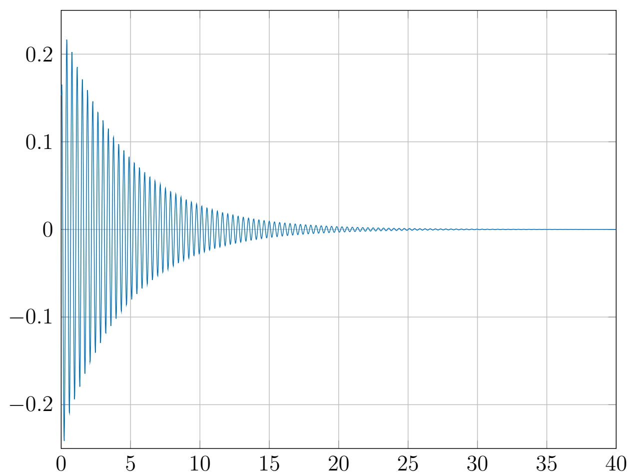

Figure 29: Angular displacement of a pendulum with a communication delay of $\tau = 0.039$ sec.

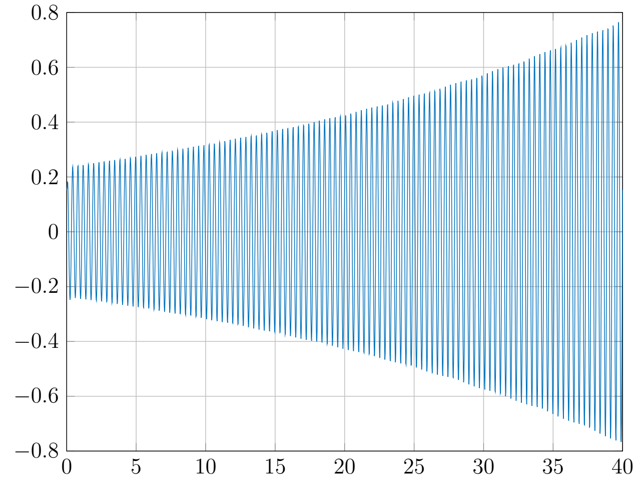

Figure 30: Angular displacement of a pendulum with a communication delay of $\tau = 0.040$ sec.

3 Part Three: Discrete Event System

3.1 Question 1

The discrete event systems that model a machine M and a buffer B are illustrated in figures 31 and 32 respectively.

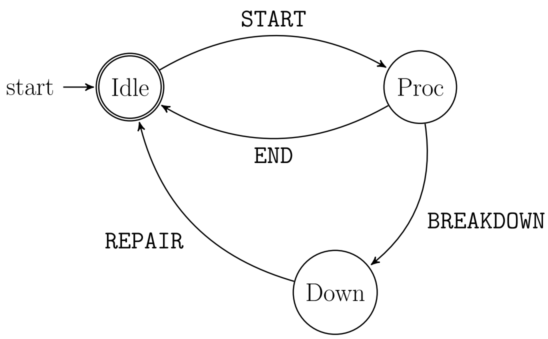

Figure 31: The DES that models machines M_1 and M_2.

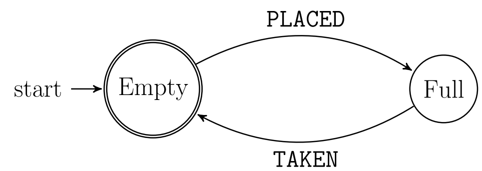

Figure 32: The DES that models buffer B.

4 Question 2

The set of states of the discrete event system that models the complete manufacturing process Q will be a subset of the set of states comprised by the combination of all states of the DES of machines M_1, M_2 and buffer M:

$$Q \subseteq Q_{M_1} \times Q_{M_2} \times Q_B$$

Table 1 features the complete set of states that result from the combination of states of M_1, M_2 and B. Here we can see that some states do not have a meaning in the process, for example state (I, I, F), where both machines are idle but the buffer is full does not correspond to real-life conditions.

state notation of the complete DES	M1	M2	B
(I,I,E)	Idle	Idle	Empty
(I,I,F)	Idle	Idle	Full
(I,P,E)	Idle	Processing	Empty
(I,P,F)	Idle	Processing	Full
(I,D,E)	Idle	Down	Empty
(I,D,F)	Idle	Down	Full
(P,I,E)	Processing	Idle	Empty
(P,I,F)	Processing	Idle	Full
(P,P,E)	Processing	Processing	Empty
(P,P,F)	Processing	Processing	Full
(P,D,E)	Processing	Down	Empty
(P,D,F)	Processing	Down	Full
(D,I,E)	Down	Idle	Empty
(D,I,F)	Down	Idle	Full
(D,P,E)	Down	Processing	Empty
(D,P,F)	Down	Processing	Full
(D,D,E)	Down	Down	Empty
(D,D,F)	Down	Down	Full

Table 1: The complete set of all state combinations between machines M_1 and M_2 and buffer B.

A Appendix

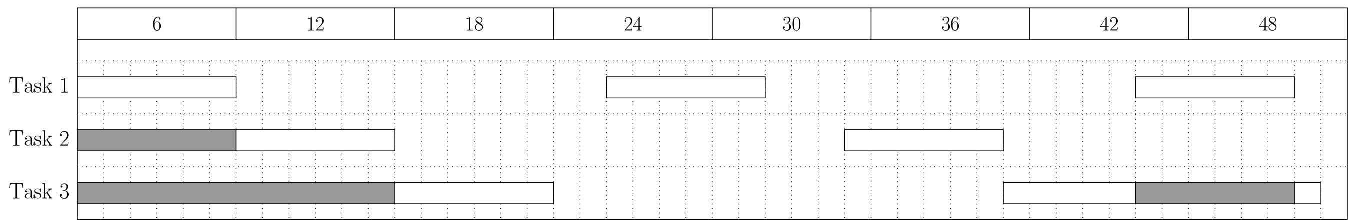

Figure 33: A portion of the RM schedule σ for tasks J_1, J_2, J_3. Shaded areas denote the waiting time.

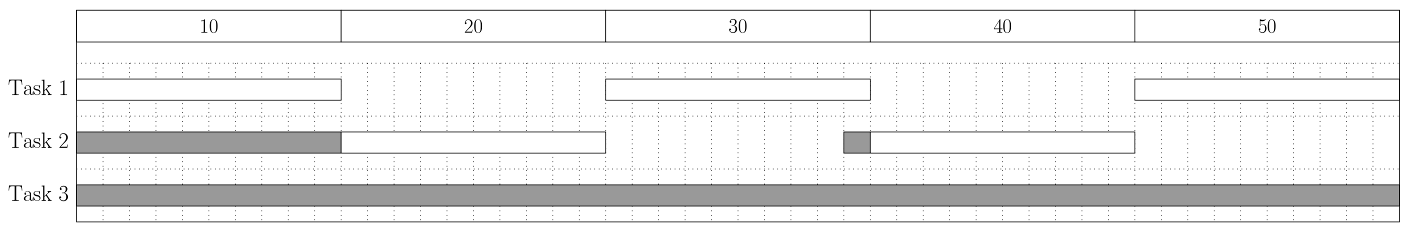

Figure 34: A portion of the RM schedule σ for tasks J_1, J_2, J_3 for $C_i = 10$ ms. Shaded areas denote the waiting time. Notice that J_3 misses its deadlines consecutively, indicative of the inability of schedulability.

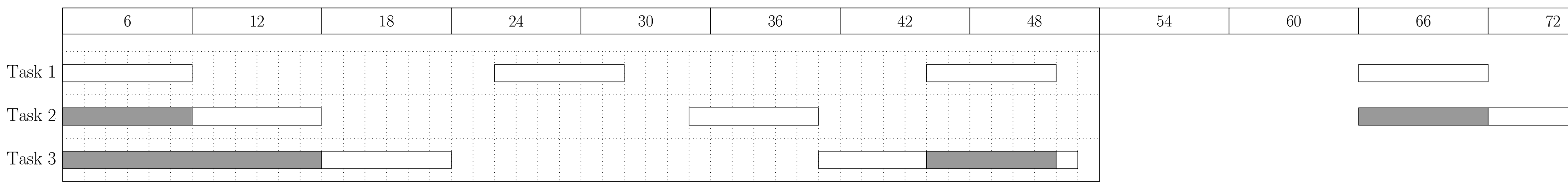

Figure 35: A portion of the EDF schedule σ for tasks J_1, J_2, J_3. Shaded areas denote the waiting time.