Homework 2 in EL2450 Hybrid and Embedded Control Systems

Javier Cerna
930917
fjch@kth.se

Roubing Li
950911
roubing@kth.se

February 11, 2020

Task 1 1/1

For the Rate Monotonic scheduling, the priority of a task J_i is fixed according to its period T_i, i.e., the rate of this task being released. The smaller the period, the higher the priority.

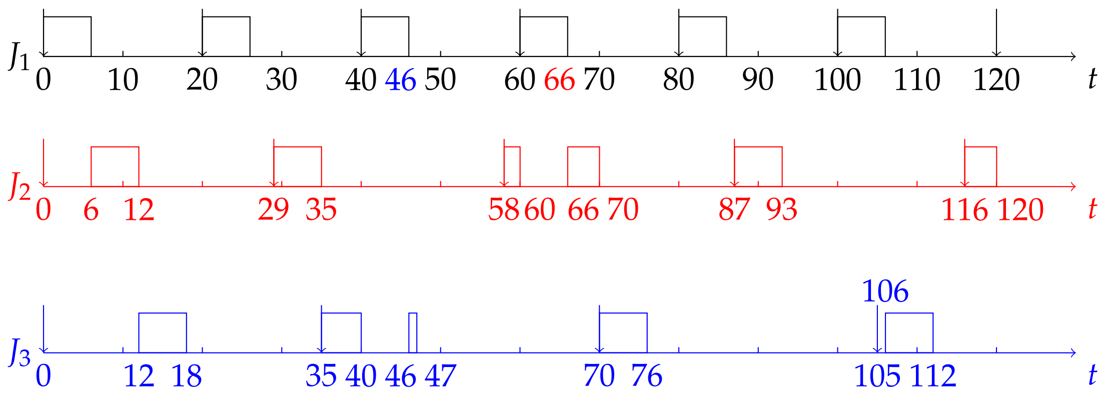
Task 2 2/2

First of all we can check the utilization factor of the three tasks:

$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{6}{20} + \frac{6}{29} + \frac{6}{35}$$
$$\approx 0.68 < n(2^{\frac{1}{n}} - 1) \approx 0.78,$$

Moreover, $U = 0.68$ is less than 0.69, which is the rule of thumb of RM, thus the tasks are schedulable with RM scheduling.

The hyper-period of the three tasks is lcm(20,29,35) = 4060. For simplicity, only part of the schedule is shown in the following diagram:


J_1
0 10 20 30 40 46 50 60 66 70 80 90 100 110 120 t
J_2
0 6 12 29 35 58 60 66 70 87 93 116 120 t
J_3
106
0 12 18 35 40 46 47 70 76 105 112 t

Task 3 1/2

The pendulum angles are plotted and shown in Figure 1. It can be seen that the final value of the three pendulums is 0, which implies that all of them are stabilized.

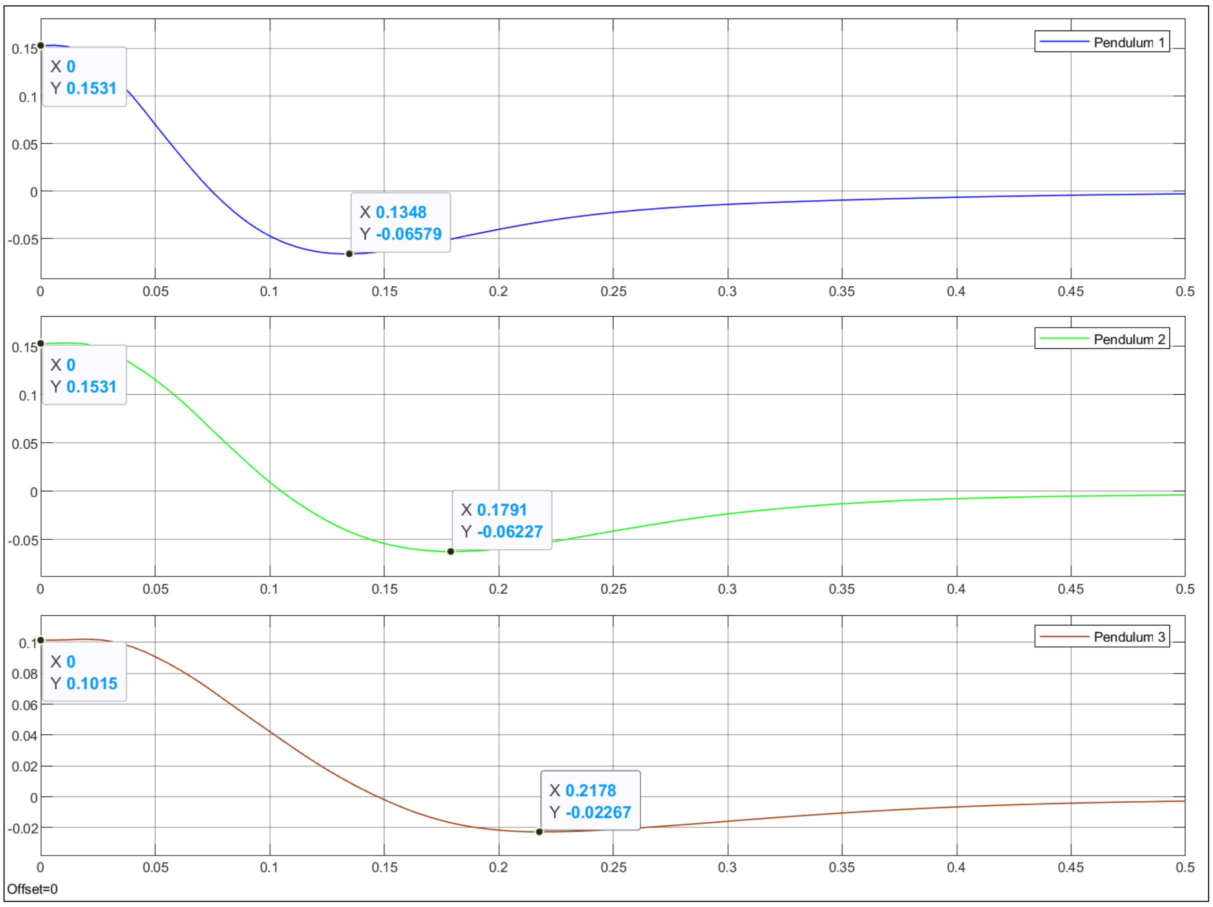


Figure 1: Pendulum angles for execution time = 6 ms

The control performance of each pendulum is characterized by falltime, settling time and undershoot in Table 1. The results are derived from Figure 1. One can conclude, from Table 1, that the control performance of Pendulum1 got the smallest falltime and settling time. That is to say the angle of Pendulum1 got stable quicker than that of the other 2 pendulums. On the other hand, Pendulum3 got the largest falltime and settling-time, while its undershoot(21.34%) is the largest comparing with others.
largest?

	Falltime(ms)	Settling-time(s)	Undershoot
Pendulum1	45.09	0.49	42.14%
Pendulum2	61.26	0.53	40.14%
Pendulum3	85.42	0.54	21.34%

Table 1: control performance of the three pendulums for RM scheduling with 6ms execution time

Task 4 2/2

The schedule plot is shown in Figure 2. As defined theoretically, Task 1 has a sampling time of 20 ms, Task 2 has a sampling time of 29 ms and Task 3 has a sampling time of 35 ms. It can be seen that Task 1 is always executed in 6 ms (it has the highest priority), which explains how it is the ”faster” pendulum to be stabilized. The opposite occurs for Task 3, because it is preempted during its execution by Task 1 and Task 2, every time one of them is released. For this reason, this task takes longer to control the pendulum, which is evidenced in the ”Falltime” of 85.42 ms.

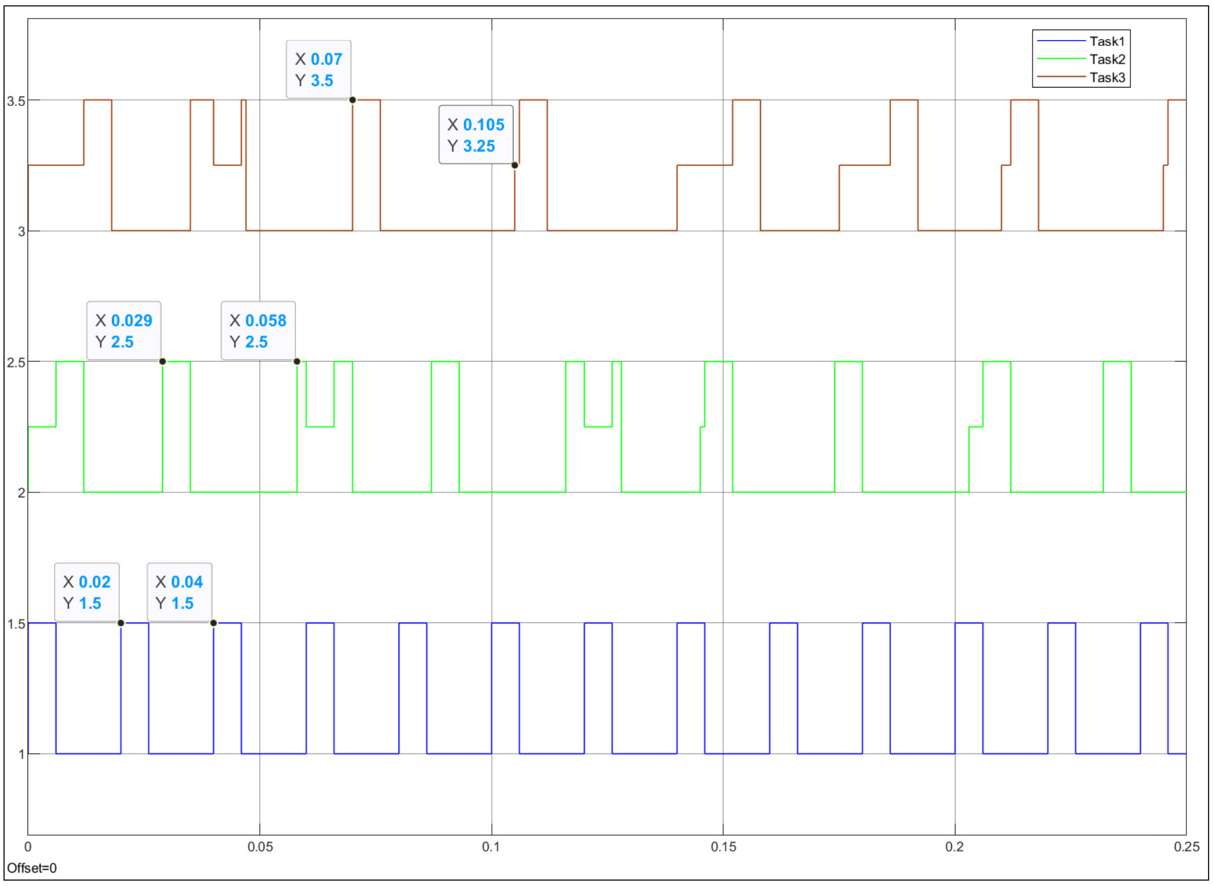


Figure 2: RM schedule for execution time = 6 ms

Finally, because the system is schedulable, all tasks get time to execute before their sampling period is over.

Task 5 3/3

Again, we can check the utilization factor of the three tasks:

$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{10}{20} + \frac{10}{29} + \frac{10}{35}$$
$$\approx 1.13 > 1,$$

Because the utilization factor is greater than 1, there is no scheduling algorithm that can schedule all three tasks. In practice, this means that it is likely that Task 3 will

never is a bit conservative.in time for each deadline is more accurate.
never run, because it will always be preempted by tasks with higher priority.

The pendulum angles are plotted and shown in Figure 3. It can be seen that the final value of Pendulum1 and Pendulum2 is 0, while the final value of Pendulum $\overset{3}{2}$ can't be determined (even plotting more time won't give an indication of the final value, which seems to keep increasing without bounds). This implies that the first two pendulums are stabilized, while the third system is unstable.

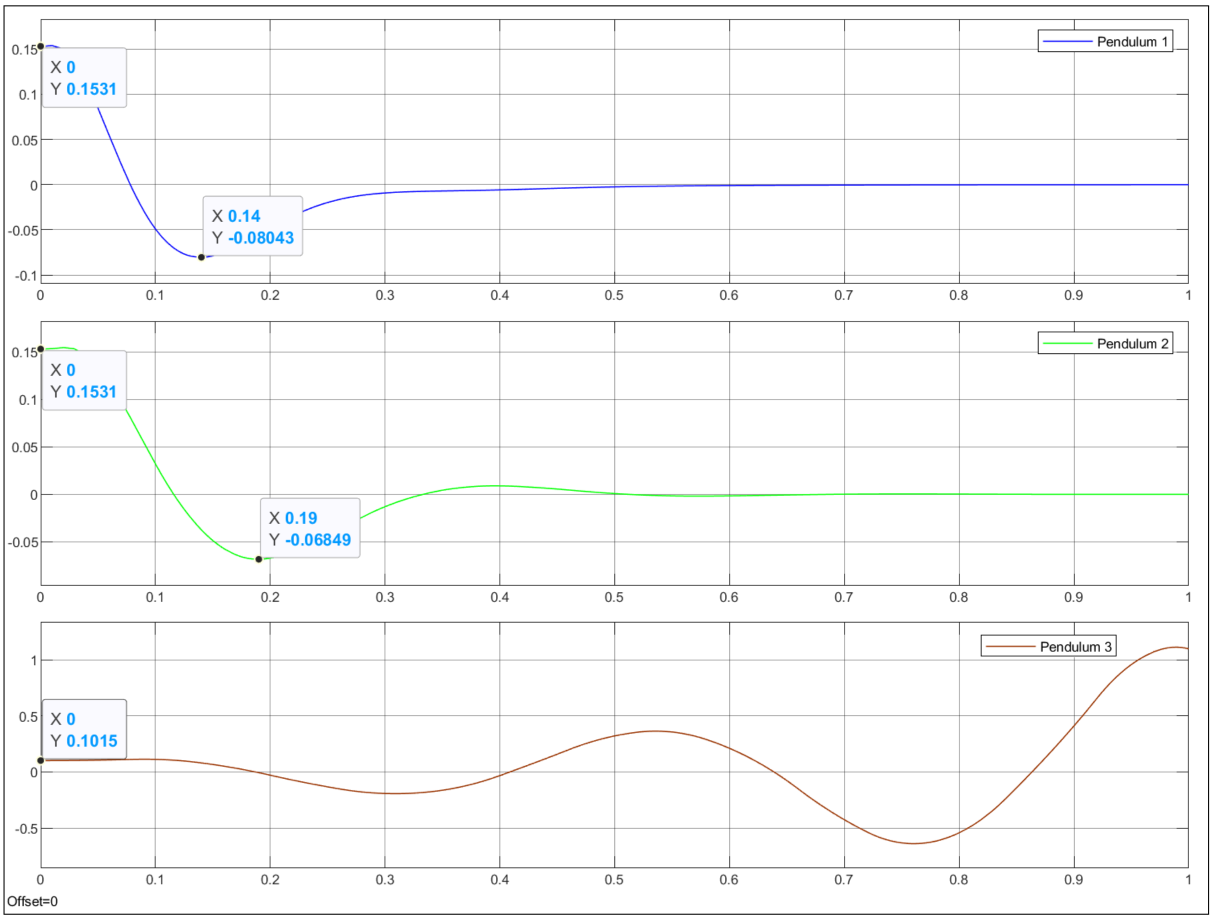


Figure 3: Pendulum angles for execution time = 10 ms

Figure 4 shows the schedule from TrueTime. It can be seen that Task1 always meets its deadline (because it always has the highest priority), and although Task2 is sometimes preempted, it also manages to meet the deadline (both Task 1 and Task 2 complete the execution time of 0.010 s). However, Task3 misses its deadline several times. One example of this is in the segment from 0.035 to 0.07 ms (second release time). Because Task3 has the lowest priority, it is preempted and only manages to complete part of its execution (0.008 s instead of 0.010). The situation worsens for subsequent task releases and this explains why the pendulum is not stabilized.

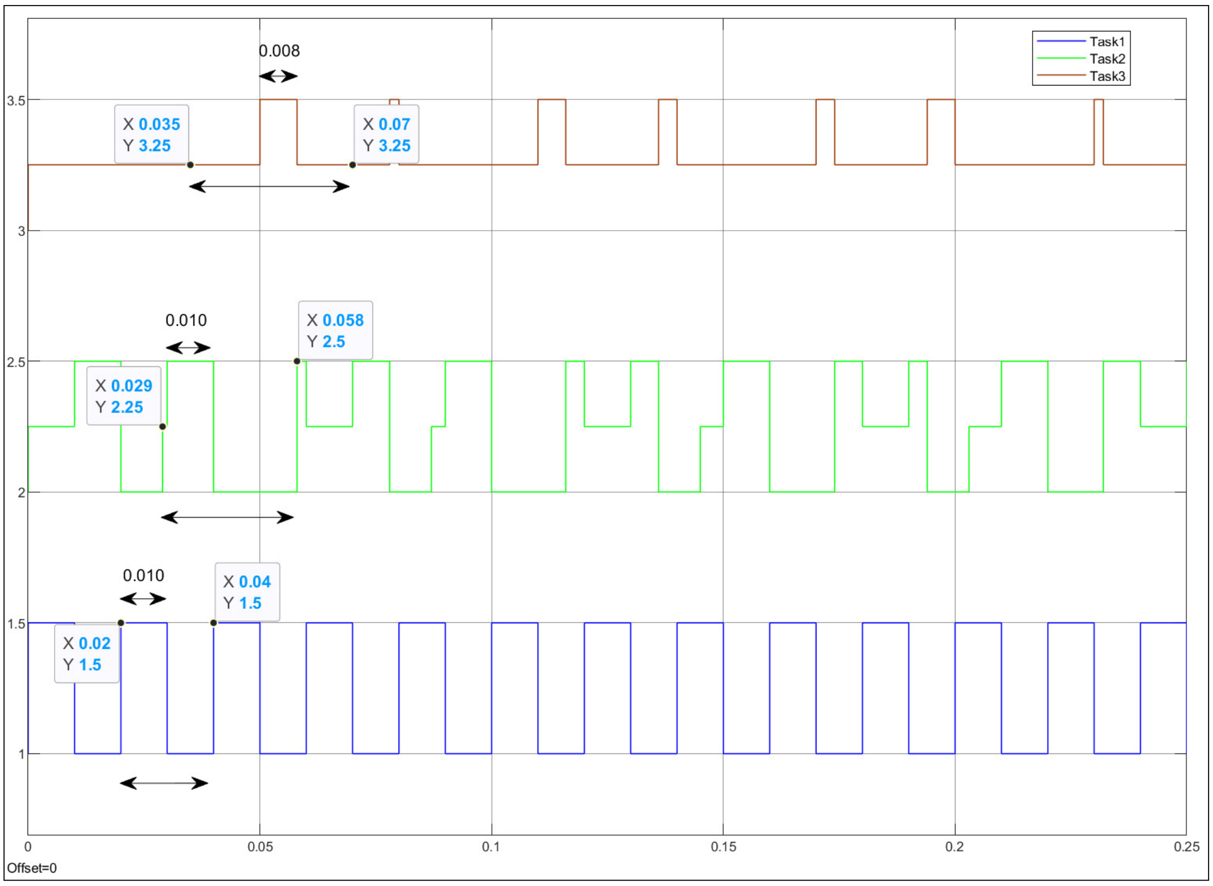
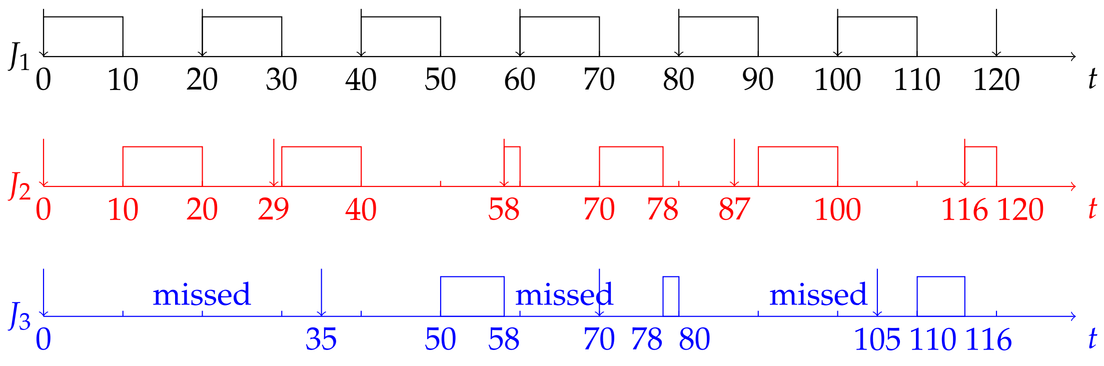


Figure 4: Schedule for execution time = 10 ms

Finally, a part of the schedule (manually plotted) is shown below. To verify correctness, Task 3 can be taken as comparison. In both cases (Figure 3 and manual plot), it can be seen that Task3 completely misses its first execution and completes 8 ms of its second execution, which means that it misses the deadline again.



Task 6 1/1

For the EDF scheduling, the priority of a task is assigned dynamically according to its absolute deadline. The task with the closest deadline will be executed first.

The advantages over RM are the following:

1. Because EDF is optimal, it outperforms RM, in the sense that for some set of tasks, EDF has a higher boundary for schedulable utilization (for EDF it is only required that $U <= 1$).

The disadvantages over RM are the following:

1. If we want to minimize the delay between release and completion of a task, EDF can't guarantee it, because the priorities are assigned dynamically and thus they can change. In RM, the highest priority task will be fixed, which means it will never be preempted.
2. Because EDF computes priorities dynamically, it is harder to predict which tasks could miss the deadline because of overload.
3. EDF is more complex to implement, because on each time step it needs to keep track of all the deadlines and assign the priorities based on the closest one.

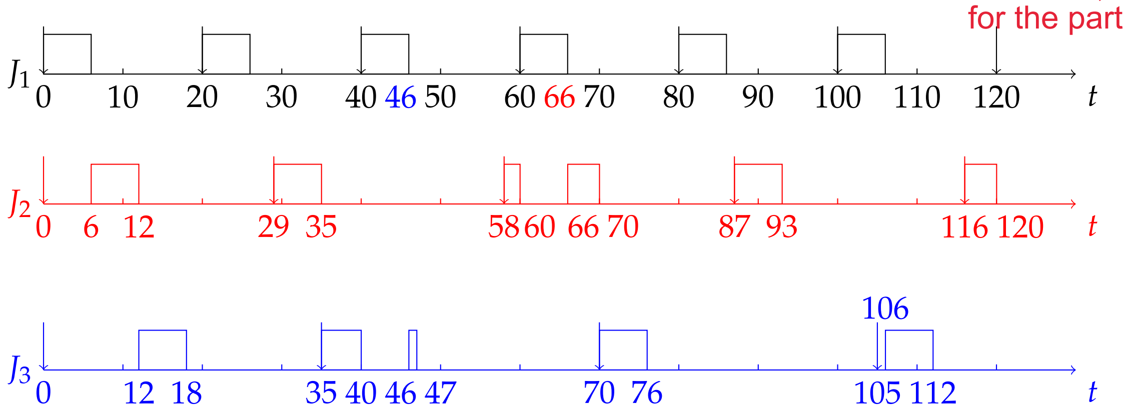
Task 7 2/2

First of all we can check the utilization factor of the three tasks:

$$
\begin{aligned}
U &= \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{6}{20} + \frac{6}{29} + \frac{6}{35} \\
&\approx 0.68 <= 1,
\end{aligned}
$$

For EDF, the utilization factor needs to be less or equal to 1, which means that the tasks are schedulable, and the corresponding schedule is shown as follows. Also note that the schedule in this case is identical with respect to RM scheduling.

note! not the entire schedule, but it is true for the part you made :)



J_1
0 10 20 30 40 46 50 60 66 70 80 90 100 110 120 t
J_2
0 6 12 29 35 58 60 66 70 87 93 116 120 t
J_3
106
0 12 18 35 40 46 47 70 76 105 112 t

Task 8 1/2

The responses of the pendulum angles are shown in Figure 5. One can notice that the parameters of curves are almost the same as those in Task 3, thus the pendulums are finally stabilized and the final values converge to 0.

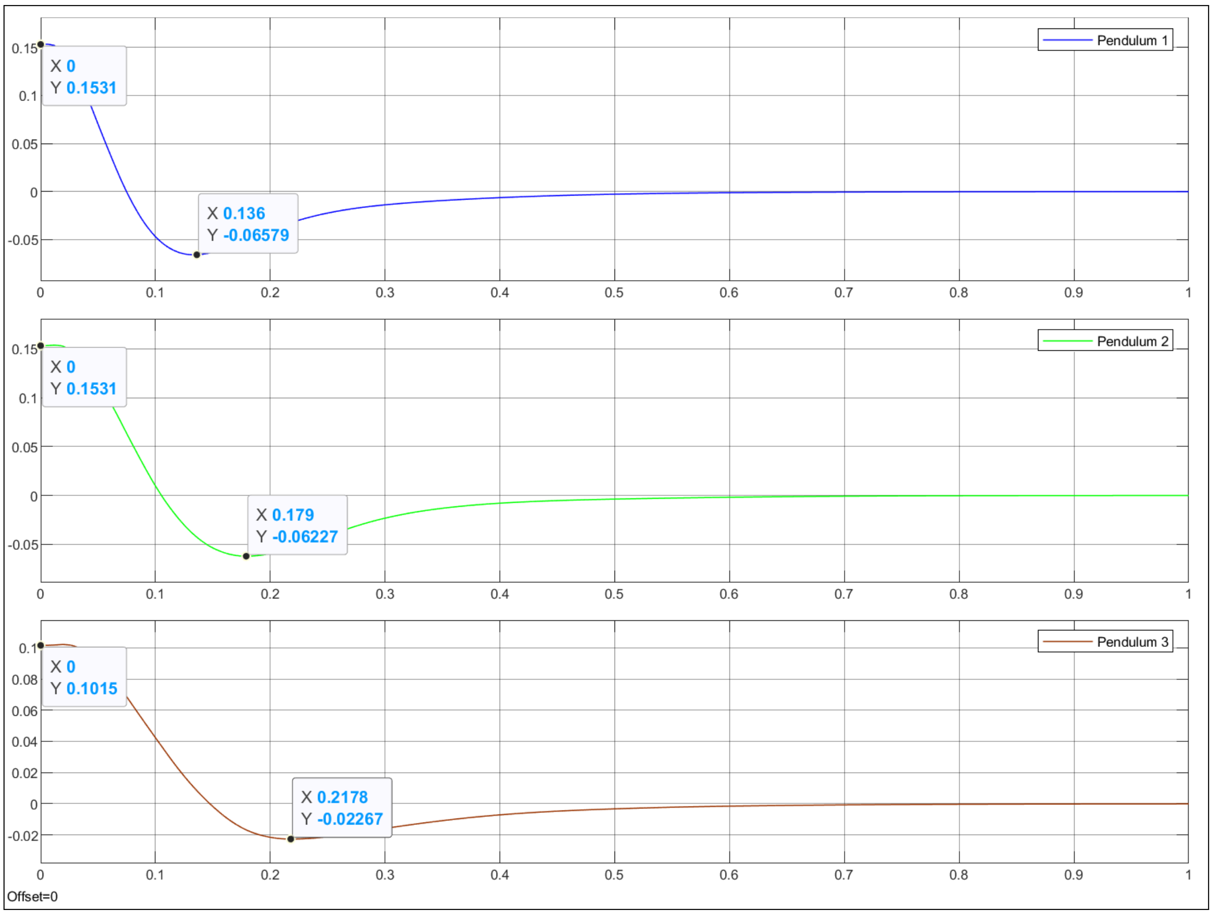


Figure 5: Pendulum angles for EDF scheduling execution-time = 6 ms

The characteristic values(Falltime, Settling-time and Undershoot) of the control performance are shown in Table 2. The control performance of Pendulum1 obtained the smallest falltime(45.09 ms) and settling time(0.49 s). That is to say the angle of Pendulum1 became stable quicker than that of the other 2 pendulums. On the other hand, Pendulum3 got the largest falltime and settling-time, while its undershoot(21.34%) is the largest comparing with others. largest?

	Falltime(ms)	Settling-time(s)	Undershoot
Pendulum1	45.09	0.49	42.14%
Pendulum2	61.26	0.53	40.14%
Pendulum3	85.42	0.54	21.34%

Table 2: control performance of the three pendulums for EDF scheduling with execution-time= 6 ms

Task 9 1/2

In the previous task (Task 8), the same results as in Task 3 were obtained. The reason is that for this particular case, both schedules (EDF and RM) are the same. Aditionally, the execution time is the same as in Task 3 (6 ms). This can be verified by the schedule plot shown in Figure 6. the schedules are not the same! You have only looked at part of it, you can't conclude what happens after.

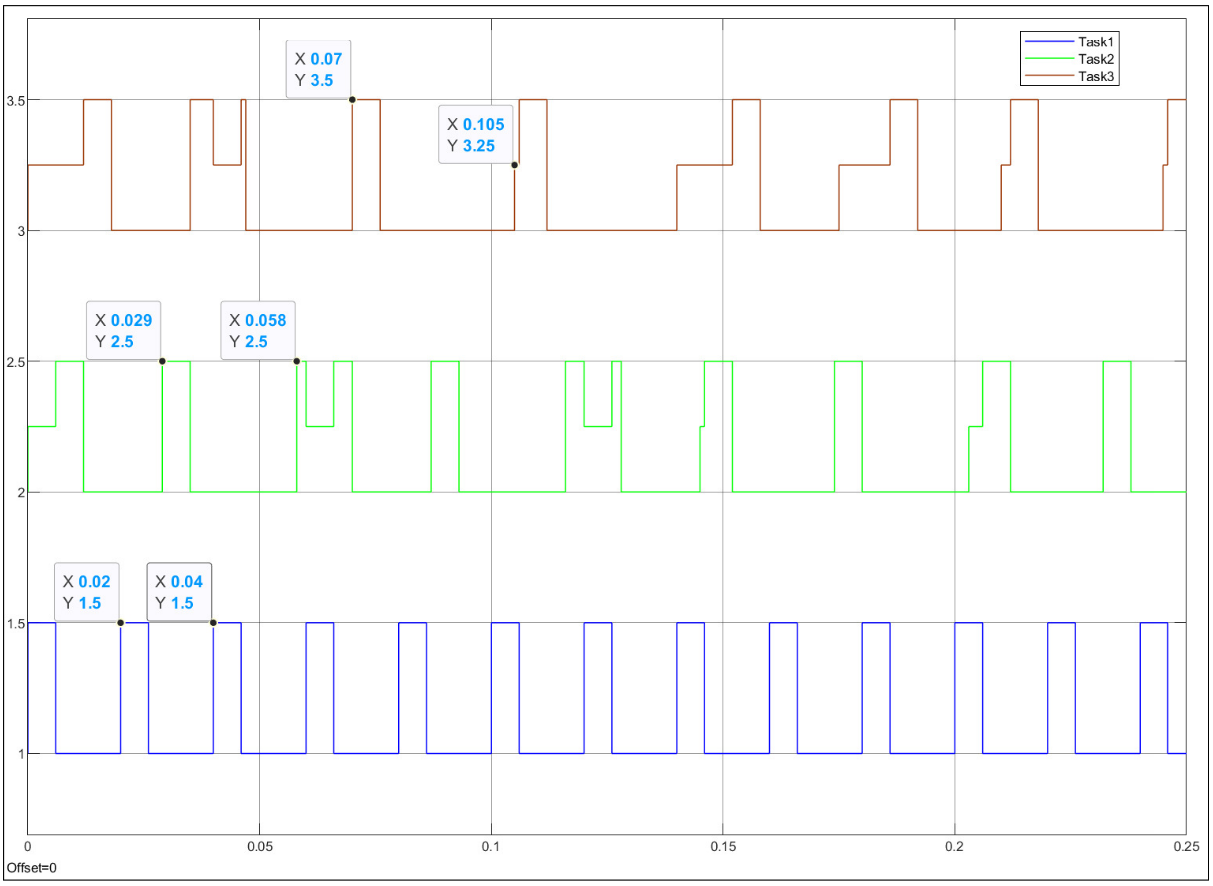


Figure 6: EDF Schedule for execution time = 6 ms

This shouldn't always be the case (in general, EDF and RM are not the same), but because of the particular setup of the tasks (Task 1 is released more frequently than Task 2, and Task 2 is released more frequently than Task 3, and all three tasks have the same execution time) this makes the deadline of Task 1 always be earlier than the deadline of Task2, and the deadline of Task2 always be earlier than the deadline of Task 3 (which essentially gives the same "fixed" priority as RM).

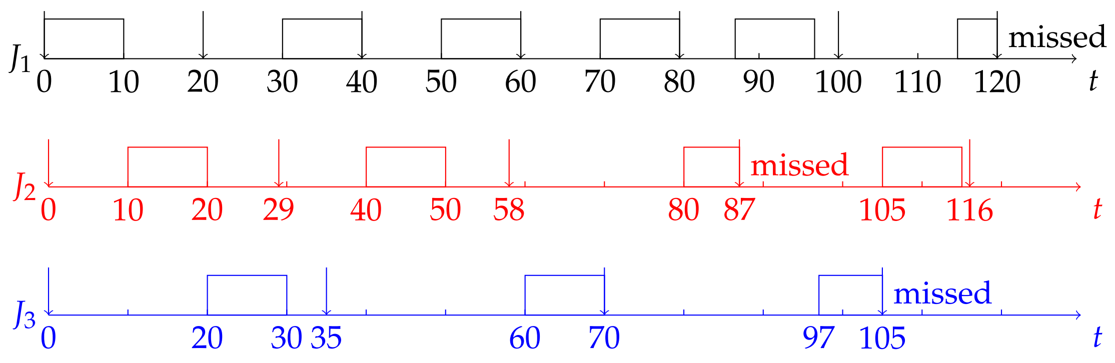
Finally, because the system is schedulable, all tasks get time to execute before their sampling period is over.

Task 10 4/4

Again, we can check the utilization factor of the three tasks:

$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} = \frac{10}{20} + \frac{10}{29} + \frac{10}{35} \approx 1.13 > 1,$$

Because the utilization factor is greater than 1, there is no scheduling algorithm that can schedule all three tasks. A part of the theoretical schedule for EDF is shown below. It can be seen that all three tasks miss its deadline at some point, because the priorities are now dinamically assigned (Task1 can now be preempted).



J_1
0 10 20 30 40 50 60 70 80 90 100 110 120 t
missed
J_2
0 10 20 29 40 50 58 80 87 105 116 t
missed
J_3
0 20 30 35 60 70 97 105 t
missed

Figure 7 shows the schedule from TrueTime. It is similar to the theoretical schedule above, but in this case, each task finishes its execution even if it misses its deadline. One example of this is Task 2, between 0.08 and 0.087 s. In the theoretical schedule, Task 2 misses its deadline and then it's preempted by Task 1, so it never completes. However, in 7 even if Task 2 misses its deadline, it immediately continues running on the next release time until it finishes its execution time (10 ms), and only then this task is preempted.

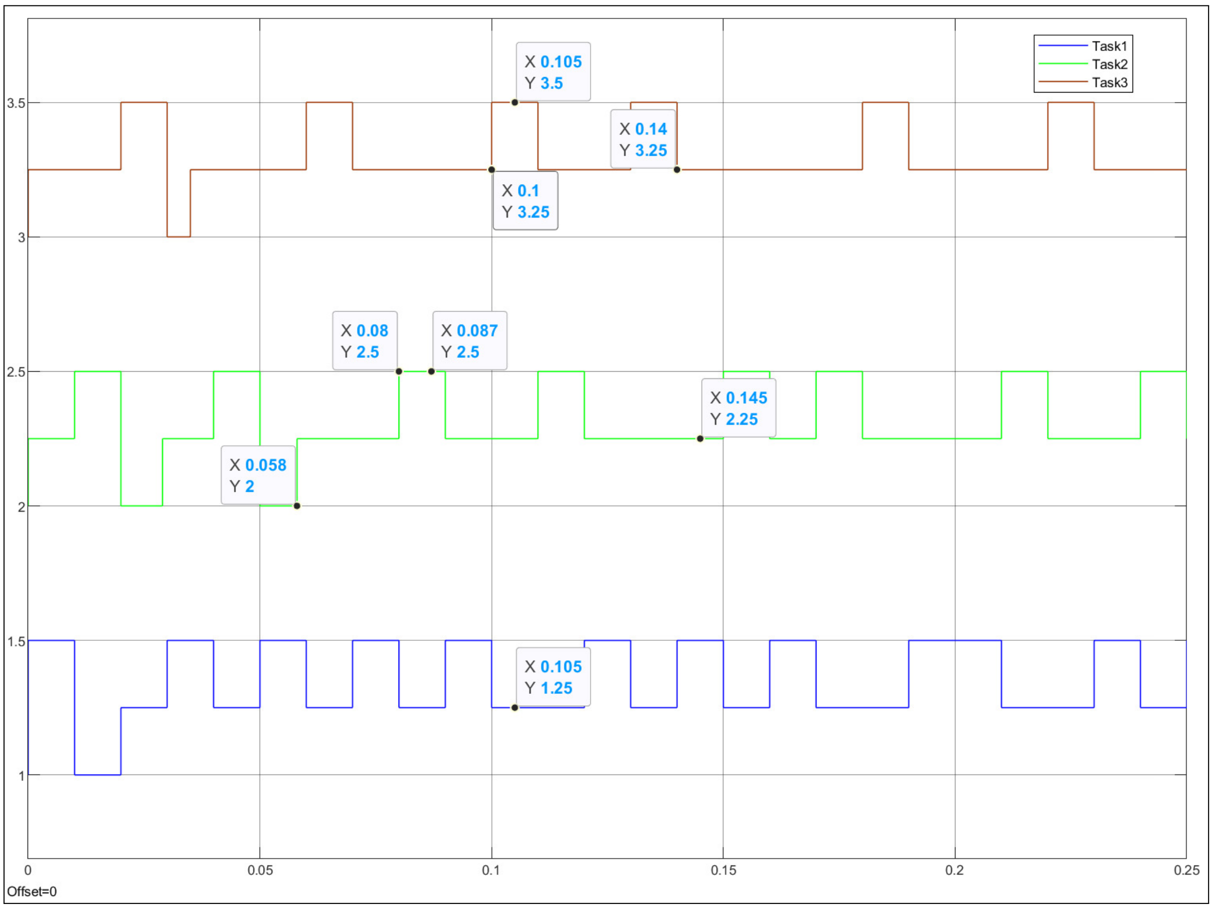


Figure 7: EDF Schedule for execution time = 10 ms

The characteristic values(Falltime, Settling-time and Undershoot) of the control performance are shown in Table 3. The control performance of Pendulum1 obtained the smallest falltime(14.06 ms) and settling time(0.51 s), at the cost of having the largest undershoot (60.94%). That is to say the angle of Pendulum1 became stable quicker than that of the other 2 pendulums. On the other hand, Pendulum3 got the largest fall-time and settling-time, while its undershoot(57.94%) is the smallest comparing with others.

	Falltime(ms)	Settling-time(s)	Undershoot
Pendulum1	14.06	0.51	60.94%
Pendulum2	53.55	0.59	63.12%
Pendulum3	68.90	0.73	57.94%

Table 3: control performance of the three pendulums for EDF scheduling with 10ms execution time

In summary, all three systems are stabilized (even if the tasks are not schedulable), at the cost that the performance of each one is worse than in the case where all of them were schedulable.

Task 11 3/4

In the case that the execution-time = 6 ms, the controllers had the same performance according to the results in Task 3 and Task 8. However, for execution-time = 10 ms, Pendulum3 can be stabilized under EDF scheduling and the simulation result is shown in Figure 8.

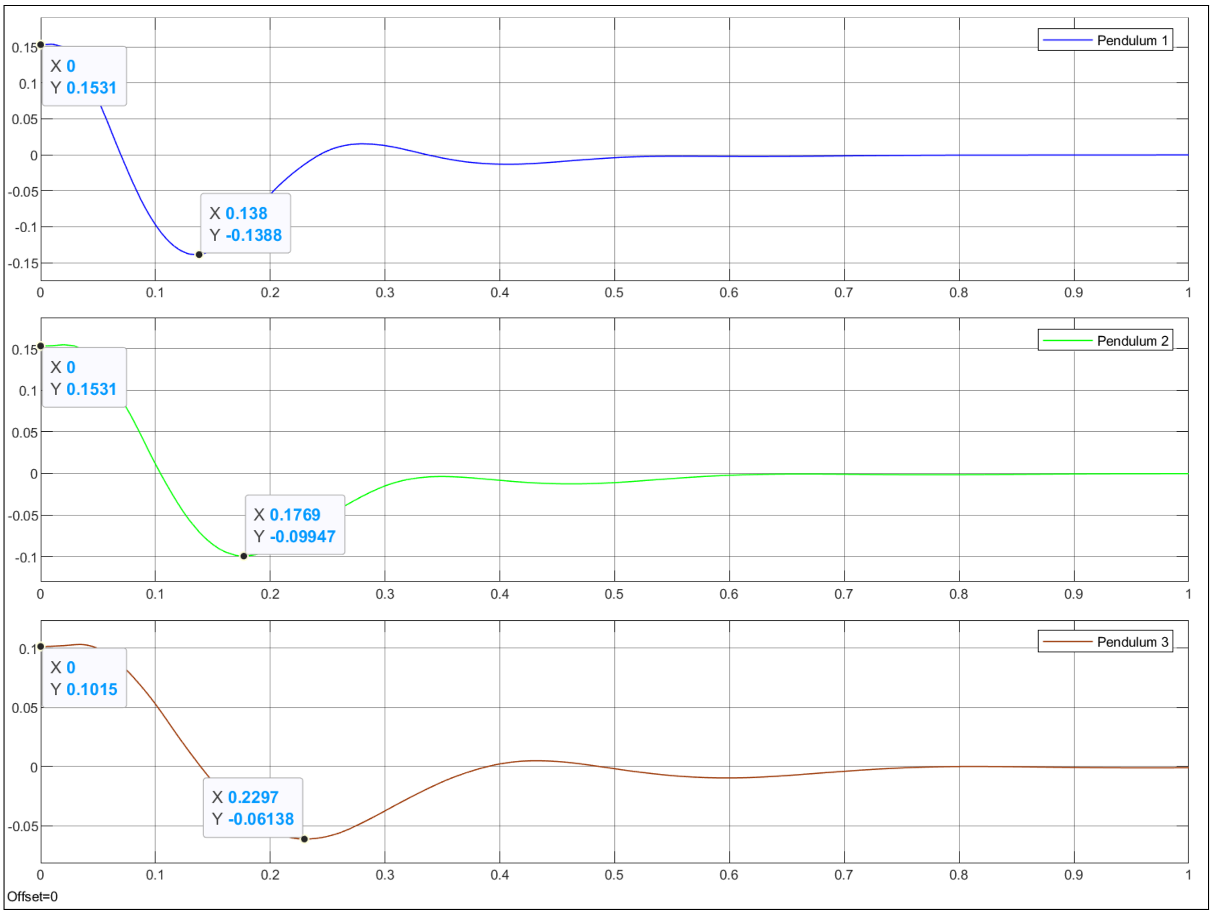


Figure 8: Pendulum angles for execution time = 10 ms

Therefore we can conclude that EDF-scheduling is better than RM-scheduling, considering that all the systems will finally become stable for EDF-scheduling. Whereas, RM-scheduling has less flexibility, thus the highest-priority task would never be preempted, while the lowest-priority task is always interrupted, which leads to the fact that ,in Task 5, the control procedure could never work for pendulum3 to make it stable. is EDF preferred for 6ms too? what shoudl be preferred when they perform equally?

Task 12 1/2

If the delays are combined and considered as $\tau = \tau_{sc} + \tau_{ca}$, then the sampled system can be defined as follows:

$$\begin{aligned} x(kh+h) &= \Phi x(kh) + \Gamma_0 u(kh) + \Gamma_1 u(kh-h) \\ y(kh) &= Cx(kh) \end{aligned}$$

where $\Phi = e^{Ah}$, $\Gamma_0 = \int_0^{h-\tau} e^{As} B ds$ and $\Gamma_1 = \int_{h-\tau}^{h} e^{As} B ds$.

In order to account for the term $u(kh-h)$, the system is augmented and a new state vector $z(kh) = [x(kh), u(kh-h)]^T$ is defined.

Finally, the new state transition matrix is the following:

$$\Phi_z = \begin{bmatrix} \Phi - \Gamma_0 L & \Gamma_1 \\ -L & 0 \end{bmatrix} \tag{1}$$

Replacing values to this equation ($\Phi = 1$, $\Gamma_0 = h - \tau$ and $\Gamma_1 = \tau$ leads to the following result:

$$\Phi_z = \begin{bmatrix} 1-(h-\tau)L & \tau \\ -L & 0 \end{bmatrix} \tag{2}$$

Which leads to the closed loop system representation:

you still have u-dependency
express using only x

$$\begin{bmatrix} x(kh+h) \\ u(kh) \end{bmatrix} = \begin{bmatrix} 1-(h-\tau)L & \tau \\ -L & 0 \end{bmatrix} \begin{bmatrix} x(kh) \\ u(kh-h) \end{bmatrix} \tag{3}$$

Task 13 2/4

The characteristic polynomial comes from setting $det(\lambda I - \Phi_z) = 0$

$$\lambda^2 + [(h-\tau)L - 1]\lambda + L\tau = 0 \tag{4}$$

Then, using Jury stability criterion, four conditions need to be tested:

1. $|a_n| < a_0 \implies \tau < 1/L$
2. $P(z)|_{z=1} > 0 \implies hL > 0$
3. $P(z)|_{z=-1} > 0 \implies \tau > \frac{hL-2}{2L}$
4. $|a_0| < a_2 \implies -1 < L\tau < 1$

what is a0, a2 and an?
or P?

Combining conditions (3) and (4), and dividing by h (h > 0), the following 4 inequalities are derived for $\frac{\tau}{h}$:

1. $\frac{\tau}{h} > \frac{1}{2} - \frac{1}{Lh}$ (condition 3)
2. $\frac{\tau}{h} < \frac{1}{Lh}$ (condition 4)
3. $\frac{\tau}{h} > 0$ (From knowledge that both τ and h are > 0)
4. $\frac{\tau}{h} <= 1$ (From definition of τ with respect to h)

what is the resulting limit then? what is f and g?

Task 14 4/4

Figure 9 shows the response of the system using different delay values (0.01 s, 0.03s and 0.04 s). The system becomes unstable for a delay of 0.04 s.

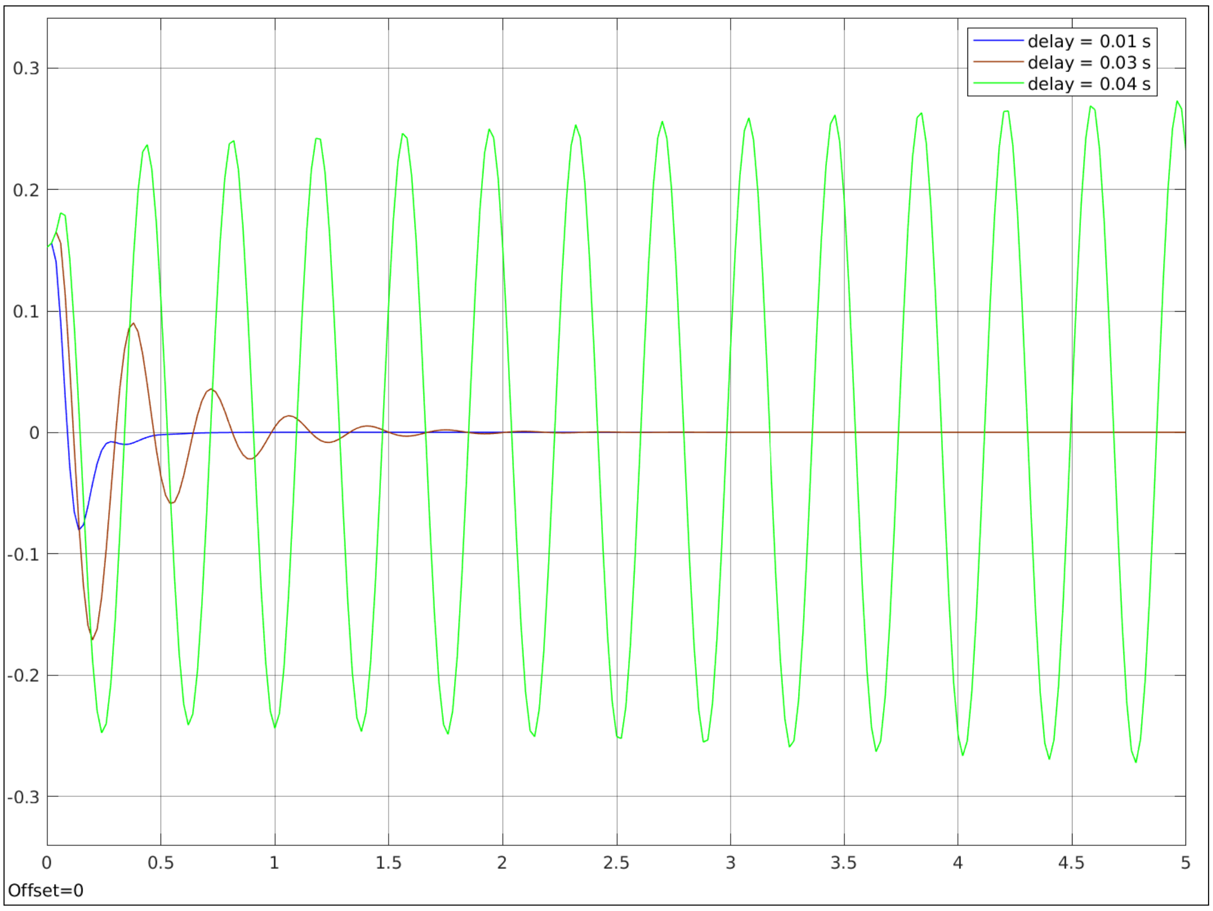


Figure 9: System response to three delay values

Task 15 2/2

Definition of T_1:

- $S = \{s_0, s_1, s_2\}$

- $S^0 = \{s_0\}$
- $\Sigma = \{a, b, c\}$
- $\rightarrow = \{(s_0, a, s_1), (s_1, c, s_0), (s_1, a, s_2), (s_2, b, s_0)\}$
- $AP = S_1 = \{s_0, s_1, s_2\}$
- $L : L(s_0) = s_0, L(s_1) = s_1, L(s_2) = s_2$

Definition of T_2:

- $S = \{t_0, t_1, t_2\}$
- $S^0 = \{t_0\}$
- $\Sigma = \{a, b, c\}$
- $\rightarrow = \{(t_0, a, t_1), (t_1, c, t_2), (t_2, c, t_0), (t_0, b, t_2)\}$
- $AP = S_2 = \{t_0, t_1, t_2\}$
- $L : L(t_0) = t_0, L(t_1) = t_1, L(t_2) = t_2$

Task 16 6/6

Definition of T_{int} (Product interleaving transition system):

- $S_{int} = \{(s_0, t_0), (s_0, t_1), (s_0, t_2), (s_1, t_0), (s_1, t_1), (s_1, t_2), (s_2, t_0), (s_2, t_1), (s_2, t_2)\}$
- $S^0_{int} = \{(s_0, t_0)\}$
- $\Sigma_{int} = \{a, b, c\}$
- $\rightarrow_{int} = \{((s_0, t_0), a, (s_1, t_0)), ((s_0, t_0), a, (s_0, t_1)), ((s_0, t_0), b, (s_0, t_2)), ((s_0, t_1), a, (s_1, t_1)),$
$((s_0, t_1), c, (s_0, t_2)), ((s_0, t_2), a, (s_1, t_2)), ((s_0, t_2), c, (s_0, t_0)), ((s_1, t_0), a, (s_2, t_0)),$
$((s_1, t_0), a, (s_1, t_1)), ((s_1, t_0), b, (s_1, t_2)), ((s_1, t_0), c, (s_0, t_0)), ((s_1, t_1), a, (s_2, t_1)),$
$((s_1, t_1), c, (s_0, t_1)), ((s_1, t_1), c, (s_1, t_2)), ((s_1, t_2), a, (s_2, t_2)), ((s_1, t_2), c, (s_0, t_2)),$
$((s_1, t_2), c, (s_1, t_0)), ((s_2, t_0), a, (s_2, t_1)), ((s_2, t_0), b, (s_0, t_0)), ((s_2, t_0), b, (s_2, t_2)),$
$((s_2, t_1), b, (s_0, t_1)), ((s_2, t_1), c, (s_2, t_2)), ((s_2, t_2), b, (s_0, t_2)), ((s_2, t_2), c, (s_2, t_0))\}$
- $AP_{int} = \{s_0, s_1, s_2, t_0, t_1, t_2\}$
- $L_{int} : L_{int}((s_0, t_0)) = \{s_0, t_0\}, L_{int}((s_0, t_1)) = \{s_0, t_1\},$
$L_{int}((s_0, t_2)) = \{s_0, t_2\}, L_{int}((s_1, t_0)) = \{s_1, t_0\},$
$L_{int}((s_1, t_1)) = \{s_1, t_1\}, L_{int}((s_1, t_2)) = \{s_1, t_2\},$
$L_{int}((s_2, t_0)) = \{s_2, t_0\}, L_{int}((s_2, t_1)) = \{s_2, t_1\},$
$L_{int}((s_2, t_2)) = \{s_2, t_2\}$

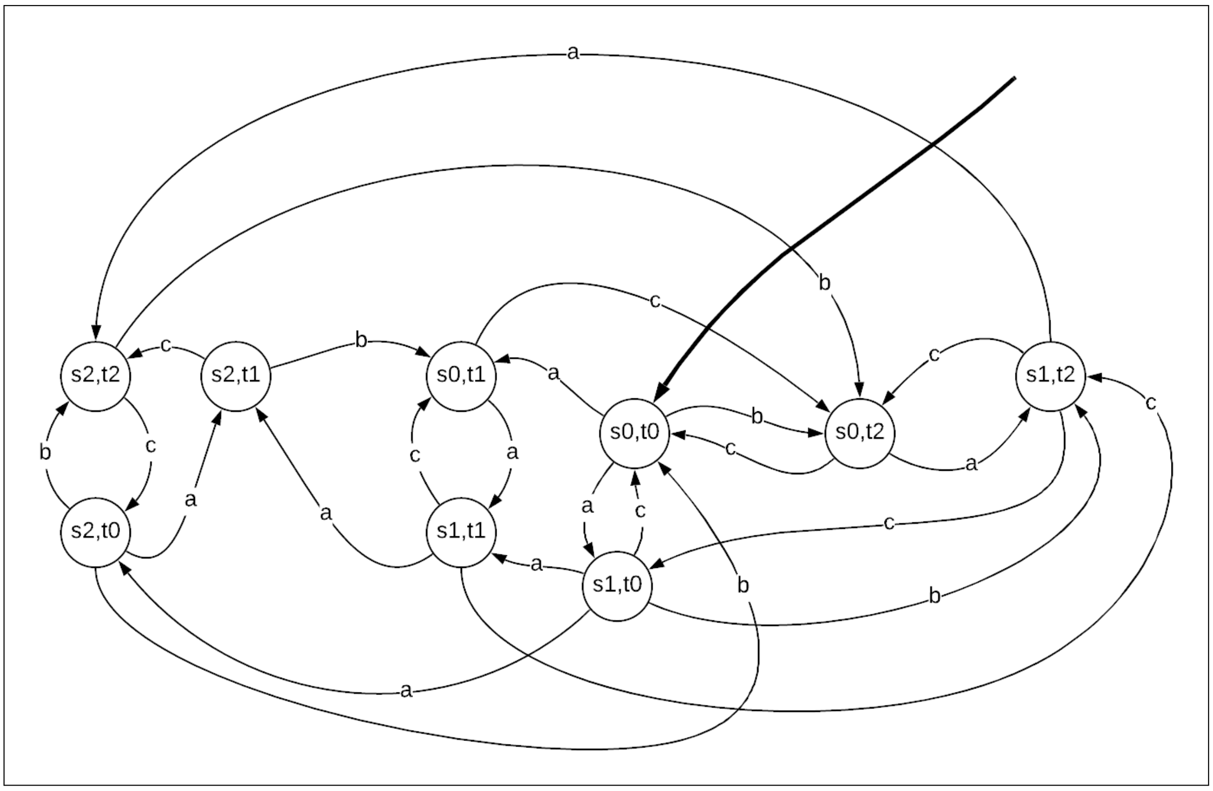


Figure 10: Product Interleaving Transition System T_{int}

Task 17 6/7

Definition of T_{hand} (Product handshaking transition system):

- $S_{hand} = \{(s_0,t_0),(s_0,t_1),(s_0,t_2),(s_1,t_0),(s_1,t_1),(s_1,t_2),(s_2,t_0),(s_2,t_1),(s_2,t_2)\}$
- $S^0_{hand} = \{(s_0,t_0)\}$
- $\Sigma_{hand} = \{a,b,c\}$
- $\rightarrow_{hand} = \{((s_0,t_0),a,(s_1,t_0)),((s_0,t_0),a,(s_0,t_1)),((s_0,t_0),b,(s_0,t_2)),((s_0,t_1),a,(s_1,t_1)),$ $((s_0,t_2),a,(s_1,t_2)),((s_1,t_0),a,(s_2,t_0)),((s_1,t_0),a,(s_1,t_1)),((s_1,t_0),b,(s_1,t_2)),$ $((s_1,t_1),a,(s_2,t_1)),((s_1,t_2),a,(s_2,t_2)),((s_2,t_0),a,(s_2,t_1)),((s_2,t_0),b,(s_0,t_0)),$ $((s_2,t_0),b,(s_2,t_2)),((s_2,t_1),b,(s_0,t_1)),((s_2,t_2),b,(s_0,t_2)),((s_1,t_1),c,(s_0,t_2)),$ $((s_1,t_2),c,(s_0,t_0))\}$
- $AP_{hand} = \{s_0,s_1,s_2,t_0,t_1,t_2\}$
- $L_{hand} : L_{hand}((s_0,t_0)) = \{s_0,t_0\}, L_{hand}((s_0,t_1)) = \{s_0,t_1\},$ $L_{hand}((s_0,t_2)) = \{s_0,t_2\}, L_{hand}((s_1,t_0)) = \{s_1,t_0\},$ $L_{hand}((s_1,t_1)) = \{s_1,t_1\}, L_{hand}((s_1,t_2)) = \{s_1,t_2\},$ $L_{hand}((s_2,t_0)) = \{s_2,t_0\}, L_{hand}((s_2,t_1)) = \{s_2,t_1\},$ $L_{hand}((s_2,t_2)) = \{s_2,t_2\}$

Set is H = {c}.

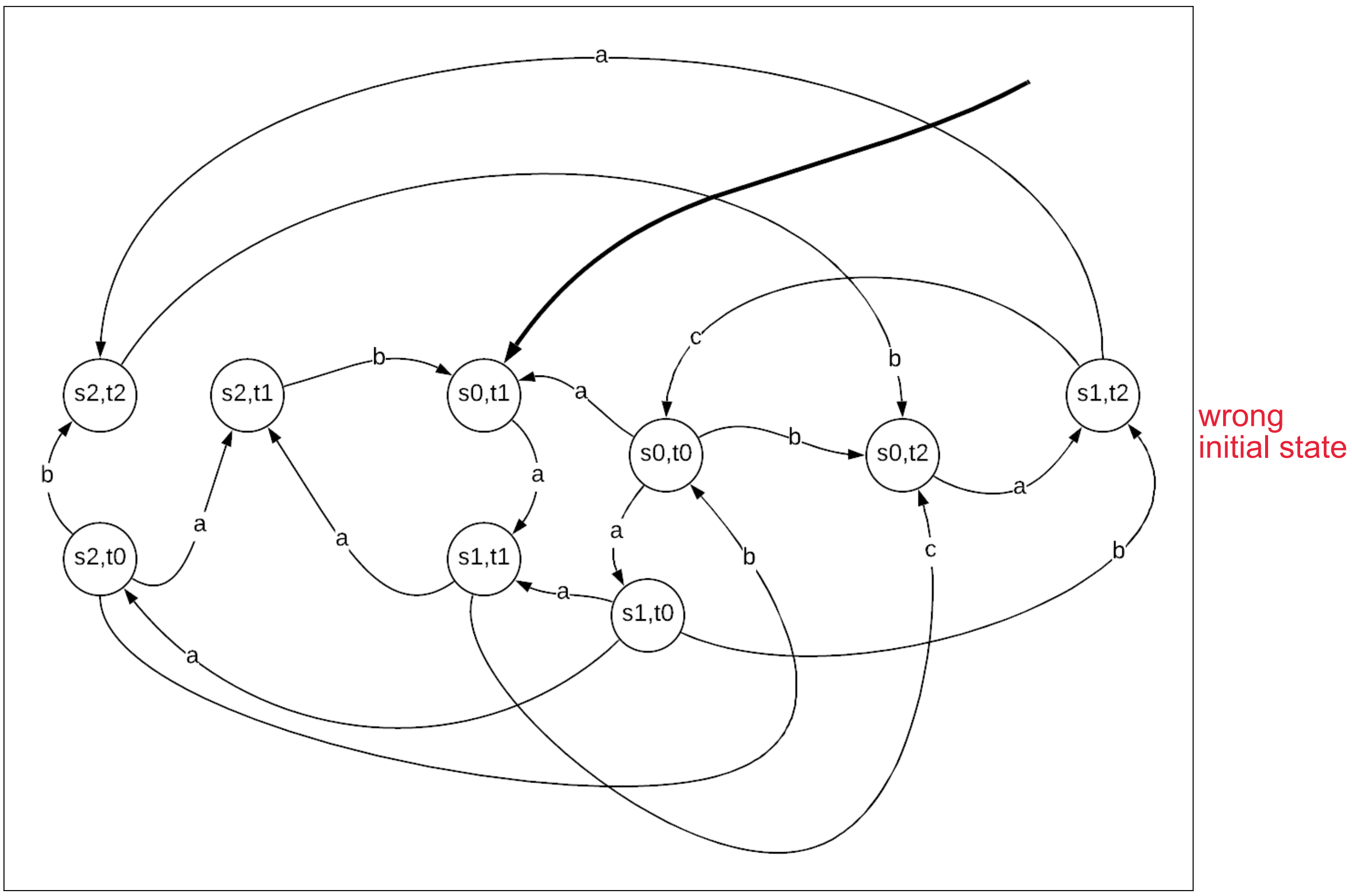


Figure 11: Product Handshaking Transition System T_{hand}