Homework 2 in EL2450 Hybrid and Embedded Control Systems

Jiayi Feng
199808180393
jiayif@kth.se

Gengwu Du
199802131459
gengwu@kth.se

2021-02-12

Instructions and Help

Please remove this part and the sample references before submitting your homework.

Read the general homework instructions available on the course homepage before starting to write the report.

Here are some additional guidelines how to write a homework report.

- Try, if possible, to work in teams of two students. The workload of the homework is for two students.
- Fill in name and personal number of all group members.
- Do not modify this template. Just write names, e-mails, personal numbers and answer the tasks below.
- Do not copy the task descriptions and use the structure below.
- Motivate your answers well and how you derived them, but be concise.
- The number of points is not necessarily related to much you need to write for task.
- Put references in the end if any.
- Do not include plots from the Simulink scope (color on black background) but export the data to Matlab for plotting.
- There is plenty of material available how to use Latex. Use a search engine of your choice to learn more.

Here are some examples how to use Latex:

- An equation with a reference (1) to it

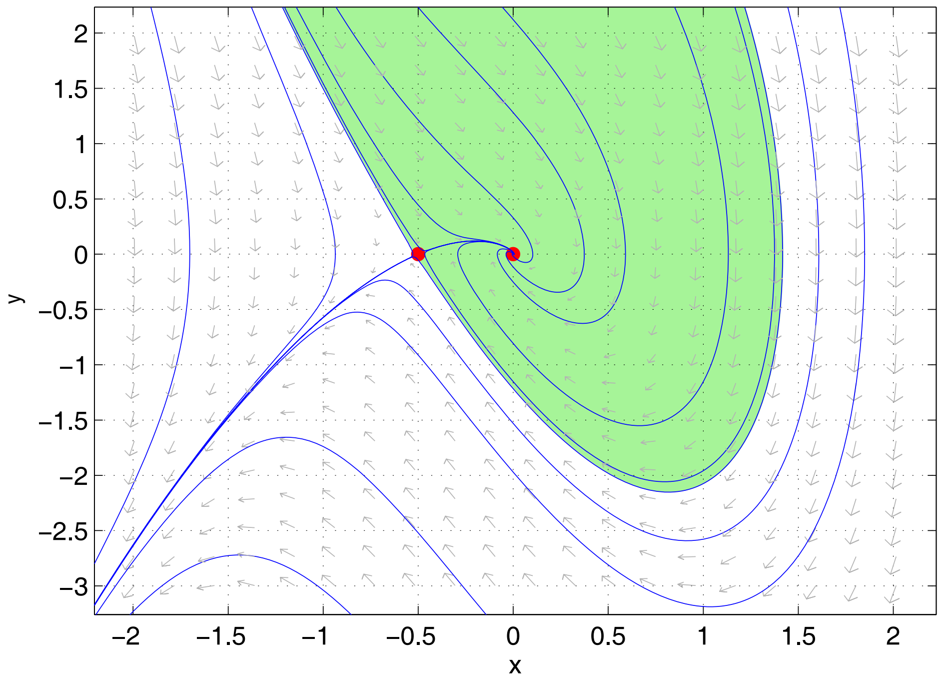
$$\dot{x} = \frac{3}{4}x. \tag{1}$$

- A multi-line equations with a reference to it

$$\begin{aligned}\hat{x} &= x - y\\ \alpha &= x + \gamma.\end{aligned}$$

- An equation in text: $\Phi = \int\limits_0^h \mathrm{e}^{A\tau} d\tau$.

- An image



- A table

-2.46	0	-1.73	0
0	-2.553	0	2.774
0	6.172	-10	7.333
1.767	-0.357	5.714	-6.074

- A citation [2]

- Display something exactly as it is written: `\frac{1}{2}_`

- Basic formating: **bold**, *italics*, `typewriter`

Task 1

Rate monotonic (RM) scheduling algorithm assigns fixed priorities to tasks, and the priority is in reverse proportional to its corresponding period. Once determined, the priority is fixed and will not be changed.

Task 2

Based on the RM Schedulability, the utilization factor

$$U = \sum_{i=1}^{3} \frac{C_i}{T_i} = \frac{6}{20} + \frac{6}{29} + \frac{6}{35} \approx 0.678 \tag{2}$$

For n=3

$$n\left(2^{\frac{1}{n}} - 1\right) = 3\left(2^{\frac{1}{3}} - 1\right) \approx 0.780 \tag{3}$$

Since the Utilization factor is smaller, $0.678 < 0.780$, the task is schedulable with RM. The corresponding manual schedule is given below:

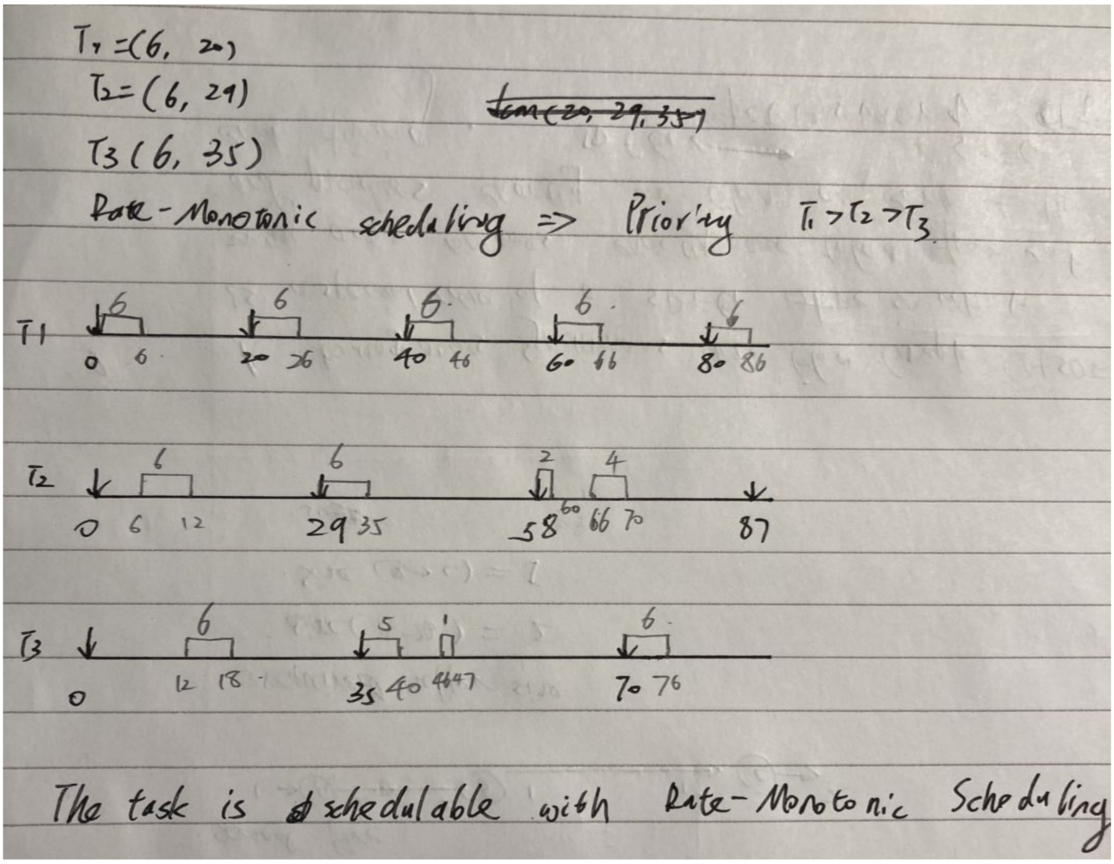


Figure 1: manual RM scheduling

Task 3

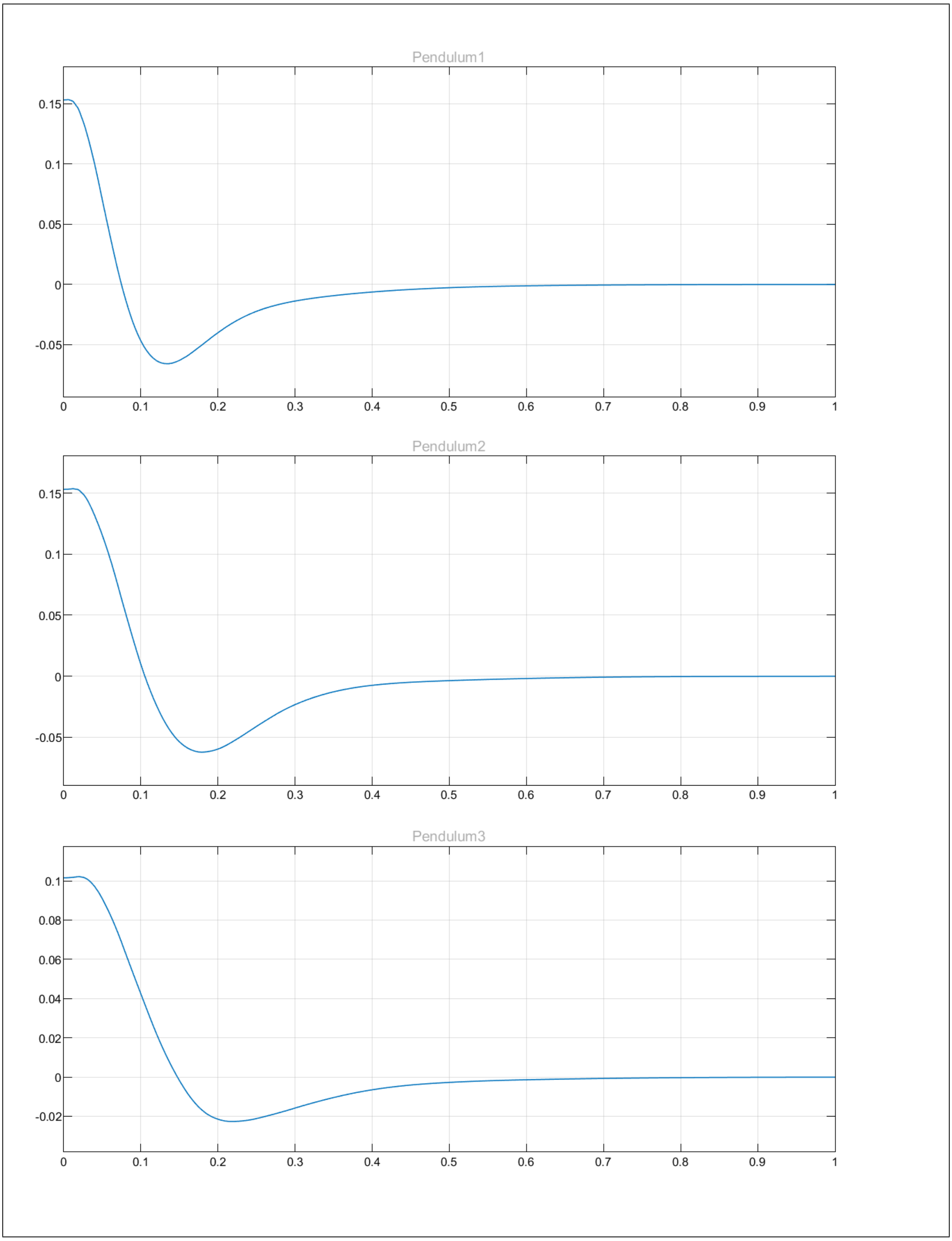


Figure 2: Pendulum angles

The Figure 2 shows that all three pendulums are stabilized. Their settling times are different between these three pendulums, ranging from 0.477, 0.515, 0.546, in contrast to pendulum1, pendulum2, pendulum3, respectively.

Control performance difference: different settling time and oscillating amplitude. Shorter pendulum has shorter settling time and bigger oscillation.

Task 4

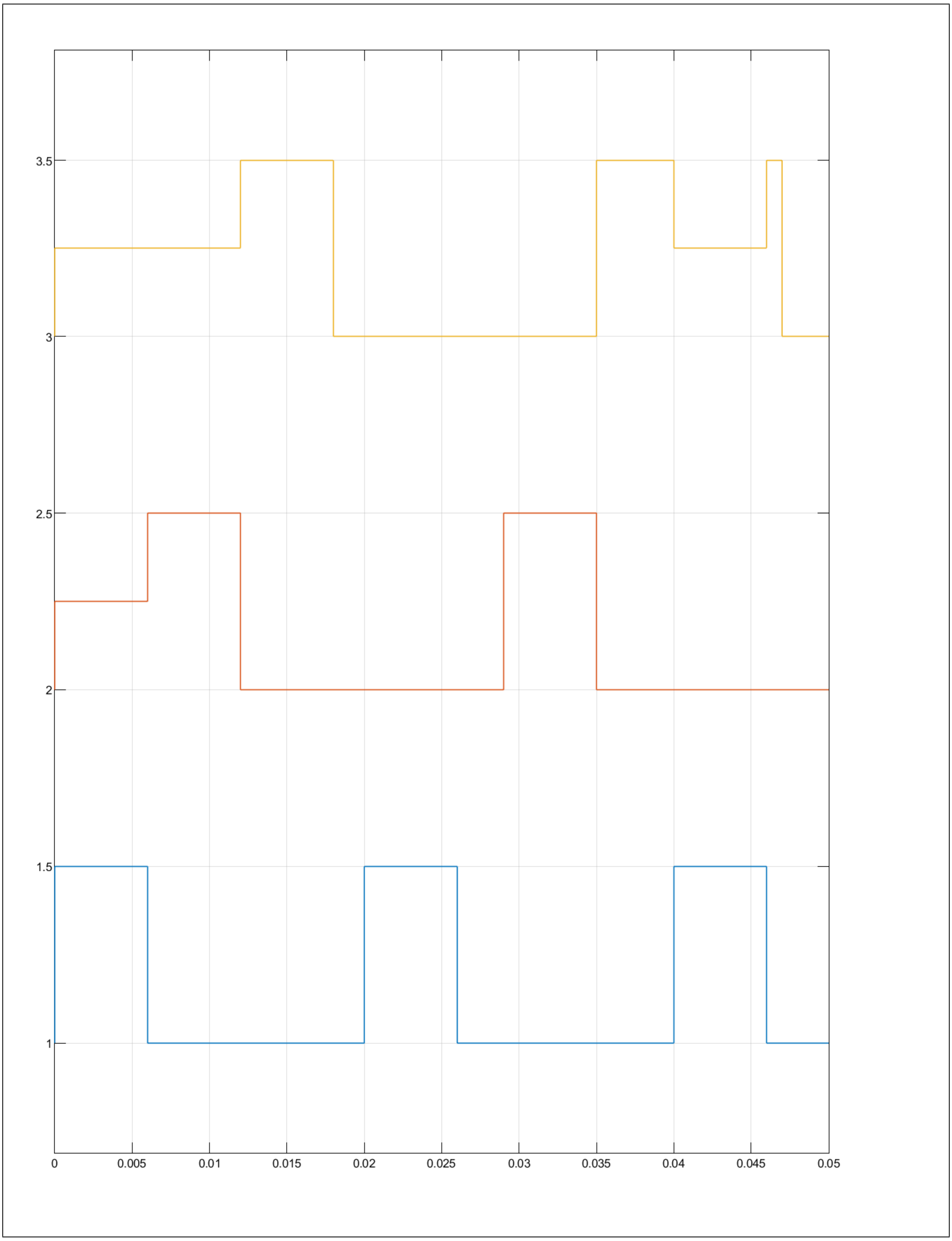


Figure 3: MATLAB RM scheduling

blue curve in the bottom for task1, red for task2, yellow for task3 This figure shows exactly the same result above. The results agree with the analysis. All the tasks get enough time to meet their deadline.

Task 5

Q2

Based on the RM Schedulability, the utilization factor

$$U = \sum_{i=1}^{3} \frac{C_i}{T_i} = \frac{10}{20} + \frac{10}{29} + \frac{10}{35} \approx 1.131 \geq 1 \tag{4}$$

Since the Utilization factor exceeds 1, RM schedule is not available now. The corresponding manual schedule is given below:

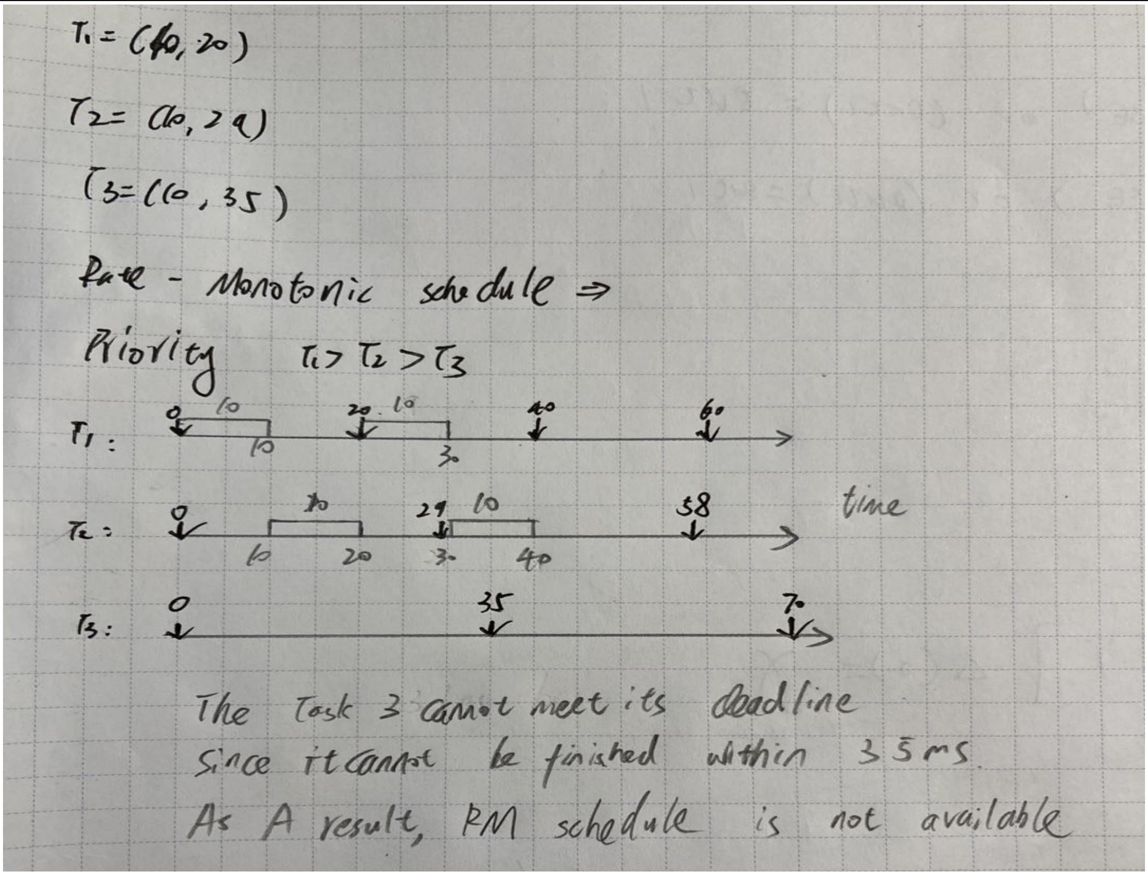


Figure 4: manual RM scheduling for 10ms execution time

Q3

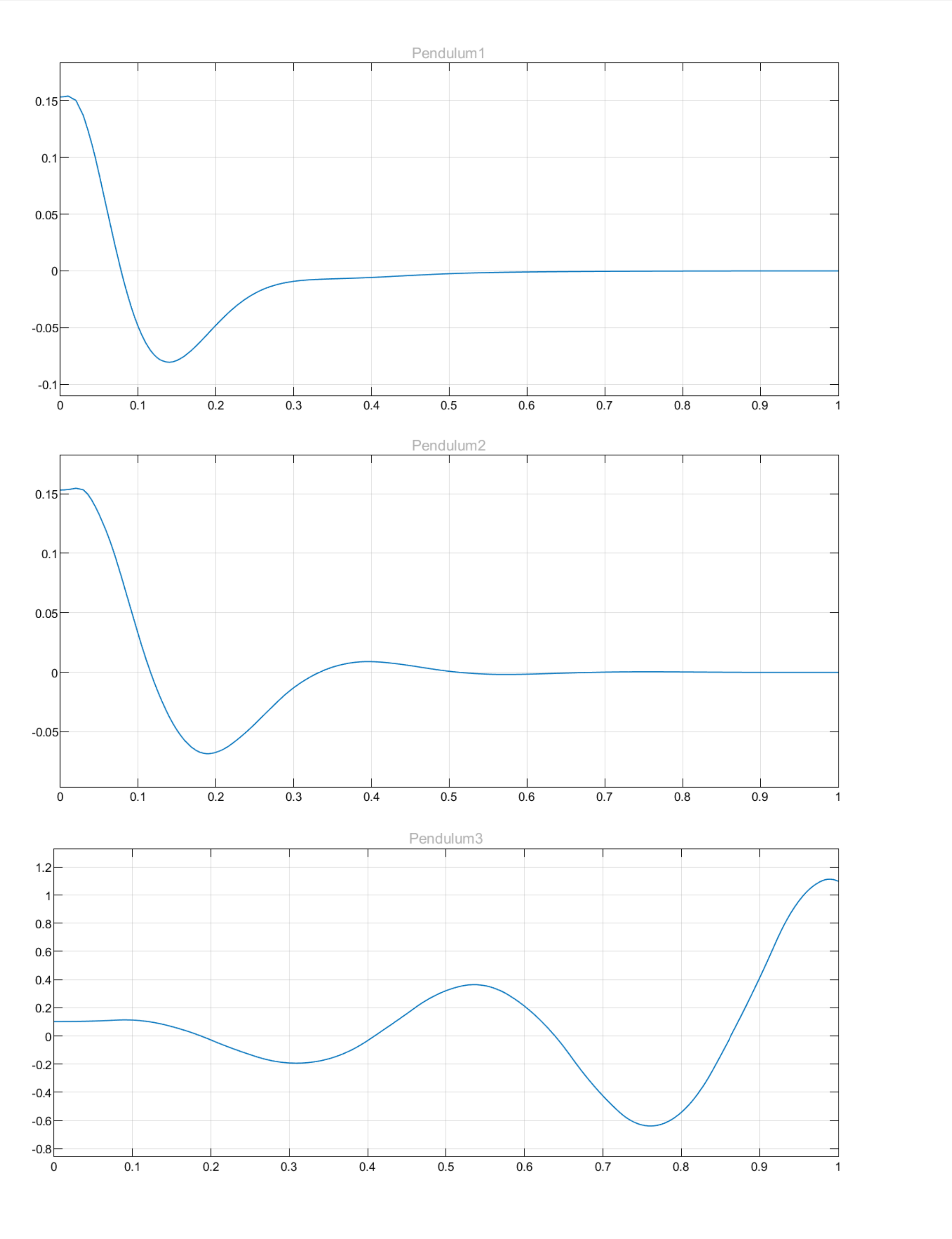


Figure 5: pendulum angles for 10ms execution time

As we can see in the figure, the pendulum3 is diverge, and it cannot be stabilized. However, both the pendulum1 and pendulum2 are stabilized with different settle time, where pendulum1 is 0.483 and pendulum2 is 0.476, almost the same.

Q4

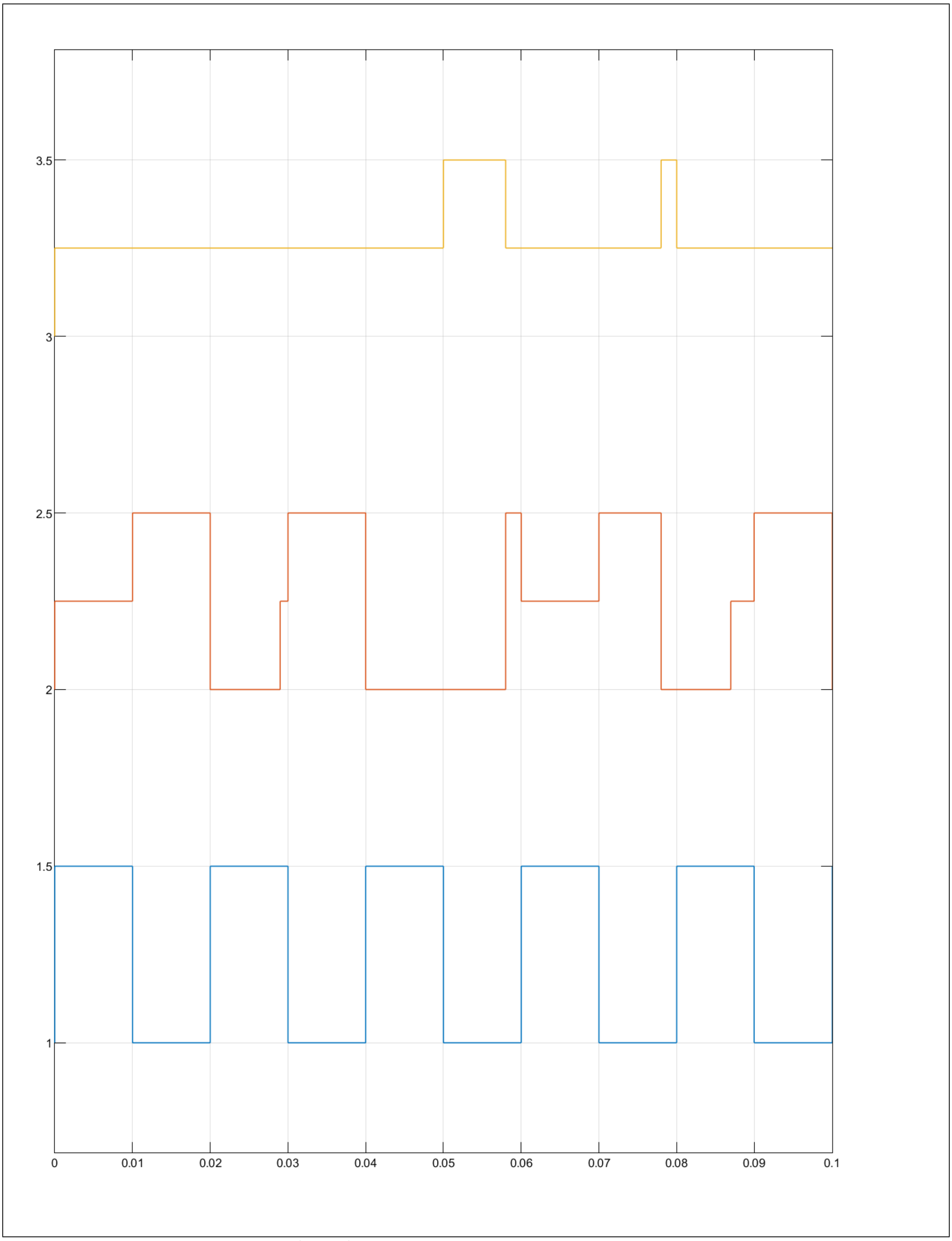


Figure 6: MATLAB RM scheduling for 10ms execution time

blue curve in the bottom for task1, red for task2, yellow for task3

Both the manual and MATLAB figures behave the same. The results agree with the analysis. However, in this case, pendulum3 cannot meet its deadline in the RM scheduling for the utilization factor exceeds 1.

Task 6

Earliest deadline first (EDF) scheduling algorithm assigns dynamic priorities to the tasks based on their absolute deadlines: Execute task with shortest time to deadline d_k.

Advantages:

Works for aperiodic tasks.

Analysable: For RM, $n\left(2^{\frac{1}{n}}-1\right)$ is used for RM scheduling, which may have valid schedulability after calculation by drawing task schedule. However, for the EDF ,once the Utilization factor bound is 1, the schedulability can be determined.

Disadvantages:

Complexity: Priorities are set dynamically, which requires higher computation resources and advanced hardware structure to monitoring and calculating which task has the shortest time to deadline at the moment.

Task 7

the utilization factor

$$U=\sum_{i=1}^{3}\frac{C_i}{T_i}=\frac{6}{20}+\frac{6}{29}+\frac{6}{35}\approx 0.678\leq 1 \tag{5}$$

As a result, tasks are schedulable with EDF. The corresponding manual schedule is given below:

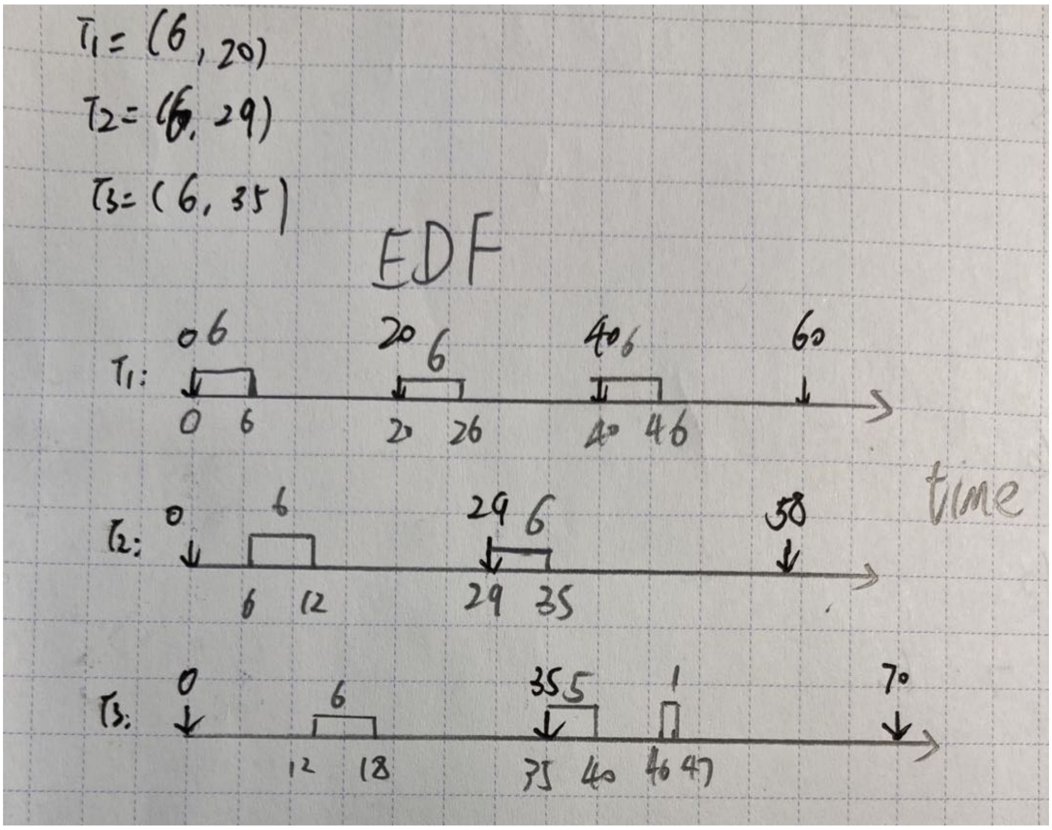


Figure 7: manual RM scheduling for 6ms execution time

Task 8

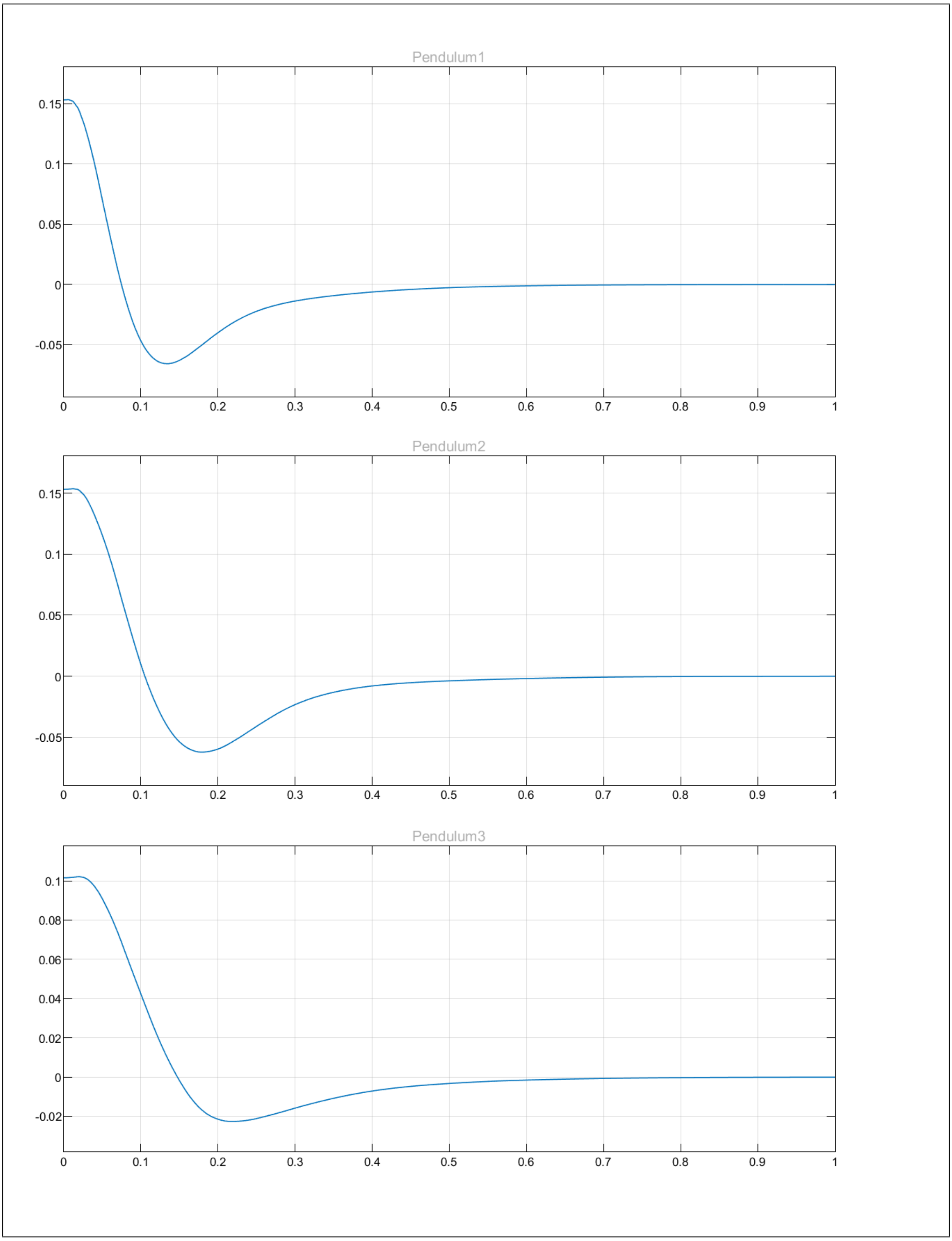


Figure 8: manual RM scheduling for 6ms execution time

The Figure shows that all three pendulums are stabilized. Their settling times are different between these three pendulums, ranging from 0.482, 0.525, 0.566, in contrast to pendulum1, pendulum2, pendulum3, respectively.

Task 9

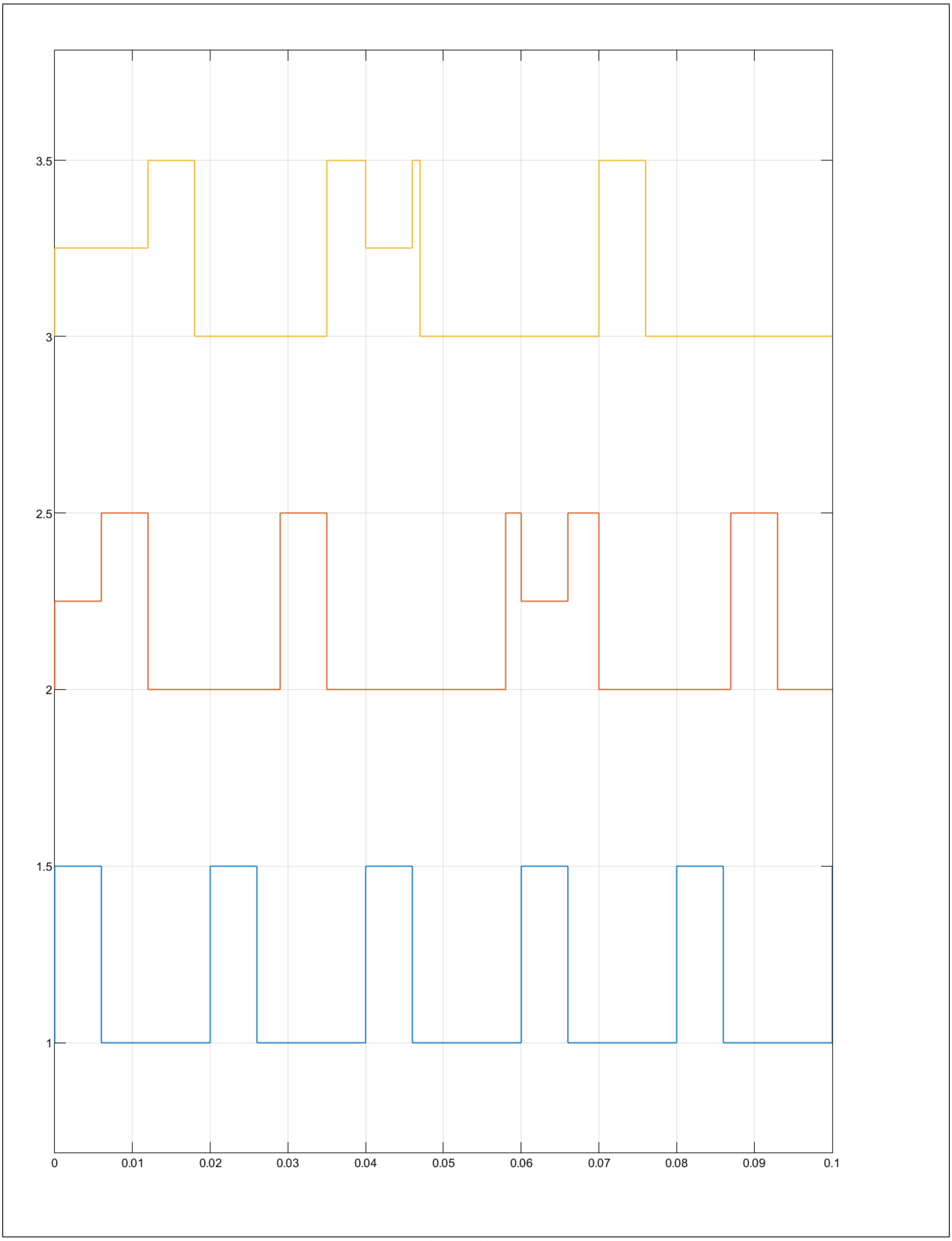


Figure 9: MATLAB RM scheduling

blue curve in the bottom for task1, red for task2, yellow for task3

This figure shows exactly the same result above. The results agree with the analysis. All the tasks get enough time to meet their deadline.

Task 10

Q7

Based on the RM Schedulability, the utilization factor

$$U = \sum_{i=1}^{3} \frac{C_i}{T_i} = \frac{10}{20} + \frac{10}{29} + \frac{10}{35} \approx 1.131 \geq 1 \tag{6}$$

Since the Utilization factor exceeds 1, EDF schedule is not available now. The corresponding manual schedule is given below:

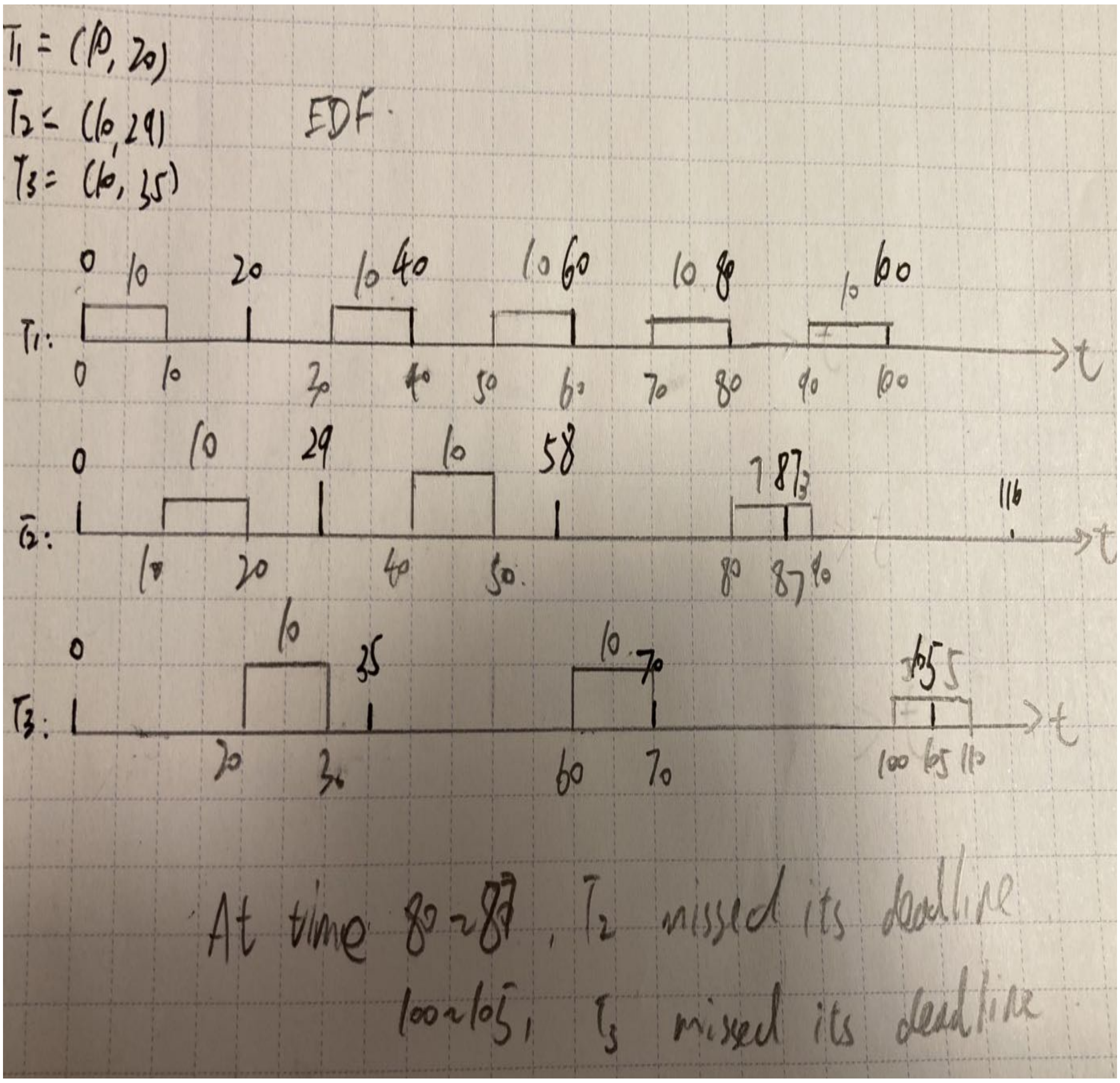


Figure 10: manual EDF scheduling for 10ms execution time

Q8

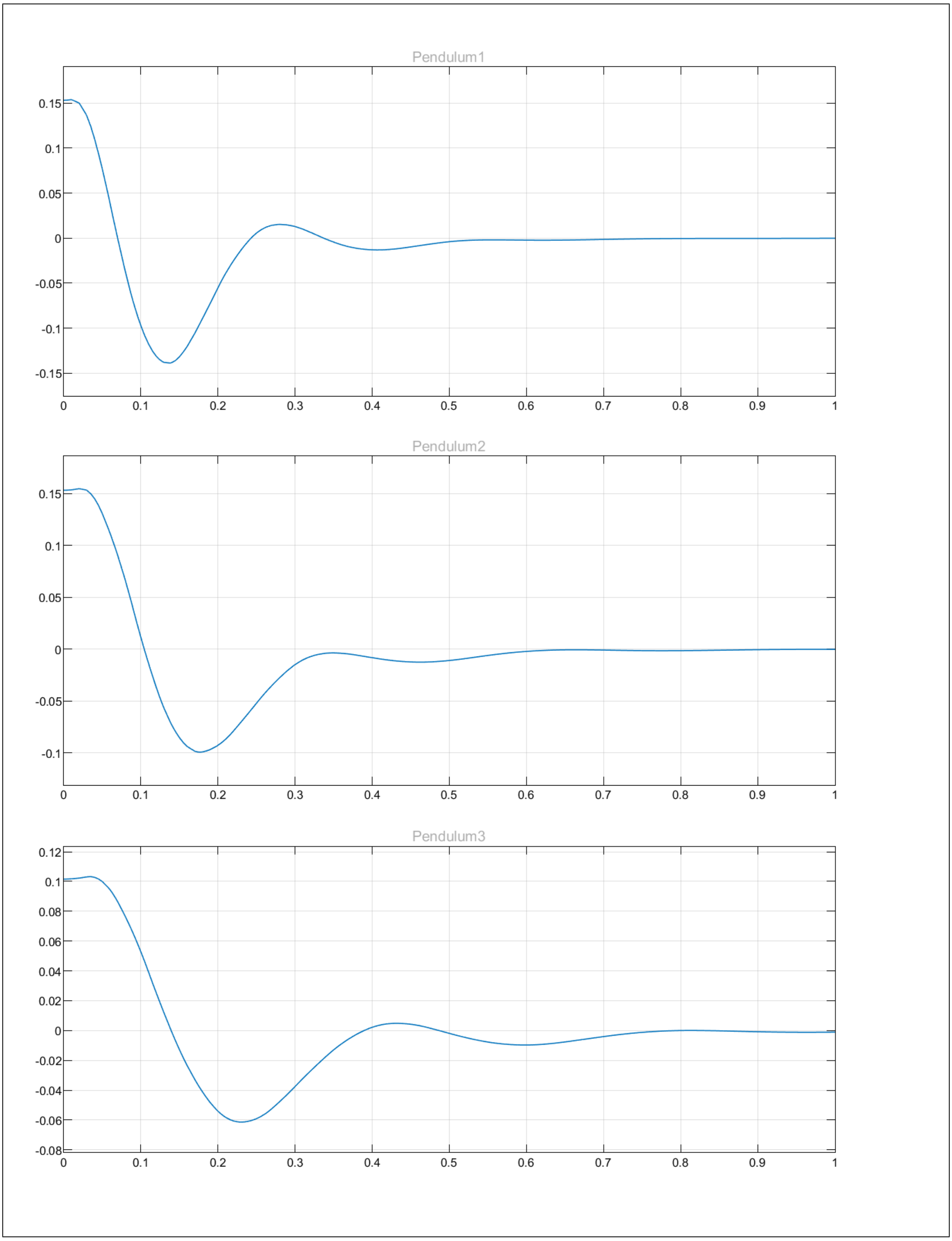


Figure 11: pendulum angles for 10ms execution time

As we can see in the figure, all of the pendulums are stabilized. Their settling times are different between these three pendulums, ranging from 0.521, 0.596, 0.784, in terms of pendulum1, pendulum2, pendulum3, respectively. However, the steady state error is larger.

Q9

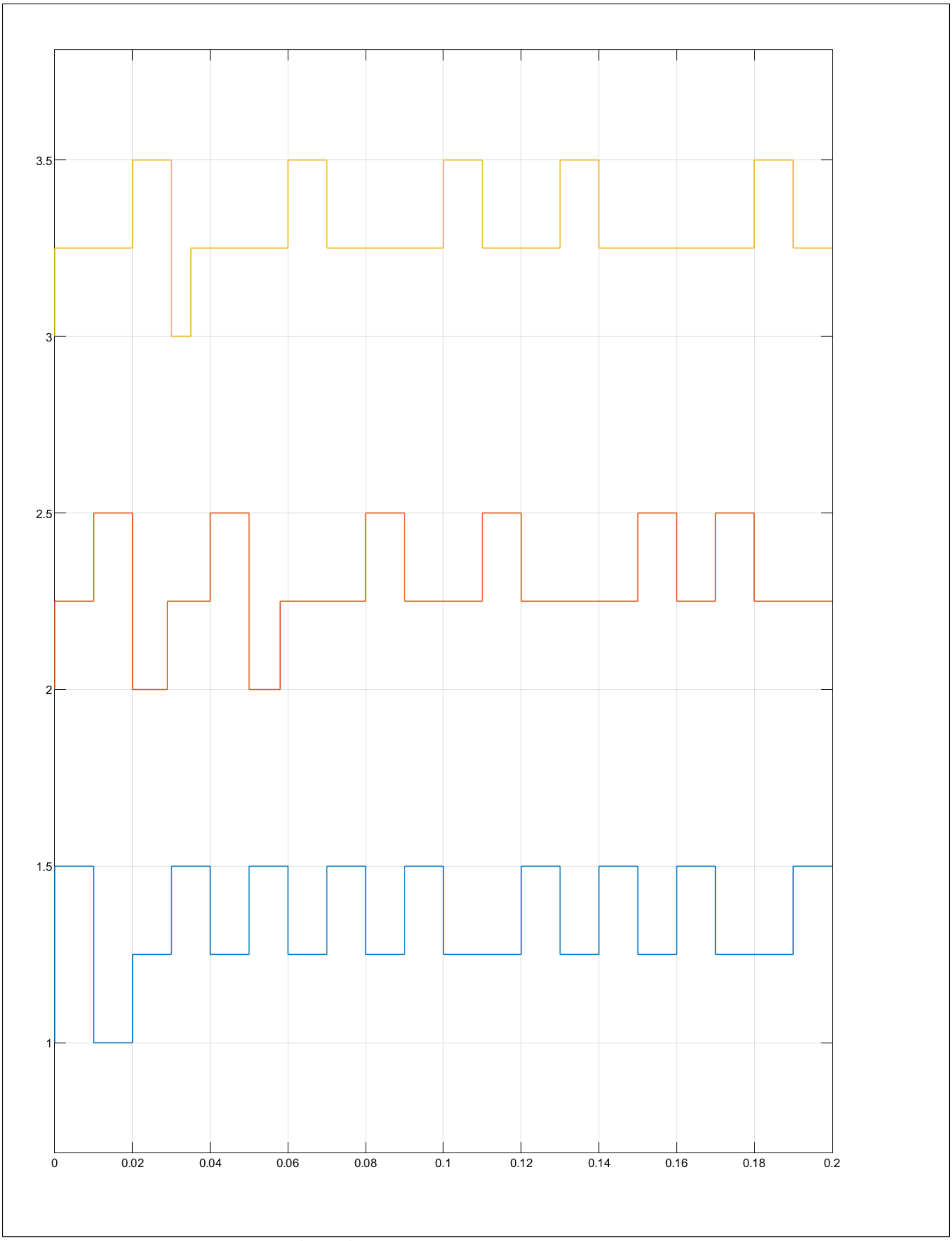


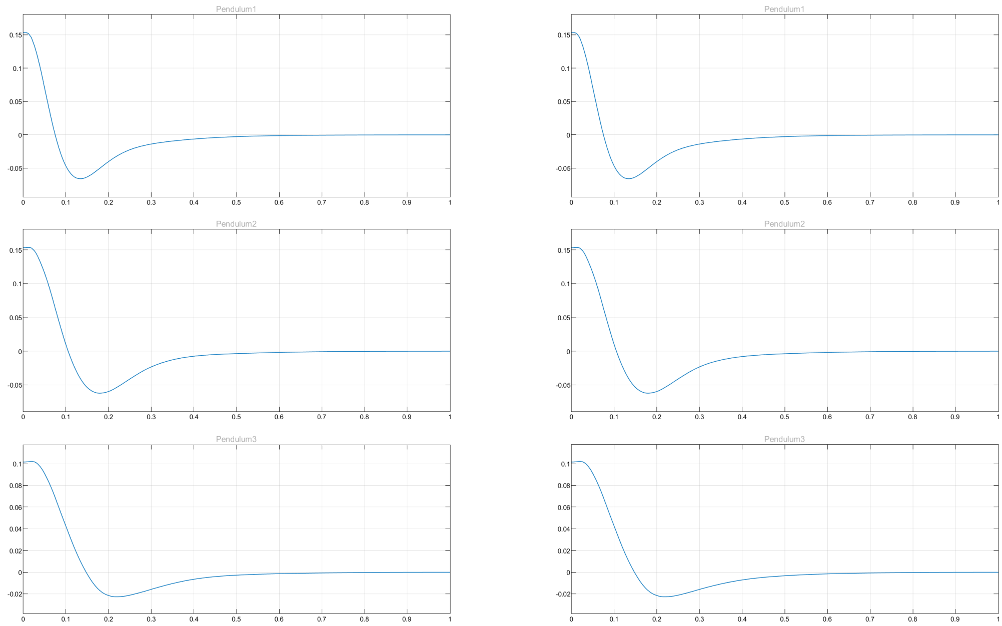
Figure 12: MATLAB EDF scheduling for 10ms execution time

blue curve in the bottom for task1, red for task2, yellow for task3

Both the manual and MATLAB figures behave the same. The results agree with the analysis. However, in this case, pendulum2 and pendulum3 cannot meet its deadline in the EDF scheduling for the utilization factor exceeds 1. Based on the figure, the pendulum2 should finish its work before 0.087s, but it is finished at 0.090s. As for the pendulum3, it should finish its work before 0.105s. But it actually finished at 0.110s. As a result, these tasks can't be scheduled.

Task 11

Yes, EDF performs better than RM. 4 sets of graphs are given below in terms of T, angle, schedule for comparison between RM and EDF. Especially, figure 15, when T = 10ms, EDF scheduling performs better in stability, since RM's pendulum3 is unstable at the same time.



[RM] [EDF]

Figure 13: angle, T=6ms

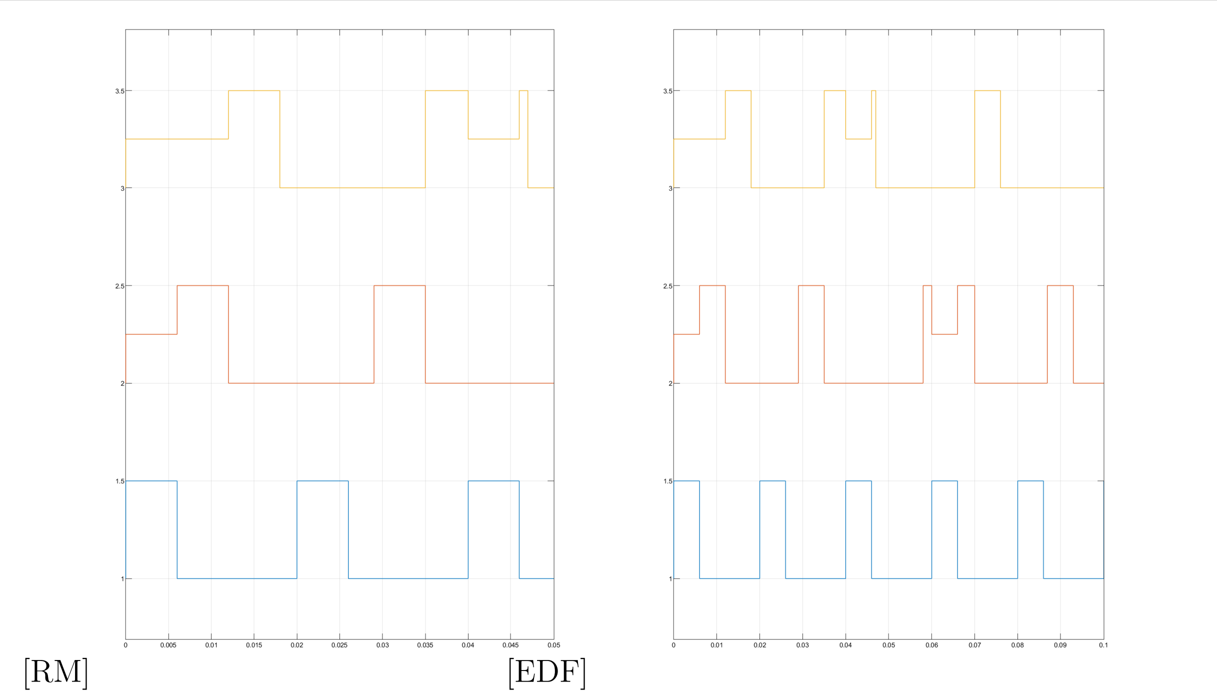


Figure 14: schedule, T=6ms

blue curve in the bottom for task1, red for task2, yellow for task3

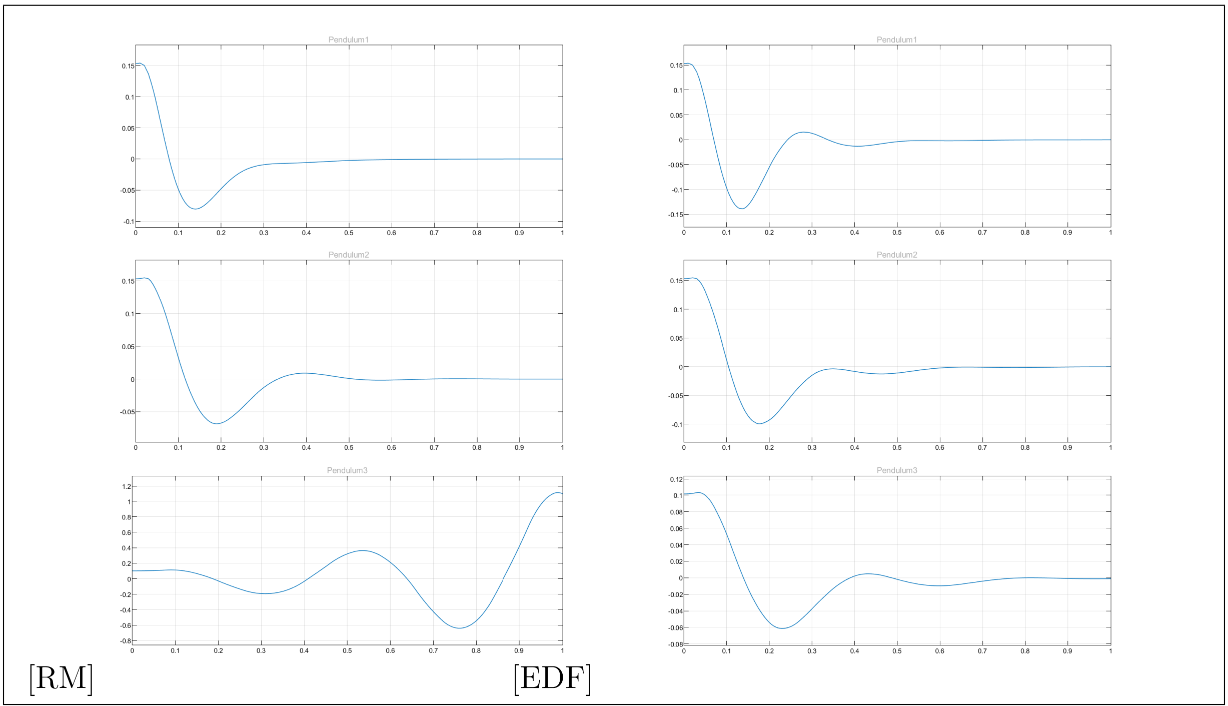


Figure 15: angle, T=10ms

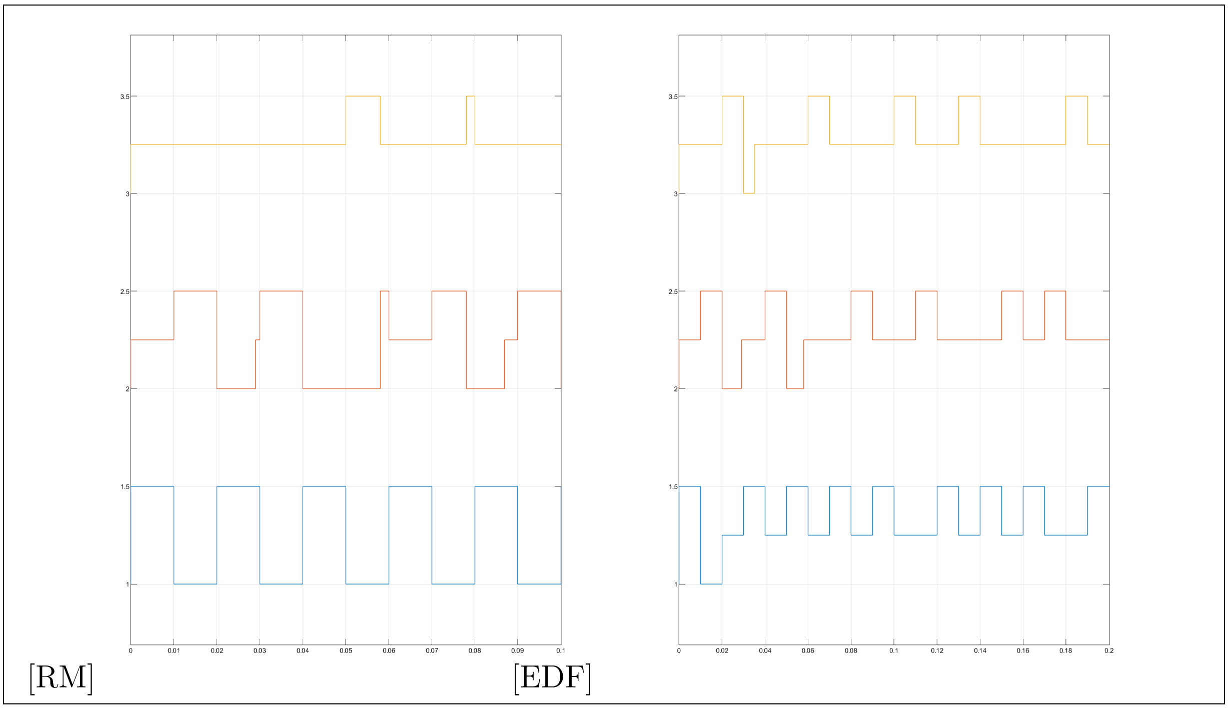


Figure 16: schedule, T=10ms

blue curve in the bottom for task1, red for task2, yellow for task3

Task 12

when delay $\tau \leq h$:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t-\tau), t \in [kh+\tau, (k+1)h+\tau) \\ u(kh) &= -Lx(kh) \end{aligned} \tag{7}$$

originated from:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \\ u(kh) &= -Lx(kh), \quad k = 0, 1, 2, \ldots \end{aligned} \tag{8}$$

Then, for discrete system:

$$\begin{aligned} x(kh+h) &= e^{Ah}x(kh) + \Gamma_0 u(kh) + \Gamma_1 u(kh-h) \\ \Gamma_0 &= \int_0^{h-\tau} e^{As} ds B \\ \Gamma_1 &= e^{A(h-\tau)} \int_0^{\tau} e^{As} ds B \end{aligned} \tag{9}$$

A=0,B=1:

$$\begin{aligned} x(kh+h) &= x(kh) - (h-\tau)Lx(kh) + \tau u(kh-h) \\ y(kh) &= Cx(kh) \end{aligned} \tag{10}$$

tf2ss:

$$
\begin{aligned}
\hat{A} &= \begin{bmatrix} 1-(h-\tau)L & \tau \\ -L & 0 \end{bmatrix} \\
z(kh) &= \begin{bmatrix} x(kh) \\ u(kh-h) \end{bmatrix} \\
z(kh+h) &= \hat{A}z(kh)
\end{aligned} \tag{11}
$$

Task 13

Solve the inequality will make the system stable.

$$
|\lambda_i(\Phi)| < 1 \tag{12}
$$

Quadratic equation gives:

$$
\begin{aligned}
x &= \frac{-b \pm \sqrt{b^2-4ac}}{2a} \\
x_1 + x_2 &= -\frac{b}{a} \\
x_1 x_2 &= \frac{c}{a}
\end{aligned} \tag{13}
$$

Then the eigenvalues will be given by:

$$
|\lambda I - \hat{A}| = \lambda^2 - (1 - hL + \tau L)\lambda + \tau L = 0 \tag{14}
$$

Two eigenvalues are:

$$
\lambda_i = \frac{1}{2}\left[(1-hL+\tau L) \pm \sqrt{(1-hL+\tau L)^2 - 4\tau L}\right], i = 1, 2 \tag{15}
$$

If they are real number, then solve the inequality:

$$
\begin{aligned}
\frac{1}{2}\left[(1-hL+\tau L) + \sqrt{(1-hL+\tau L)^2 - 4\tau L}\right] &< 1 \\
\frac{1}{2}\left[(1-hL+\tau L) - \sqrt{(1-hL+\tau L)^2 - 4\tau L}\right] &> -1 \\
|\lambda_i(\Phi)| &< 1
\end{aligned} \tag{16}
$$

the result will be:

$$
\frac{\tau}{h} > \frac{1}{2} - \frac{1}{Lh} \tag{17}
$$

However, if the eigenvalues have imaginary part, then the inequality will be:

$$
|\lambda_1 \lambda_2| = |\tau L| < 1 \tag{18}
$$

$$
\frac{\tau}{h} < \frac{1}{Lh} \tag{19}
$$

Since it is known that $0 < \tau < h$, it can be concluded that the system is stable when:

$$
\max\left\{0, \frac{1}{2} - \frac{1}{Lh}\right\} < \frac{\tau}{h} < \min\left\{1, \frac{1}{Lh}\right\} \tag{20}
$$

Task 14

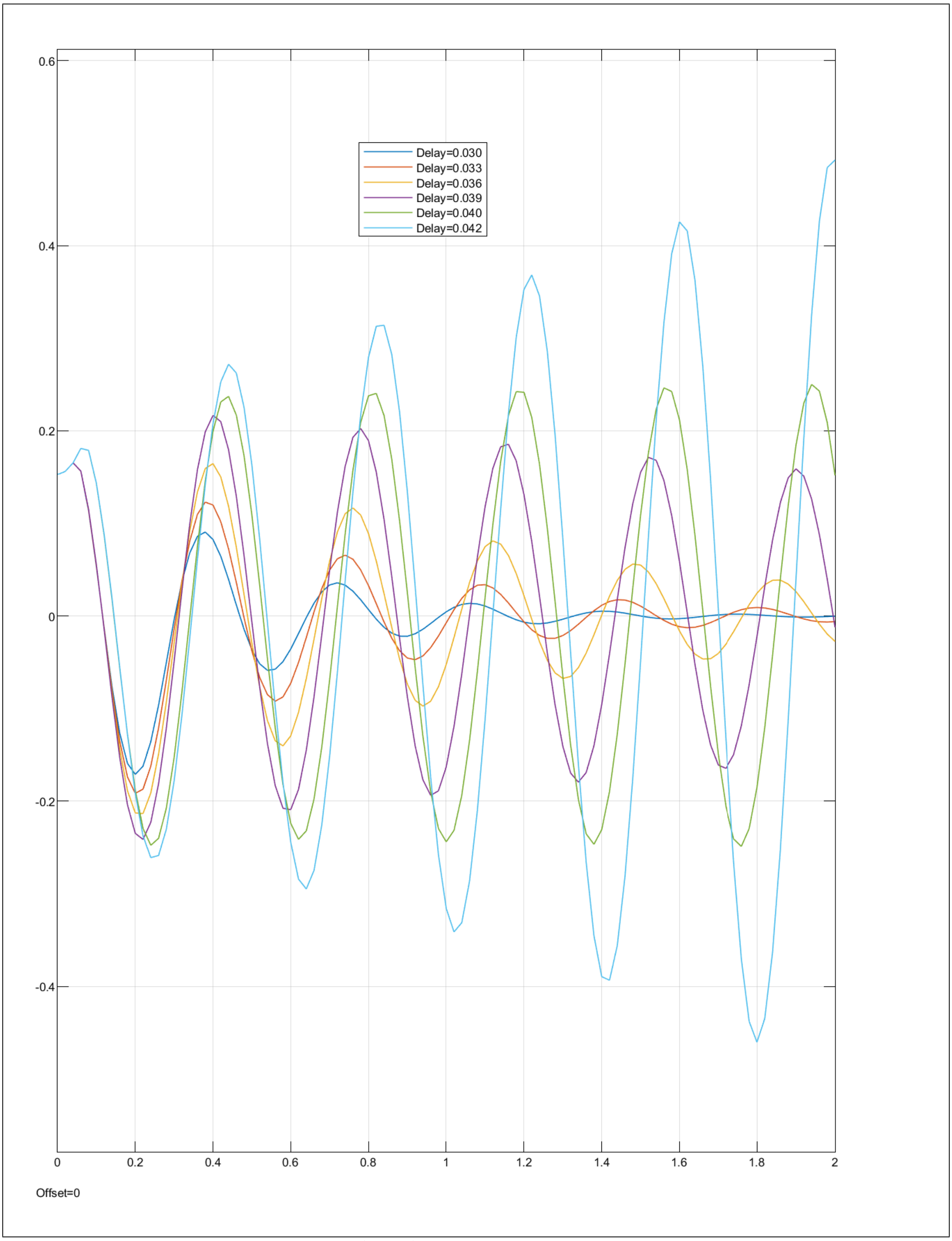


Figure 17: MATLAB simulation of different delays

As the figure shows, if the delay is greater than 0.04, the system becomes unstable, for delay = 0.04 brings a constant amplitude oscillation, which is also unstable. However, if delay is less than 0.039, then the system becomes stable.

Task 15

$\mathcal{T}_1 = (S_1, S_1^0, \Sigma_1, \rightarrow_1, AP_1, L_1)$ where:

- $S_1 = \{s_0, s_1, s_2\}$
- $S_1^0 = \{s_0\}$
- $\Sigma_1 = \{a, b, c\}$
- $\rightarrow_1 = \{(s_0, a, s_1), (s_1, c, s_0), (s_2, b, s_0), (s_1, a, s_2)\}$
- $AP_1 = S_1 = \{s_0, s_1, s_2\}$
- $L_1(s) = \{s\} =$ identity map

$\mathcal{T}_2 = (S_2, S_2^0, \Sigma_2, \rightarrow_2, AP_2, L_2)$ where:

- $S_2 = \{t_0, t_1, t_2\}$
- $S_2^0 = \{t_0\}$
- $\Sigma_2 = \{a, b, c\}$
- $\rightarrow_2 = \{(t_0, a, t_1), (t_1, c, t_2), (t_2, c, t_0), (t_0, b, t_2)\}$
- $AP_2 = S_2 = \{t_0, t_1, t_2\}$
- $L_2(t) = \{t\} =$ identity map

Task 16

$\mathcal{T}_{\text{int}} = (S_{\text{int}}, S_{\text{int}}^0, \Sigma_{\text{int}}, \rightarrow_{\text{int}}, AP_{\text{int}}, L_{\text{int}})$ where:

- $S_{\text{int}} = S_1 \times S_2 = \{s_0, s_1, s_2\} \times \{t_0, t_1, t_2\} = \{(s_i, t_j) \quad i, j \in \{0, 1, 2\}\}$
- $S_{\text{int}}^0 = S_1^0 \times S_2^0 = \{s_0\} \times \{t_0\} = \{(s_0, t_0)\}$
- $\Sigma_{\text{int}} = \Sigma_1 \cup \Sigma_2 = \{a, b, c\}$
- $\rightarrow_{\text{int}}$:

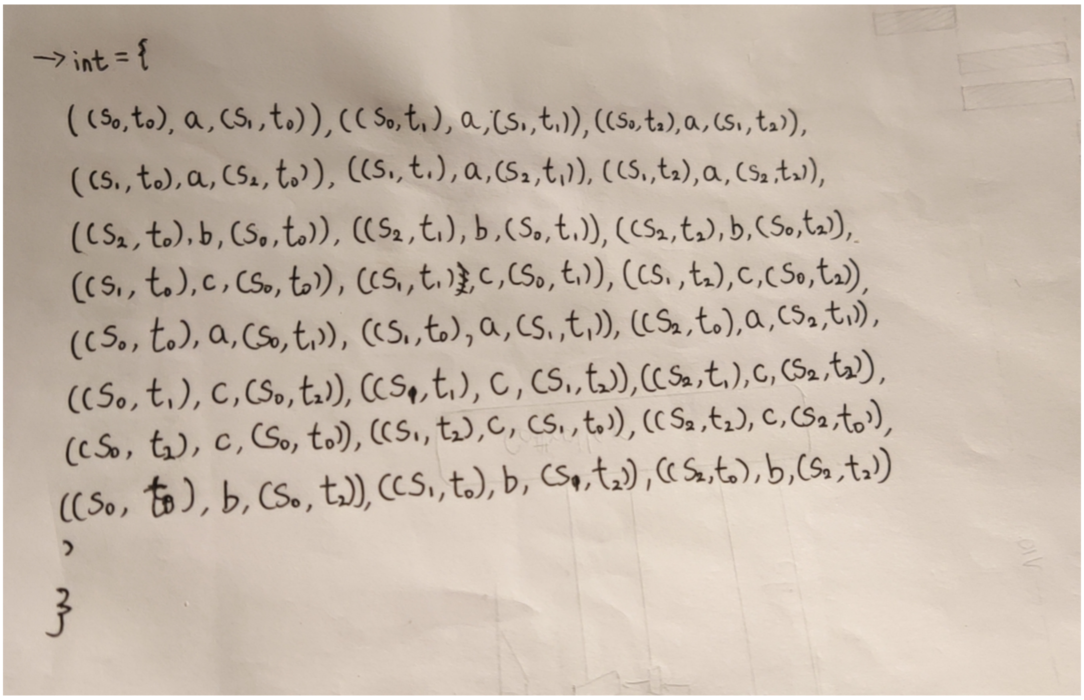


Figure 18: Define the Product Interleaving Transition System

- $AP_{\text{int}} = AP_1 \cup AP_2 = \{s_0, s_1, s_2, t_0, t_1, t_2\}$
- $L_{\text{int}}(s,t) = L_1(s) \cup L_2(t) = \{s\} \cup \{t\} = \{s,t\}$ still some kind of identity map

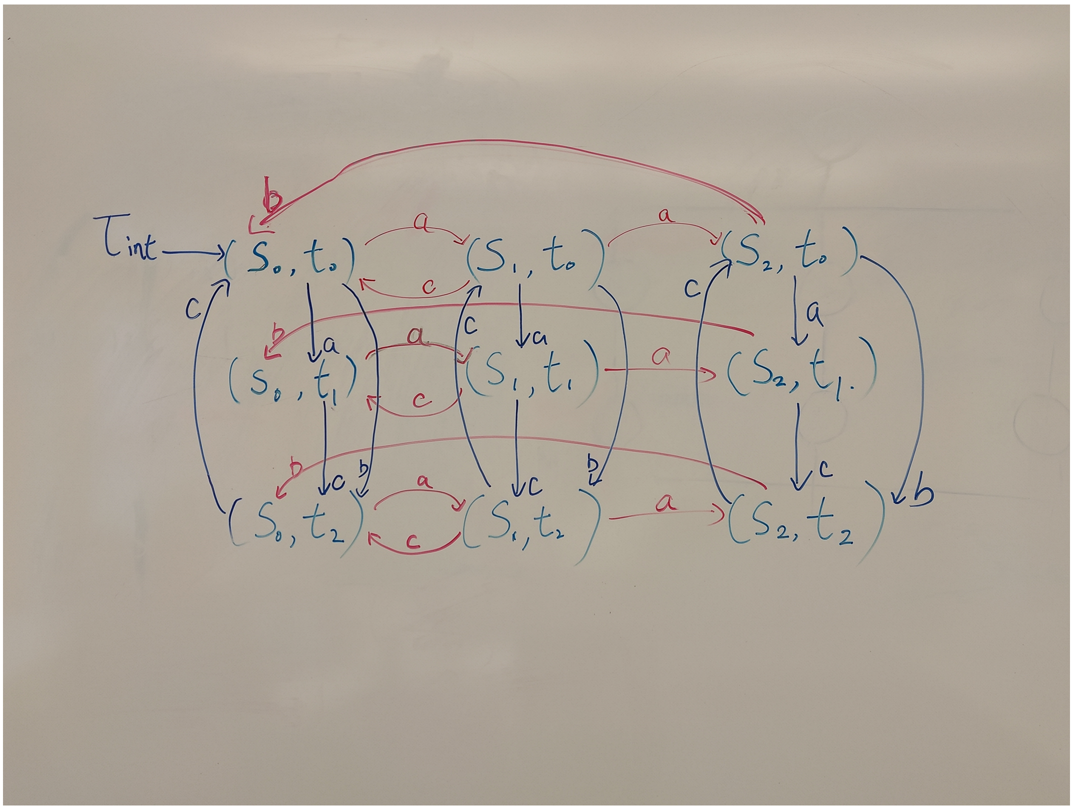
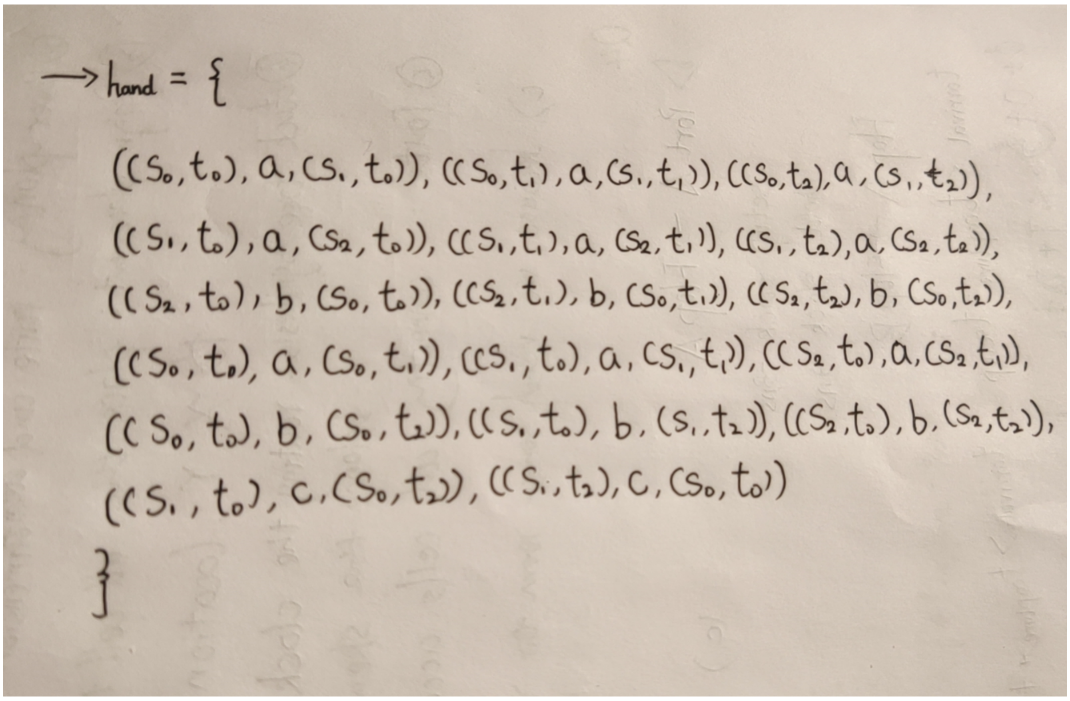


Figure 19: Draw the Product Interleaving Transition System

Task 17

$\mathcal{T}_{\text{hand}} = (S_{\text{hand}}, S^0_{\text{hand}}, \Sigma_{\text{hand}}, \rightarrow_{\text{hand}}, AP_{\text{hand}}, L_{\text{hand}})$ where:

- $S_{\text{hand}} = S_1 \times S_2 = \{s_0, s_1, s_2\} \times \{t_0, t_1, t_2\} = \{(s_i, t_j) \quad i, j \in \{0, 1, 2\}\}$
- $S^0_{\text{hand}} = S^0_1 \times S^0_2 = \{s_0\} \times \{t_0\} = \{(s_0, t_0)\}$
- $\Sigma_{\text{hand}} = \Sigma_1 \cup \Sigma_2 = \{a, b, c\}$
- $H = \{c\} \subseteq \Sigma_1 \cap \Sigma_2 = \{a, b, c\}$
- $\rightarrow_{\text{hand}}$:



$\rightarrow_{hand} = \{$

$((s_0,t_0),a,(s_1,t_0)), ((s_0,t_1),a,(s_1,t_1)), ((s_0,t_2),a,(s_1,t_2)),$

$((s_1,t_0),a,(s_2,t_0)), ((s_1,t_1),a,(s_2,t_1)), ((s_1,t_2),a,(s_2,t_2)),$

$((s_2,t_0),b,(s_0,t_0)), ((s_2,t_1),b,(s_0,t_1)), ((s_2,t_2),b,(s_0,t_2)),$

$((s_0,t_0),a,(s_0,t_1)), ((s_1,t_0),a,(s_1,t_1)), ((s_2,t_0),a,(s_2,t_1)),$

$((s_0,t_0),b,(s_0,t_2)), ((s_1,t_0),b,(s_1,t_2)), ((s_2,t_0),b,(s_2,t_2)),$

$((s_1,t_0),c,(s_0,t_2)), ((s_1,t_2),c,(s_0,t_0))$

$\}$

Figure 20: Define the Product Handshaking Transition System

- $AP_{\text{hand}} = AP_1 \cup AP_2 = \{s_0, s_1, s_2, t_0, t_1, t_2\}$
- $L_{\text{hand}}(s, t) = L_1(s) \cup L_2(t) = \{s\} \cup \{t\} = \{s, t\}$ still some kind of identity map

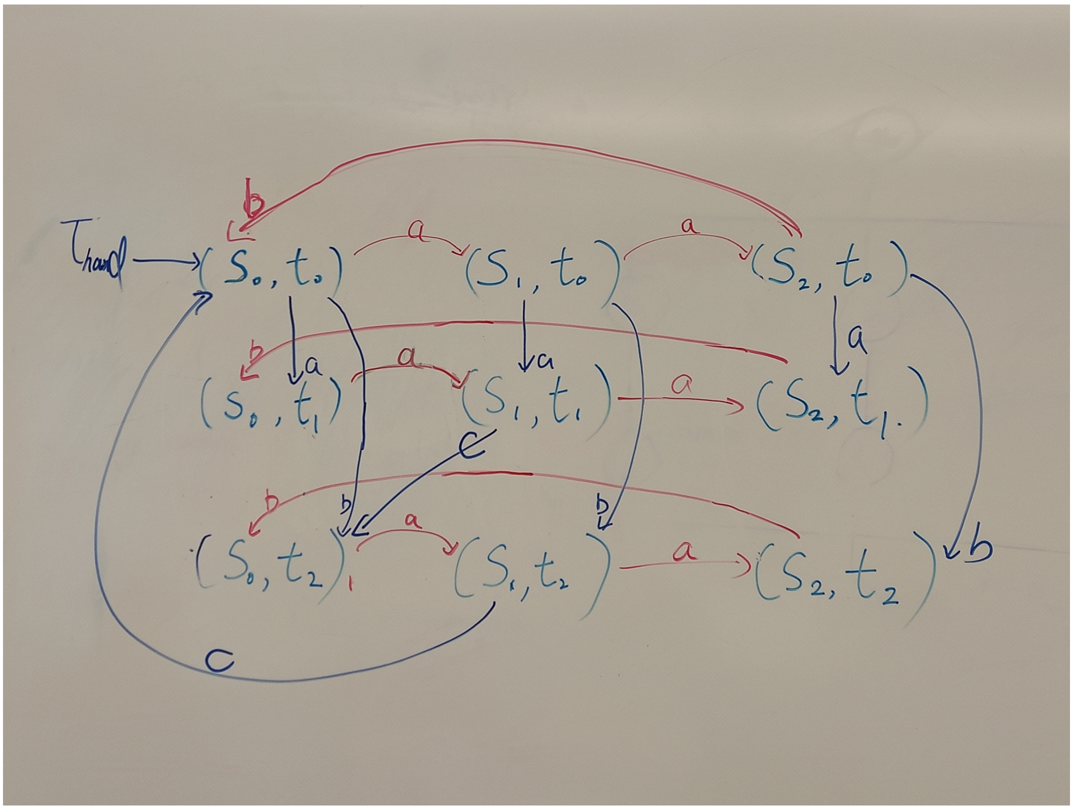


Figure 21: Draw the Product Handshaking Transition System

References

[1] Hassan K Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle river, 3. edition, 2002. ISBN 0-13-067389-7.

[2] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *The Not So Short Introduction to LaTeX 2ε*. Oetiker, OETIKER+PARTNER AG, Aarweg 15, 4600 Olten, Switzerland, 2008. http://www.ctan.org/info/lshort/.

[3] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer, New York, N.Y., 1999. ISBN 0-387-98513-1.