Solutions to the exam in Real-Time Systems 180531

These solutions are available on WWW:
http://www.control.lth.se/course/FRTN01/

1.

a. ZOH-sampling with $h = 1$ gives

$$
\begin{aligned}
x(k+1) &= ex(k) + 2(e-1)u(k) \\
y(k) &= x(k)
\end{aligned}
$$

The corresponding pulse transfer function $H(z)$ then becomes

$$H(z) = C(zI - \Phi)^{-1}\Gamma = \frac{2(e-1)}{z-e}$$

with pole $z_p = e$ and no zeros.

b. The closed loop system $H_{cl}(z)$ becomes

$$H_{cl}(z) = \frac{K\dfrac{2(e-1)}{z-e}}{1 + K\dfrac{2(e-1)}{z-e}} = \frac{2K(e-1)}{z - e + 2K(e-1)}$$

with the closed loop system pole in $z_p = e - 2K(e-1)$. Thus for stability we require

$$-1 \le e - 2K(e-1) \le 1$$

with the solution

$$\frac{1}{2} \le K \le \frac{1+e}{2(e-1)}$$

c. For $K = 1$ and the closed loop system $Y(z) = H_{cl}(z)R(z)$, we get

$$H_{cl}(z) = \frac{2(e-1)}{z+e-2}$$

As the closed loop system was proven to be stable for $K = 1$ in the previous subproblem, we can use the finite value theorem or simply look at the static gain of $H_{cl}(z)$ to find what value y will converge to if the step had been of size 1. When the size of the step is 2 the answer is simply twice that amount, i.e., the answer is

$$y(\infty) = 4$$

2.

a.

$$\sum_i \frac{C_i}{D_i} = 0.9333$$

So it is schedulable under EDF.

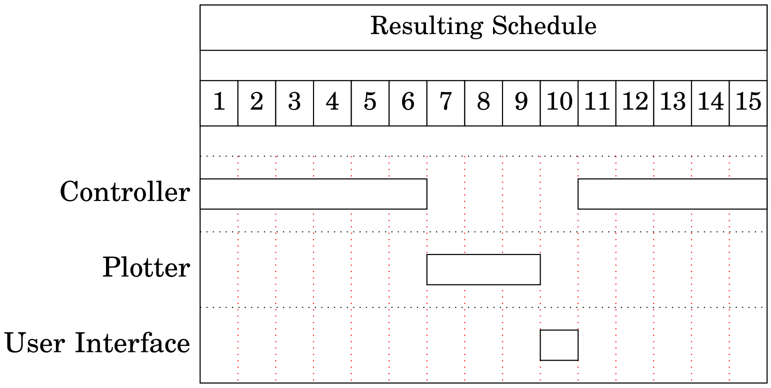

Figure 1 Schedule for the first 15 time units for Problem **??**.

b. The taskset will not meet the deadlines under this policy. Specifically, either the Plotter or the User Interface will miss their deadline already at the first activation. Figure 1 shows the first 15 time units using such a policy. As can be seen, in the first 10 units, 6 are given to the controller, then the plotter and the user interface have 4 units before the controller gets activated again. However, they should execute for 5 time units before their deadline at time 15. Suppose that the plotter is picked first and the user interface is picked second (as shown in Figure 1). The user interface is preempted at time 10 when the controller is ready to execute again. Since the controller executes until time 16, there is no possibility for the user interface to have the additional time unit it would have needed to complete its work before the deadline. Alternatively, if the user interface is selected first, it is the plotter task which cannot complete before its deadline.

3.

a. Converting to the fixed point *Q 4.3* format, the coefficients of $C(z)$ become:

$$\begin{aligned}
-0.38_{4,3} &: \text{round}\left(-0.38 \cdot 2^3\right) = -3 \\
0.0345_{4,3} &: \text{round}\left(0.0345 \cdot 2^3\right) = 0 \\
-1.63_{4,3} &: \text{round}\left(1.63 \cdot 2^3\right) = -13 \\
0.6552_{4,3} &: \text{round}\left(0.6552 \cdot 2^3\right) = 5 \\
1_{4,3} &: \text{round}\left(1 \cdot 2^3\right) = 8
\end{aligned}$$

b. The controller is represented in fixed point *Q 4.3* format as

$$C_{4,3}(z) = \frac{8z^2 - 3z}{8z^2 - 13z - 5}.$$

Dividing all coefficients by 2^3, we get the controller $\hat{C}(z)$ with rounded coefficients as

$$\hat{C}(z) = \frac{z^2 - 0.375z}{z^2 - 1.625z - 0.625},$$

with the poles $z_p^1 = 0.625$, $z_p^2 = 1.00$ and zeros $z_z^1 = 0.00$, $z_z^2 = 0.375$.

c. The control law can be realized using the series or parallel forms instead, with much better numerical properties.

4.

a. The problem is that several threads are sharing the common window resource for printing without proper synchronization. Since the threads don't synchronize on a common object (each thread synchronizes on `this`, i.e. itself), they will always consider the window resource as available for printing.

Another possible cause for strange printouts is if the `get`-method of the `AnalogIn`-class is not thread-safe.

b. A solution to the synchronization problem is to make sure that all threads synchronize on a common object. You can do this by e.g.:

- Create a dummy class and pass a reference to the same instance of this dummy class to all instances of `PrintMeasurement`.
- Declare a static object within `PrintMeasurement` and synchronize on the static object.
- Synchronize directly on the class lock of `PrintMeasurement`, i.e. use `synchronized(this.getClass())`.

5.

a. The rotation frequency of the drill f_d is between $4000/60 = 66.7$ Hz and $5000/60 = 80.3$ Hz, while the sampling frequency is $f_s = 1/0.02 = 50$ Hz. Since the rotation frequency of the drill is above the Nyquist frequency ($f_N = f_s/2 = 25$ Hz), the peak at 20 Hz is most likely due to aliasing of the measured rotational angle of the drill bit.

An anti-aliasing filter should be introduced to solve this problem. The filter could be either a fixed analog filter or a digital filter. The digital filter is more flexible if we expect changes in the sampling period, but must be combined with a fixed analog prefilter.

b. Assuming that the fundamental alias frequency is $f = 20$ Hz, we have:

$$f = |(f_d + f_N)\text{mod}(f_s) - f_N|.$$

There are two possible solutions for $66.7 \leq f_d \leq 80.3$ Hz, given by

$$f_d = f + f_N + f_s - f_N = f + f_s = 20 + 50 = 70\text{Hz},$$

and

$$f_d = -f - f_N + 2f_s + f_n = -f + 2f_s = -20 + 2 \cdot 50 = 80\text{Hz}.$$

Thus the exact frequency of the rotation of the drill was either 70 or 80 Hz.

6.

a. A Bode plot of the filters for α_1 and α_2 respectively are shown in Figure 2. We note that the measurement α_1 is low-pass filtered while α_2 is high-pass filtered. This means that high-frequency noise will be filtered away from α_1 while low-frequency noise will be filtered away from α_2. We should thus pick the accelerometer for α_1 and the gyroscope for α_2.

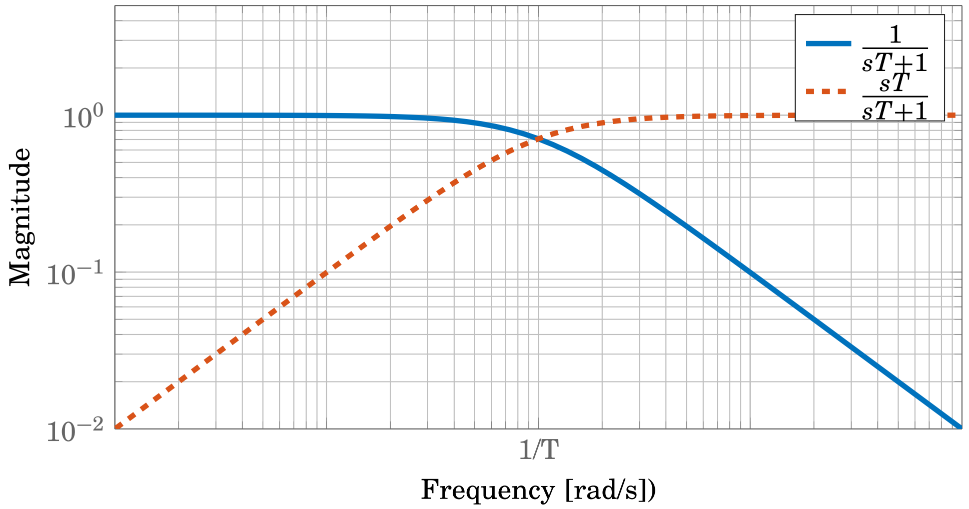

Figure 2 Bode plot of the filters for measurements α_1 (solid) and α_2 (dashed).

b. We perform the forward difference approximation by setting $s = (z-1)/h$ in the filters, which gives

$$\hat{\alpha}(z) = \frac{1}{\frac{z-1}{h}T + 1}\alpha_1(z) + \frac{\frac{z-1}{h}T}{\frac{z-1}{h}T + 1}\alpha_2(z)$$

$$\Leftrightarrow$$

$$\hat{\alpha}(z)(z - 1 + \frac{h}{T}) = \frac{h}{T}\alpha_1(z) + (z-1)\alpha_2(z).$$

Inverse $\mathcal{Z}$-transform and some re-arrangements then give the difference equation

$$\hat{\alpha}(k+1) = (1 - \frac{h}{T})\hat{\alpha}(k) + \frac{h}{T}\alpha_1(k) + \alpha_2(k+1) - \alpha_2(k).$$

c. An implementation which minimizes the latency between reading the measurements and transmitting the estimate is given below (the added code is highlighted in blue):

```
private double alpha_hat = 0, alpha_1 = 0, alpha_2 = 0;
private double h = 0.01, T = 0.33;
private double a = 1 - h/T, b = h/T, temp = 0;
while(true) {
  alpha_1 = readAlpha1();
  alpha_2 = readAlpha2();
  alpha_hat = temp + alpha_2;
  transmitEstimate(alpha_hat);
  temp = a*alpha_hat + b*alpha_1 - alpha_2;
  sleep();
}
```

In this implementation we minimize the latency by pre-computing the coefficients ("a" and "b") and the part that only depends on the previous iteration ("temp").

7.

a. The figure shows a PI controller with reference weighting and anti-windup. a is the reference weighting, used to tune the response to setpoint changes; b is the proportional gain of the controller; c is the integral gain of the controller; d is the anti-windup gain.

b. Suggested solution:

```
// Code executed in the Regul thread

time = getCurrentTime();
i_part = 0.0;

while (true) {
    y = getY();
    r = getRef();

    p_part = b*(a*r - y); // using reference weighting
    v = p_part + i_part;

    if (v > u_max) u = u_max;
    else if (v < u_min) u = u_min;
    else u = v;

    setU(u);

    i_part = i_part + h*c*(r - y) + h*d*(u - v); // using anti-windup

    time = time + h;
    waitUntil(time)
}
```

8.

a. The relationship between continuous-time poles (here denoted p_c) and discrete-time poles (denoted p_d) is

$$p_d = e^{hp_c}$$

or

$$p_c = \frac{\ln p_d}{h}$$

The continuous-time open-loop poles will be -6.9312. The requirement that the closed loop system should be twice as fast means that the continuous-time poles instead should be in -13.8629 and the corresponding discrete-time poles in 0.25.

Using state feedback the closed loop system becomes

$$\begin{aligned} x(k+1) &= (\Phi - \Gamma L)x(k) \\ &= \begin{pmatrix} 1 - l_1 & -(l_2 + 0.25) \\ 1 & 0 \end{pmatrix} x(k) \end{aligned}$$

The characteristic polynomial of the closed loop system becomes

$$z^2 - (1 - l_1)z + (0.25 + l_2)$$

which should be equal to

$$(z - 0.25)^2 = x^2 - 0.5z + 0.0625$$

From this follows that $l_1 = 0.5$ and $l_2 = -0.1875$.

b. The equation for an observer with direct term is

$$\hat{x}(k) = (I - KC)(\Phi\hat{x}(k-1) + \Gamma u(k-1)) + Ky(k)$$

which gives that the reconstruction error is

$$\tilde{x}(k) = (\Phi - KC\Phi)\tilde{x}(k-1)$$

The dynamics for the reconstruction error is

$$\tilde{x}(k) = \begin{pmatrix} 1 - k_1 & -0.25 \\ 1 - k_2 & 0 \end{pmatrix} \tilde{x}(k-1)$$

The characteristic polynomial of the closed loop system becomes

$$z^2 - (1 - k_1)z + 0.25(1 - k_2)$$

which should be equal to

$$(z - 0.2)^2 = z^2 - 0.4z + 0.04$$

from which follows that $k_1 = 0.6$ and $k_2 = 0.84$.

c. The model and feedforward generator is given by

$$\begin{aligned} x_m(k+1) &= \Phi x_m(k) + \Gamma u_{ff}(k) \\ y_m(k) &= C x_m(k) \end{aligned}$$

with

$$u_{ff}(k) = -L_m x_m(k) + L_c u_c(k)$$

Here L_m is obtained from **a.** and L_c from

$$L_c = \frac{1}{C(I - \Phi + \Gamma L_m)^{-1}\Gamma} = 0.5625$$

The block diagram is obtained by combining Figures 30 and 31 in IFAC PB.

9.

a. A possible solution is shown in Figure 3.

b. A possible solution is shown in Figure 4.

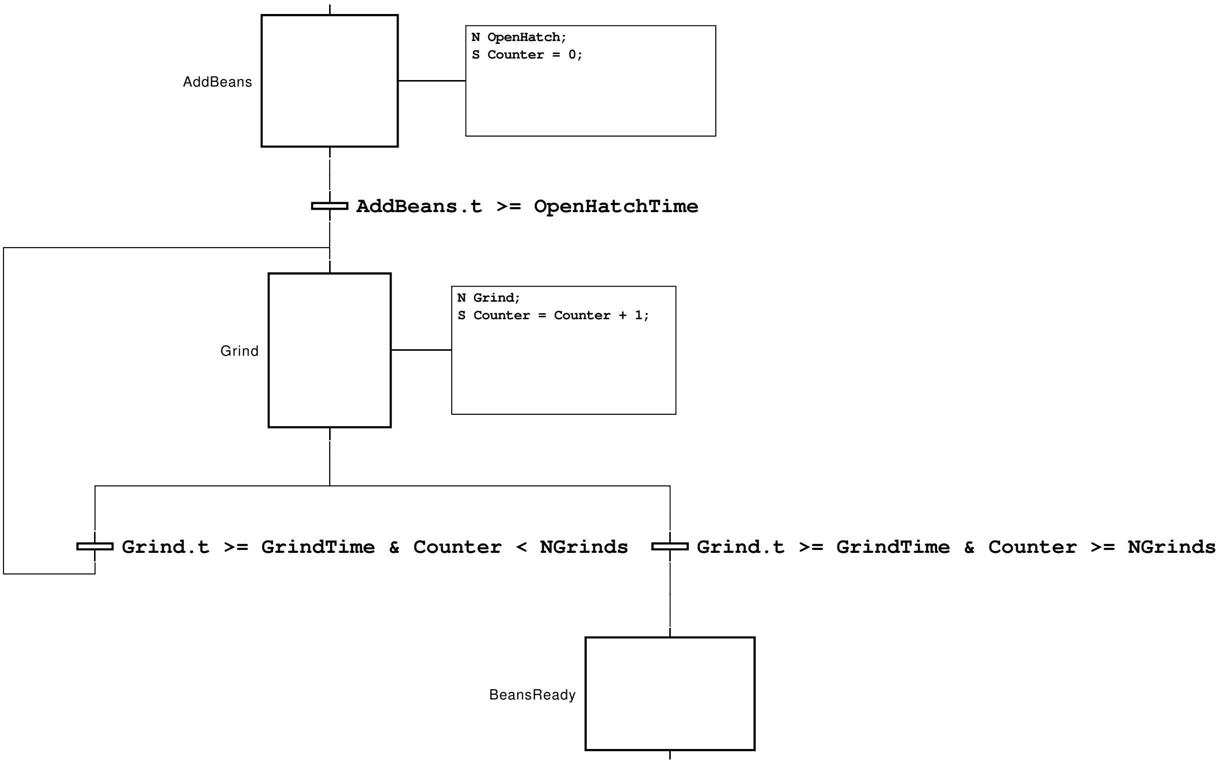
AddBeans
N OpenHatch;
S Counter = 0;
AddBeans.t >= OpenHatchTime
Grind
N Grind;
S Counter = Counter + 1;
Grind.t >= GrindTime & Counter < NGrinds
Grind.t >= GrindTime & Counter >= NGrinds
BeansReady

Figure 3 A possible solution for the coffee-grinding task.

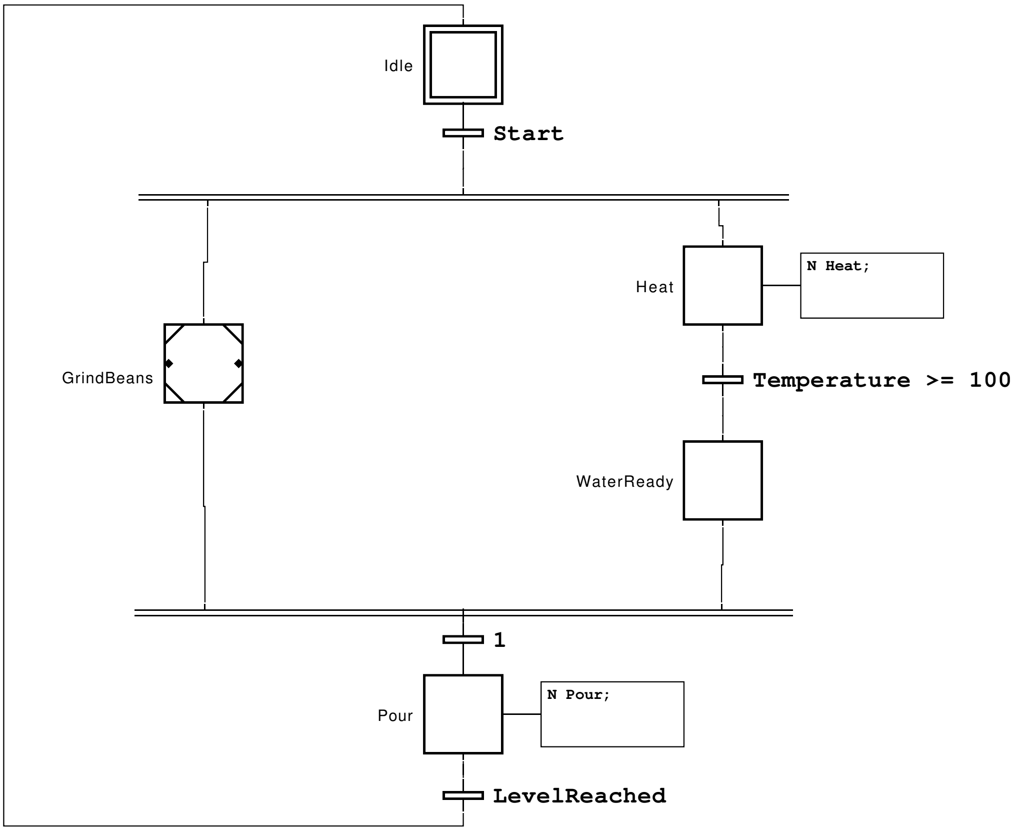

Figure 4 A possible solution for the complete coffee-making task.