LUND INSTITUTE
OF TECHNOLOGY
Lund University

Department of
AUTOMATIC CONTROL

Real-Time Systems

Exam April 10, 2012, hours: 14.00–19.00

Points and grades

All answers must include a clear motivation and a well-formulated answer. Answers may be given in **English or Swedish**. The total number of points is 25. The maximum number of points is specified for each subproblem.

Accepted aid

Standard mathematical tables and authorized formula sheet. Pocket calculator.

Results

The result of the exam will be posted on the notice-board at the Department. The result as well as solutions will be available on WWW:
http://www.control.lth.se/course/FRTN01/

1. Consider the system

$$G(s) = \frac{1}{1+sT}$$

a. Sample the system using zero-order hold and the sampling interval h. (0.5 p)

b. Discretize the system using backward differences and the sampling interval h. (0.5 p)

c. Explain when zero-order hold sampling should be used and when discretization using e.g. backward differences should be used. (1 p)

2. A model of a ball rolling on a beam is given by

$$\frac{dx(t)}{dt} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 10 \\ 0 \end{pmatrix} u(t-0.02)$$
$$y(t) = \begin{pmatrix} 0 & 1 \end{pmatrix} x(t)$$

The input time delay models the total delay induced by the implementation (computational delay, network delay, etc.).

Sample the system assuming zero-order hold inputs and the sampling interval $h = 0.1$. (2 p)

3. In Fig. 1 and Fig. 2 two schedules are shown for the same periodic task set, but for different scheduling policies. An up-arrow indicates the arrival

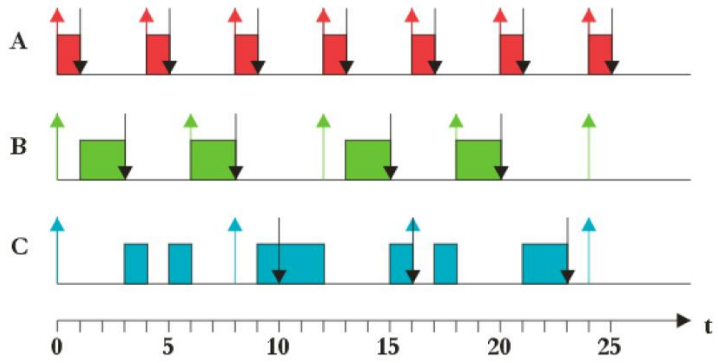

Figure 1 Execution schedule.

time for a new instance (job) of a task and an down-arrow indicates that the execution associated with an instance is completed. For all of the tasks it holds that $D_i = T_i$.

a. Calculate the CPU utilization for the task set. (1 p)

b. Determine which scheduling policies that are used in the two cases. Motivate your answers (1 p)

c. Verify the schedulability or non-schedulability of the task set under the two scheduling policies using quantitative calculations. (2 p)

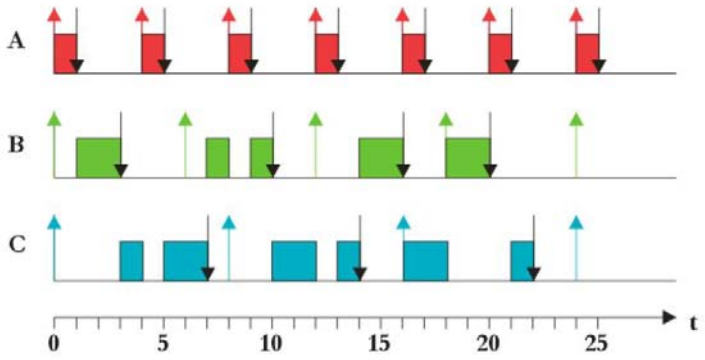

Figure 2 Execution schedule.

4. The following continuous-time state-space system is given:

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$
$$y = (0 \quad 1)x$$

a. Design a discrete-time deadbeat state-feedback controller $u = -Lx$ for the ZOH-sampled discrete-time system, assuming that the sampling time $h = 1$. (2 p)

b. If the B-matrix instead would be

$$B = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

would it still be possible to design a deadbeat controller? If Yes, perform the calculations. If No, explain why this is not possible. (1 p)

c. Give at least one reason for why it might be a bad idea to use a deadbeat controller. (1 p)

5. A discrete time filter is to be implemented on a microcontroller platform that does not support floating point arithmetic. The filter has the discrete time transfer function

$$H(z) = \frac{0.405z + 0.3529}{z^3 - 1.912z^2 + 1.583z - 0.6703} \tag{1}$$

The microcontroller supports 8-bit and 16-bit signed integers. We choose to use the 8-bit size for declared variables and 16-bits for intermediate results. For simplicity, we want to use the same number of fractional bits everywhere.

a. Determine the maximum number of fractional bits possible for coefficients on the described platform and calculate values of the coefficients for this number of fractional bits. (2 p)

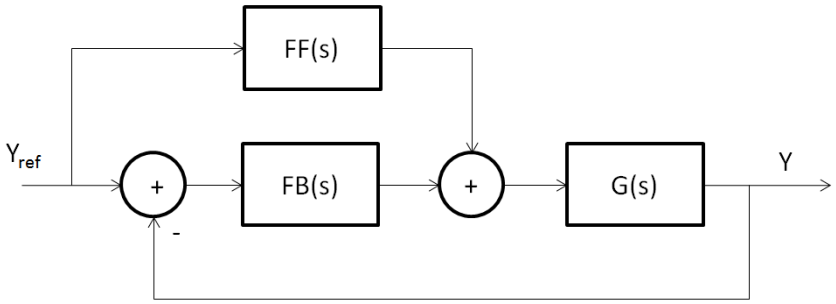

Figure 3 2-DOF controller structure.

b. Assume that the transfer function is written as

$$H(z) = \frac{b_1 z + b_0}{z^3 + a_2 z^2 + a_1 z + a_0} \tag{2}$$

Provide a C implementation of the filter function `int filter(int u)` so that it realizes the filter described above. It is OK if you realize the transfer function on direct form, even if this, normally, is not recommended for numerical reasons.

The input signal `u` uses Q6.0 encoding and the output from the filter should do the same. A skeleton implementation is provided below.

```
int8_t filter(int8_t u) {
    static int8_t b1 = ..., b0 = ..., ... ;
    static int8_t yold1 = 0, yold2 = 0, yold3 = 0;
    static int8_t uold1 = 0 uold2 = 0, uold3 = 0;
    int16_t y16;
    int8_t y;

    /* Calculate the output */

    /* Perform suitable limitation of the return value */

    return y;
}
```

(2 p)

6. Consider the following PID controller:

$$U(s) = K(\beta Y_{ref}(s) - Y(s) + \frac{1}{sT_I}E(s) + sT_D(\gamma Y_{ref}(s) - Y(s)) \tag{3}$$

a. Write the controller on 2-degree of freedom (2-DOF) form, i.e., so that it contains on feedback part ($FB(s)$) and one feedforward part ($FF(s)$) according to the figure below: (1 p)

b. Assume the following parameter values, $K = 1, T_I = 5, T_D = 0.2$. Where are the poles and zeros of $FB(s)$ located? (Hint: View $FB(s)$ as a single transfer function.) (1 p)

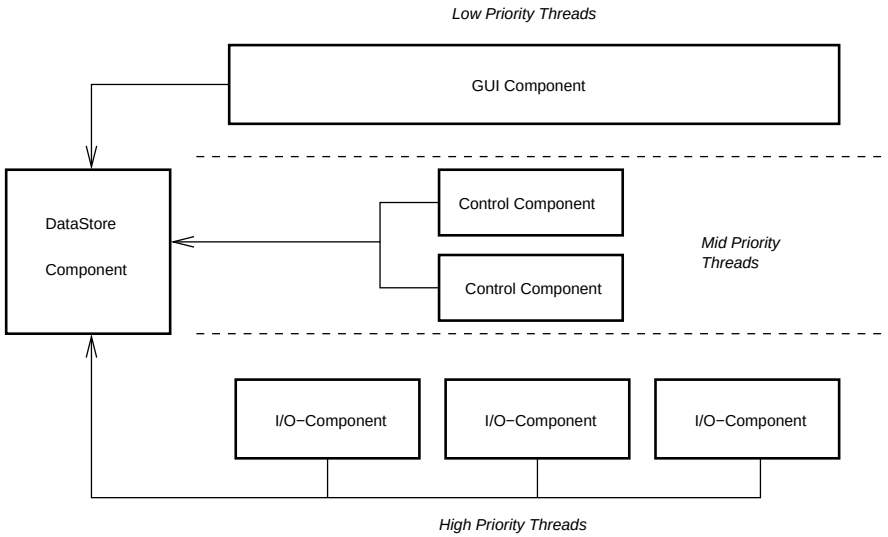

Figure 4 The layered control architecture

c. Approximate $FB(s)$ with a Tustin approximation with $h = 0.1$. Where are the poles of the discrete-time version of $FB(s)$ located? (1 p)

7. A control system implemented in Java uses a three-layered approach to divide its functionality (see Figure 4). The lowest layer performs I/O with the physical control plant and is executed as a number of high priority threads. The middle layer implements reference generation and control algorithms and is run with mid priority. The highest layer is the GUI and is run at the lowest priority. Data is accessed through a globally referenced data structure.

a. The desired interface for the central data structures can be seen below. Provide an implementation so that the operations `putData()` and `getData()` are properly synchronized and so that the GUI can wait for new data to be available through `waitForData()` without polling. The implementation must also make sure that no there are no external references to the privately held data object.

```
public class DataStore {
    private Cloneable data;  /* Private data */

    /* Updates the data with a local copy,
       notifies all threads that are waiting
       for the updated message. */
    public void putData(Cloneable newData);

    /* Returns a private data object. */
    public Object getData();

    /* Blocks until the data is updated. */
    public void waitForData()
        throws InterruptedException;
}
```

i	T_i [ms]	f_i [Hz]	C_i [ms]
1	10	100	1
2	20	50	3
3	20	50	6
4	40	25	2
5	40	25	2
6	40	25	4
7	40	25	5

Table 1 Task period time T_i, frequency f_i and worst case execution time C_i for the avionics system tasks τ_i, $i \in \{1, \ldots, 7\}$.

(2 p)

b. The priority scheme used can result in unwanted behavior. Describe the phenomenon, why it can occur and at least one method to resolve it. (1 p)

8. In avionics systems, static scheduling is a common solution to the real-time problem. In this problem we consider an avionics subsystem with seven tasks τ_i, $i \in \{1, \ldots, 7\}$ executing at three harmonic frequencies, with f_1=100 [Hz] being the base frequency. The task characteristics are given in Table 1.

a. As a first approach to solve the real-time problem the engineers used a static schedule implemented as fixed procedure cycles as shown in Figure 5. The

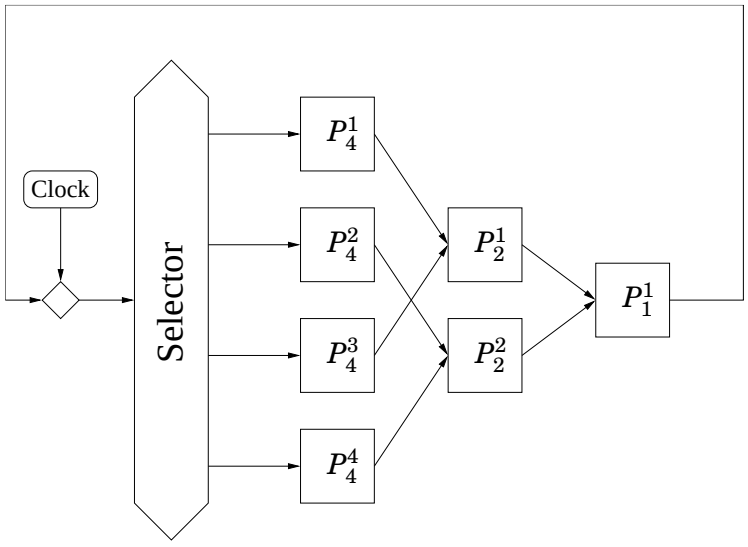

Figure 5 Static scheduling scheme based on procedure chains with tasks executing at harmonic frequencies.

program traverses the loop at the rate of the base frequency. The selector executes each of the 4 procedure chains once every 4th cycle in the loop. As a result the procedures labeled P_4^i executes at the frequency $f_1/4$, P_2^j at

the frequency $f_1/2$, and P_1^1 at the frequency f_1. Schedule the seven tasks in Table 1, τ_i, $i \in \{1,\ldots,7\}$, by uniquely assigning them the labels P_1^1, P_2^i, $i \in \{1,2\}$, P_4^j, $j \in \{1,\ldots,4\}$ in Figure 5. (1 p)

b. Calculate the jitter for each of the tasks given in your previous task assignments. Assume that the execution times given in Table 1 are worst case execution times and that the best case execution times are 0.5 time units shorter than the worst case values. Define jitter as the difference between the worst case (latest) starting time (within the period) and the best case (earliest) starting time of a task. (2 p)