LUND INSTITUTE
OF TECHNOLOGY
Lund University

Department of
AUTOMATIC CONTROL

Real-Time Systems

Exam August 23, 2016, 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

The textbooks Real-Time Control Systems and Computer Control: An Overview - Educational Version. Standard mathematical tables and authorized "Real-Time Systems 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/

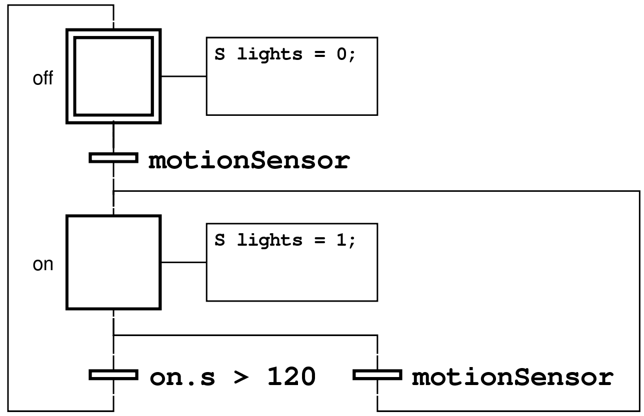

Figure 1

1. Consider the following discrete-time system

$$x(k+1) = \begin{pmatrix} 0.5 & 0 \\ -1 & 0 \end{pmatrix} x(k) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(k)$$
$$y(k) = \begin{pmatrix} 0 & 1 \end{pmatrix} x(k)$$

a. Design a full-order dead-beat observer for this system. (1 p)

b. The observer will be used together with a state-feedback controller $u(k) = -L\hat{x}(k)$. Write the pseudo-code for the observer. Use the following functions:

```
getU();  // Returns u(k)
getY();  // Returns y(k)
setU(u); //  Outputs the control signal
```

In order to get full-points for the problem the latency between the reading of the inputs and the output of the control signal should be as small as possible. Also, in order to get full points you have to write the observer in scalar form, i.e., matrix expressions are not allowed. (2 p)

2. The lighting of an attic is triggered by motion sensors only. The lights should be on when a motion has been detected during the last two minutes. An application for this has been implemented in Grafcet, see Figure 1, but it contains an error. Explain what is wrong and correct the application.

Note: <`stepName`>`.s` *is the number of seconds a named step has been active since the last activation. It is reset to* 0 *when the step is deactivated.* (2 p)

3. In Java, it is very common to use the word synchronized when you need some form of mutual exclusion synchronization.

a. Explain in general terms three cases when you need mutual exclusion synchronization. (1 p)

b. There is a number of ways in which the word synchronized can be used in Java. Explain the differences (especially what kind of lock you obtain) between the following constructions:

- call to a synchronized non-static method

- synchronized(this)...
- synchronized(this.getClass())...
- synchronized(a reference to an object != this)...

Every correct answer will give 0.5 points, missing or wrong answers will give −0.5 points. The total point will be at least 0 and at most 2. (2 p)

4. Consider a system with the continuous-time transfer function

$$G(s) = \frac{e^{-s}}{(s+1)(s+2)}$$

Sample the system using zero-order hold and h = 0.5 to obtain the pulse-transfer function $H(z)$. You may give the answer as a sum of rational functions. (2 p)

5. Consider the following continuous-time system

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

a. Design a continuous-time state-feedback controller, $u(t) = -Lx(t)$, such that the closed loop system has a double pole in −1. If this is not possible explain why. (0.5 p)

b. ZOH-sample the system with sampling period $h = \pi$. (1 p)

c. Design a discrete-time state-feedback controller such that the closed loop system has a double pole in $e^{-\pi}$. If this is not possible explain why. (1 p)

6. Each of the following pulse-transfer functions 1-6 corresponds to one of the pole diagrams a-e in Fig. 2 and one of the step responses A-E in Fig. 3. Determine the correspondences and don't forget to motivate your answers. (3 p)

$$H_1(z) = \frac{1}{z^2 + 0.8z + 0.07}$$
$$H_2(z) = \frac{1}{z^2 - 0.8z + 0.07}$$
$$H_3(z) = \frac{1}{z^2 - 0.2z + 0.01}$$
$$H_4(z) = \frac{1}{z - 1}$$
$$H_5(z) = \frac{1}{z^2 - 0.8z + 0.52}$$
$$H_6(z) = \frac{1}{z - 1.2}$$

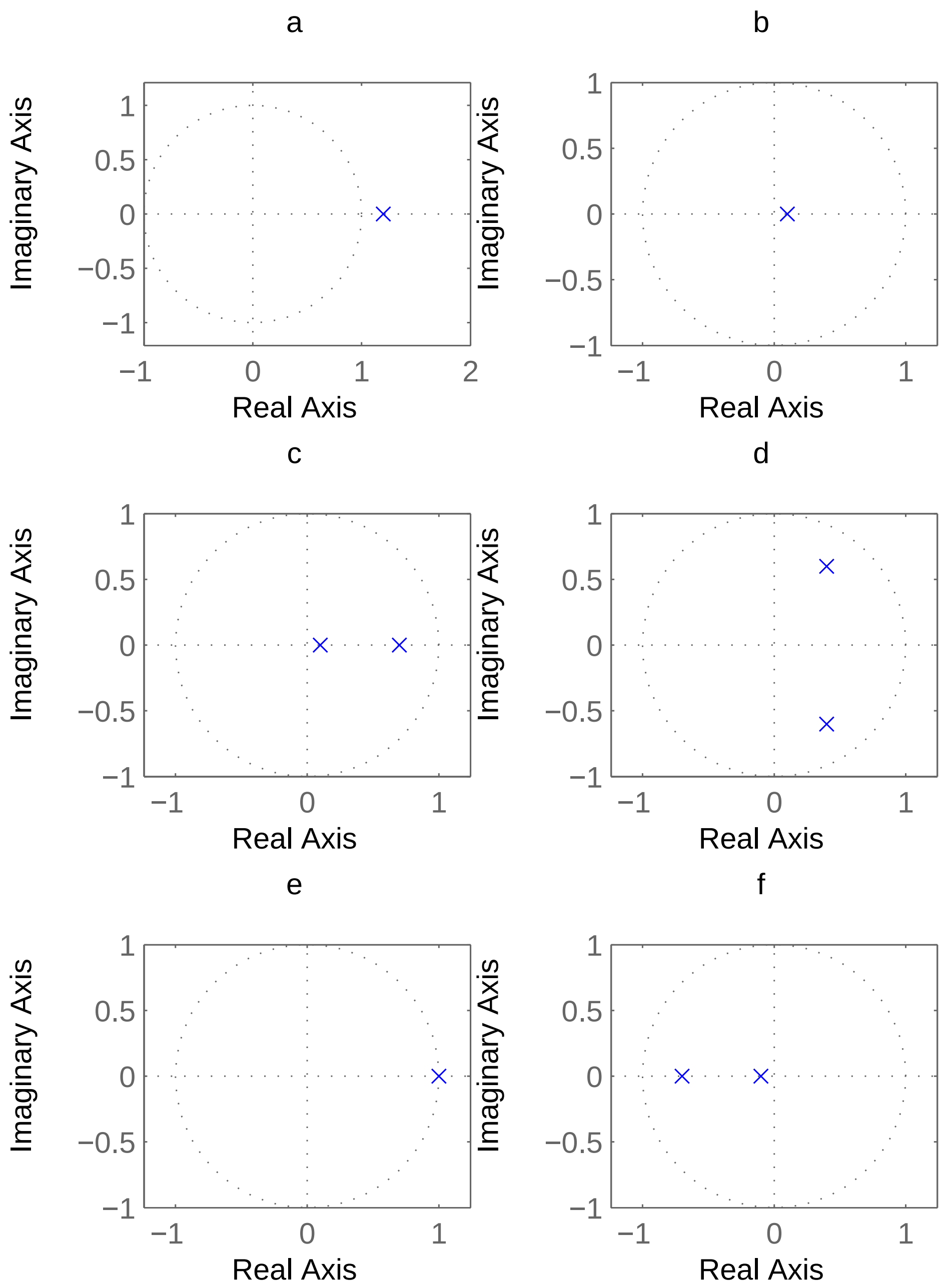

Figure 2 Pole diagrams for problem 6

7. A camera is used to capture images with a frame rate of approximately 50 FPS (frames per second) with an exposure time of 1 ms. The room where the images are captured is illuminated by fluorescent lights, flickering with a frequency of 100 Hz, due to the 50 Hz AC voltage powering them. In the video captured by the camera it appears as if the illumination of the room is varying with a period of 10 s. What actually happens is that the lights shine with different intensity depending on when the image is captured.

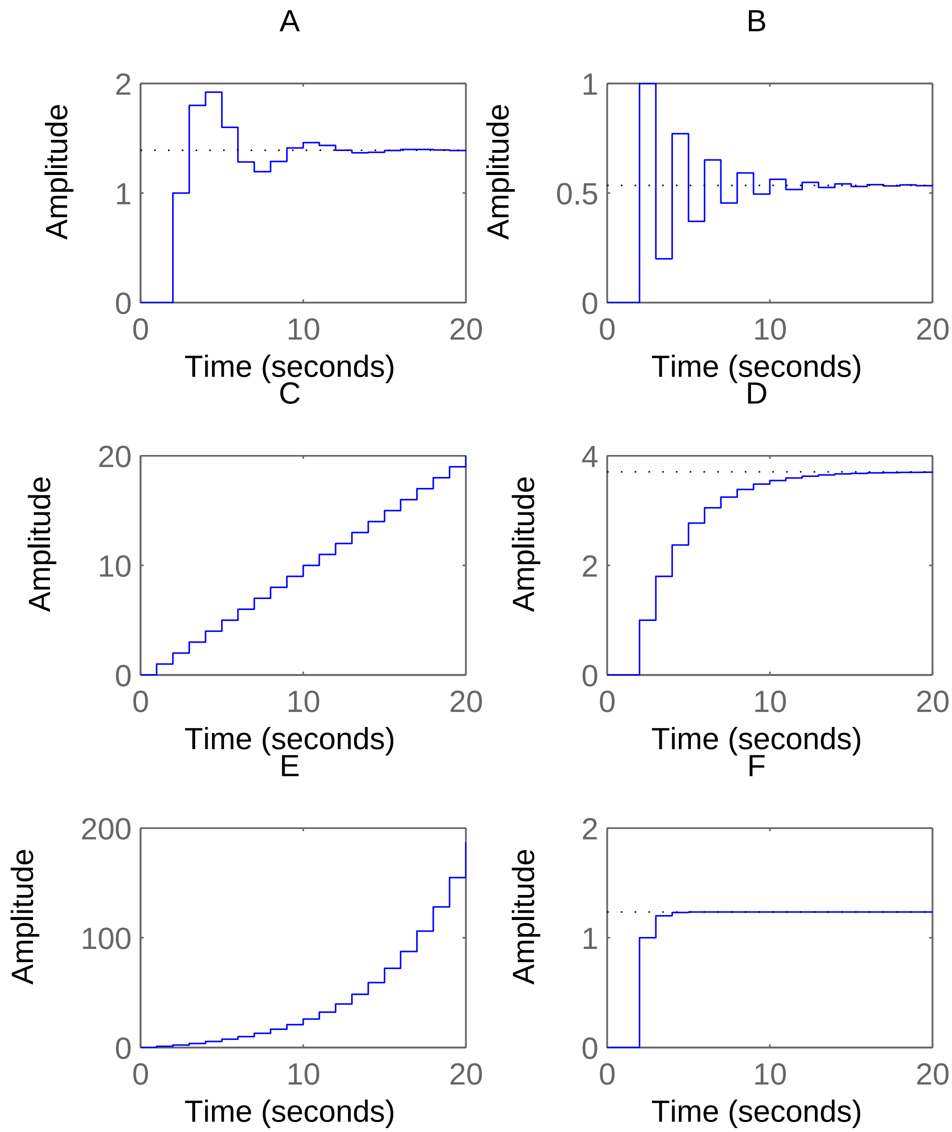

Figure 3 Step responses for problem 6

Assume that the frequency of the AC voltage to the lights is exactly 50 Hz. What are the two possible actual frame rates of the cameras? (1 p)

8. A system is given by

$$
\begin{aligned}
x(k+1) &= \begin{pmatrix} 0.82 & 0.86 \\ 0.07 & 0.50 \end{pmatrix} x(k) + \begin{pmatrix} 3.39 \\ 2.08 \end{pmatrix} u(k) \\
y(k) &= \begin{pmatrix} 5.11 & 7.42 \end{pmatrix} x(k)
\end{aligned}
$$

The system will be implemented using Q4.3 fixed-point numbers.

a. Approximate the system, i.e. convert to fixed-point numbers and then convert back again to real numbers. Write the approximated system on state-space form. (1 p)

b. How are the poles of the system affected by the approximation? Are there any significant changes? (1 p)

9. The table below describes three processes with their respective periods, relative deadlines and worst case execution times.

Task	T_i	D_i	C_i
A	3	3	1
B	8	8	2
C	20	20	5

a. Show that the task set is schedulable under Earliest-Deadline-First scheduling. (0.5 p)

b. Decide if the task set is schedulable under fixed-priority scheduling with rate-monotonic priority assigment. (1 p)

c. What is the shortest period that Task B may have in order for the task set to still be schedulable under fixed-priority rate-monotonic scheduling? The period may take arbitrary values, i.e. not only be integers. We still assume that the relative deadline is equal to the period. (Hint: Trying to calculate this with response time analysis can be very cumbersome and may not be worth the time it takes. So do the best you can with the means that you have available. One thing that you may possibly make use of is the following upper bound, $\bar{R}_i$, on the response time R_i.)

$$\bar{R}_i = \frac{C_i + \sum_{\forall j \in hp(i)} C_j(1 - U_j)}{1 - \sum_{\forall j \in hp(i)} U_j}$$

(1.5 p)

10. Assume a producer-consumer example in Java with multiple producer threads and consumer threads that communicate using a buffer of length 1. The producer processes enter an integer into the buffer in each call. Assume that there are two types of consumer threads, ConsumerA threads are only interested in reading integers with a value that is less than 0 and ConsumerB threads are only interested in reading values that are greater than or equal to 0.

Consider the following implementation of the buffer:

```
public class Buffer {
  private int data;
  private boolean full = false;


  public synchronized void put(int inData) {
```

```
    while (full) {
      try {
        wait();
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
    data = inData;
    full = true;
    notifyAll();
  }

  public synchronized int getNegative() {
    while (!full || (data >= 0)) {
      try {
        wait();
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
    full = false;
    notifyAll();
    return data;
    }

  public synchronized int getPositive() {
    while (!full || (data < 0)) {
      try {
        wait();
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
    full = false;
    notifyAll();
    return data;
    }
}

public class Producer extends Thread {
  private Buffer b;

  public Producer(Buffer bu) {
    b = bu;
  }

  public void run() {
    int data;
    while (true) {
      // Generate integer data in some way
      b.put(data);
    }
```

```
  }
}

public class ConsumerA extends Thread {
  private Buffer b;

  public ConsumerA(Buffer bu) {
    b = bu;
  }

  public void run() {
    int data;
    while (true) {
      data = b.getNegative();
      // Use data
    }
  }
}

public class ConsumerB extends Thread {
  private Buffer b;

  public ConsumerB(Buffer bu) {
    b = bu;
  }

  public void run() {
    int data;
    while (true) {
      data = b.getPositive();
      // Use data
    }
  }
}

public class Main {

  public static void main(String[] args) {

    Buffer b = new Buffer();
    Producer w;
    ConsumerA rA;
    ConsumerB rb;

    for (int i=0; i<10; i++) {
      w = new Producer(b);
      w.start();
    }
    for (int j=0; j<5; j++) {
      rA = new ConsumerA(b);
      rA.start();
    }
```

```
    for (int k=0; k<5; k++) {
      rB = new ConsumerB(b);
      rB.start();
    }
  }
}
```

a. This solution has an undesired effect related to efficiency. What is the problem? (1 p)

b. Modify the buffer so that the problem is removed. You may, e.g., use the classes Semaphore and ConditionVariable, defined in RTCS. (2.5 p)