Real-Time Control Systems

Exam 2008-08-25

14:00 – 18:00: Halls at "Väg och Vatten"

Course code: SSY190

Teacher: Knut Åkesson, phone 0701-749525

The teacher will visit examination halls twice to answer questions. This will be done approximately one hour after the examination started and one hour before it ends.

The exam comprises 30 credits. For the grades 3, 4, and 5, is respectively required 15, 20 and 25 credits.

Solutions and answers should be complete, written in English and be unambiguously and well motivated. In the case of ambiguously formulated exam questions, the suggested solution with possible assumptions must be motivated. The examiner retains the right to accept or decline the rationality of assumptions and motivations.

Solutions will be announced on the course web-page on the first week-day after the exam date. *Exam results* will be announced on the department notice board on the latest 2008-09-08 at 12:30. *The results* are open for review 2008-09-08, 12:30-13:30 at the department.

Allowed aids:

- An A4 sheet with handwritten notes on both pages. The name of the student must be on the sheet. You should hand-in your notes together with your solutions.
- A pocket calculator with erased memory.
- Dictionary (paper and electronic) between English and the students native language.

1

a) Describe what we mean by mutual exclusion?

(1p)

b) Why do processes have different priorities in a real-time operating system, explain how a real-time system use the different priorities?

(1p)

c) Assume that senors and actuators are connected to a controller through a network, discuss how the properties of the network may influence the control performance.

(1p)

d) Explain why it is necessary to low-pass filter all analog signals before sampling.

(1p)

2

A common industrial requirement is that controllers might be running in two modes, manual and automatic. When switching between these two modes it is often required that the control signal does not make large changes. To achieve this the controller need to be extended with support for bumpless transfer. In addition, if the controller includes an integrator it is necessary to include anti-windup support in the controller.

a) Describe the problems with windup and explain why it should be avoided.

(1p)

b) Describe how to implement bumpless transfer and anti-windup for a PI-controller. To get full credit you need to present a detailed block-diagram and explain how it is working.

(3p)

3

In the course book you find the following relation

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq n(2^{1/n} - 1).$$

a) Explain the purpose of the relation and what the symbols C_i, T_i, and n denotes. (2p)

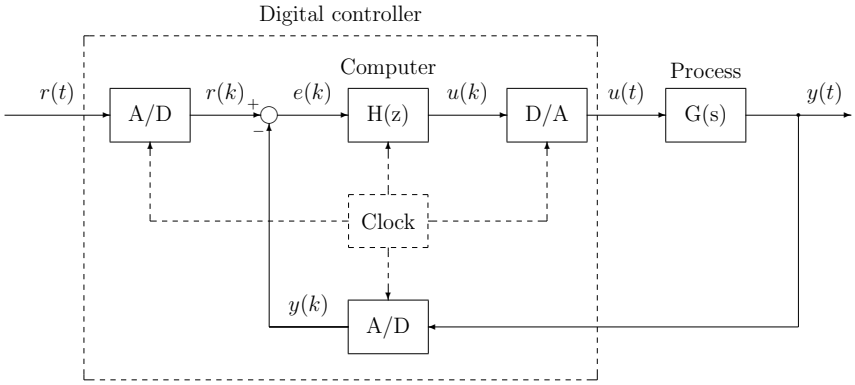
b) What conclusions might be drawn if

i) the relation hold. (1p)

ii) the relation does not hold. (1p)

4

Consider the control setup below.



The process can be described by the following transfer function

$$G(s) = \frac{3}{1 + 60s}$$

a) Determine a discrete time controller, i.e. $H(z)$, such that the closed-loop system is stable and the steady-state control error for step-formed setpoint changes are zero.

(3p)

b) Write pseudo-code for how to implement the controller using a language like C, Java or Modula 2. Ensure that the input-output latency is minimized.

(3p)

5

A continuous PD-controller may be written as

$$F(s) = K(1 + \frac{sT_d}{1 + sT_d/N})$$

Discretize the controller $F(s)$ using the sampling time h and the backward difference approximation. Show how the control signal $u(k)$ at sample k can be calculated from the control errors $(r(k)-y(k), r(k-1)-y(k-1), \ldots)$ and previous values of the control signal and controller states. Determine which part of the control signal $u(k)$ that can be precomputed at sample $k-1$.

(3p)

6

The code below shows the logical semantics of the semaphore primitives Wait and Signal in the kernel discussed in the course book.

```
Wait(sem); <--->

IF semˆ.counter = 0 THEN
  insert Running in the waiting queue of the semaphore;
ELSE
  semˆ.counter = semˆ.counter - 1;


Signal(sem); <--->

IF waiting queue is not empty THEN
  move first process in waiting queue to ReadyQueue;
ELSE
  semˆ.counter = semˆ.counter + 1;
```

a) Two tasks, A and B, are sharing a common resource that is protected by a semaphore, mutex. Task A has higher priority than task B. Consider the following execution sequence

```
Task A (high)        Task B (low)
-------------        ------------

Wait(mutex);         ...
...                  Wait(mutex)
Signal(mutex);       ...
Wait(mutex);         ...
```

Explain what happens in the kernel during the execution. Which task will be holding the semaphore after the execution?

(2p)

b) Modify the semantics above so that it is guaranteed that the highest priority task ends up holding the semaphore.

(2p)

7

Solve the following optimization problems.

a) Find the optimal solution.

(4p)

$$\begin{array}{ll} \max & 2x_1 - 3x_2 + 4x_3 \\ \text{s.t.} & 4x_1 - 3x_2 + x_3 \leq 3 \\ & x_1 + x_2 + x_3 \leq 10 \\ & 2x_1 + x_2 - x_3 \leq 10 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

b) Find the optimal solution, note that the variables can only take integer values.

(1p)

$$\begin{array}{ll} \max & x_1 + 5x_2 \\ \text{s.t.} & x_1 + 10x_2 \leq 20 \\ & x_1 \leq 2 \\ & x_1, x_2 \geq 0 \text{ and integer} \end{array}$$

Good Luck!