Real-Time Control Systems

Exam 2011-03-16

14:00 – 18:00: Halls at “Maskin”

Course code: SSY190

Teacher: Knut Åkesson

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, be unambiguous, and be 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.

Exam results will be reported in Ladok. *The results* are open for review 2011-03-30, 12:30-13:30 at the department.

Allowed aids:

- A pocket calculator with erased memory.
- Formula Sheet: “Real-Time Control Systems (SSY190), Formula Sheet, 2011”
- Dictionary (paper and electronic) between English and the students native language.

1

a) Explain the difference between soft and hard real-time systems.

(1p)

b) What is priority inversion? Explain by giving a simple example.

(1p)

c) Explain shortly how the priority inheritance protocol works.

(1p)

d) Describe the main parts of the CAN-bus protocol and explain the arbitration procedure.

(2p)

e) Describe how TTCAN (Time-Triggered CAN) can be used to decrease the latency and jitter for real-time control applications.

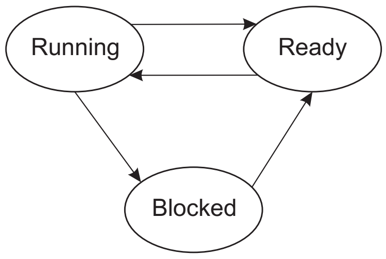
(2p)

f) Compare Ethernet and EtherCAT, explain what make EtherCAT more suitable for distributed control systems than standard Ethernet.

(2p)

2

Each process in a real-time operating system can be in the following three states.



Explain shortly what characterizes each of the three states and explain for each of the four transitions, in the figure above, what could cause the process to switch state.

(3p)

3

Consider the processes

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

controlled by the PI-controller

$$F(s) = \frac{1}{3} + \frac{1}{27s}.$$

a) Choose a suitable sampling time h based on the system dynamics, motivate your answer. Discretize the controller using Tustin discretization. Determine the final difference equation for the discretized controller.

(2p)

b) Write pseudo-code (C or java like syntax) for the final controller implementation. Optimize the code to minimize the time before the control signal is generated.

(2p)

c) Assume there is a network between the controller and the acturator. Motivate in which situations it is resonable to assume that there is a constant delay between the release time of the control task and the time the actuator effectuates the control signal.

(1p)

d) Use the above assumption about a constant delay. Determine the largest constant delay you can allow before the closed-loop system becomes unstable. To simplify your calculations, only consider the continous models of the plant and the controller.

(2p)

4

Suppose we have N threads with identities $0, 1, \ldots, N-1$. The threads share a critical region of code that only one of the threads is allowed to execute at a time, in addition they are required to repeatedly cycle through this section in the natural order $(0, 1, \ldots, N-1)$.

a) Let `turn` be a shared integer variable initialized to zero and let `current_pid` be the identity of the currently executing process. Each thread executes the code below. Assume that reading or writing to a single variable is an atomic operation. Can a race-condition occur? Motivate your answer!

```
while( TRUE )
{
    while( turn != current_pid ) /* wait */ ;
    critical_region();
    turn = (turn + 1) % N;
    noncritical_region();
}
```

Note: x % N returns the remainder of division of x by N.

(2p)

b) Develop a solution using semaphores for the problem described above - you are not allowed to used a turn variable or a similar solution all coordination should be based on sempahores. Clearly specify the initial value of each semapahore. Write pseduo-code for the solution. Notation: if `x` is a sempahore then `wait(x)` and `signal(x)` are the semaphore operations.

(2p)

5

In many embedded application the processor do not have a floating-point unit thus the calculations for implementing a control algorithm for example has to be implemented using fixed-point arithmetics. You would like to implement a first-order digital controller

$$\begin{aligned} x(k+1) &= 0.78x(k) - 1.23y(k) \\ u(k) &= 0.91x(k) \end{aligned}$$

using fixed-point arithmetic.

The microprocessor is an 8-bit microprocessor.

a) You are using a small 8-bit processor so you decide to use $N = 8$ word length. Determine the largest possible number of fractional bits n and convert the coefficients to fixed-point representation.

(1p)

b) Write C-code for implementing the controller calculations using fixed-point arithmetics. You can assume that the input and output are done with the functions

```
int8_t readInput();
void writeOutput(int8_t u);
```

(2p)

6

We have three processes that execute on a single processor with an ideal preemptive real-time kernel. T is the period of the task, D is the deadline of the task, C is the worst-case computation time.

Task	T	D	C
P_1	10	2	1
P_2	3	3	1
P_3	5	5	2

a) Determine priorities (low, medium and high) for the three processes according the the rate monotonic principle and deadline monotonic principle.

(1p)

b) Compute the worst case response time for the deadline and rate monotonic cases above, determine if the deadlines can be met?

(3p)

Good Luck!

1) See book

2) See book

3/ $G(s) = \frac{1}{s(s+1)}$ $\qquad F(s) = \frac{1}{3} + \frac{1}{27s} = \frac{9s+1}{27s}$

a/ Choose suitable sampling time and discretize the controller using Tustin.

From formula sheet: $h \cdot w_c \approx 0.05$ to 0.15

Determine w_c (Can be done by drawing a Bodediagram)

$$|L(jw_c)| = 1 \Rightarrow \frac{1}{w_c\sqrt{w_c^2+1}} \frac{\sqrt{1+81w_c^2}}{27w_c} = 1$$

$$\Rightarrow 729w_c^6 + 729w_c^4 - 81w_c^2 - 1 = 0 \Rightarrow$$ positive real root in $\underline{\underline{w_c = 0.33}}$

Let $h \cdot w_c = 0.1 \Rightarrow h = \frac{0.1}{w_c} = \frac{0.1}{0.33} \approx \underline{\underline{0.3 \text{ seconds.}}}$

Tustin $s' = \frac{2}{h}\frac{z-1}{z+1}$

$$F(s) = \frac{9s+1}{27s} \Rightarrow F(z) = \frac{9\left(\frac{2}{h}\frac{z-1}{z+1}\right)+1}{27\frac{2}{h}\frac{z-1}{z+1}} = \frac{\frac{18+h}{54}z + \frac{h-18}{54}}{z-1} =$$

$$= \frac{0.34z - 0.33}{z-1}$$

$$\begin{tikzcd} \arrow[r, "e"] & \boxed{F(z)} \arrow[r, "u"] & {} \end{tikzcd} \Rightarrow U(z) = F(z) \cdot E(z)$$

$$\Rightarrow (z-1)U(z) = (0.34z - 0.33)E(z)$$

$$\Rightarrow u(k+1) - u(k) = 0.34e(k+1) - 0.33e(k)$$

$$\Rightarrow u(k) = u(k-1) + 0.34e(k) - 0.33e(k-1)$$

3b/

```
void controller()
{
    currentTime(t);
    while (TRUE)
    {
        y = AD_read("y");
        r = AD_read("r");
        e = r - y;
        u = upre + 0.34·e;
        AD_write("u", u);
        upre = u - 0.33·e;
        incTime(t, 0.3);
        waitUntil(t);
    }
```

3c/ i) The controltask is the only task on the microprocessor

ii) No collisions on the network will occur, for example by using TTCAN or by having exclusive access to the network.

3d/ $w_c = 0.33$ rad/s

Compute phase margin (φ_m)

$\arg L(jw_c) = -90^\circ - \arctan(w_c) - 90^\circ + \arctan(9w_c) \overset{w_c=0.33}{=} -127^\circ$

$\Rightarrow \varphi_m = -127^\circ + 180^\circ = 53^\circ$

Let Td be the constant delay $\Rightarrow \tilde{L}(jw) = F(jw)G(jw)e^{-jwTd}$

$\arg \tilde{L}(jw) = \arg L(jw) - wTd \cdot \frac{180^\circ}{\pi}$

$|\tilde{L}(jw)| = |L(jw)| \Rightarrow w_c = 0.3$ for $\tilde{L}$ as well.

Compute max Td such that closed-loop system is stable.

$\Rightarrow w_c \cdot Td \cdot \frac{180}{\pi} = 53^\circ \Rightarrow Td = \frac{53 \cdot \pi}{180 \cdot 0.33} = \underline{\underline{2.8 \text{ seconds.}}}$

4/a) No race condition occur. Thread i must change the value of turn before process i+1 can escape its busy-wait loop, thus only one process can be in the critical region.

b) You need i semaphores.

```
i = current_pid
while (TRUE)
{
    wait(semaphore[i]);
    critical_region();
    signal(semaphore[(i+1)%N]);
    noncritical_region();
}
```

Semaphore 0 is initialized to 1, all others to 0.

5/ a) k should be stored as $K = \text{round}(k \cdot 2^N)$ where N is the number of fractional bits

8-bit signed ints ⇒ range [−128, 127]

Largest coefficient is 3.262 ⇒ $\log_2\left(\frac{128}{3.262}\right) = 5.29$ fractional bits ⇒ Use N = 5

$A = \text{round}(0.8967 \cdot 2^5) = 29$ $B = \text{round}(0.2332 \cdot 2^5) = 7$

$C = \text{round}(-3.262 \cdot 2^5) = -104$ $C = \text{round}(-0.8484 \cdot 2^5) = -27$

b/

```
#define A 29
#define B 7
#define C -104
#define D -27
#define N 5
int8_t y, x, u;
int16_t x16=0, u16=0;

y = readInput();
u16 = u16+ ((int16_t)D × (int16_t)y) >>N;
if (u16>127)
{
  u=127;
}
else if (u16< -128)
{
  u=-128;
}
else
  u=u16;
}
writeOutput(u);
x16 = ((int16_t)A× (int16_t)x + (int16_t)B*(int16_t)y)>>N;
if (x16 >127) {x=127;}
else if (x16<-128) {x=-128;}
else {x=x16;}
}
u16= ((int16_t)C * (int16_t)x) >>N;
```

6/ a)

	T	D	C
P_1	10	2	1
P_2	3	3	1
P_3	5	5	2

Rate monotonic (consider T)

P_1 low prio

P_2 high prio

P_3 medium prio

Deadline monotonic (consider D)

P_1 high prio

P_2 medium prio

P_3 low prio

b/ Compute worst-case response time.

$$R_i = C_i + \sum_{k_j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil \cdot C_j$$

Rate monotonic:

$hp(P_1) = \{P_2, P_3\}$

$hp(P_2) = \{\}$

$hp(P_3) = \{P_2\}$

Deadline monotonic:

$hp(P_1) = \{\}$

$hp(P_2) = \{P_1\}$

$hp(P_3) = \{P_1, P_2\}$

6b)

RM

step 1 : $R_1 = 1$
$R_2 = 1$
$R_3 = 2$

step 2 : $R_1 = 1 + \left\lceil \frac{1}{3} \right\rceil \cdot 1 + \left\lceil \frac{1}{5} \right\rceil \cdot 2 = 4$

$R_2 = 1$

$R_3 = 2 + \left\lceil \frac{2}{3} \right\rceil \cdot 1 = 3$

step 3 : $R_1 = 1 + \left\lceil \frac{4}{3} \right\rceil \cdot 1 + \left\lceil \frac{4}{5} \right\rceil \cdot 2 = 5$

$R_2 = 1$

$R_3 = 2 + \left\lceil \frac{3}{3} \right\rceil \cdot 1 = 3$

step 4 :

$R_1 = 1 + \left\lceil \frac{5}{3} \right\rceil \cdot 1 + \left\lceil \frac{5}{5} \right\rceil \cdot 2 = 5$

$R_2 = 1$

$R_3 = 3$

$R_1 > D_1 \Rightarrow$ Thus the deadline for P_1 will not be met.

6c) DM

step 1: $R_1 = 1$

$R_2 = 1$

$R_3 = 2$

step 2: $R_1 = 1$

$$R_2 = 1 + \left\lceil \frac{1}{10} \right\rceil \cdot 1 = 2$$

$$R_3 = 2 + \left\lceil \frac{2}{10} \right\rceil \cdot 1 + \left\lceil \frac{2}{3} \right\rceil \cdot 1 = 4$$

step 3: $R_1 = 1$

$$R_2 = 1 + \left\lceil \frac{1}{10} \right\rceil \cdot 1 = 2$$

$$R_3 = 2 + \left\lceil \frac{4}{10} \right\rceil \cdot 1 + \left\lceil \frac{4}{3} \right\rceil \cdot 1 = 5$$

step 4: $R_1 = 1$

$R_2 = 2$

$$R_3 = 2 + \left\lceil \frac{5}{10} \right\rceil \cdot 1 + \left\lceil \frac{5}{3} \right\rceil \cdot 1 = 5$$

$$\left.\begin{array}{l} R_1 \le D_1 \\ R_2 \le D_2 \\ R_3 \le D_3 \end{array}\right\} \Rightarrow \text{All deadlines met!}$$