

# Esame Software Engineering (AA 2022/23)

24 Marzo 2023

*Enrico Tronci*

*Computer Science Department, Sapienza University of Rome  
Via Salaria 113 - 00198 Roma - Italy*

`tronci@di.uniroma1.it`

`http://mclab.di.uniroma1.it`

## Esercizio 5 (15 punti)

L'unità di tempo per questo esercizio è il secondo.

Un modello semplificato del carico (domanda di calcolo) in un data center è:

$$x(t+1) = x(t) + Tw(t) \quad (1)$$

$$r(t+1) = r(t)u(t) \quad (2)$$

$$z(t) = e^{x(t)} - r(t) \quad (3)$$

$$(4)$$

dove:

1.  $T$  è il sampling time. Potete assumere  $T = 1$ .
2.  $w(t)$  è una variabile aleatoria uniformemente distribuita in  $[-1, 1]$ .
3.  $x(t)$  modella la domanda di risorse al tempo  $t$ .
4.  $z(t)$  modella il backlog al tempo  $t$ .
5.  $r(t) \geq 0$  modella le risorse di calcolo messe a disposizione al tempo  $t$ .
6.  $u(t) \geq 0$  regola le risorse al  $t + 1$ .

Inizialmente abbiamo:

$$x(0) = 0 \quad (5)$$

$$r(0) = 1 \quad (6)$$

$$(7)$$

Si vuole minimizzare il backlog  $z(t)$  ed al contempo minimizzare le risorse usate. Quindi si sceglie  $u(t)$  in modo da mantenere  $z(t)$  piccolo ed  $r(t)$  non-negativo e piccolo.

A tal fine viene realizzato un software di controllo che ogni  $T$  secondi (*sampling and holding*) calcola il valore di  $u$  in modo da soddisfare il requisito di cui sopra. Una semplice strategia può essere (ma potete usarne una qualsiasi a vostra scelta): aumentare  $u(t)$  del 10% se  $z(t)$  è positivo, diminuire  $u(t)$  del 10% se  $z(t)$  è negativo o nullo.

Si progetti il software di controllo

## Parametri del Modello

Il modello conterrà i seguenti parametri:

1. **HORIZON**, contenente l'orizzonte di simulazione, cioè il tempo simulato. Potete usare **HORIZON = 1000**.
2.  $T = 1$

## Modello

Se sviluppi un modello Modelica consistente di almeno i seguenti blocchi:

1. Blocco **Plant** nel file **plant.mo** che modella  $x, z, r$ .
2. Blocco **Controller** nel file **ctr.mo** che modella il software di controllo che calcola  $u$  in funzione di  $z$ .
3. Blocco **Monitor** nel file **monitor.mo** che calcola dei KPI (*Key Performance Indicators*) rispetto ai desiderata. Nello specifico, ogni  $T$  secondi il monitor calcola:

$$e_1(t) = z(t) \tag{8}$$

$$e_2(t) = r(t) \tag{9}$$

$$\tag{10}$$

## Output della simulazione

Si usi l'istruzione Modelica **terminate** per terminare la simulazione quando la variabile Modelica **time** ha un valore maggiore del parametro **HORIZON**.

Alla terminazione si stampino nel file **outputs.txt** i valori medi e le deviazioni standard per tutti le componenti.

**AvgErr1 StdDevErr1 AvgErr2 StdDevErr2 (ID = aaa, MyMagicNumber = bbb, HORIZON = ccc, time = ddd)**

dove:

- `aaa` è il valore del parametro `ID`,
- `bbb` è il valore del parametro `MyMagicNumber`,
- `ccc` è il valore del parametro Modelica `HORIZON`,
- `ddd` è il valore della variabile Modelica `time` quando la simulazione viene terminata dal comando `terminate`.

Le altre righe hanno il seguente formato:

`< Valore medio di  $e_1$  > < Deviazione standard di  $e_1$  > < Valore medio di  $e_2$  > < Deviazione standard di  $e_2$  >`

Si avranno quindi, a parte la prima riga di intestazione, 1 riga.

Si usi un orizzonte di simulazione molto grande. In particolare si verifichi che l'orizzonte di simulazione sia maggiore del valore del `time` quando la simulazione viene terminata dal comando `terminate`. Se questo non è verificato il modello è sbagliato. Questo valore di `time` è visibile su `stdout`.

## NOTA

Si vedano le istruzioni ed in particolare la sezione *NOTA BENE* delle istruzioni.