

Esame Software Engineering (AA 2021/22)

06 Settembre 2022

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 2 (20 punti)

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

Un impianto industriale è monitorato attraverso una rete di sensori (IoT).

Il monitoraggio consiste nel leggere periodicamente i valori dei sensori e nel caso aggiornare i parametri di controllo dell'impianto in modo da mantenere i valori dei sensori entro valori normali.

L'impianto consiste di N sottosistemi, numerati da 1 ad N . Per ogni sottosistema ci sono Q sensori, numerati da 1 a Q che ne monitorano il comportamento.

I valori dei sensori sono memorizzati in una matrice X a valori reali di dimensione $N \times Q$. Il valore $X[i, j]$ contiene il valore del sensore j per il sottosistema i .

I parametri di controllo dell'impianto sono memorizzati in una matrice U a valori interi di dimensione $N \times Q$. Il valore $U[i, j]$ contiene la strategia di controllo in uso per la dinamica monitorata dal sensore j del sottosistema i .

La dinamica di ciascun sensore è modellata con una *Discrete Time Markov Chain* (DTMC). Sia, per ogni t , $r(t)$ una variabile random reale uniformemente distribuita in $[0,1]$. Ogni T_{plant} secondi, per ogni $i \in \{1, \dots, N\}$, $j \in \{1, \dots, Q\}$, il valore di $X[i, j]$ è aggiornato come segue:

$$X[i, j](t+1) = \begin{cases} X[i, j](t) + T0.01(-2 + 2r(t)), & \text{if } (U[i, j] = -1) \\ X[i, j](t) + T0.01r(t)2, & \text{if } (U[i, j] = 1) \\ X[i, j](t) + T0.01(-1 + 2r(t)), & \text{otherwise} \end{cases}$$

dove:

$$X[i, j](0) = \alpha(i, j) \tag{1}$$

$$\alpha(i, j) = Q(i-1) + j \tag{2}$$

Potete assumere $T_{plant} = 10$.

L'ambiente **Env** per il software consiste di un server che, oltre alle matrici di cui sopra, mantiene una variabile p che definisce il sottosistema sotto osservazione ed una v che definisce il sensore sotto osservazione.

Il server **Env** riceve comandi nella forma (c, d) , dove c e d sono interi. Se $c = 0$ allora il server assegna alla variabile p il valore d . Se $c = 1$ allora il server assegna alla variabile v il valore d . Se $c = 2$ allora il server assegna alla variabile

$U[p, v]$ il valore d , cioè aggiorna la strategia di controllo per il sottosistema p , relativamente al sensore v . Se $c = 3$ allora il server ritorna in output il valore reale $X[p, v]$.

Il server **Env** è connesso al client di monitoraggio attraverso due fifo: una dal server al client (che invia al client il valore $X[p, v]$) e l'altra dal client al server (che invia al server la coppia (c, d)).

Ogni T_{ctr} secondi il client legge il valore di un sensore per un sottosistema in modo che dopo $N * Q * T_{ctr}$ secondi il client ha letto i valori di tutti i sensori per tutti i sottosistemi. Deve essere $N * Q * T_{ctr} \leq T_{plant}$. Quindi deve essere $T_{ctr} \leq \frac{T_{plant}}{N * Q}$. Potete leggere i sensori in un ordine a piacere.

Sulla base delle letture di cui sopra, ogni T_{ctr} secondi il client aggiorna la strategia di controllo in modo da mantenere il valore del sensore j per il sottosistema i nell'intervallo $[\alpha(i, j) - 0.5, \alpha(i, j) + 0.5]$.

Nello specifico, il client manda al server -1 (cioè richiede $U[i, j] = -1$) se il valore del sensore è maggiore di $\alpha(i, j) + 0.5$, manda 1 (cioè richiede $U[i, j] = 1$) se il valore del sensore è minore di $\alpha(i, j) - 0.5$, manda 0 (cioè richiede $U[i, j] = 0$) in tutti gli altri casi.

Questo esercizio si focalizza sulla modellazione del server **Env** di cui sopra. A tal fine si sviluppano i seguenti blocchi.

1. Blocco **Env** nel file `env.mo` che modella l'environment descritto sopra.

Parametri del Modello

Il vostro modello conterrà i seguenti parametri positivi:

1. N, Q .

Output della simulazione

Si usi l'istruzione Modelica **terminate** per terminare la simulazione quando per tutti i teams, la deviazione standard del tempo di completamento è minore od uguale a $0.1 * \mu$, dove μ è il valor medio del tempo di completamento.

Alla terminazione si stampino nel file `outputs.txt` i tempi ed i costi per ogni team nel seguente formato.

La prima riga (di *intestazione*) del file `outputs.txt` contiene:

```
ID MyMagicNumber A B C D F G Team AvgTime AvgCost StdDevTime StdDevCost (time = xxx)
```

dove: **xxx** è il valore della variabile Modelica **time** quando la simulazione viene terminata dal comando **terminate**.

Le altre righe hanno il seguente formato:

<Valore del parametro ID> <Valore del parametro A> <Valore del parametro B> <Valore del parametro C> <Valore del parametro D> <Valore del parametro F> <Valore del parametro G> <ID del Team> <Valore di AvgTime> <Valore di AvgCost> <Valore di StdDevTime> <Valore di StdDevCost>

Si avranno quindi, a parte la prima riga di intestazione, W righe, una per ogni team.

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.