

Esame Software Engineering (AA 2021/22)

21 Luglio 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 (25 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 segnalare verso l'operatore:

- -1 , quando il valore rilevato è inferiore del 20% rispetto alla media mobile;
- $+1$, quando il valore rilevato è superiore del 10% rispetto alla media mobile;
- 0 in tutti gli altri casi.

Per ogni sensore, la media mobile, cioè la media degli ultimi K valori rilevati, è calcolata con una finestra di valore $K = 10$.

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 .

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

$$X[i, j](t+1) = X[i, j](t) + T(-1 + 2r(t) + U[i, j](t)) \quad (1)$$

dove:

$$X[i, j](0) = \alpha(i, j) \quad (2)$$

$$\alpha(i, j) = Q(i-1) + j \quad (3)$$

Potete assumere $T = 10$ (10 secondi).

L'ambiente **Env** per il software consiste di un server che, oltre alle matrici di cui sopra, mantiene una matrice $U[i, j]$ che definisce il drift sul sensore j del sottosistema i .

Sia, per ogni t , $U[i, j](t)$ una variabile aleatoria reale.

Ogni $5T$ secondi, per ogni $i \in \{1, \dots, N\}$, $j \in \{1, \dots, Q\}$, il valore di $U[i, j](t)$ è aggiornato scegliendolo uniformemente a random nell'intervallo $[-1, 1]$. $U[i, j](t)$ rimane costante per ogni t fino al nuovo aggiornamento.

Inizialmente $U[i, j](t)$ è 0. Cioè:

$$U[i, j](0) = 0 \quad (4)$$

Ogni secondo il client legge il valore di un sensore per un sottosistema in modo che dopo $W = N * Q$ secondi ha letto i valori di tutti i sensori per tutti i sottosistemi. Su questa base il client aggiorna il valore della media mobile per ogni sensore e manda in output all'operatore i segnali $\{-1, 0, 1\}$ come descritto sopra.

Le letture dei sensori vengono effettuate nel seguente ordine. Prima si considerano i sensori per il sottosistema 1, poi quelli per il sottosistema 2, etc fino al sottosistema N . I sensori vengono considerati dal più piccolo (indice 1) al più grande (indice Q)

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, cioè $X[i, j](t)$ ed $U[i, j](t)$.
2. Blocco **Monitor** nel file **monitor.mo** che calcola la media mobile per ogni sensore di ogni sottosistema.

Si usi l'istruzione Modelica **terminate** per terminare la simulazione quando il valore di **time** è maggiore di $500 * N * Q$.

Alla terminazione si stampino nel file **outputs.txt**, per ogni sensore, l'ultimo valore calcolato per la media mobile.

Il file **outputs.txt** ha il seguente formato.

La prima riga del file **outputs.txt** contiene:

Sottosistema Sensore MobileAvg

Le altre righe hanno il seguente formato:

<ID del sottosistema> <ID sensore> <Valore della media mobile>

NOTA BENE

1. Tutti i parametri del vostro modello devono essere contenuti nel record `Prm` nel file `parameters.mo`. Oltre a quelli menzionati nel testo dell'esercizio potete aggiugnere dei vostri parametri, ma non dovete in alcun caso rimuovere quelli che ci sono poichè vengono usati per la correzione.
2. Il modello `System` nel file `system.mo` deve essere esteso come serve, ma non devono essere rimosso il contenuto già presente poichè viene usato per la correzione.
3. Potete aggiungere file a vostra discrezione ed estendere a vostra discrezione il contenuto dei file che vi sono forniti.
4. Salvo esplicita istruzione in senso contrario, non potete modificare in alcun modo il contenuto già presente nei file che vi sono forniti. Questi vengono usati per interfacciarsi con gli script di correzione. Una modifica delle interfacce fornite rende impossibile la correzione e quindi l'esercizio riceverà 0 punti.
5. Prima di consegnare accertarsi che il vostro modello compili. I modelli che non compilano ricevono 0 punti.
6. Prima di consegnare accertarsi che l'orizzonte di simulazione sia maggiore del valore di *time* quando il comando `terminate` termina la simulazione. Se questo non accade la simulazione è sbagliata e l'esercizio riceve 0 punti.
7. Prima di consegnare accertarsi che il file `outputs.txt` (se richiesto) sia presente e valorizzato come richiesto. Se questo file manca oppure è vuoto l'esercizio riceve 0 punti.