

Automazione

13 settembre 2021

Esercizio 1

Si consideri un sistema di automazione in cui, a livello di coordinamento, è necessario gestire le attività di un drone operante nel settore dell'agricoltura di precisione. Le attività sono definite dai seguenti task periodici:

- ogni 4 t.u. eseguire la geo-referenziazione del drone, impiegando 1 t.u.;
- ogni 16 t.u. eseguire l'analisi della scansione termica, impiegando 4 t.u.;
- ogni 8 t.u. eseguire l'analisi della scansione fotografica, impiegando 1 t.u.

Ai task suddetti si aggiunge un task aperiodico di verifica della sensoristica di bordo del drone che interviene con un intervallo minimo di 32 t.u. e che dura al massimo 8 t.u. Tutti i task fin qui descritti devono essere gestiti con una modalità di scheduling hard real time.

Il sistema di automazione viene supervisionato da un tecnico il quale, attraverso un'interfaccia uomo-macchina costituita da un touch screen, può visualizzare una schermata di stato del drone. La prima occorrenza di tale task aperiodico ha activation time $a_5(1) = 20$ t.u., computation time $C_5(1) = 6$ t.u., e deadline assoluta $D_5(1) = 32$ t.u. Rispetto ai task precedenti, questo task aperiodico deve essere servito con una modalità di scheduling soft real time.

Si chiede di risolvere i seguenti punti.

1. Verificare le condizioni necessarie e sufficienti per lo scheduling secondo RMPO.
2. Mostrare il risultato dello scheduling RMPO dei task da gestire in modalità hard real time sommato ad un servizio in background FIFO per gestire le occorrenze del task soft real time.
3. Determinare se il task soft real time viene eseguito entro la deadline assoluta.

Esercizio 2

Mostrare che la rete di Petri in Fig. 1 può andare in deadlock. Verificare che un supervisore che imponga i seguenti due vincoli sulle marcature raggiungibili

$$x(p_2) + x(p_5) \leq 1, \quad x(p_4) \leq 1,$$

risolve il problema, garantendo anche la reversibilità della rete supervisionata. Progettare tale supervisore e disegnare la rete di Petri risultante.

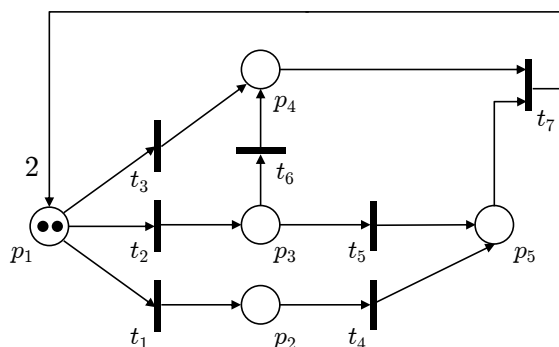


Figura 1: Una rete di Petri.

[60 minuti; libri aperti]

Soluzione

5 luglio 2021

Esercizio 1

La verifica di non inammissibilità si effettua calcolando il fattore di utilizzazione dei task periodici hard real time:

$$U = \frac{1}{4} + \frac{3}{16} + \frac{1}{8} + \frac{8}{32} = \frac{8 + 6 + 4 + 8}{32} = \frac{26}{32} = 0.8125 < 1.$$

Verificata la condizione necessaria, controlliamo se esiste almeno una condizione sufficiente:

$$U_{lsm}(RMPO) = n \left(2^{1/n} - 1 \right) = 4 \left(2^{1/4} - 1 \right) \simeq 0.7568.$$

Dato che $U > U_{lsm}$ questa condizione sufficiente non è verificata. Si può però notare che i 4 task sono legati tra loro da relazioni armoniche. Pertanto, abbiamo individuato una condizione sufficiente. La soluzione dello scheduling RMPO + SERVIZIO IN BACKGROUND FIFO è riportata in Fig. 2. Da questa si evince che il task soft real time può essere portato a termine entro la deadline assoluta.

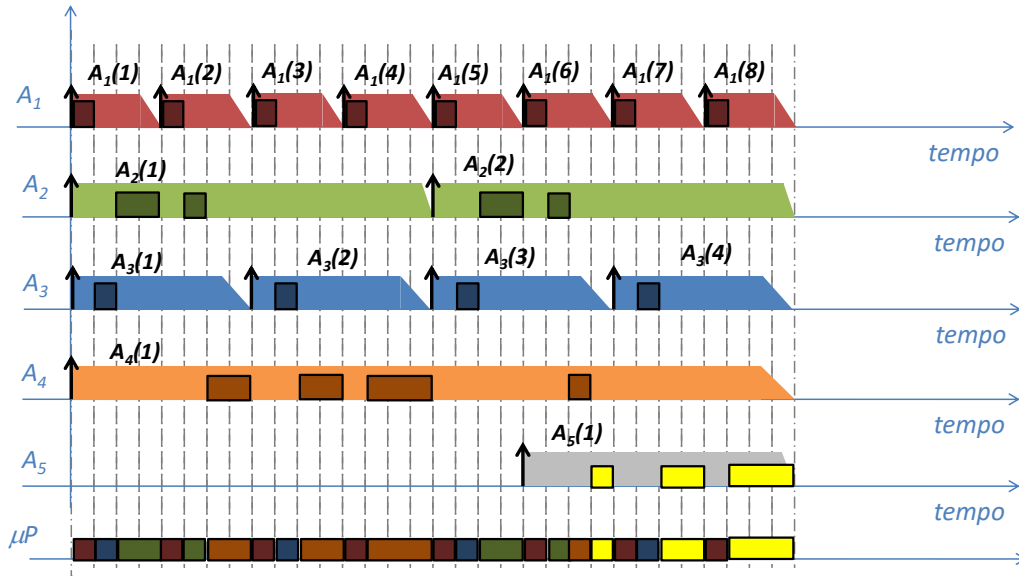


Figura 2: Scheduling ottenuto.

Esercizio 2

La matrice di incidenza e la marcatura iniziale della rete di Petri di Fig. 1 sono

$$C = \begin{pmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{pmatrix}, \quad x_0 = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Una possibile sequenza di deadlock è data da

$$\mathbf{x}_0 \rightarrow t_3 \rightarrow \mathbf{x}_1 = (1 \ 0 \ 0 \ 1 \ 0)^T \rightarrow t_3 \rightarrow \mathbf{x}_{dead} = (0 \ 0 \ 0 \ 2 \ 0)^T.$$

I vincoli di disuguaglianza da imporre evitano che i due token si accumulino tutti su una parte della rete. Il progetto del supervisore che impone tali disuguaglianze su tutte le marcature raggiungibili della rete è definito dalle relazioni

$$\begin{aligned} h_1^T \mathbf{x} &= (0 \ 1 \ 0 \ 0 \ 1) \mathbf{x} = x(p_2) + x(p_5) \leq 1 = k_1, \\ h_2^T \mathbf{x} &= (0 \ 0 \ 0 \ 1 \ 0) \mathbf{x} = x(p_4) \leq 1 = k_2, \end{aligned}$$

da cui

$$\mathbf{C}_m = \begin{pmatrix} C_{m1} \\ C_{m2} \end{pmatrix} = \begin{pmatrix} -h_1^T \mathbf{C} \\ -h_2^T \mathbf{C} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 \end{pmatrix}$$

e le marcature iniziali dei due posti monitor

$$x_0(p_{m1}) = k_1 - h_1^T \mathbf{x}_0 = 1, \quad x_0(p_{m2}) = k_2 - h_2^T \mathbf{x}_0 = 1.$$

La rete di Petri così supervisionata è mostrata in Fig. 3.

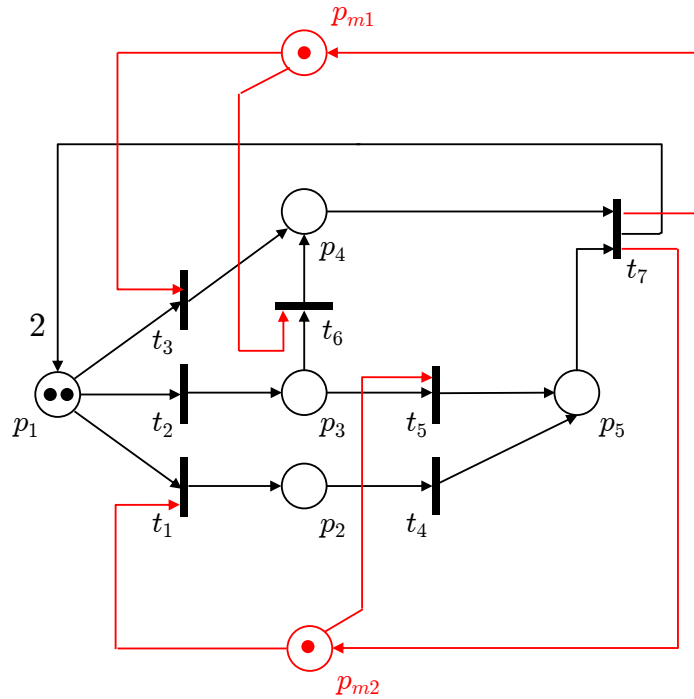


Figura 3: La rete di Petri di Fig. 1 supervisionata per evitare deadlock. I posti e gli archi del supervisore sono riportati in rosso. La rete complessiva è ora anche reversibile.

E' facile verificare che tutte le sequenze ammissibili di scatto (ad esempio, $\{t_1, t_2, t_4, t_6, t_7\}$) riportano la rete nello stesso stato iniziale esteso (reversibilità)

$$(\mathbf{x}_0^T \ \mathbf{x}_{m,0}^T) = (2 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1).$$
