

Automazione

18 marzo 2021

Esercizio 1

In Fig. 1 è mostrata l'uscita rilevata sperimentalmente di un processo stabile asintoticamente (che si assume abbia una dinamica approssimativamente lineare) in risposta a un gradino in ingresso di ampiezza $U = 3$, applicato all'istante $t = 0$ a partire da una situazione di riposo. Si richiede di progettare un regolatore di tipo PID sull'errore di uscita che garantisca il mantenimento della stabilità asintotica ad anello chiuso, errore nullo a regime rispetto ad arbitrari riferimenti costanti, tempo di assestamento simile a quello ad anello aperto e transitorio non eccessivamente oscillatorio. Procedere a una sintesi dei guadagni del regolatore basata sulle regole di Ziegler-Nichols, fornendo i valori numerici dei guadagni. Quale può essere un passo di campionamento consono nel caso di una implementazione digitale del regolatore?

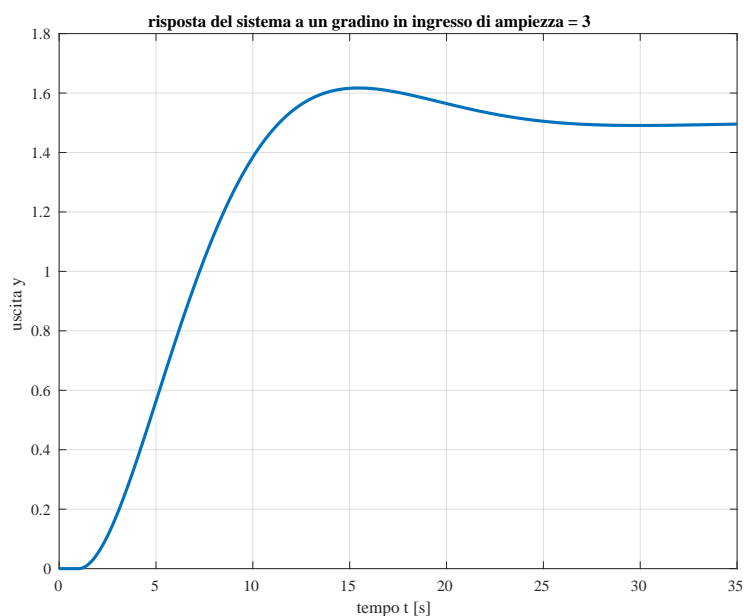


Figura 1: Risposta del processo ad anello aperto a un ingresso a gradino di ampiezza $U = 3$.

Esercizio 2

Si consideri un sistema di automazione di una azienda farmaceutica in cui, a livello di coordinamento, è necessario gestire i seguenti task periodici:

- ogni 4 t.u. viene effettuata l'azione di riempimento delle fiale, impiegando 1 t.u.;
- ogni 6 t.u. viene effettuata l'azione di stampa delle etichette delle fiale in 2 t.u.;
- ogni 9 t.u. viene eseguita l'azione di confezionamento delle fiale in 3 t.u.

C'è inoltre un task aperiodico di controllo della qualità, che ha un activation time 2 t.u., una deadline relativa di 30 t.u., e un computation time di 3 t.u. Si ipotizzi che tutti i task descritti siano indipendenti, dal punto di vista funzionale, l'uno dall'altro. I task periodici devono essere gestiti con una modalità di scheduling hard real time. Si chiede di risolvere i seguenti punti.

1. Verificare se sussiste la condizione necessaria di schedulabilità dei task periodici.
2. Verificare se sussiste almeno una delle condizioni sufficienti di schedulabilità, utilizzando l'algoritmo RMPO.
3. Eseguire lo scheduling RMPO. Nel caso in cui RMPO non sia in grado di schedulare i task periodici in maniera hard real time, eseguire lo scheduling utilizzando l'algoritmo EDF.
4. Utilizzando l'algoritmo di scheduling scelto nel punto precedente, verificare se il task aperiodico riesce ad essere eseguito in maniera hard real time utilizzando:
 - uno scheduling in background;
 - oppure, un processo polling server caratterizzato da $TSRV = 12$ t.u. e $CSRVR = 1$ t.u.;
 - oppure, un processo deferring server caratterizzato da $TSRV = 12$ t.u. e $CSRVR = 1$ t.u.

Esercizio 3

Una cella flessibile è dotata di due macchine M_1 e M_2 che possono lavorare tutti i tipi di pezzi che arrivano in ordine casuale in ingresso, con tempi di lavorazione variabili e non fissati a priori. La logica di assegnazione dei pezzi alle due macchine è molto semplice: un pezzo viene assegnato alla macchina M_1 se questa è libera; viene invece assegnato alla macchina M_2 se questa è libera e la macchina M_1 è occupata. Si suppone che ci siano sempre pezzi da lavorare, ma che questi entrino nella cella attraverso un buffer di capacità unitaria (quindi solo quando il pezzo entrato in precedenza è assegnato a una delle due macchine). Modellare il funzionamento della cella mediante una rete di Petri che utilizzi, se necessario, anche archi *inibitori* e sia opportunamente inizializzata con una marcatura x_0 . La logica di (dis)abilitazione delle transizioni da parte di archi inibitori è riassunta per comodità in Fig. 2. Anche in caso di presenza di tali archi è possibile parlare di reversibilità, vivezza e limitatezza della rete, come pure determinare tutte le marcature raggiungibili da x_0 . Definire quindi la rete di Petri che modella il processo di assegnazione dei pezzi alle macchine della cella e la loro lavorazione in modo che la rete sia limitata, viva e reversibile, costruendo il relativo albero di raggiungibilità e l'insieme delle marcature $\mathcal{R}(PN)$ per verificare tali proprietà.

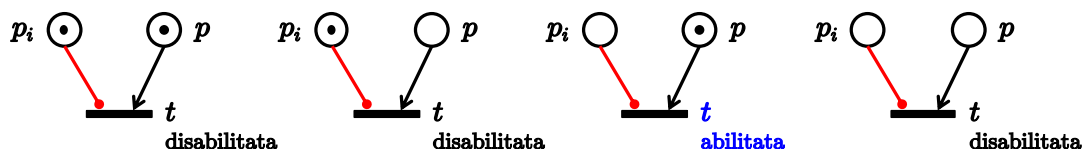


Figura 2: Illustrazione dell'abilitazione di una transizione in presenza di archi inibitori.

Al termine si consideri la seguente estensione. Sia presente nella cella anche una terza macchina M_3 con le stesse caratteristiche delle altre due. La logica di assegnazione dei pezzi è completata in tal caso dalla seguente regola: un pezzo viene assegnato alla macchina M_3 se questa è libera e le macchine M_1 e M_2 sono entrambe occupate. Modificare la rete di Petri precedente per gestire anche questa situazione.

[150 minuti; libri aperti]

Soluzione

18 marzo 2021

Esercizio 1

Dal grafico della risposta in Fig. 1 si ottengono i parametri

$$\text{guadagno } K = \frac{y_\infty}{U} = \frac{1.5}{3} = 0.5, \quad \text{ritardo finito } \theta = 2.5 \text{ [s]}, \quad \text{costante di tempo } \tau = 5.5 \text{ [s]},$$

da inserire nel modello di riferimento usato nel progetto dei guadagni dei regolatori PID secondo la prima regola di Ziegler-Nichols,

$$P_{rif}(s) = \frac{K}{1 + \tau s} e^{-\theta s}. \quad (1)$$

Il confronto tra l'andamento dell'uscita di tale modello e del processo originale in risposta allo stesso ingresso è mostrato in Fig. 3. Il grafico tratteggiato in rosso è quello utilizzato per ricavare i valori dei parametri di progetto K , θ e τ .

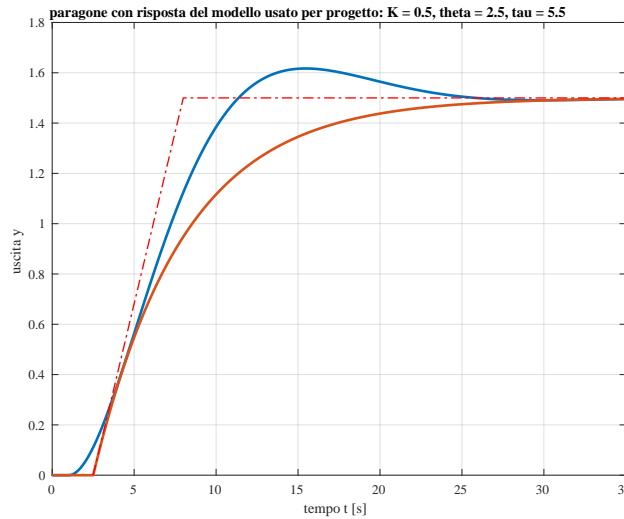


Figura 3: Risposte del processo e del modello (1) (in rosso) a un gradino di ampiezza $U = 3$.

Il processo originale non contiene un'azione integrale, altrimenti l'uscita forzata da un ingresso costante, al netto della parte transitoria, sarebbe cresciuta indefinitamente in modo lineare. Quindi per avere errore nullo a regime nel sistema ad anello chiuso è necessario avere un'azione integrale nel regolatore. Si opta inoltre per un regolatore che includa anche un'azione derivativa (eventualmente filtrata) in modo da poter limitare le possibili oscillazioni nel transitorio. Il regolatore sarà quindi un PID completo, con tutte e tre le azioni proporzionale, integrale e derivativa:

$$u = K_p e + K_i \int e(\sigma) d\sigma + K_d \frac{de}{dt} = K_p \left[e + \frac{1}{T_i} \int e(\sigma) d\sigma + T_d \frac{de}{dt} \right], \quad e = y_d - y. \quad (2)$$

Dalla tabella del primo metodo di Ziegler-Nichols per un regolatore PID si ha

$$K K_p = 1.2 \left(\frac{\theta}{\tau} \right)^{-1}, \quad \frac{T_i}{\tau} = 2 \left(\frac{\theta}{\tau} \right), \quad \frac{T_d}{\tau} = 0.5 \left(\frac{\theta}{\tau} \right),$$

da cui si ricavano i seguenti valori dei guadagni:

$$K_p = 5.28, \quad T_i = 5 \Rightarrow K_i = \frac{K_p}{T_i} = 1.056, \quad T_d = 1.25 \Rightarrow K_d = K_p T_d = 6.6.$$

La Fig. 4 mostra la risposta in uscita a un gradino di ampiezza $U = 3$ del sistema ad anello chiuso risultante. L'errore a regime è nullo mentre le altre specifiche di massima sul comportamento transitorio sono ragionevolmente soddisfatte. La Fig. 5 mostra invece lo sforzo di controllo u dato dalla (2). Poiché le dinamiche in gioco ad anello chiuso hanno costanti di tempo dell'ordine di 8-10 secondi (intervallo tra i primi due picchi, positivo e negativo, della risposta al gradino), un passo di campionamento di un ordine di grandezza più piccolo ($T_c \simeq 0.8 \div 1$ [s]) risulta consona.

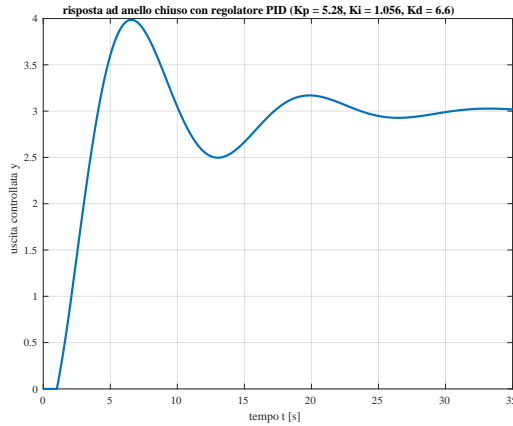


Figura 4: Risposta del sistema ad anello chiuso a un gradino di ampiezza $U = 3$.

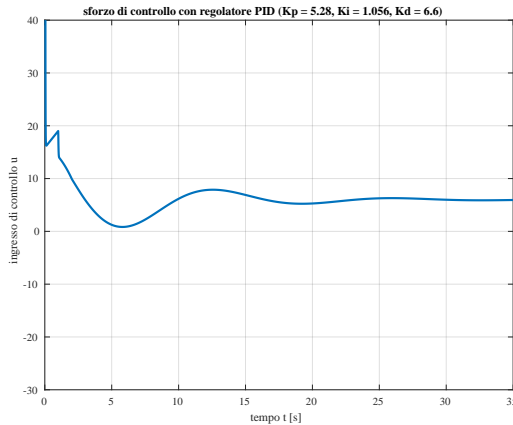


Figura 5: Sforzo di controllo ad anello chiuso nella risposta a un gradino di ampiezza $U = 3$.

Nota: Nel caso si desiderasse fare altre prove in simulazione, il processo originario $P(s)$ che si è considerato è un sistema lineare del secondo ordine con un ritardo finito in catena diretta, chiuso in un anello di retroazione unitaria. La sua funzione di trasferimento complessiva è

$$G(s) = \frac{k_g}{(1 + \tau_g s)^2} e^{-\theta_g s} \quad \Rightarrow \quad P(s) = \frac{G(s)}{1 + G(s)} = \frac{k_g e^{-\theta_g s}}{k_g e^{-\theta_g s} + (1 + \tau_g s)^2},$$

con $k_g = 1$, $\tau_g = 5$ [s] e $\theta_g = 1$ [s]. Il guadagno del processo è $k_P = P(0) = k_g / (k_g + 1) = 0.5$ (come già ricavato dalla risposta in Fig. 1).

Esercizio 2

Per verificare la condizione necessaria si calcola il fattore di utilizzazione dei task periodici hard real time:

$$U = \frac{1}{4} + \frac{2}{6} + \frac{3}{9} = \frac{9 + 12 + 12}{36} = \frac{33}{36} \simeq 0.92.$$

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

$$U_{lsm}(RMPO) = n \left(2^{1/n} - 1 \right) = 3 \left(2^{1/3} - 1 \right) \simeq 0.78.$$

Dato che $U > U_{lsm}$ questa condizione sufficiente non è verificata. Inoltre i tre task non sono legati tra loro da relazioni armoniche. Non possiamo quindi dire a priori se RMPO è in grado di schedulare i task. La soluzione dello scheduling RMPO è riportata in Fig. 6. Da questa si evince che RMPO non è in grado di schedulare in maniera hard real time i task periodici come richiesto.

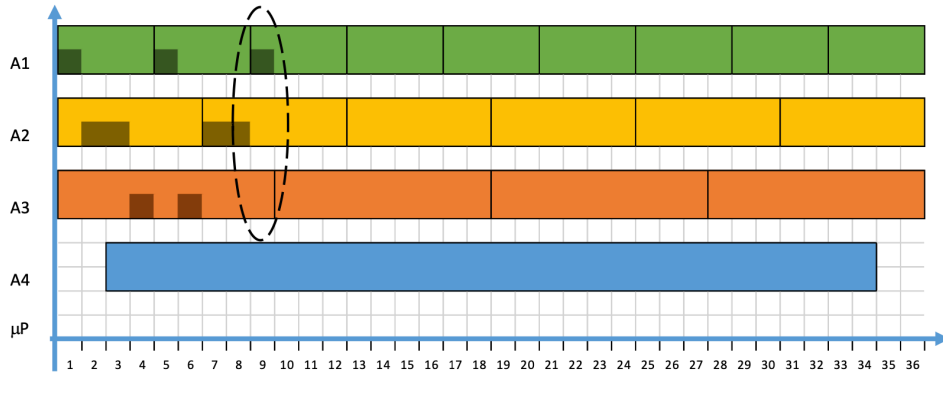


Figura 6: Scheduling RMPO.

La soluzione dello scheduling EDF è riportata in Fig. 7. Il task aperiodico (A4) non riesce ad essere eseguito in tempo se si utilizza una politica in background.

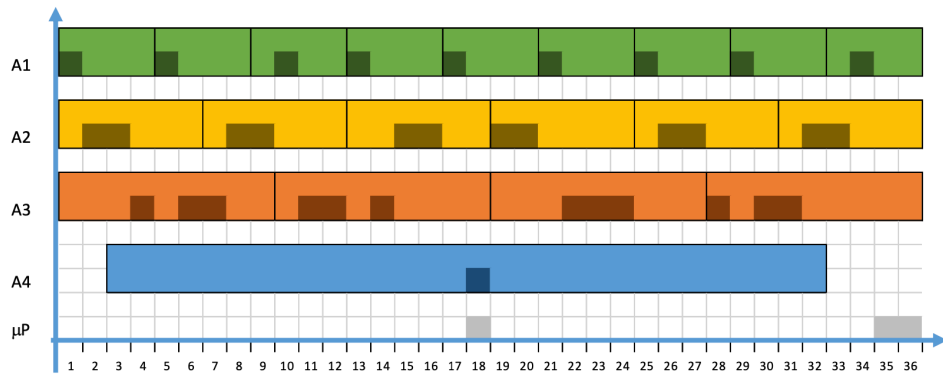


Figura 7: Scheduling EDF con politica in background del task aperiodico.

Aggiungendo il processo server (ASRV), il fattore di utilizzazione diventa

$$U = \frac{1}{4} + \frac{2}{6} + \frac{3}{9} + \frac{1}{12} = \frac{9 + 12 + 12 + 3}{36} = \frac{36}{36} = 1.$$

Verificata la condizione necessaria e sufficiente per EDF, la soluzione dello scheduling è riportata in Fig. 8. Da cui si evince che le 3 t.u. del task aperiodico non possono essere eseguite entro la deadline assoluta.

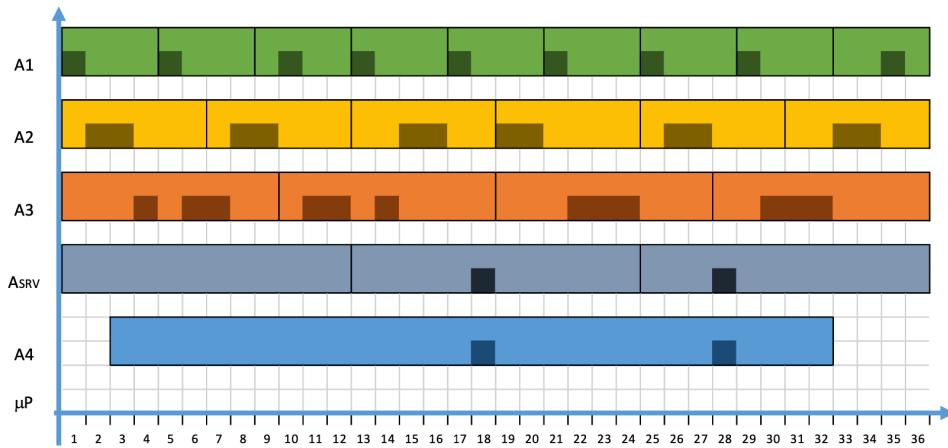


Figura 8: Scheduling EDF con politica polling server.

Infine, se invece di utilizzare una politica Polling Server, si utilizza una politica Deferring Server, allora si ottiene lo scheduling di Fig. 9. Da questa si evince che le 3 t.u. del task aperiodico possono essere eseguite entro la deadline assoluta.

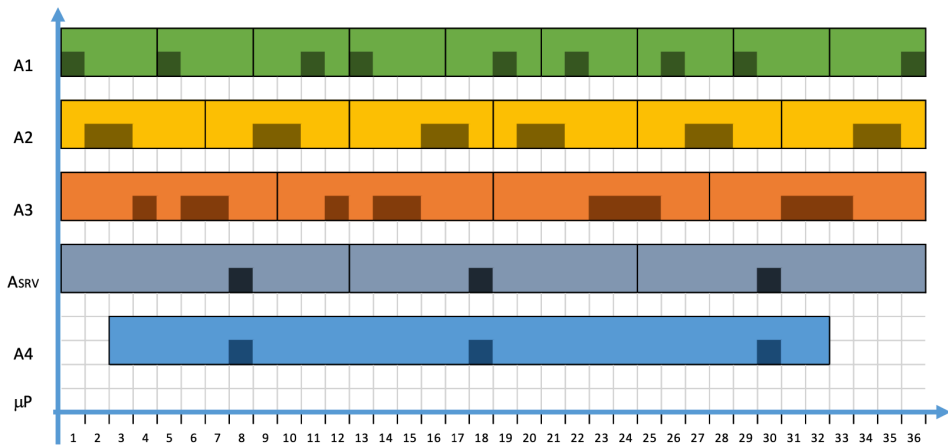


Figura 9: Scheduling EDF con politica deferring server.

Esercizio 3

La rete di Petri in Fig. 10 modella in modo accurato la logica di assegnazione dei pezzi e la loro lavorazione nella cella con 2 macchine. Si utilizzano 6 posti, 5 transizioni e due archi inibitori. Posti e transizioni hanno il seguente significato:

- p_1 = pezzo in attesa di entrare nella cella (token sempre presente)
- p_2 = pezzo nel buffer d'ingresso della cella in attesa di assegnazione
- p_3 = macchina M_1 libera
- p_4 = macchina M_1 occupata
- p_5 = macchina M_2 libera
- p_6 = macchina M_2 occupata
- t_1 = ingresso di un pezzo nella cella
- t_2 = pezzo assegnato alla macchina M_1 e inizio lavorazione
- t_3 = termine della lavorazione sulla macchina M_1
- t_4 = pezzo assegnato alla macchina M_2 e inizio lavorazione
- t_5 = termine della lavorazione sulla macchina M_2

L'arco inibitore dal posto p_2 alla transizione di ingresso t_1 tiene conto della capacità unitaria del buffer di ingresso che non permette a un nuovo pezzo di entrare nella cella. L'arco inibitore dal posto p_3 alla transizione t_4 realizza facilmente la priorità di impiego della macchina M_1 (se libera) rispetto alla macchina M_2 (quando libera). La marcatura iniziale

$$\mathbf{x}_0 = (1 \ 0 \ 1 \ 0 \ 1 \ 0)^T$$

corrisponde alla cella libera (le due macchine libere e il buffer vuoto) con un pezzo che richiede l'ingresso. infine, l'auto-anello tra il posto p_1 e la transizione t_1 e viceversa (in ingresso e uscita) risulta funzionale alla reversibilità della rete.

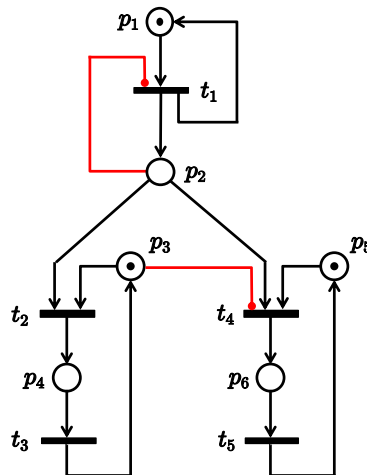


Figura 10: La rete di Petri con archi inibitori che modella la cella con due macchine M_1 e M_2 .

L'albero di raggiungibilità della rete di Petri a partire dalla marcatura \mathbf{x}_0 è mostrato in Fig. 11. Al solito, l'albero è costruito espandendo le transizioni abilitate in ordine lessicografico e procedendo in profondità (regola *depth first*). E' immediato verificare la limitatezza della rete (numero finito

di marcature raggiungibili con numero di token sempre limitato), la sua vivezza (ogni transizione può essere abilitata da una sequenza di scatto ammissibile a partire da qualsiasi marcatura raggiungibile) e la reversibilità (è possibile ritornare alla marcatura iniziale x_0 con una sequenza di scatto ammissibile a partire da qualsiasi marcatura raggiungibile).

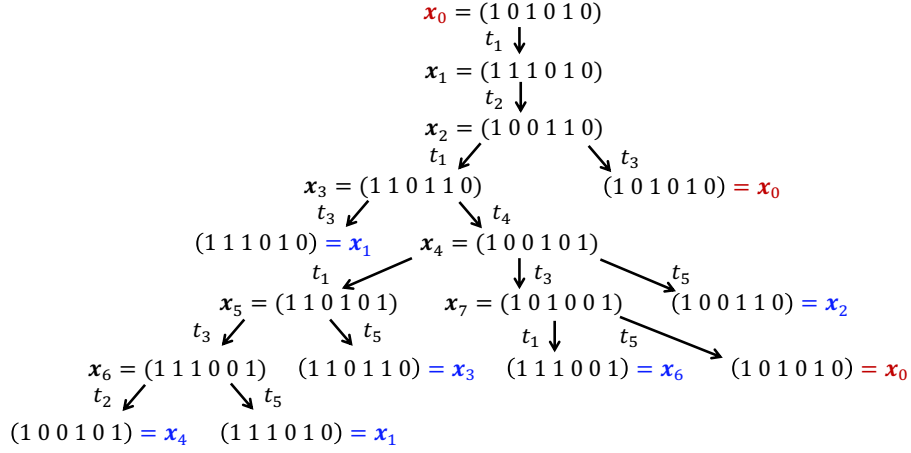


Figura 11: L'albero di raggiungibilità della rete di Petri in Fig. 10.

L'insieme delle marcature raggiungibili da x_0 è dato da

$$\mathcal{R}(PN) = \left\{ \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7

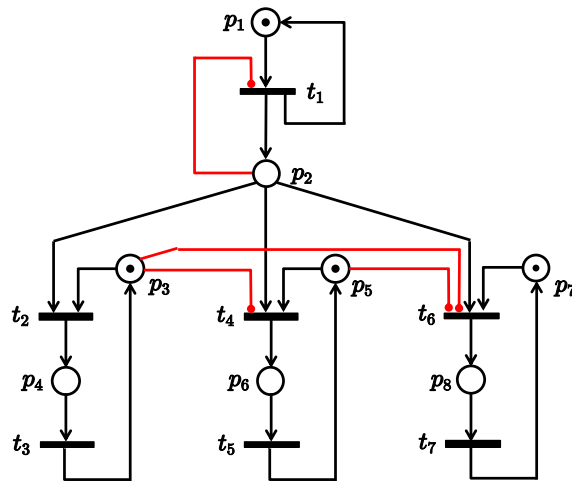


Figura 12: La rete di Petri che modella la cella con tre macchine M_1 , M_2 e M_3 .

L'estensione al caso di 3 macchine è a questo punto immediata. Basta aggiungere una struttura analoga per la macchina M_3 (2 posti e 2 transizioni, rispettivamente di macchina libera/occupata e inizio/fine lavorazione) e due archi inibitori della transizione di inizio lavorazione t_6 a partire dai posti p_3 e p_5 . La rete di Petri modificata è riportata in Fig. 12.

* * * * *