

# Automazione

19 gennaio 2024

**Esercizio 1** (gli studenti con un progetto saltano questo esercizio)

Si consideri un sistema di automazione SCADA (Supervisory Control And Data Acquisition) che opera a livello di conduzione e in cui è necessario analizzare:

- ogni 6 t.u., un pacchetto ricevuto sull'interfaccia di rete Ethernet, impiegando 1 t.u.;
- ogni 9 t.u., un pacchetto ricevuto sull'interfaccia di rete Wifi, impiegando 1 t.u.;
- ogni 12 t.u., un pacchetto ricevuto sull'interfaccia di rete Bluetooth, impiegando 5 t.u.;
- ogni 18 t.u., un pacchetto ricevuto sull'interfaccia seriale Modbus RS232, impiegando 5 t.u..

I suddetti task sono periodici e devono essere gestiti in modalità hard real time. C'è inoltre un task aperiodico da gestire in modalità soft real time e relativo all'interfaccia uomo-macchina HDMI, che ha un activation time di 1 t.u., una deadline relativa di 70 t.u., e un computation time di 2 t.u.. Si ipotizzi che tutti i task descritti siano indipendenti, dal punto di vista funzionale, l'uno dall'altro.

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 dei task periodici equivalenti, utilizzando l'algoritmo RMPO.
3. Eseguire lo scheduling dei task periodici equivalenti con RMPO.
4. Se RMPO non è soluzione del punto precedente, eseguire lo scheduling con EDF, altrimenti proseguire.
5. Utilizzando l'algoritmo di scheduling derivato dai punti precedenti verificare se il processore è completamente utilizzato, motivando la risposta.
6. Utilizzando l'algoritmo di scheduling derivato dai punti precedenti verificare se sia possibile gestire il task aperiodico in maniera tale che sia eseguito entro la deadline assoluta, applicando le seguenti politiche esattamente nell'ordine in cui sono proposte:
  - a) BACKGROUND;
  - b) POLLING SERVER così caratterizzato:  $TSRV = 36$  t.u. e  $CSRV = 1$  t.u.;
  - c) DEFERRING SERVER così caratterizzato:  $TSRV = 36$  t.u. e  $CSRV = 1$  t.u.

## Esercizio 2

Un robot collaborativo lavora in una cella nella quale un (unico) utente può entrare liberamente. La cella è monitorata da un laser scanner (con apertura a  $360^\circ$ ) posto alla base del robot. Questo sensore misura ogni  $T = 0.5$  secondi la distanza  $r$  dell'utente dal robot (a una data quota dal pavimento). Inoltre, una fotocellula segnala l'accesso o l'uscita dell'utente dalla cella. Se l'utente è fuori dalla cella o è all'interno ma a distanza superiore a  $r_{\max}$  (**zona verde**), il robot svolge le sue operazioni come da programma (*normal operation*). Per distanze intermedie,  $r \in [r_{\min}, r_{\max}]$  (**zona arancione**), il robot ha una ridotta velocità operativa (*slow motion*). Se la distanza è ravvicinata, ossia  $r < r_{\min}$  (**zona rossa**), il robot si ferma (stato di *stop*, con il controllo attivo). Se il sensore laser continua a fornire letture a distanza ravvicinata, il robot passa in uno stato di *hold* (in cui i motori vengono frenati meccanicamente). Per motivi di sicurezza, letture anomale del sensore laser (come un 'salto' di distanza misurata dalla zona verde alla rossa) o della fotocellula (ad esempio, due rilevazioni di ingresso senza uscita intermedia) devono porre il robot in uno stato terminale di *abort* (freni attivati, motori non alimentati; è necessario un intervento per riavviare il sistema).

Descrivere il comportamento del sistema (robot, utente e sensori) mediante automi modulari, identificando i relativi stati ed eventi. Sintetizzare quindi un supervisore come automa di Mealy avente i soli stati complessivi ammissibili, con gli eventi associati in ingresso (informazioni sensoriali) e in uscita (di comando per il robot). Nello stato iniziale, l'utente è fuori dalla cella e il robot è in condizioni operative normali.

### Esercizio 3

Si consideri la rete di Petri ordinaria in Fig. 1.

- Questa rete appartiene a una classe particolare? Se sì, quale?
- Dimostrare che la rete è viva, illimitata e reversibile, utilizzando gli strumenti analitici di studio di tali proprietà strutturali.
- Progettare un supervisore basato sui P-invarianti che renda la rete limitata, senza perdere le proprietà di vivezza e reversibilità.
- Per la rete così supervisionata, costruire l'albero di raggiungibilità.

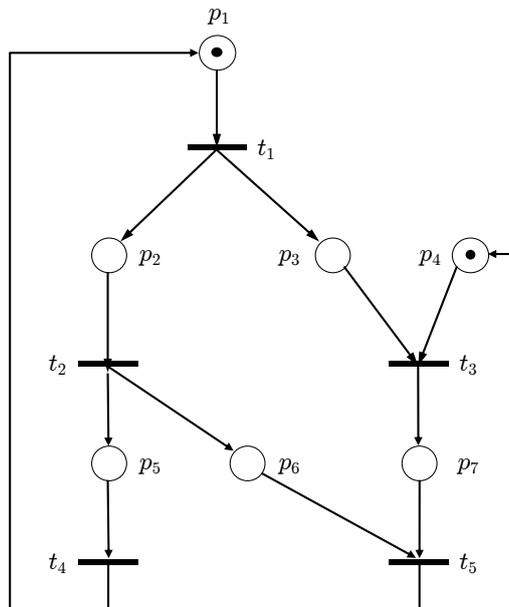


Figura 1: Una rete di Petri ordinaria con 7 posti e 5 transizioni.

### Esercizio 4

La Fig. 2 riporta la risposta indiciale ad anello aperto del processo

$$P(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2 + 2.5s + 0.25}, \quad (1)$$

che risulta priva di oscillazioni, ma molto lenta (oltre 50 s di transitorio) e con un errore a regime ( $y_\infty = 4 \neq 1 = y_d$ ). Si utilizza quindi un primo anello di controllo  $u = K_t(v - y)$ , proporzionale all'errore rispetto al nuovo ingresso di riferimento  $v$ , con guadagno  $K_t = 1$ . La Fig. 3 mostra la risposta a un gradino unitario  $v$  del sistema ottenuto (si nota una riduzione del tempo di risposta).

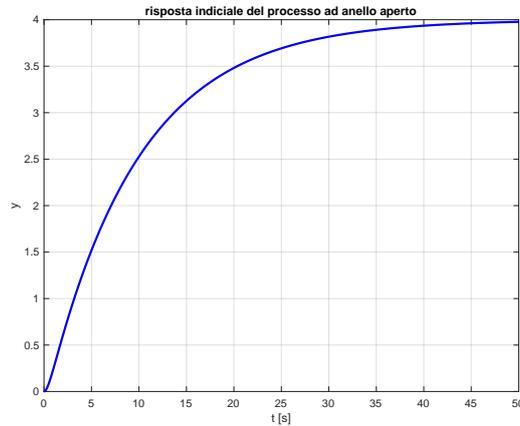


Figura 2: Risposta indiciale del processo  $P(s)$  ad anello aperto.

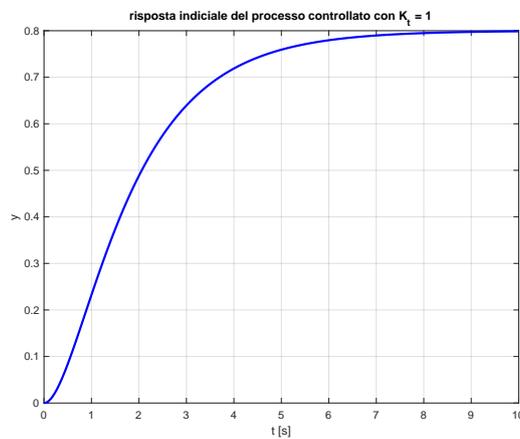


Figura 3: Risposta indiciale in presenza di un primo regolatore proporzionale con  $K_t = 1$ .

- Per il nuovo ingresso  $v$ , si progetti un regolatore di tipo PID (o una sua sottoclasse) in modo che l'uscita  $y$  assuma a regime permanente il valore desiderato  $y_d$ , con ridotta sovravelazione e limitate oscillazioni della risposta transitoria.
- Utilizzando MATLAB, fornire l'andamento della risposta indiciale del sistema finale ottenuto.
- Come si comporta a regime permanente il sistema così controllato se il riferimento è una rampa unitaria ( $y_d = t$ )? L'errore diverge, converge a un valore finito o si annulla?
- Per riprodurre asintoticamente in uscita l'andamento tempo-variante  $y_d(t) = A(1 - \cos\omega t)$ , con  $A > 0$  e  $\omega > 0$ , quale segnale di feedforward  $u_{ffw}(t)$  occorre aggiungere all'ingresso originale di controllo  $u$ ? Qual è il valore iniziale  $u_{ffw}(0)$ ? Specificare le condizioni iniziali che rendono la riproduzione di  $y_d(t)$  esatta fin dall'istante  $t = 0$ .

[4 ore (per tutti gli esercizi), 3 ore (senza Esercizio 1); libri aperti]

# Soluzione

19 gennaio 2024

## Esercizio 1

### Punto 1)

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

$$U = \frac{1}{6} + \frac{1}{9} + \frac{5}{12} + \frac{5}{18} = \frac{6 + 4 + 15 + 10}{36} = \frac{35}{36} = 0.97\bar{2} < 1.$$

Dal fattore di utilizzazione si può dedurre che ai task aperiodici soft real time può essere dedicata al massimo 1 t.u. ogni 36.

### Punto 2)

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

- sicuramente  $U > \ln 2 \simeq 0.69$ ;
- inoltre,  $U > U_{ism}(RMPO) = n(2^{1/n} - 1) = 4(2^{1/4} - 1) \simeq 0.76$ ;
- infine, i task non sono legati tra loro da relazioni armoniche.

Non possiamo quindi dire a priori se RMPO è in grado di schedulare i task.

### Punto 3)

L'esecuzione dello scheduling RMPO è mostrata in Fig. 4. Da questa si evince che RMPO non è in grado di schedulare in maniera hard real time i task periodici equivalenti, in quanto la prima istanza del task A4 non riesce a essere eseguita entro la deadline assoluta.

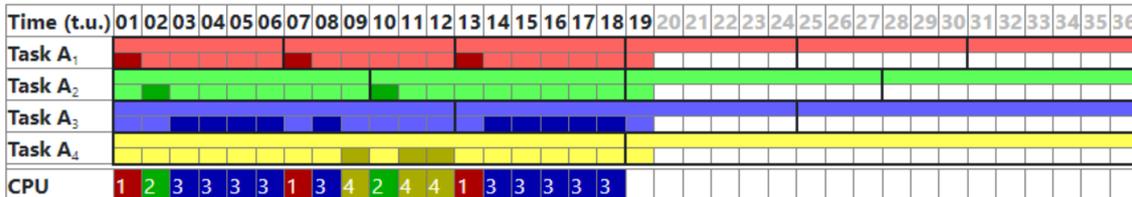


Figura 4: Scheduling con RMPO.

### Punto 4)

Dato che RMPO non è soluzione, eseguiamo lo scheduling con EDF riportato in Fig. 5.

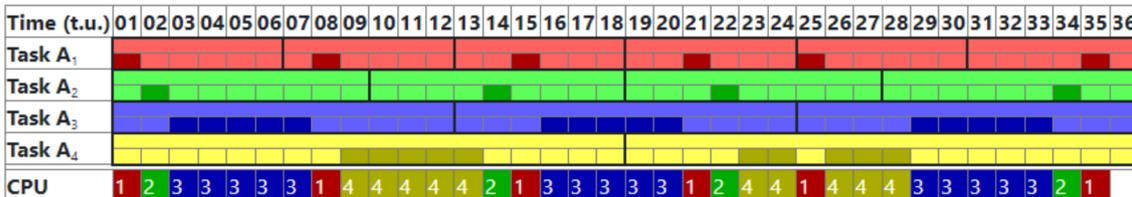


Figura 5: Scheduling con EDF.

**Punto 5)**

Per definizione, dato un algoritmo di scheduling e un insieme di task periodici, il processore viene detto **completamente utilizzato** se la schedulazione è fattibile e se un aumento comunque piccolo di uno qualsiasi dei computation time la rende impossibile. Osservando la soluzione del precedente punto, si può facilmente trovare un contro-esempio che violi la condizione di completa utilizzazione. Infatti, aumentando di un  $\epsilon$  sufficientemente piccolo uno qualsiasi dei computation time (in particolare, con  $\epsilon$  minore o uguale del reciproco del numero di occorrenze del task avente il maggior numero di occorrenze nella trama di scheduling; nel caso specifico, sono le 6 occorrenze del task A1 e pertanto  $\epsilon \leq 1/6$ ), lo schema viene semplicemente traslato senza che nessuna istanza di ogni task superi la propria deadline assoluta. Pertanto, il processore non è completamente utilizzato.

**Punto 6.a)**

Osservando la soluzione in Fig. 5 del punto 4), si nota che l'unica t.u. libera (su 36) è l'ultima. Quindi RMPO concederà alla politica di BACKGROUND la 36-esima t.u., poi la 72-esima t.u. e così via. Tuttavia il task aperiodico soft real time entra nella coda di attesa dopo una t.u. e ha una deadline assoluta di 71 t.u.. Pertanto la politica in BACKGROUND non permette di eseguire il task aperiodico entro la deadline assoluta, in quanto la seconda t.u. sarà messa a disposizione del servizio in background solo nel corso della 72-esima t.u..

**Punto 6.b)**

Dato che il task aperiodico ha activation time pari a 1 t.u. e che il processo server ha un periodo di attivazione di 36 t.u., la prima istanza del processo server avrà computation time nullo. Ma la seconda istanza del processo server va dalla t.u. 37 fino alla t.u. 72 e metterà a disposizione del task aperiodico solo una t.u. delle 2 che deve eseguire. Pertanto, anche la politica POLLING SERVER non permette di eseguire il task aperiodico entro la deadline assoluta.

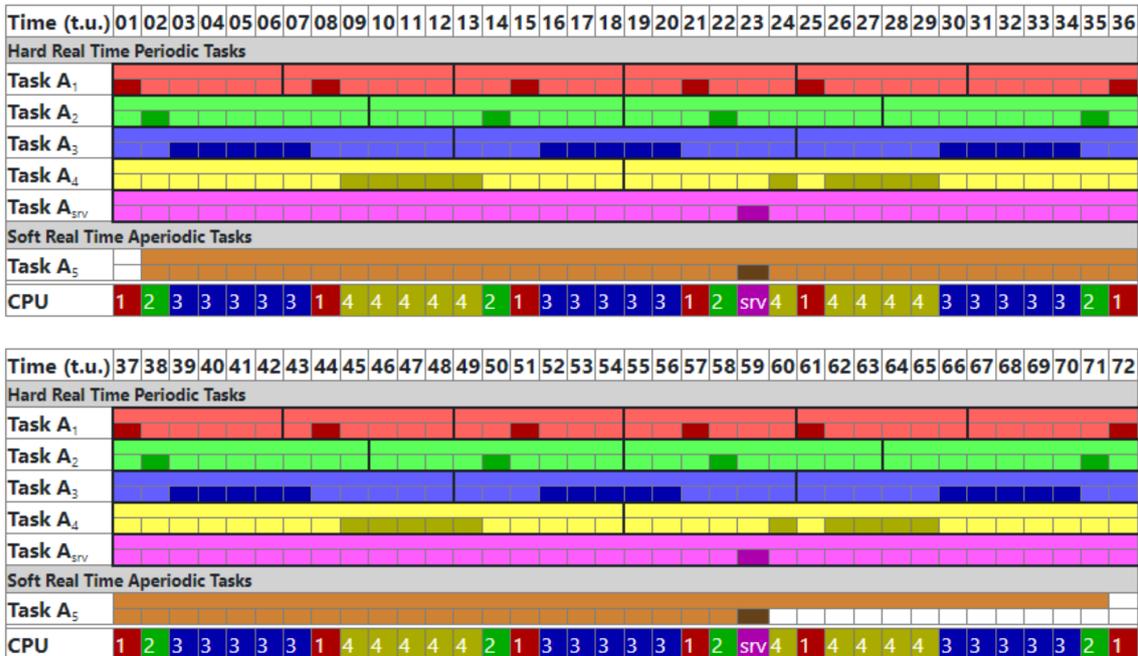


Figura 6: Scheduling con politica DEFERRING SERVICE.

**Punto 6.c)**

Con una politica di tipo DEFERRING SERVER si ottiene il risultato in Fig. 6, da cui si evince che la tale politica permette di eseguire il task aperiodico entro la deadline assoluta.

**Esercizio 2**

Si considerano inizialmente due automi,  $A_{robot}$  che descrive gli stati del robot (Fig. 7) e  $A_{utente}$  che descrive lo stato dell'utente, ossia la zona in cui si trova rispetto al robot (Fig. 8). Le transizioni del primo automa sono guidate dalle misure (da  $u_1$  a  $u_4$ ) del solo sensore laser, mentre quelle del secondo automa sono guidate anche dai segnali ( $u_5$  e  $u_6$ ) della fotocellula di accesso alla cella.

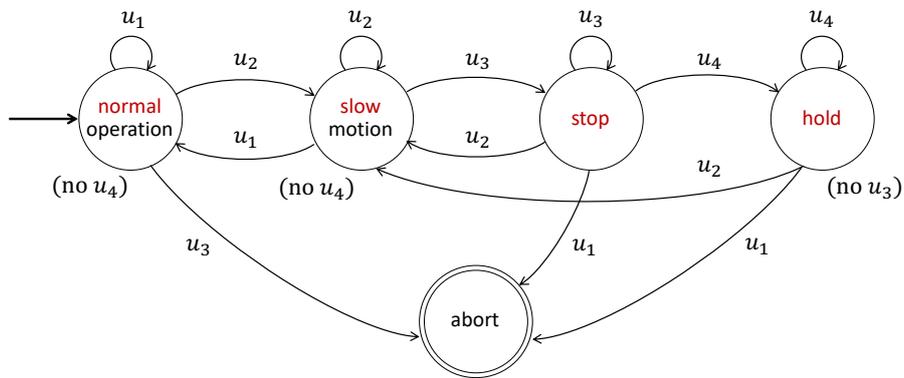


Figura 7: Automa del robot.

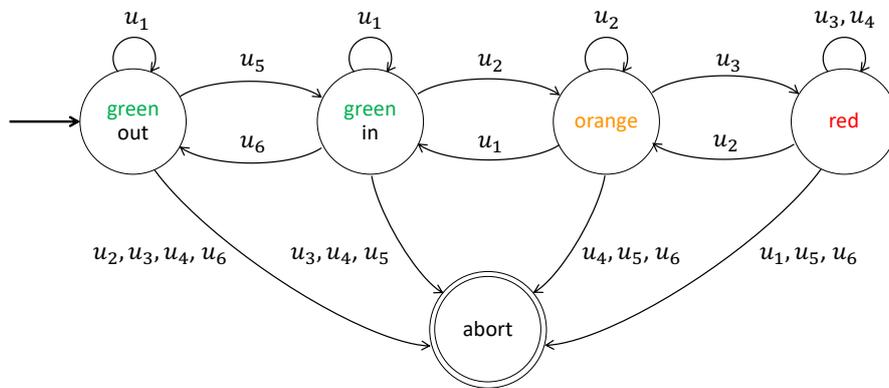


Figura 8: Automa della localizzazione dell'utente.

In particolare, gli stati di  $A_{robot}$  sono così definiti:

- *normal operation*: è lo stato di normale funzionamento del robot;
- *slow motion*: è lo stato in cui la velocità di esecuzione dei compiti è opportunamente rallentata;
- *stop*: il robot è fermo sotto un comando di controllo (con i motori alimentati);
- *hold*: il robot è fermo, con i freni attivati (senza comando di controllo);
- *abort*: è lo stato terminale in cui il robot si pone, con freni attivati e motori non alimentati.

Gli stati di  $A_{\text{utente}}$  sono invece i seguenti:

- *green out*: l'utente è all'esterno della cella;
- *green in*: l'utente è all'interno della cella, ma a distanza  $r > r_{\max}$  dal robot;
- *orange*: l'utente è all'interno della cella, a una distanza  $r \in [r_{\min}, r_{\max}]$  dal robot;
- *red*: l'utente è all'interno della cella, vicino al robot ( $r < r_{\min}$ );
- *abort*: è lo stato terminale in cui l'utente non collabora più con il robot a causa di errori.

Gli eventi sensoriali  $u_i$ ,  $i = 1, \dots, 6$ , sono i seguenti:

- $u_1$ : il laser legge una distanza  $r > r_{\max}$  (o un valore convenzionale  $r_{\infty} \gg r_{\max}$ , se l'utente è fuori);
- $u_2$ : il laser legge una distanza  $r \in [r_{\min}, r_{\max}]$ ;
- $u_3$ : il laser legge una distanza  $r < r_{\min}$ ;
- $u_4$ : il laser fornisce un numero  $\geq 2$  di letture consecutive a distanza ravvicinata ( $r < r_{\min}$ );
- $u_5$ : la fotocellula segnala l'ingresso dell'utente nella cella;
- $u_6$ : la fotocellula segnala l'uscita dell'utente dalla cella.

Si noti che l'evento  $u_3$  non è compatibile con stato *hold* nell'automa  $A_{\text{robot}}$ , perché il robot si trova in questo stato solo quando ci siano già state ripetute letture  $r < r_{\min}$  (quindi, eventi  $u_4$ ). In modo simile, l'evento  $u_4$  non è compatibile con gli stati *normal operation* e *slow motion*. Si vedrà che anche questi eventi incompatibili sono gestiti nell'automa supervisore e si traducono in situazioni di errore dei sensori che portano allo stato di *abort* delle normali operazioni.

La Fig. 9 mostra la sintesi del supervisore, nella forma di un automa di Mealy. Gli stati ammissibili  $S_i$ ,  $i = 1, \dots, 6$ , sono un sottoinsieme del prodotto cartesiano di  $5 \times 5 = 25$  possibili stati per il sistema integrato dei due automi modulari. Il significato dei singoli stati è evidente.

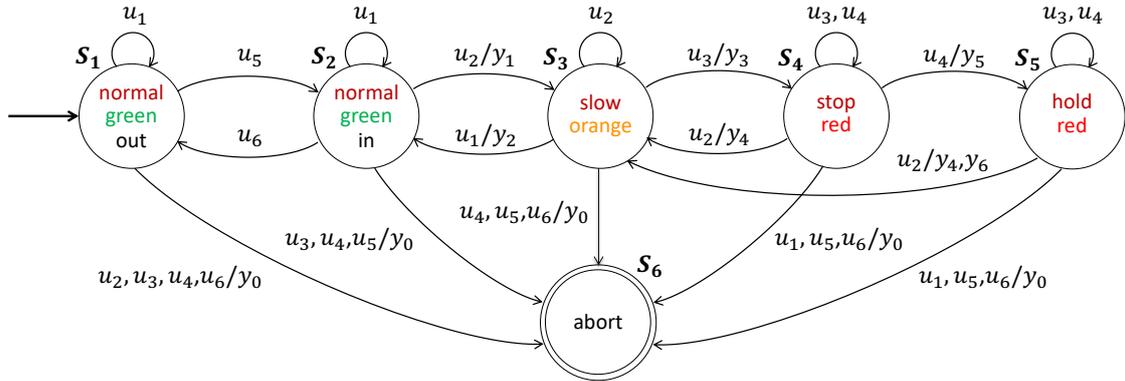


Figura 9: Automa supervisore della cella collaborativa.

L'automa contiene anche gli eventi di uscita  $y_i$ ,  $i = 0, \dots, 6$ , così definiti:

- $y_0$ : comando di arresto di tutte le operazioni del robot, con distacco dell'alimentazione dei motori e attivazione dei freni (quest'ultima non necessaria dallo stato  $S_5$ ) — il supervisore si porta nello stato terminale  $S_6 = \text{abort}$ ;
- $y_1$ : comando di riduzione della velocità operativa del robot a un valore di sicurezza;

- $y_2$ : comando di ripristino della normale velocità operativa del robot;
- $y_3$ : comando di arresto del moto del robot (gestito dalla legge di controllo);
- $y_4$ : comando di riavvio del moto del robot a velocità operativa ridotta;
- $y_5$ : comando di attivazione dei freni (con robot già fermo);
- $y_6$ : comando di disattivazione dei freni.

Si noti che in ogni stato  $\mathbf{S}_i$  del supervisore **tutti** gli eventi in ingresso (dati provenienti dai sensori) sono ammissibili. Le misure dal laser scanner o i segnali dalla fotocellula che sono incompatibili con lo stato corrente del supervisore sono trattati come errori e generano quindi, come richiesto, il comando  $y_0$  che abortisce le operazioni del robot. Due transizioni di stato (da  $\mathbf{S}_1$  a  $\mathbf{S}_2$  e viceversa) sono prive di evento in uscita (il supervisore non genera nessun comando per il robot). Infine la transizione dallo stato  $\mathbf{S}_5$  allo stato  $\mathbf{S}_3$  è associata ai due comandi  $y_4$  e  $y_6$ , i quali vengono in pratica eseguiti in sequenza stretta: prima si disattivano i freni ( $y_6$ ), poi si riavvia il moto del robot a velocità ridotta ( $y_4$ ).

### Esercizio 3

La rete di Petri in Fig. 1 è un grafo marcato: ogni posto ha al più una sola transizione in ingresso e una sola in uscita. Poiché nella marcatura iniziale  $\mathbf{x}_0 = (1, 0, 0, 1, 0, 0, 0)$  è presente (almeno) un token in ogni ciclo di posti ( $\{p_1, p_2, p_5\}$  e  $\{p_4, p_7\}$ ), possiamo già concludere la *vivezza* della rete.

La matrice di incidenza  $\mathbf{C}$  della rete, di dimensioni  $p \times t = 7 \times 5$ , è

$$\mathbf{C} = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 \end{pmatrix}.$$

È facile verificare (anche senza MATLAB) che la matrice  $\mathbf{C}$  ha rango  $\rho = 4$ : sommando infatti tutte le colonne si ottiene il vettore  $\mathbf{0}$  (le 5 colonne non sono quindi linearmente indipendenti); cancellando poi l'ultima colonna e le ultime tre righe di  $\mathbf{C}$ , si ottiene una sotto-matrice  $4 \times 4$  (la  $\mathbf{C}(1:4, 1:4)$ ) il cui determinante è pari a 1.

È inoltre immediato il calcolo dell'unico T-invariante (essendo  $t - \rho = 5 - 4 = 1$ ):

$$\mathbf{C} \boldsymbol{\eta} = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{\eta} = \mu (1 \ 1 \ 1 \ 1 \ 1)^T, \quad \mu \in \mathbb{N}^+. \quad (2)$$

Per avere reversibilità è quindi necessario che tutte le transizioni scattino un numero uguale di volte. Una possibile sequenza ammissibile di transizioni a partire da  $\mathbf{x}_0$  che riporta alla marcatura iniziale è  $\{t_1, t_2, t_4, t_3, t_5\}$ . Tuttavia questa situazione deve sussistere a partire da tutte le marcature raggiungibili della rete; occorre quindi approfondire l'analisi.

Per determinare ulteriori proprietà, calcoliamo i P-invarianti della rete. Si ha

$$\boldsymbol{\gamma}^T \mathbf{C} = \mathbf{0}^T \quad \Leftrightarrow \quad \mathbf{C}^T \boldsymbol{\gamma} = \mathbf{0} \quad \Leftrightarrow \quad \begin{aligned} \gamma_2 + \gamma_3 - \gamma_1 &= 0 \\ \gamma_5 + \gamma_6 - \gamma_2 &= 0 \\ \gamma_7 - \gamma_3 - \gamma_4 &= 0 \\ \gamma_1 - \gamma_5 &= 0 \\ \gamma_4 - \gamma_6 - \gamma_7 &= 0. \end{aligned} \quad (3)$$

Essendo la dimensione dello spazio nullo di  $C^T$  pari a  $p - \rho = 7 - 4 = 3$ , proviamo a esprimere tutte le soluzioni non banali di queste 5 equazioni lineari omogenee in termini di tre parametri interi e non negativi,  $\alpha$ ,  $\beta$  e  $\delta$ , che possiamo assegnare rispettivamente alle tre componenti

$$\gamma_2 = \alpha, \quad \gamma_3 = \delta, \quad \gamma_4 = \beta.$$

Dalla prima, terza e quarta equazione in (3), ricaviamo quindi

$$\gamma_1 = \gamma_2 + \gamma_3 = \alpha + \delta, \quad \gamma_7 = \gamma_3 + \gamma_4 = \delta + \beta, \quad \gamma_5 = \gamma_1 = \alpha + \delta.$$

Dalla quinta equazione, utilizzando anche l'espressione di  $\gamma_7$ , si ha però

$$\gamma_6 = \gamma_4 - \gamma_7 = \gamma_4 - (\gamma_3 + \gamma_4) = -\gamma_3.$$

Poiché siamo interessati solo a soluzioni intere e non negative, da questa identità segue necessariamente che  $\gamma_6 = \gamma_3 = 0$  e quindi  $\delta = 0$ . Tutte le soluzioni di interesse sono pertanto parametrizzate dai soli due parametri  $\alpha$  e  $\beta$  come

$$\boldsymbol{\gamma}^T = ( \alpha \quad \alpha \quad 0 \quad \beta \quad \alpha \quad 0 \quad \beta ), \quad \alpha, \beta \in \mathbb{N}^+.$$

La terza dimensione dello spazio nullo di  $C^T$  non contiene soluzioni diverse da zero che abbiano le componenti non nulle appartenenti a  $\mathbb{N}$ .

Due P-invarianti canonici e a supporto minimo sono quindi

$$\boldsymbol{\gamma}_A^T = ( 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 ), \quad \boldsymbol{\gamma}_B^T = ( 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 ).$$

Gli insiemi di supporto di questi P-invarianti rivelano che i posti  $p_3$  e  $p_6$  non sono ricoperti e che quindi vi si potrebbero accumulare token in numero illimitato. Viceversa, dalle due equazioni di invarianza

$$\boldsymbol{\gamma}_A^T \mathbf{x} = x_1 + x_2 + x_5 = 1 = \boldsymbol{\gamma}_A^T \mathbf{x}_0, \quad \boldsymbol{\gamma}_B^T \mathbf{x} = x_4 + x_7 = 1 = \boldsymbol{\gamma}_B^T \mathbf{x}_0, \quad \forall \mathbf{x} \in R(\text{PN}),$$

ritroviamo che il numero di token nei due cicli  $\{p_1, p_2, p_5\}$  e  $\{p_4, p_7\}$  si conserva ed è pari a 1 in ciascuno di essi.

Dall'analisi svolta, si possono trarre le seguenti conclusioni.

- La rete è *illimitata* nei posti  $p_3$  e  $p_6$ : una sequenza ammissibile di transizioni che accumula esattamente  $h$  token in questi due posti, con  $h$  arbitrariamente grande, è data dalla  $\{t_1, t_2, t_4\}^h$ .
- La rete è *reversibile*. Consideriamo infatti una qualsiasi marcatura raggiungibile nella quale si siano accumulati un numero generico e arbitrariamente grande  $h$  di token sia nel posto  $p_3$  che nel posto  $p_6$ . Contemporaneamente, a causa della conservatività, deve essere presente uno e un solo token in ciascuno dei cicli  $\{p_1, p_2, p_5\}$  e  $\{p_4, p_7\}$ . Senza perdita di generalità, immaginiamo di essere nella marcatura  $\bar{\mathbf{x}} = (1, 0, h, 1, 0, h, 0)$ . Il ritorno alla marcatura iniziale  $\mathbf{x}_0 = (1, 0, 0, 1, 0, 0, 0)$  avviene con la sequenza ammissibile di transizioni  $\{t_3, t_5\}^h$ .
- I due passi precedenti verificano evidentemente la condizione necessaria di reversibilità ottenuta con il T-invariante  $\boldsymbol{\eta}$ , che richiede un numero  $\mu$  di scatti uguale per tutte le transizioni; nella situazione considerata, si avrà  $\mu = 2h$  in (2).

Un supervisore che limita a un valore finito il numero di token nei posti  $p_3$  e  $p_6$  senza portare la rete in deadlock si può ottenere con la tecnica dei P-invarianti, ovvero imponendo un'opportuna

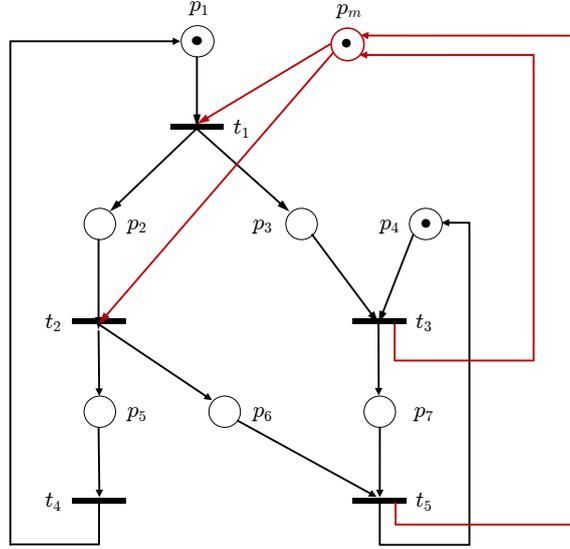


Figura 10: La rete di Petri di Fig. 1 con il supervisore progettato.

disuguaglianza che elimina le marcature indesiderate dall'insieme delle marcature raggiungibili. Una possibile soluzione si ottiene imponendo

$$x_3 + x_6 \leq 1, \quad \forall \mathbf{x} \in R(\text{PN}). \quad (4)$$

Si noti che la marcatura iniziale  $\mathbf{x}_0$  soddisfa (strettamente) tale disuguaglianza. La disuguaglianza (4) si riscrive nella forma generale

$$\mathbf{h}^T \mathbf{x} = (0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0) \mathbf{x} \leq 1 = k, \quad \forall \mathbf{x} \in R(\text{PN}). \quad (5)$$

Il supervisore si traduce in un singolo posto monitor  $p_m$  collegato alle transizioni della rete con archi orientati, come specificato dalla riga aggiuntiva della matrice di incidenza calcolata come

$$\mathbf{c}_m = -\mathbf{h}^T \mathbf{C} = (-1 \ -1 \ 1 \ 0 \ 1)$$

e con un numero iniziale di token dato da

$$x_{m,0} = k - \mathbf{h}^T \mathbf{x}_0 = 1 - 0 = 1.$$

La rete supervisionata risultante è riportata in Fig. 10, dove il posto monitor e gli archi aggiunti sono indicati in colore. La rete risultante non è più un grafo marcato, ma una rete a scelta libera (Free Choice (FC)) di tipo asimmetrico: i posti con transizioni in uscita (insiemi  $O(p_i)$ ) in comune sono  $p_1$  con  $p_m$  e  $p_2$  con  $p_m$ , ma valgono le inclusioni  $O(p_1) = \{t_1\} \subset \{t_1, t_2\} = O(p_m)$  e  $O(p_2) = \{t_2\} \subset O(p_m)$ .

L'albero di raggiungibilità della rete supervisionata è riportato in Fig. 11, da cui è immediato verificare la vivezza, limitatezza e reversibilità della soluzione ottenuta.

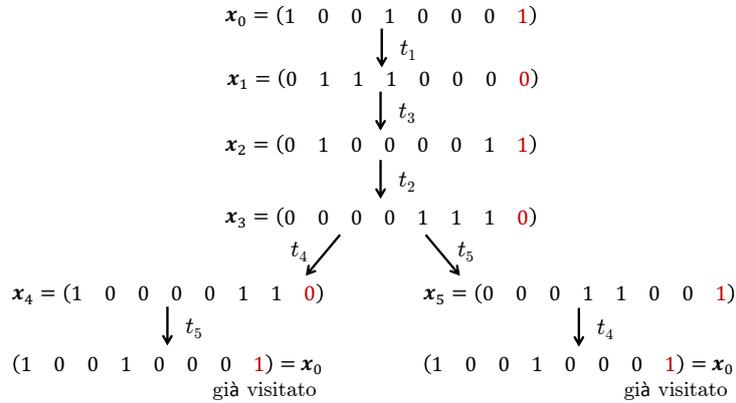


Figura 11: Albero di raggiungibilità della rete di Petri di Fig. 10.

#### Esercizio 4

Il processo  $P(s)$  in (1) ha due poli reali in  $p_1 = -0.1044$  (polo dominante lento) e in  $p_2 = -2.3956$ . Dopo la chiusura del primo anello proporzionale con  $K_t = 1$ , il sistema diventa

$$P'(s) = \frac{y(s)}{v(s)} = \frac{K_t P(s)}{1 + K_t P(s)} = \frac{K_t n_P(s)}{d_P(s) + K_t n_P(s)} = \frac{1}{s^2 + 2.5s + 1.25}, \quad (6)$$

con poli reali ma spostati in  $p'_1 = -0.6910$  (polo dominante reso più rapido) e in  $p'_2 = -1.8090$ .

Il sistema ha ancora un errore a regime permanente ( $e_\infty = 0.2$ ) e quindi necessita dell'azione integrale nel regolatore, che sarà quindi di tipo PI o PID. L'aggiunta del termine derivativo può servire a ridurre la sovraelongazione e/o le oscillazioni nel transitorio, anche se ritarda in generale il raggiungimento della condizione di regime. Lo schema a blocchi del sistema di controllo risultante è mostrato in Fig. 12.

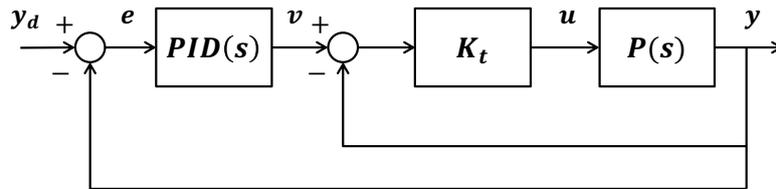


Figura 12: Schema a blocchi del sistema di controllo.

Dalla risposta al gradino unitario sull'ingresso  $v$  riportata in Fig. 3, si deducono i parametri necessari per la scelta dei guadagni utilizzando il 1° metodo di Ziegler-Nichols. Con riferimento alla Fig. 13 (che contiene una maschera sovrapposta alla Fig. 13), si ha che:

- il valore a regime  $y_\infty$  dell'uscita fornisce il guadagno  $K = 0.8$ ;
- in assenza di un istante di attraversamento del valore di regime, per la costante di tempo dominante  $\tau$  si può utilizzare il tempo di salita dell'uscita tra l'istante in cui assume il 10% del valore di regime ( $t_1 = 0.5$  s) e quello in cui raggiunge il 90% del valore di regime ( $t_2 = 4$  s), ottenendo  $\tau = t_2 - t_1 = 3.5$  s;
- di conseguenza, un'approssimazione del tempo di ritardo finito è calcolata come  $\theta = t_1 = 0.5$  s.

Dalle tabelle si ottengono quindi i seguenti valori dei guadagni:

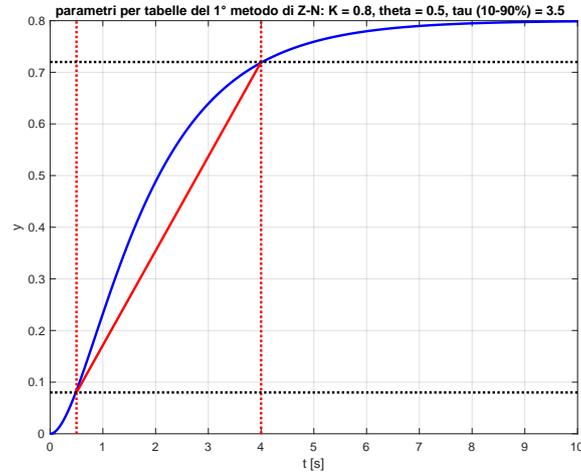


Figura 13: Valutazione dei parametri  $K$ ,  $\tau$  e  $\theta$  per il 1° metodo di Ziegler-Nichols.

- per un regolatore PI della forma

$$v_{PI}(t) = K_P e(t) + K_I \int_0^t e(s) ds, \quad e(t) = y_d - y(t), \quad (7)$$

si ha

$$K_P = \frac{0.9}{K \left( \frac{\theta}{\tau} \right)} = 7.875, \quad T_i = 3.33 \left( \frac{\theta}{\tau} \right) \tau = 1.665 \Rightarrow K_I = \frac{K_P}{T_i} = 4.7297;$$

- per un regolatore PID (ideale) della forma

$$v_{PID}(t) = K_P e(t) + K_I \int_0^t e(s) ds + K_D \frac{de(t)}{dt}, \quad e(t) = y_d - y(t), \quad (8)$$

$$K_P = \frac{1.2}{K \left( \frac{\theta}{\tau} \right)} = 10.5, \quad T_i = 2 \left( \frac{\theta}{\tau} \right) \tau = 1 \Rightarrow K_I = \frac{K_P}{T_i} = 10.5,$$

$$T_d = 0.5 \left( \frac{\theta}{\tau} \right) \tau = 0.25 \Rightarrow K_D = K_P T_d = 2.625,$$

La risposta indiciale del sistema complessivo è riportata nelle Figg. 14 e 15, rispettivamente quando si utilizza il regolatore PI o quello PID (ideale). Si può notare il miglioramento ottenuto in termini di oscillazioni (assenti in Fig. 15) e di massima sovraelongazione (ridotta da 1.30 a 1.13) grazie all'aggiunta dell'azione derivativa.

Questi grafici si possono facilmente ottenere in MATLAB con le seguenti istruzioni, ad esempio per il regolatore PID e la Fig. 15:

```
% processo già con il primo controllore proporzionale (vedi eq. (6))
nPprime=[1]; dPprime=[1 2.5 1.25];
% parametri Z-N
K=0.8; theta=0.5; tau=3.5;
```

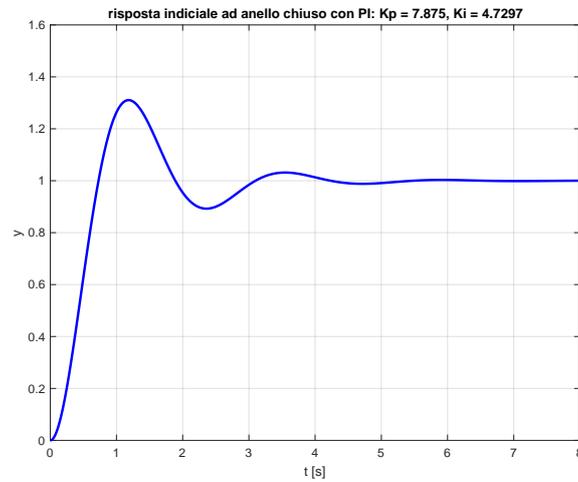


Figura 14: Risposta indiciale del sistema finale con un regolatore PI.

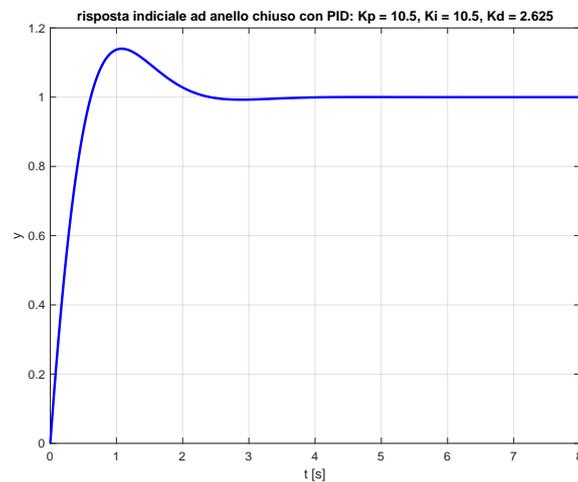


Figura 15: Risposta indiciale del sistema finale con un regolatore PID (ideale).

```

tht=theta/tau;
% valori guadagni PID dalla tabella Z-N
Kp=1.2/(K*tht);
Ti=2*tht*tau; Ki=Kp/Ti;
Td=0.5*tht*tau; Kd=Kp*Td;
% controllore
nC_PID=Ki*[Ti*Td Ti 1], dC_PID=[1 0]
% sistema ad anello chiuso
nW=nPprime*nC_PID, dW=conv(dPprime,dC_PID)+[0 conv(nPprime,nC_PID)]
% plot
tw=[0:0.01:8];
yw=step(nW,dW,tw);
plot(tw,yw,'b','Linewidth',2);grid; xlabel('t [s]');ylabel('y');
title(['risposta indiciale ad anello chiuso con PID: Kp = ',num2str(Kp),', Ki = ',num2str(Ki),', Kd = ',num2str(Kd)])

```

Per quanto riguarda il comportamento del sistema finale in risposta a un riferimento a rampa unitaria ( $y_d = t$ ), l'errore a regime permanente convergerà a un valore finito e diverso da zero; il sistema di controllo è infatti di tipo 1, grazie all'integratore nel regolatore. Tale errore a regime si può calcolare dalla (6) e, ad esempio, dalla (8) come

$$e_{\text{rampa},\infty} = \frac{1}{K_{G,P'} \cdot K_{G,PID}} = \frac{1}{0.8 \cdot 10.5} = 0.119,$$

dove  $K_{G,F}$  è il guadagno di una funzione di trasferimento  $F(s)$ .

Infine, per riprodurre almeno asintoticamente l'andamento  $y_d(t) = A(1 - \cos \omega t)$  in uscita al sistema, il segnale di feedforward<sup>1</sup>  $u_{\text{ffw}}(t)$  si calcola invertendo il processo originario  $P(s)$  nel dominio di Laplace e poi antitrasformando nel tempo. Si ha

$$u_{\text{ffw}}(s) = P^{-1}(s) y_d(s) = (s^2 + 2.5s + 0.25) y_d(s),$$

da cui

$$\begin{aligned} u_{\text{ffw}}(t) &= \ddot{y}_d(t) + 2.5 \dot{y}_d(t) + 0.25 y_d(t) \\ &= A \omega^2 \cos \omega t + 2.5 A \omega \sin \omega t + 0.25 A (1 - \cos \omega t), \end{aligned}$$

il cui valore iniziale è  $u_{\text{ffw}}(0) = A \omega^2$ .

Per ottenere poi la riproduzione di  $y_d(t)$  esatta fin dall'istante  $t = 0$ , lo stato iniziale del processo deve essere agganciato alle condizioni iniziali della traiettoria desiderata. In questo caso, essendo il processo del secondo ordine, la variabile di uscita  $y$  e la sua derivata prima  $\dot{y}$  possono essere prese come variabili di stato, ovvero  $\mathbf{x} = (y, \dot{y})$ . Poichè  $y_d(0) = \dot{y}_d(0) = 0$ , si ha che anche lo stato iniziale del processo deve essere interamente nullo:  $\mathbf{x}(0) = \mathbf{x}_d(0) = \mathbf{0}$ .

Con l'aggiunta del termine di feedforward, lo schema del sistema di controllo di Fig. 12 si modifica come mostrato in Fig. 16.

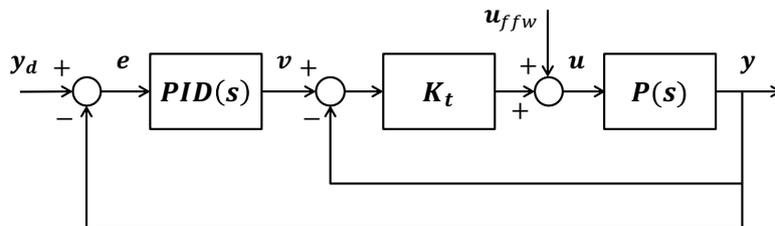


Figura 16: Schema a blocchi del sistema di controllo con l'aggiunta del termine di feedforward.

\* \* \* \* \*

<sup>1</sup>Il testo richiede l'aggiunta del feedforward direttamente sull'ingresso originario  $u$  al processo  $P(s)$  e non sul nuovo ingresso  $v$  al processo modificato  $P'(s)$ .