

# Automazione

20 gennaio 2025

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

Si consideri un sistema di automazione caratterizzato dai seguenti requisiti:

- un task periodico hard real time  $A_1$  deve essere eseguito ogni 24 millisecondi (ms) e impiega un massimo di 9 ms per essere portato a termine;
- un task periodico hard real time  $A_2$  deve essere eseguito ogni 15 ms e impiega un massimo di 6 ms per essere portato a termine;
- un task aperiodico hard real time  $A_3$  caratterizzato dalle cinque occorrenze passate in Tab. 1;

Occorrenza	1	2	3	4	5
Activation time	0	63	123	189	250
Computation time	12	11	9	7	12

Tabella 1: Dati storici del task  $A_3$  (tempi in ms).

- un task aperiodico soft real time  $A_4$  con  $a_4(1) = 6$  ms,  $D_4(1) = 186$  ms,  $C_4(1) = 6$  ms.

Ipotizzando che i task siano indipendenti, si chiede di risolvere i seguenti punti.

1. Spiegare come verrà gestito il task  $A_3$  aperiodico hard real time.
2. In base ai dati del problema, stabilire un valore (in ms) della t.u. da utilizzare nel corso dello svolgimento dell'esercizio.
3. Verificare se sussiste la condizione necessaria di schedulabilità dei task hard real time.
4. Verificare se sussiste almeno una delle condizioni sufficienti di schedulabilità dei task hard real time, utilizzando l'algoritmo RMPO.
5. Eseguire lo scheduling dei task hard real time con algoritmo RMPO. Nel caso in cui RMPO non sia in grado di schedulare i task hard real time, eseguire lo scheduling utilizzando l'algoritmo EDF.
6. Scrivere la definizione di “processore completamente utilizzato” e, utilizzando l'algoritmo di scheduling scelto nel punto 3, verificare se il processore risulti essere completamente utilizzato.
7. Utilizzando l'algoritmo di scheduling scelto nel punto 3, verificare se il task aperiodico soft real time riesce ad essere eseguito entro la propria deadline assoluta utilizzando uno scheduling in background. In caso ciò non sia possibile, verificare se il task aperiodico può essere eseguito entro la propria deadline assoluta utilizzando un processo deferring server caratterizzato da  $T_{SRV} = 120$  ms e  $C_{SRV} = 3$  ms.
8. Spiegare cosa sarebbe successo se il task aperiodico fosse stato eseguito utilizzando un processo polling server con gli stessi valori  $T_{SRV} = 120$  ms e  $C_{SRV} = 3$  ms.

**Esercizio 2**

Si consideri un semplice sistema di analisi di testi composti in lingua italiana che identifica tutte le parole che terminano in ONE (ad esempio, AUTOMAZIONE). In ingresso c'è una sequenza di simboli tratti da un insieme costituito dalle 21 lettere dell'alfabeto (A, B, ..., Z) e dal simbolo  $\Delta$  che indica uno spazio (o un segno di interpunkzione) tra due parole successive. L'uscita del sistema è binaria, con il valore 1 destinato a segnalare la presenza di una parola che termina in ONE e il valore 0 altrimenti. Descrivere il comportamento dinamico di questo sistema di identificazione mediante un automa di Mealy, con eventi in ingresso (uno dei 22 simboli) e in uscita (bit 0 o 1), a partire da uno stato iniziale neutro. Per compattezza, si indichi con W un qualsiasi evento in ingresso diverso dalle tre lettere O, N ed E, nonché dal simbolo  $\Delta$ .

### Esercizio 3

Si consideri la rete di Petri  $PN$  in Fig. 1.

- Costruire l'albero di raggiungibilità/copertura. La rete è reversibile? E' viva?
- Calcolare i  $P$ -invarianti canonici della rete. La rete è conservativa?
- Scrivere le equazioni di invarianza e determinare l'insieme delle soluzioni. Esistono in questo caso soluzioni di tutte le equazioni di invarianza che *non* sono marcature raggiungibili?
- Calcolare i  $T$ -invarianti della rete e individuare, se esistono, le relative sequenze di scatto ammissibili.
- Progettare due supervisori che impongano, per ogni marcatura  $\mathbf{x} \in R(PN)$ , il vincolo  $\mathbf{x}(p_2) \leq k$ , rispettivamente con  $k = 1$  e con  $k = 2$ . Studiare le proprietà delle due reti supervisionate così ottenute.

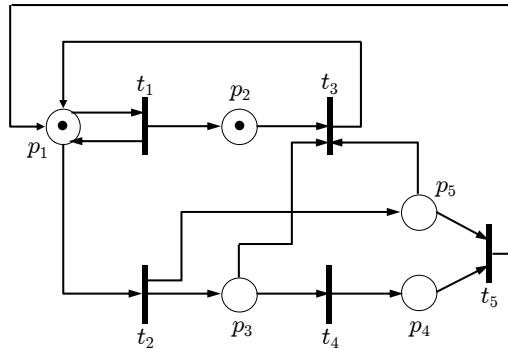


Figura 1: Una rete di Petri ordinaria con 5 posti e 5 transizioni.

### Esercizio 4

- i) La Fig. 2 mostra la risposta indiciale di un processo ad anello aperto. Volendo ottenere un sistema ad anello chiuso asintoticamente stabile, con errore nullo a regime in risposta a ingressi costanti e con transitori possibilmente smorzati, progettare un regolatore di tipo PID usando il primo metodo di Ziegler–Nichols.

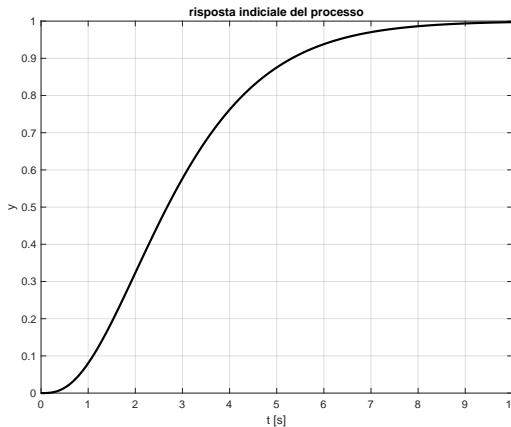


Figura 2: Risposta indiciale del processo ad anello aperto.

- ii) Viene successivamente ricavato anche il modello del processo, che risulta essere  $P(s) = 1/(s+1)^3$ . Progettare un nuovo regolatore di tipo PID con le stesse specifiche precedenti, usando questa volta il secondo metodo di Ziegler–Nichols. Commentare la struttura del regolatore progettato rispetto al precedente e in relazione al modello ricavato per il processo. E' possibile in questo caso una sintesi diretta del regolatore?

*Nota: La parte b) può svolgersi indifferentemente in modo analitico oppure utilizzando MATLAB.*

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

# Soluzione

20 gennaio 2025

## Esercizio 1

1. Dovendo gestire in maniera hard real time il task aperiodico  $A_3$ , è necessario verificare se sia possibile trasformarlo in un task periodico equivalente, studiandone i dati storici. Dall'analisi dei dati presenti in Tab. 1 è immediato notare che il task ha un minimo tempo di inter-arrivo di due consecutive occorrenze di 60 ms (tra l'occorrenza 2 e la 3) e il massimo tra i computation time di 12 ms (occorrenze 1 e 5). Pertanto, il task  $A_3$  può essere trattato alla stregua di un task periodico equivalente avente le seguenti caratteristiche:  $T_4 = 60$  ms,  $C_4 = 12$  ms.

2. Dovendo scegliere un valore (in millisecondi) da poter attribuire alla time unit (t.u.) da utilizzare nel corso dello svolgimento dell'esercizio, è immediato notare che tutti i dati del problema siano espressi come multipli di 3 ms. Pertanto, si può ipotizzare di stabilire che 1 t.u. = 3 ms.

3. Avendo stabilito che 1 t.u. = 3 ms, il problema di scheduling di task misti può essere trasformato in un problema equivalente caratterizzato dai seguenti task periodici:

- $T_1 = 8$  t.u.,  $C_1 = 3$  t.u.;
- $T_2 = 5$  t.u.,  $C_2 = 2$  t.u.;
- $T_3 = 20$  t.u.,  $C_3 = 4$  t.u.

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

$$U = \frac{3}{8} + \frac{2}{5} + \frac{4}{20} = \frac{15 + 16 + 8}{40} = \frac{39}{40} = 0.975.$$

Dato che  $U < 1$ , la condizione necessaria ( $U \leq 1$ ) è verificata.

4. Verificata la condizione necessaria, controlliamo se esiste almeno una condizione sufficiente. Dato che  $U > \ln 2$ , calcoliamo il limite superiore minimo per l'algoritmo RMPO per  $n$  task periodici:

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

Dato che  $U > U_{lsm}$  e che i 3 task non sono legati tra loro da relazioni armoniche, nessuna delle tre condizioni sufficienti è verificata e quindi non possiamo dire a priori se RMPO è in grado di schedulare i task.

5. La soluzione dello scheduling RMPO è riportata in Fig. 3.

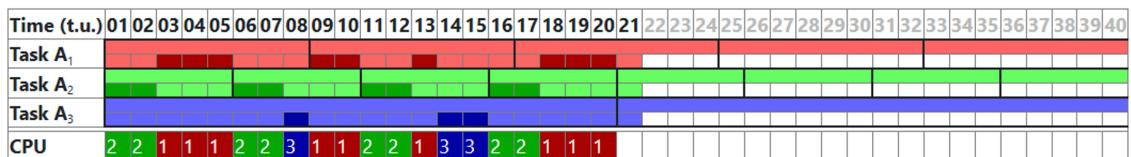


Figura 3: Scheduling con RMPO.

Da questa si evince che al termine della t.u. 20, la deadline del task  $A_3$  è mancata in quanto tale slot è stato riservato all'esecuzione del task  $A_1$ . Pertanto, RMPO non è in grado di schedulare in maniera hard real time i task come richiesto.

La soluzione dello scheduling EDF è riportata in Fig. 4.

Time (t.u.)	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
Task A <sub>1</sub>																																												
Task A <sub>2</sub>																																												
Task A <sub>3</sub>																																												
CPU	2	2	1	1	1	2	2	3	1	1	2	2	1	3	3	3	2	2	1	1	1	2	2	3	1	2	2	1	1	3	2	2	3	3	1	1	1	2	2					

Figura 4: Scheduling con EDF.

6. 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 rende la schedulazione impossibile. Osservando la soluzione dello scheduling EDF presentata in Fig. 4, si può facilmente trovare un contro-eSEMPIO che viola la condizione di completa utilizzazione. Infatti, aumentando di un  $\epsilon$  sufficientemente piccolo (in particolare, con  $\epsilon$  minore o uguale al reciproco del numero di occorrenze del task avente il maggior numero di occorrenze nella trama di scheduling, che nel caso specifico sono le 8 occorrenze del task A<sub>2</sub>, pertanto  $\epsilon \leq 1/8$  t.u. = 3/8 ms) uno qualsiasi dei computation time, lo schema viene semplicemente traslato senza che alcuna istanza di alcun task superi la propria deadline assoluta. Pertanto, il processore **non è completamente utilizzato**.

7. Dallo scheduling secondo l'algoritmo EDF in Fig. 4 si evince che il processore rimane inutilizzato in tutte le t.u. multiple di 40 (40, 80, 120, ecc.). Il task aperiodico soft real-time non riesce ad essere eseguito entro la sua deadline se si utilizza una politica in background, in quanto attivandosi alla t.u. 2, dovrà terminare entro la t.u. 64, ma in tale intervallo ricade solo una t.u. di processore libero e non 2 t.u. come richiesto dal task.

Aggiungendo il processo (deferring) server (A<sub>4</sub>), il coefficiente di utilizzazione diventa:

$$U = \frac{3}{8} + \frac{2}{5} + \frac{4}{20} + \frac{1}{20} = 1.$$

Verificata la condizione necessaria e sufficiente per EDF, la soluzione dello scheduling è riportata in Fig. 5.

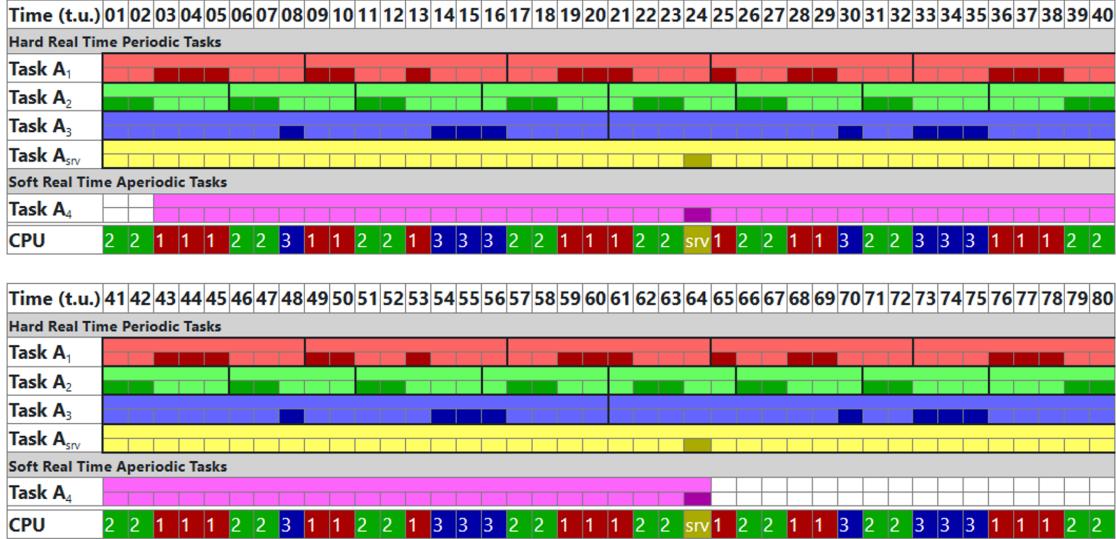


Figura 5: Scheduling con politica deferring server.

Pertanto, il processo deferring server mette a disposizione le t.u. 24 e 64 per l'esecuzione del task aperiodico soft real time. È facile verificare il task aperiodico soft real time viene eseguito entro la deadline assoluta  $d_4(1) = a_4(1) + D_4(1) = 64$  t.u.

8. Se il processo server è di tipo polling, dato che all'inizio dello scheduling la coda dei task soft real time è vuota (in quanto  $a_4(1) = 2$  t.u.), ne consegue che il computation time della prima istanza del task

server è 0. Solo la seconda e la terza istanza del processo server hanno computation time pari a 1, quindi il processore sarebbe disponibile, per il task soft real time, nella t.u. 64 e 104. Pertanto, in questo caso, il task aperiodico soft real time non verrebbe eseguito entro la deadline assoluta.

### Esercizio 2

Un automa di Mealy che risolve il problema è mostrato in Fig. 6. L'automa ha quattro stati così definiti:

- $\alpha$ : è lo stato iniziale dell'automa ed è anche lo stato neutro di funzionamento, ossia quando i precedenti eventi in ingresso non hanno avvito il possibile riconoscimento della sequenza di terminazione delle parole cercate;
- $\beta$ : è lo stato in cui l'ultimo simbolo in ingresso è stata la lettera O;
- $\gamma$ : è lo stato in cui l'ultimo simbolo in ingresso è stata la lettera N e il penultimo la O;
- $\delta$ : è lo stato in cui l'ultimo simbolo in ingresso è stata la E, il penultimo la N e il terzultimo la O.

Se a partire dallo stato  $\delta$  il prossimo simbolo in ingresso è  $\Delta$ , il sistema riconoscerà una parola che è terminata in ONE emettendo in uscita l'evento 1. In tutte le altre transizioni di stato l'uscita è sempre 0.

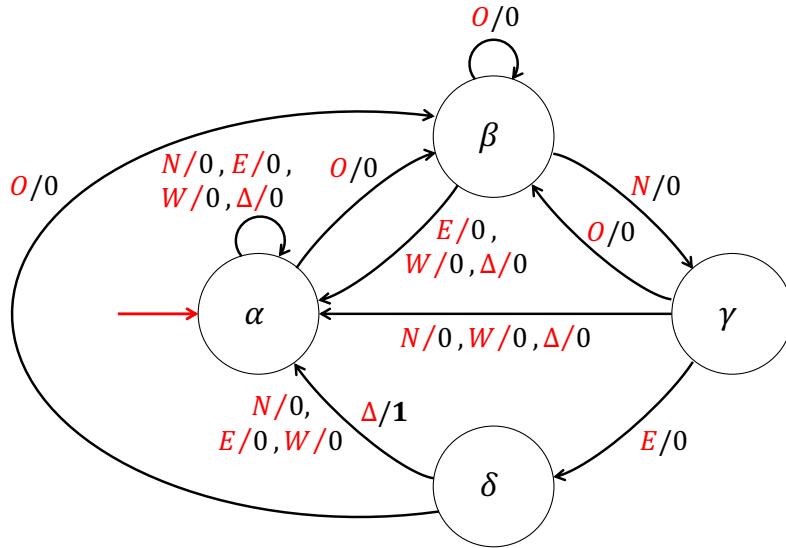


Figura 6: Automa del sistema di identificazione di parole che terminano in ONE.

### Esercizio 3

La matrice di incidenza (di dimensioni  $p \times t = 5 \times 5$ ) della rete di Petri in Fig. 1 è

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

Si noti che l'elemento in prima riga, prima colonna è 0 per la presenza di un autoanello: esiste sia un arco tra il posto  $p_1$  e la transizione  $t_1$  ( $\text{Pre}(p_1, t_1) = 1$ ) sia un arco tra la transizione  $t_1$  e il posto  $p_1$  ( $\text{Post}(t_1, p_1) = 1$ ), il che fornisce  $C(1, 1) = \text{Post}(t_1, p_1) - \text{Pre}(p_1, t_1) = 1 - 1 = 0$ .

E' facile verificare che la transizione  $t_1$  è abilitata allo scatto in  $x_0$  e che il suo scatto riporta il token in  $p_1$  mantenendo l'abilitazione della stessa  $t_1$ . Questo meccanismo porta a un numero illimitato di token

nel posto  $p_2$  ( $k$  token nella sequenza ammissibile di scatto  $\{t_1^k\}$ ). La rete è quindi illimitata. Possiamo verificarlo formalmente costruendo l'albero di copertura<sup>1</sup>, come in Fig. 7.

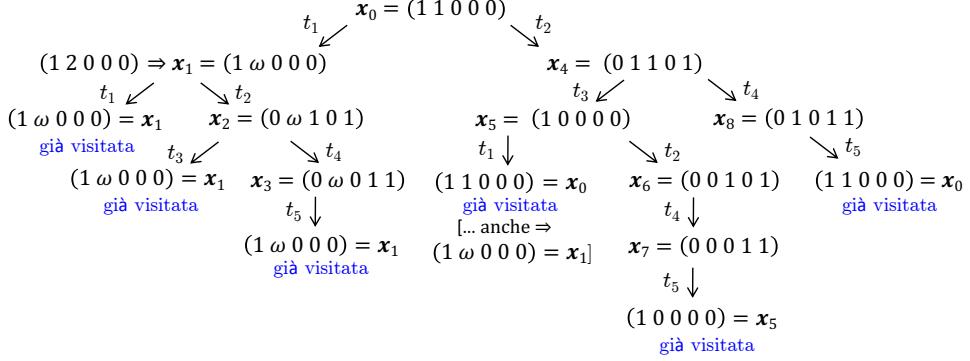


Figura 7: Albero di copertura della rete di Petri in Fig. 1.

Dall'analisi dell'albero, le marcature raggiungibili sono

$$R(PN) = \left\{ \begin{pmatrix} 1 \\ \omega \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \omega \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \omega \\ 0 \\ 1 \\ 1 \end{pmatrix} \right\}, \quad \text{con } \omega \text{ arbitrario.}$$

La rete è dunque illimitata (come aspettato, nel posto  $p_2$ ). La rete in questo caso è anche reversibile: infatti, quale che sia il numero  $\omega$  di token nel posto  $p_2$ , è sempre possibile portarsi nella marcatura  $\mathbf{x}_1$  e attivare la sequenza di scatto  $s = \{(t_2, t_3)^{\omega-1}\}$  per 'scaricare' il posto  $p_2$  di tutti i suoi token meno uno, tornando alla fine nella marcatura iniziale  $\mathbf{x}_0$ . Con considerazioni analoghe si può inoltre concludere che ogni transizione può essere abilitata allo scatto tramite una sequenza ammissibile di scatti, a partire da qualsiasi marcatura raggiungibile: la rete è dunque viva.

Calcoliamo ora i P-invarianti della rete. La matrice  $\mathbf{C}$  ha rango  $\rho = 3$ , quindi la dimensione dello spazio nullo di  $\mathbf{C}^T$  (come pure di  $\mathbf{C}$ ) è pari a 2. Si ha

$$\gamma^T \mathbf{C} = \mathbf{0}^T \Leftrightarrow \mathbf{C}^T \gamma = \mathbf{0} \Leftrightarrow \begin{cases} \gamma_2 = 0 \\ -\gamma_1 + \gamma_3 + \gamma_5 = 0 \\ \gamma_1 - \gamma_2 - \gamma_3 - \gamma_5 = 0 \\ -\gamma_3 + \gamma_4 = 0 \\ \gamma_1 - \gamma_4 - \gamma_5 = 0 \end{cases} \Rightarrow \begin{cases} \gamma_2 = 0 \\ \gamma_1 = \gamma_3 + \gamma_5 \\ \gamma_3 = \gamma_4. \end{cases}$$

Due P-invarianti indipendenti e canonici sono dunque

$$\gamma_1^T = (1 \ 0 \ 1 \ 1 \ 0) \quad \gamma_2^T = (1 \ 0 \ 0 \ 0 \ 1).$$

Nella rete sono quindi presenti due parti conservative (la somma del numero di token presenti nei posti  $(p_1, p_3, p_4)$  e nei posti  $(p_1, p_5)$  è sempre costante in ogni marcatura raggiungibile), ma ovviamente la rete non è conservativa perché gli insiemi di supporto di questi invarianti non ricoprono anche il posto  $p_2$  (che, come visto, è infatti illimitato).

Le due equazioni di invarianza  $\gamma_i^T \mathbf{x} = \gamma_i^T \mathbf{x}_0$ , per  $i = 1, 2$ , sono rispettivamente

$$x_1 + x_3 + x_4 = 1 \quad x_1 + x_5 = 1.$$

<sup>1</sup>A differenza di quanto accade con l'albero di raggiungibilità, le proprietà di vivezza e reversibilità di una rete di Petri illimitata non sono immediatamente deducibili dall'albero di copertura così costruito e vanno analizzate con cautela utilizzando ragionamenti ad hoc, come nel caso presente.

Gli insiemi di soluzioni (delle singole equazioni e delle due equazioni simultaneamente) hanno la struttura

$$I_{\gamma_1} = \left\{ \begin{pmatrix} 1 \\ * \\ 0 \\ 0 \\ * \end{pmatrix}, \begin{pmatrix} 0 \\ * \\ 1 \\ 0 \\ * \end{pmatrix}, \begin{pmatrix} 0 \\ * \\ 0 \\ 1 \\ * \end{pmatrix} \right\} \quad I_{\gamma_2} = \left\{ \begin{pmatrix} 1 \\ * \\ * \\ * \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ * \\ * \\ * \\ 1 \end{pmatrix} \right\} \quad \Rightarrow \quad I_{\gamma_1 \cap \gamma_2} = \left\{ \begin{pmatrix} 1 \\ * \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ * \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ * \\ 0 \\ 1 \\ 1 \end{pmatrix} \right\},$$

dove il simbolo \* rappresenta un valore qualsiasi. In generale si ha che  $R(PN) \subseteq I_{\gamma_i}$ . E' chiaro che qui l'insieme di soluzioni  $I_{\gamma_1 \cap \gamma_2} \equiv R(PN)$ . Non ci sono quindi soluzioni 'spurie', ossia che non sono marcature raggiungibili della rete.

Per il calcolo dei T-invarianti si ha

$$C \boldsymbol{\eta} = \mathbf{0} \quad \Leftrightarrow \quad \begin{cases} -\eta_2 + \eta_3 + \eta_5 = 0 \\ \eta_1 - \eta_3 = 0 \\ \eta_2 - \eta_3 - \eta_4 = 0 \\ \eta_4 - \eta_5 = 0 \\ \eta_2 - \eta_3 - \eta_5 = 0 \end{cases} \quad \Rightarrow \quad \begin{cases} \eta_2 = \eta_3 + \eta_5 \\ \eta_1 = \eta_3 \\ \eta_4 = \eta_5. \end{cases}$$

Due T-invarianti indipendenti sono dunque

$$\boldsymbol{\eta}_1^T = (1 \ 1 \ 1 \ 0 \ 0) \quad \boldsymbol{\eta}_2^T = (0 \ 1 \ 0 \ 1 \ 1).$$

Le sequenze di scatto ammissibili sono rispettivamente  $s_{11} = \{t_1, t_2, t_3\}$ ,  $s_{12} = \{t_2, t_1, t_3\}$  e  $s_{13} = \{t_2, t_3, t_1\}$  per il primo T-invariante e  $s_2 = \{t_2, t_4, t_5\}$  per il secondo.

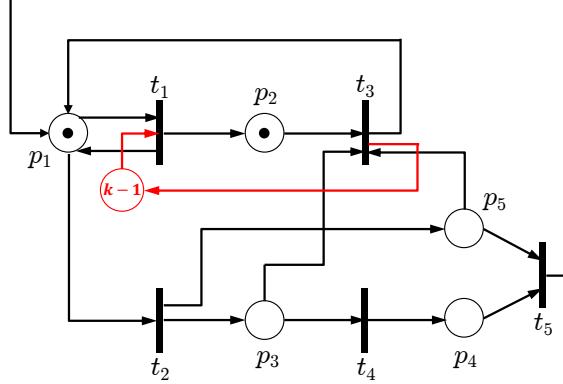


Figura 8: La rete di Petri di Fig. 1 con il supervisore progettato in cui  $k = 1$  o  $k = 2$ .

Il progetto di un supervisore che imponga il vincolo  $x_2 (= \mathbf{x}(p_2)) \leq k$  per ogni marcatura  $\mathbf{x} \in R(PN)$  si basa sulla tecnica dei P-invarianti e si può svolgere per un qualsiasi valore positivo di  $k$  (deve essere necessariamente  $k \geq 1$  in modo che la marcatura iniziale  $\mathbf{x}_0$  soddisfi il vincolo). L'equazione di progetto è

$$\mathbf{h}^T \mathbf{x} = (0 \ 1 \ 0 \ 0 \ 0) \mathbf{x} = x_2 \leq k$$

da cui si aggiunge un solo posto  $p_m$  con connessione alle transizioni della rete specificate dalla riga aggiuntiva della matrice di incidenza

$$\mathbf{c}_m^T = -\mathbf{h}^T \mathbf{C} = (-1 \ 0 \ 1 \ 0 \ 0) \quad \Rightarrow \quad \mathbf{C}^m = \begin{pmatrix} \mathbf{C} \\ \mathbf{c}_m^T \end{pmatrix}$$

e dalla marcatura iniziale

$$x_{m,0} = k - \mathbf{h}^T \mathbf{x}_0 = k - 1 \quad \Rightarrow \quad \mathbf{x}_0^m = \begin{pmatrix} \mathbf{x}_0 \\ x_{m,0} \end{pmatrix}.$$

La rete supervisionata è mostrata in Fig. 8. Il posto monitor non ha quindi token nel caso  $k = 1$  e ha un token nel caso  $k = 2$ .

Per  $k = 1$ , l'albero di raggiungibilità della rete supervisionata è riportato in Fig. 9; è immediato verificare che la rete è ora 1-limitata (safe), reversibile e viva. Il calcolo degli invarianti non è più necessario, ma per completezza si riportano qui i risultati. La matrice di incidenza  $5 \times 4$  della rete supervisionata

$$\mathbf{C}_m = \begin{pmatrix} \mathbf{C} \\ \mathbf{c}_m^T \end{pmatrix}$$

ha ancora rango pari a 3, in quanto la riga aggiunta è uguale e opposta alla seconda riga della  $\mathbf{C}$ . I T-invarianti che soddisfano  $\mathbf{C}_m \boldsymbol{\eta} = \mathbf{0}$  sono quindi gli stessi della rete originaria. Per i P-invarianti canonici, oltre ai due derivati dalla rete originaria (aggiungendo uno zero in ultima posizione)

$$\boldsymbol{\gamma}_{m,1}^T = (1 \ 0 \ 1 \ 1 \ 0 \ 0) \quad \boldsymbol{\gamma}_{m,2}^T = (1 \ 0 \ 0 \ 0 \ 1 \ 0),$$

si avrà anche

$$\boldsymbol{\gamma}_{m,3}^T = (h^T \ 1) = (0 \ 1 \ 0 \ 0 \ 0 \ 1).$$

Come atteso, l'insieme di supporto dei tre invarianti ricopre tutti i sei posti della rete supervisionata, da cui la sua conservatività e limitatezza.

Per  $k = 2$ , l'albero di raggiungibilità è riportato in Fig. 10. La rete è ora 2-limitata, reversibile e viva; le marcature raggiungibili sono però cresciute da 6 a 9. I P-invarianti e i T-invarianti rimangono ovviamente gli stessi del caso precedente, dato che non dipendono dal numero di marche presenti inizialmente nel posto monitor.

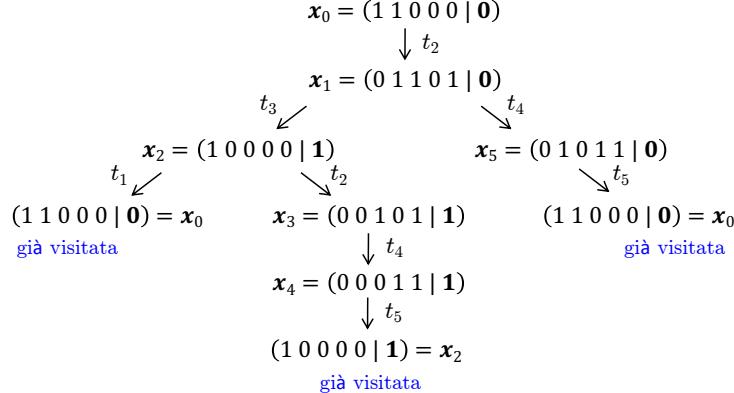


Figura 9: Albero di raggiungibilità della rete di Petri supervisionata di Fig. 8 per  $k = 1$ .

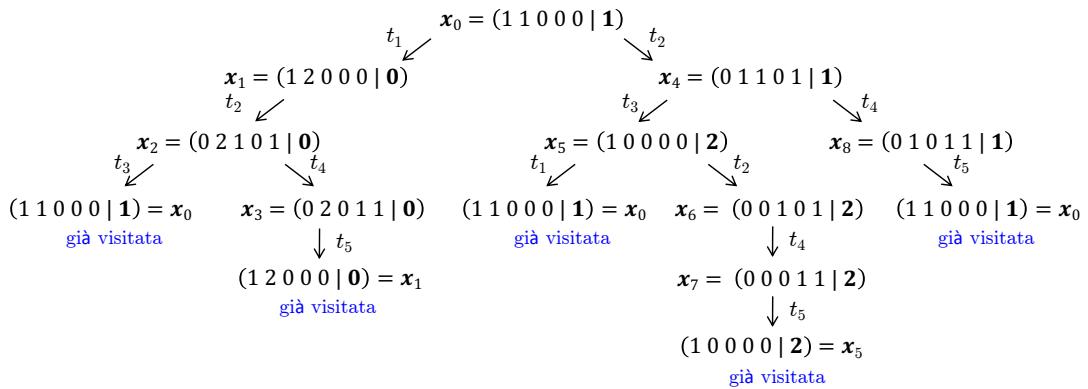


Figura 10: Albero di raggiungibilità della rete di Petri supervisionata di Fig. 8 per  $k = 2$ .

#### Esercizio 4

Poiché il processo non ha poli nell'origine (altrimenti la risposta indiciale ad anello aperto divergerebbe), il regolatore utilizzerà anche un'azione integrale oltre a quella proporzionale. In aggiunta, la presenza del termine derivativo dovrebbe aiutare ad avere transitori con ridotte oscillazioni. Si adotterà quindi per il progetto un regolatore PID completo (e ideale) nella forma

$$PID(s) = K_p \left( 1 + \frac{1}{sT_i} + sT_D \right) = \frac{K_p T_d s^2 + K_p s + K_p/T_i}{s} = \frac{K_p}{T_i} \frac{T_i T_d s^2 + T_i s + 1}{s},$$

ovvero nella forma equivalente

$$PID(s) = \frac{K_d s^2 + K_p s + K_i}{s}, \quad \text{con } K_d = K_p T_d, \quad K_i = \frac{K_p}{T_i}.$$

Al livello di implementazione occorrerà rendere realizzabile questo regolatore, altrimenti non causale, aggiungendo un polo sufficientemente in alta frequenza nel termine derivativo (derivata in banda), ossia

$$PID_r(s) = K_p \left( 1 + \frac{1}{sT_i} + \frac{sT_D}{1 + \tau s} \right) = \frac{K_p}{T_i} \frac{T_i T_d (1 + (1/N)) s^2 + (T_i + (T_d/N)) s + 1}{s (1 + (T_d/N)s)}, \quad \tau = \frac{T_d}{N} \ll 1,$$

avendo posto l'intero  $N$  sufficientemente grande. Questo aspetto non verrà ulteriormente discusso.

i) Dalla risposta del processo ad anello aperto al gradino unitario, si possono ricavare i parametri  $K$ ,  $\theta$  e  $\tau$  necessari per la scelta dei guadagni del PID quando si utilizza il **1° metodo di Ziegler–Nichols**. Mentre il parametro  $K$  è univocamente specificato dal valore raggiunto a regime dall'uscita (qui è  $K = 1$ ), gli altri due parametri  $\theta$  e  $\tau$  possono essere dedotti in diversi modi, come illustrato graficamente in Fig. 11 dove si riporta anche il confronto tra la risposta indiciale del processo considerato e quella del modello guadagno/polo/ritardo finito assunto come riferimento nelle tabelle di Ziegler–Nichols, ossia

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

a) Nel primo caso (Fig. 11, in alto) si usa la tangente nel punto di flesso (ossia il punto a massima pendenza) della risposta indiciale del processo; le intercette di questa retta con l'asse dei tempi e con il valore dell'uscita a regime determinano rispettivamente i tempi  $\theta$  e  $\tau$ :

$$K = 1, \quad \theta = 0.9 \text{ s}, \quad \tau = 3.5 \text{ s}.$$

Dalla tabella relativa al regolatore PID completo si ha

$$K_p = 1.2 \frac{\tau}{\theta} = 4.6667, \quad T_i = 2\theta = 1.8 \text{ s} \Rightarrow K_i = 2.5926, \quad T_d = 0.5\theta = 0.45 \text{ s} \Rightarrow K_d = 2.1.$$

b) Nel secondo caso (Fig. 11, al centro) si usa il tempo di salita tra il 10% e il 90% del valore della risposta a regime per la costante di tempo  $\tau$ , con  $\theta$  che segue di conseguenza:

$$K = 1, \quad \theta = 1.1 \text{ s}, \quad \tau = 4.2 \text{ s}.$$

Dalla tabella relativa al regolatore PID completo si ha in questo caso

$$K_p = 1.2 \frac{\tau}{\theta} = 4.5818, \quad T_i = 2\theta = 2.2 \text{ s} \Rightarrow K_i = 2.0826, \quad T_d = 0.5\theta = 0.55 \text{ s} \Rightarrow K_d = 2.52.$$

c) Nel terzo caso (Fig. 11, in basso) si usano i parametri  $\theta$  e  $\tau$  che minimizzano la differenza tra la risposta indiciale del processo e quella del modello  $P_m(s)$ , che risultano essere:

$$K = 1, \quad \theta = 1 \text{ s}, \quad \tau = 2 \text{ s}.$$

L'istante  $\theta + \tau$  indica anche quando si ha circa il 63% della risposta a regime. Dalla tabella segue:

$$K_p = 1.2 \frac{\tau}{\theta} = 2.4, \quad T_i = 2\theta = 2 \text{ s} \Rightarrow K_i = 1.2, \quad T_d = 0.5\theta = 0.5 \text{ s} \Rightarrow K_d = 1.2.$$

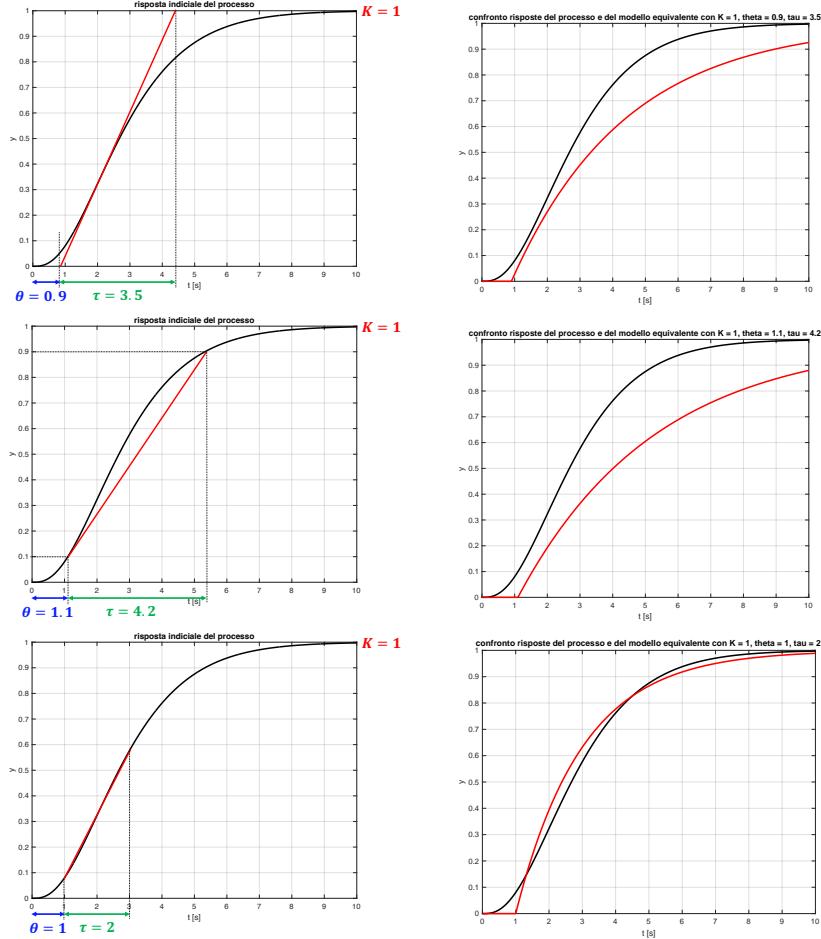


Figura 11: [A sinistra] Diversi metodi di valutazione dei parametri  $\tau$  e  $\theta$  per il 1° metodo di Ziegler–Nichols: **a)** usando la tangente nel punto di flesso (massima pendenza) della risposta (in alto); **b)** usando il tempo di salita tra il 10% e il 90% della risposta a regime (al centro); **c)** usando i parametri che minimizzano la differenza tra la risposta indiciale del processo e quella del modello  $P_m(s)$  (in basso). [A destra] Confronto tra risposta indiciale del processo (in nero) e quella del modello  $P_m(s)$  (in rosso) ottenuto con i tre metodi.

I tre regolatori (ideali) PID sono quindi

$$\begin{aligned} \text{PID}_{1a}(s) &= \frac{2.1s^2 + 4.667s + 2.593}{s} = 2.1 \frac{(s + 1.1111)^2}{s} \\ \text{PID}_{1b}(s) &= \frac{2.52s^2 + 4.582s + 2.083}{s} = 2.52 \frac{(s + 0.9091)^2}{s} \\ \text{PID}_{1c}(s) &= \frac{1.2s^2 + 2.4s + 1.2}{s} = 1.2 \frac{(s + 1)^2}{s}, \end{aligned}$$

dove si è anche messa in evidenza la localizzazione degli zeri a numeratore. Si noti che tali coppie di zeri sono sempre coincidenti e prossimi al valore  $-1$ , fatto che tornerà utile nella discussione successiva. Le risposte indiciali ottenute ad anello chiuso<sup>2</sup> con i tre suddetti regolatori sono mostrate in Fig. 12.

<sup>2</sup>Questi grafici sono stati ottenuti in MATLAB, utilizzando il modello del processo  $P(s) = 1/(s + 1)^3$  ricavato successivamente.

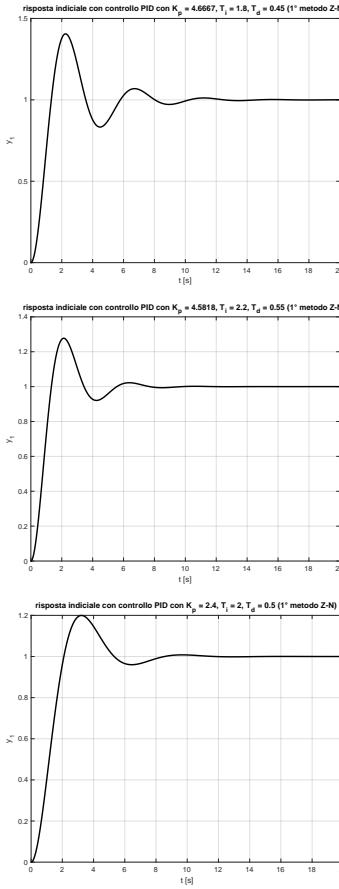


Figura 12: Risposte indiciali ad anello chiuso ottenute usando i tre regolatori (ideali)  $\text{PID}_{1a}(s)$ ,  $\text{PID}_{1b}(s)$  e  $\text{PID}_{1c}(s)$  progettati con il  $1^{\circ}$  metodo di Ziegler–Nichols.

I comportamenti ottenuti sono tutti abbastanza soddisfacenti. Il regolatore  $\text{PID}_c(s)$  è però chiaramente superiore sia per la minore sovraelongazione (pari al 20%), legata all'uso di un guadagno proporzionale  $K_p$  più basso, sia per la ridotta oscillazione della risposta prima di andare al valore desiderato di regime.

**ii)** La seconda parte dell'esercizio richiede di portare il sistema ad anello chiuso al limite di stabilità. sotto un controllo proporzionale con guadagno critico  $K_c$ . Avendo ora a disposizione il modello del processo

$$P(s) = \frac{1}{(s+1)^3},$$

si può procedere anche in modo analitico. Il sistema ad anello chiuso è infatti

$$W(s) = \frac{K_c}{(s+1)^3 + K_c} = \frac{K_c}{s^3 + 3s^2 + 3s + (K_c + 1)}.$$

Dalla tabella di Routh per il polinomio a denominatore

$$\begin{array}{c|cc}
 3 & 1 & 3 \\
 2 & 3 & K_c + 1 \\
 1 & \frac{8 - K_c}{3} \\
 0 & K_c + 1
 \end{array}$$

segue che il limite di stabilità si ottiene per  $K_c = 8$ . Dal luogo delle radici di  $P(s)$  (vedi Fig. 13), per tale valore due poli ad anello chiuso attraversano l'asse immaginario in corrispondenza ai punti  $(0, \pm j\omega_c)$ . Pertanto, quando  $K_c = 8$ , il denominatore della  $W(s)$  si fattorizza come

$$s^3 + 3s^2 + 3s + 9 = (s + p)(s + j\omega_c)(s - j\omega_c) = (s + p)(s^2 + \omega_c^2) = s^3 + ps^2 + \omega_c^2 s + p\omega_c^2$$

che fornisce quindi  $\omega_c = \sqrt{3} = 1.7321$  rad (e  $p = 3$ ).

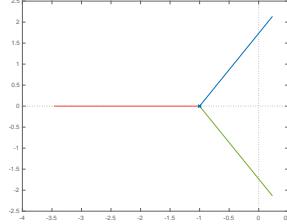


Figura 13: Luogo delle radici di  $P(s)$ .

In corrispondenza al valore critico del guadagno proporzionale, la risposta indiciale del sistema ad anello chiuso  $W(s)$  andrà in oscillazione a regime permanente con i valori

$$K_c = 8 \quad \Rightarrow \quad P_c = \frac{2\pi}{\omega_c} = 3.6276 \text{ s.}$$

Questa previsione è confermata dalla simulazione numerica in MATLAB mostrata in Fig. 14.

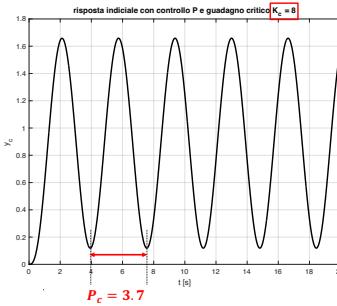


Figura 14: Valutazione dei parametri  $K_c$  e  $P_m$  per il 2° metodo di Ziegler–Nichols.

Procedendo invece graficamente, il periodo di oscillazione a regime è stimato pari approssimativamente a

$$P_c = 3.7 \text{ s.}$$

Con questo valore, la tabella del regolatore PID del **2° metodo di Ziegler–Nichols** fornisce i guadagni

$$K_p = 0.6 K_c = 4.8, \quad T_i = 0.5 P_c = 1.85 \text{ s} \Rightarrow K_i = 2.5946, \quad T_d = 0.125 P_c = 0.4625 \text{ s} \Rightarrow K_d = 2.22$$

e quindi il regolatore

$$\text{PID}_2(s) = \frac{2.22s^2 + 4.8s + 2.595}{s} = 2.22 \frac{(s + 1.0811)^2}{s}.$$

Anche in questo caso la coppia di zeri a numeratore è prossima al valore  $-1$ , mentre il guadagno proporzionale  $K_p$  è simile a quello dei due regolatori  $\text{PID}_{1a}(s)$  e  $\text{PID}_{1b}(s)$ . La risposta indiciale ad anello chiuso con il regolatore  $\text{PID}_2(s)$  mostrata in Fig. 15 è molto simile a quella ottenuta con il  $\text{PID}_{1a}(s)$ .

Dal confronto tra tutte le risposte ad anello chiuso ottenute, si può concludere che la migliore prestazione è ottenuta con il regolatore  $\text{PID}_{1c}(s)$ . Il motivo apparente è una coincidenza dei due zeri del regolatore con due dei tre poli in  $-1$  del processo, insieme a un guadagno proporzionale ridotto rispetto agli altri

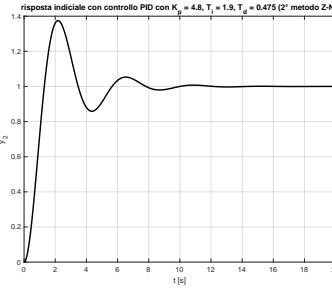


Figura 15: Risposta indiciale del sistema ad anello chiuso ottenuta usando il regolatore (ideale)  $\text{PID}_2(s)$  progettato con il 2° metodo di Ziegler–Nichols.

casi (il che evita un'eccessiva sovraelongazione). Proseguendo nella stessa direzione, si consideri allora un controllore (sempre ideale) della forma

$$\text{PID}_3(s) = K \frac{(s+1)^2}{s} = K \frac{s^2 + 2s + 1}{s},$$

con guadagno pari a  $K$ , che opera una **cancellazione diretta** di due dei tre poli asintoticamente stabili del processo. Questo controllore è ancora della forma PID con guadagni

$$K_p = K \quad K_i = 2K \quad K_d = K \quad \Rightarrow \quad W(s) = \frac{K}{s^2 + s + K}.$$

La funzione di trasferimento ad anello chiuso ha due poli con parte reale  $-0.5$  e parte immaginaria (se presente) pari a  $\pm 0.5\sqrt{4K-1}$ . Per  $K = K_p = 1.2$  si ritrova lo stesso regolatore  $\text{PID}_{1c}(s)$  già visto nel primo metodo di Ziegler–Nichols. Per  $K = K_p = 0.25$  i due poli sono entrambi reali e coincidenti in  $-0.5$  e la risposta è completamente sovrasmorzata, come mostrato in Fig. 16.

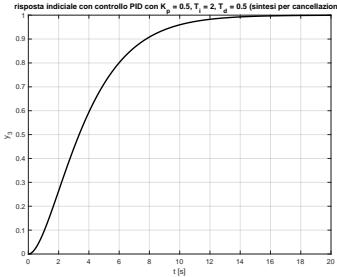


Figura 16: Risposta indiciale del sistema ad anello chiuso ottenuto con il regolatore (ideale)  $\text{PID}_3(s)$  progettato per cancellazione diretta poli-zeri e guadagno  $K = 0.25$ .

\* \* \* \*