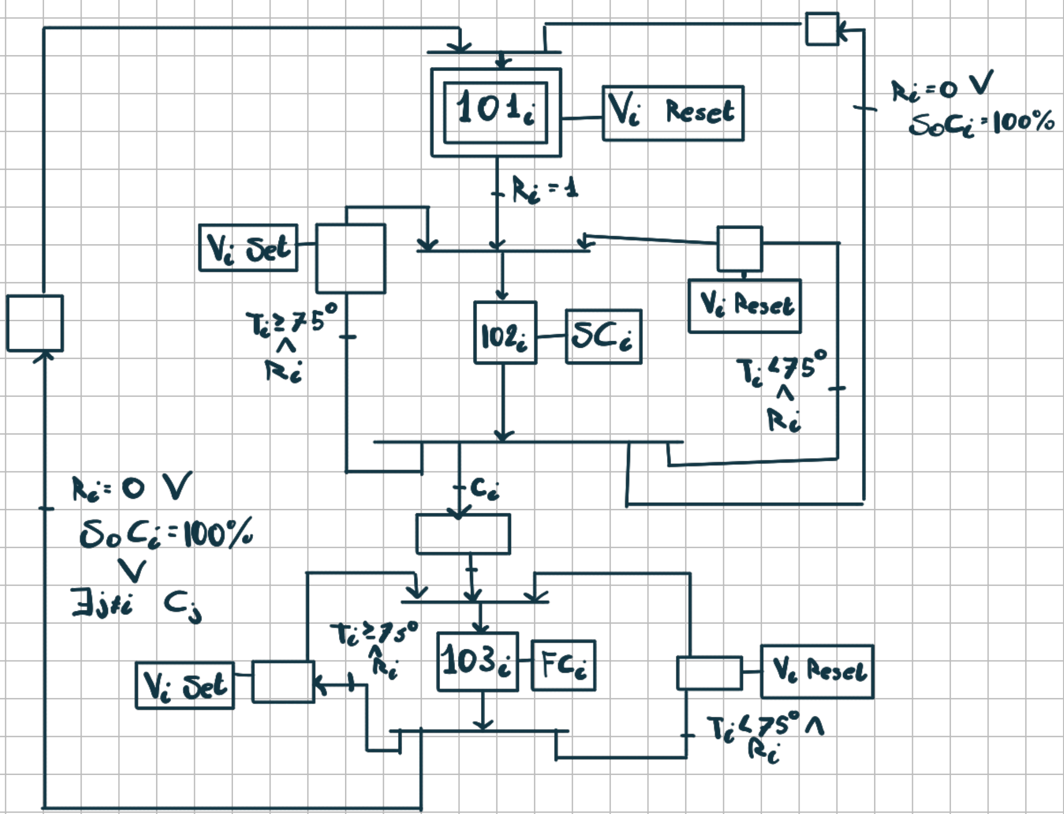
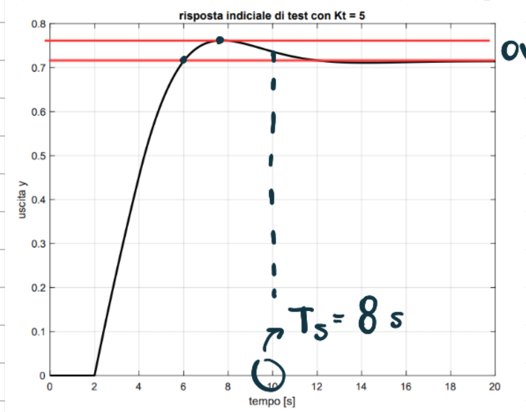
Es 1) Le proposizioni logiche sono scritte in FOL.



$$C_i = R_i \wedge \left(\forall_{j \neq i} \left[102_j \rightarrow i < j \right] \right) \wedge \left(\exists_j \; 103_j \wedge j > i \rightarrow 103_j.T \geq 1 \text{ minuto} \right) \wedge SoC_i \neq 100\,\%$$

Es 2) Il sistema e' del secondo ordine. L'errore a regime non e' nullo quindi la legge PID piu' semplice necessaria e' P-I. Se $P(s)$ e' il processo, la funzione ad anello chiuso e'

$\frac{5P(s)}{1+5P(s)}$, posso stimarla in base alla risposta indiciale.



$$W(s) = \frac{\omega_n^2}{s(s+2\zeta\omega_n)+\omega_n^2}$$

Overshoot: 5 % Stimo ω_n e ζ

$$5 = 100 \cdot \exp\left(-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right) \Rightarrow \zeta = 0.69$$

$$T_s = 8 = \frac{4}{\zeta\omega_n} \Rightarrow \omega_n = \frac{1}{2\zeta} = 0.72$$

analizzo il grafico

$$\Rightarrow W(s) = \frac{0.51 \cdot 0.71}{s(s+0.99)+0.51}$$

$$\Rightarrow \frac{5 \cdot P(s)}{1+5P(s)} = \frac{0.51 \cdot 0.71}{s(s+0.99)+0.51} \Rightarrow P(s) = \frac{0.7242}{10s^2+9.9s+1.479} = \frac{0.7242}{(s+0.806)(s+0.183)}$$

Scelgo un regolatore $\frac{K_p s + K_I}{s} = \frac{(s+0.183)}{s}$ elimino un polo lento

$\Rightarrow K_p = 1 \quad K_I = 0.183$

Es 3) Dalla matrice di incidenza e dalla trasposta trovo gli invarianti

$$C = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad C^T = \begin{bmatrix} -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 & -1 \end{bmatrix}$$

$$\begin{cases} -\gamma_1+\gamma_3+\gamma_4 = 0 \\ -\gamma_2+\gamma_3+\gamma_5 = 0 \\ \gamma_1-\gamma_3 = 0 \\ \gamma_2-\gamma_3-\gamma_4-\gamma_5 = 0 \end{cases} \quad \begin{cases} \gamma_1 = \gamma_3+\gamma_4 \Rightarrow \gamma_4 = 0 \\ \gamma_1 = \gamma_3 \\ \gamma_2 = \gamma_3+\gamma_5 \end{cases} \Rightarrow Ker(C^T) = \{ [a \;\; a{+}b \;\; a \;\; 0 \;\; b]^T \text{ con } a,b \in \mathbb{R} \}$$

I due invarianti canonici sono $\gamma_1 = [11100]^T$ e $\gamma_2 = [01001] \Rightarrow P_4$ non e' limitato. Quindi la rete non e' limititata. Le equazioni di invarianza (date da $x\gamma_i^T = x_0\gamma_i^T$) descrivono i posti conservativi:

$$\begin{aligned} \gamma_1^T \cdot x &= x(P_2)+x(P_5) = [01001]\cdot[11000]^T = 1 \\ \gamma_2^T x &= x(P_1)+x(P_2)+x(P_3) = [11100]\cdot[11000]^T = 2 \end{aligned} \Rightarrow \nexists K \text{ t.c. } x(P_4) \leq K$$

Per il resto, la rete e' reversibile, tutti i token accumulati su P_4 si possono scaricare con la sequenza $t_2 t_4$. Dato che

$\psi, \beta, \alpha \in \{0,1\}$

$\phi \in \mathbb{Z}^+$ $\bullet[\alpha \;\; 1 \;\; \beta \;\; \phi \;\; \psi] \xrightarrow{t_2} \bullet[\alpha \;\; 1 \;\; \beta{+}1 \;\; \phi \;\; \psi{+}1] \xrightarrow{t_4} \bullet[\alpha \;\; 1 \;\; \beta \;\; \phi{-}1 \;\; \psi]$

Non ci sono quindi deadlock, studio $Ker(C)$

$$\begin{cases} -\eta_1+\eta_3 = 0 \\ -\eta_2+\eta_4 = 0 \\ \eta_1+\eta_2-\eta_3-\eta_4 = 0 \\ \eta_1-\eta_4 = 0 \\ \eta_2-\eta_4 = 0 \end{cases} \Rightarrow Ker(C) = \left\{ \begin{bmatrix} x \\ x \\ x \\ x \end{bmatrix} x \in \mathbb{R} \right\} \Rightarrow \text{il T-invariante e' } \eta = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$\Rightarrow$ le 4 transizioni, se scattano un eguale numero di volte, portano la rete da x_0 a x_0. Progetto un supervisore che impone la limitatezza di p_4, provo con il metodo degli invarianti

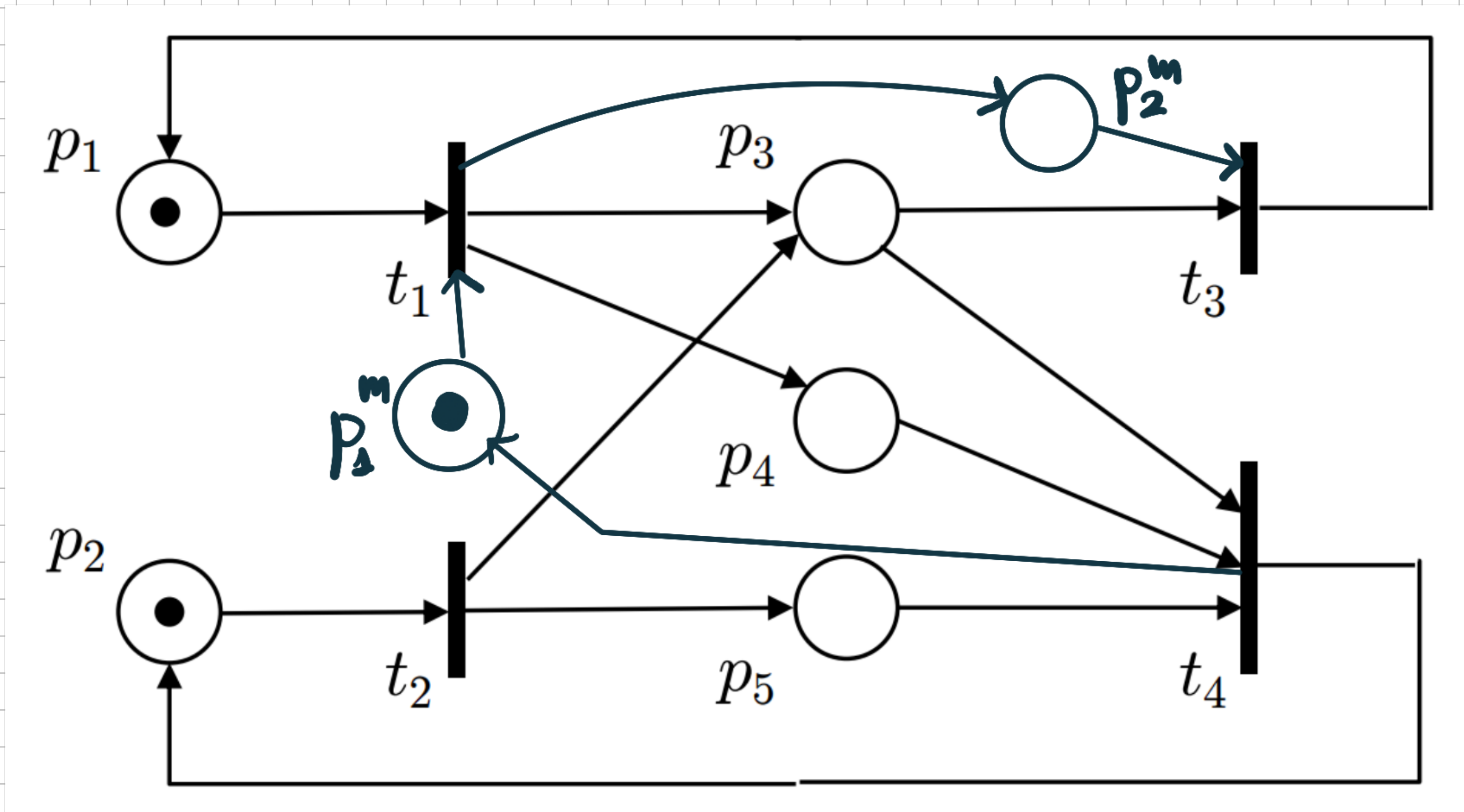
$[00010]^T x \leq 2 \Rightarrow$ Aggiungo la riga

$$C_m = -[00010]\, C = [-1 \;\; 0 \;\; 0 \;\; 1] \Rightarrow C = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

$$x_0(p^m) = 2 - [00010] \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 2$$

Questo supervisore non va bene perche' rende la rete NON priva di deadlock.

Devo far si che t_3 possa essere attivata solo in seguito a t_1, aggiungo quindi un nuovo posto monitor descritto dalla riga $C_2^m = [1\ 0\ {-1}\ 0]$. La rete finale e':



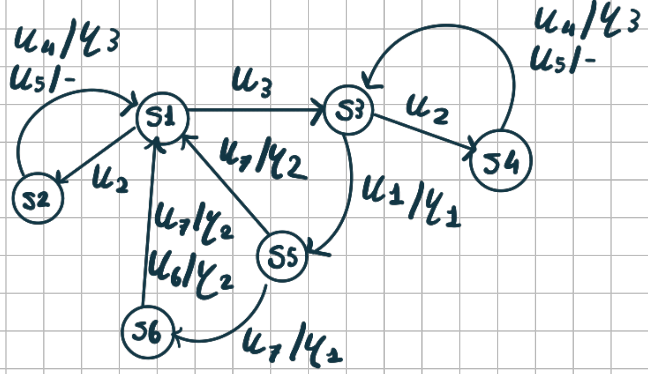
Tutte le proprieta' richieste sono soddisfatte.

Es 4) Definisco i seguenti input ed output

u_1 : riceve il token libero
u_2 : riceve un messaggio (mezzo utilizzato)
u_3 : ha un messaggio da trasmettere
u_4 : e' il dest. del messaggio ricevuto.
u_5 : non e' il dest.
u_6 : riceve un messaggio DIVERSO dall'ultimo inviato
u_7 : riceve uguale all'ultimo inviato

y_1 : trasmette messaggio con token in stato OCCUPATO
y_2 : trasmette (reintroduce) il token libero.
y_3 : legge messaggio

omessi i cappi:



S1 : in ascolto
S2, S3 : in ascolto e ho ricevuto
S4 : ho un mess. da trasmettere
S5 : ho trasmesso
S6 : ho ri-trasmesso in seguito di un errore.

Il protocollo è decentralizzato perché non c'è un access point che si occupa di gestire gli accessi o di calcolare il percorso che devono fare i pacchetti.