

# Automazione

Leonardo Ganzaroli

## Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Processi industriali</b>	<b>3</b>
1.1 Sistemi di produzione (discreti) . . . . .	4
1.1.1 Linee di trasferta . . . . .	5
1.1.2 Flow Shop . . . . .	7
1.1.3 Altri tipi . . . . .	8
1.2 CIM . . . . .	9
1.2.1 Livelli . . . . .	10
1.2.2 Reti di comunicazione . . . . .	10
<b>2 Attuazione e controllo del moto</b>	<b>12</b>
<b>3 Sistemi di controllo Real-time</b>	<b>13</b>
3.1 Scheduling . . . . .	14
3.1.1 Algoritmi . . . . .	15
3.2 Implementazione . . . . .	16
<b>4 Linguaggi per PLC</b>	<b>17</b>
4.1 SFC . . . . .	17
4.1.1 Strutture di collegamento . . . . .	18
<b>5 DEDS</b>	<b>20</b>
5.1 Reti di Petri . . . . .	21
5.1.1 Analisi matriciale . . . . .	25
5.1.2 Modellazione con reti . . . . .	26

## Introduzione

Questi appunti sono derivanti principalmente dalle dispense del corso di *Automazione* che ho seguito durante la laurea Triennale di informatica all'università "La Sapienza".

**N.B** Alcune parti del programma non sono presenti.

# 1 Processi industriali

Un sistema di produzione automatizzato è composto da:

- **Processo produttivo**

Una combinazione di operazioni e trasformazioni fisico/chimiche che permettono di ottenere il prodotto finale

- **Sistema di controllo**

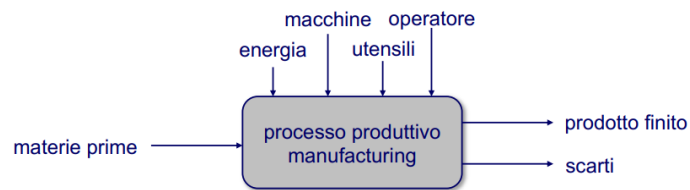
Dispositivo che scambia informazioni e azioni con il processo per cambiarne il comportamento, lo fa con:

- **Sensori**
- **Trasduttori**
- **Attuatori**

Un impianto industriale è composto da:

- Macchinari
- Strutture
- Edifici
- Componenti

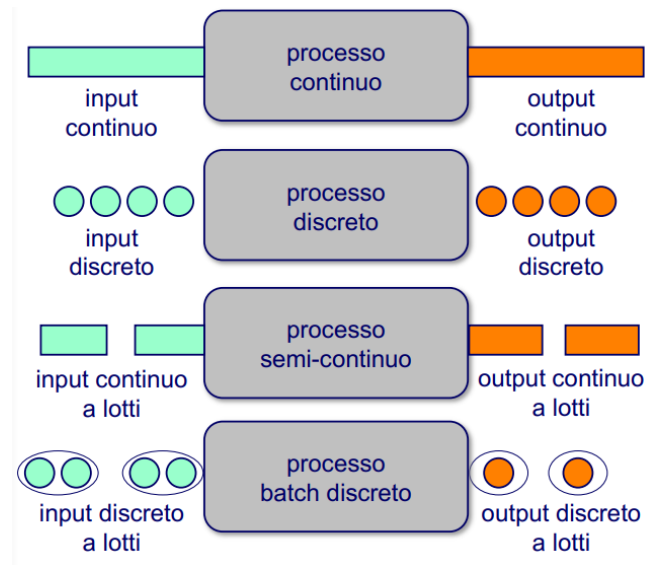
**Definizione** Il manufacturing è l'insieme dei processi produttivi applicati alle materie prime per ottenere il prodotto finale.



Un processo produttivo è composto da una sequenza di operazioni elementari raggruppabili come:

- Di lavorazione
- Di assemblaggio
- Di trasporto e stoccaggio
- Di test
- Di coordinamento e controllo

Un'altra classificazione riguarda la gestione dell'I/O:



I sistemi di controllo invece possono essere:

- **Logici**

Lavorano con variabili logiche che assumono valori in un insieme numerabile (solitamente finito).

- **Diretti**

Lavorano direttamente con i segnali.

## 1.1 Sistemi di produzione (discreti)

I principali tipi sono:

- **Linee di trasferta**
- **Flow Shop**
- **Job Shop**
- **Celle di produzione**
- **FMS**

### 1.1.1 Linee di trasferta

- Insieme di macchine/stazioni connesse in linea da un sistema di trasporto
- Sequenza prefissata di lavorazioni
- Flusso continuo di singoli pezzi
- Linee sincrone o asincrone

La legge di Little si può applicare a linee deterministiche e mono-prodotto:

$$WIP = \text{Throughput} \times \text{Tempo di attraversamento}$$

(In regime stazionario)

Per dimensionare opportunamente una linea di trasferta si segue questo procedimento:

- Data una linea con  $N$  stazioni, ognuna con il suo carico  $C_i$  (tempo necessario)
1. Il tasso di produzione è dato dalla stazione con il carico massimo
  2. Il carico massimo teorico ( $CMT$ ) è dato dal prodotto richiesto nel periodo
  3. Si crea un grafo delle precedenze delle lavorazioni
  4. Si trova un'assegnazione ammissibile delle lavorazioni alle stazioni tale che:
    - I vincoli di precedenza del grafo siano rispettati
    - Il numero di stazioni venga minimizzato
    - $\forall i \in [1, N] \quad C_i \leq CMT$

Per fare ciò si usa l'euristica RPWT:

- (a) Per ogni lavorazione si crea l'insieme  $S_i$  delle lavorazioni ad essa successive
- (b) Per ogni lavorazione si calcola il peso  $PW_i = T_i + \sum_{k \in S_i} T_k$
- (c) Si ordina per peso decrescente
- (d) Fino ad esaurimento si assegna la lavorazione con peso maggiore alla prima stazione disponibile

---

Esempio:

Una linea di assemblaggio di computer richiede 14 lavorazioni e si vogliono 300 computer ogni 7 ore.

lavorazione	A	B	C	D	E	F	G	H	I	J	K	L	M	N
tempo	55	30	50	42	20	25	45	60	36	42	30	40	36	40
$S_i$	B C D G ... N	H I ... N			G H I ... N		H I ... N	I ... N	N	L M N	N	N	N	-
$PW_i$ (sec)	506	314	334	326	349	354	329	284	76	158	70	80	76	40

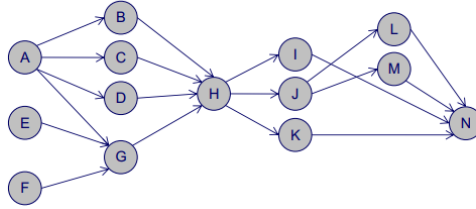


Figura 1: Costi e grafo

- $CMT = (\frac{300}{7*3600})^{-1} = 84$  [s/pezzo]
- $T_{TOT} = 551$  s
- Macchine minime =  $\frac{551}{84} \approx 7$

lavorazione	A	F	E	C	G	D	B	H	J	L	I	M	K	N
tempo	55	25	20	50	45	42	30	60	42	40	36	36	30	40
$PW_i$	506	354	349	334	329	326	314	284	158	80	76	76	70	40
stazione	1	1	2	2	3	4	4	5	6	6	7	7	8	8

Figura 2: Assegnamenti

Sommando il tempo mancante ad ogni stazione per raggiungere il  $CMT$  e dividendolo per il numero di stazioni si ottiene lo sbilanciamento medio:

$$\frac{111}{8} = 13.875 \text{ s} = 16.5\%$$

### 1.1.2 Flow Shop

- Stazioni/macchine disposte in linea
- Più prodotti ma stesse lavorazioni
- Una macchina esegue una lavorazione

In questo caso è importante minimizzare il tempo totale di completamento, nel caso di 2 macchine si segue la regola di Johnson per ogni macchina:

- $t_{i1}, t_{i2}$  sono i tempi di lavorazione del prodotto  $i$  sulle macchine 1 e 2
1. Si crea l'insieme 1 con i jobs  $t_{i1} \leq t_{i2}$
  2. Si crea l'insieme 2 con i jobs  $t_{i1} > t_{i2}$
  3. Si eseguono quelli nel primo in ordine crescente
  4. Si eseguono quelli nel secondo in ordine decrescente

---

Esempio:

Job\Prodotti	A	B	C	D	E	$T_{TOT}$
$t_{i1}$	5	3	8	10	7	33
$t_{i2}$	2	6	4	7	12	31

- $S_1 = \{B, E\}$
- $S_2 = \{A, C, D\}$
- Sequenza  $B, E, D, C, A$

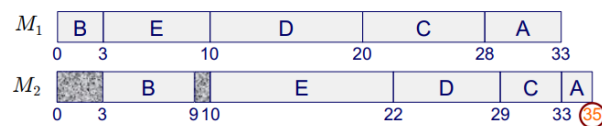


Figura 3: Diagrammi di Gantt delle macchine

- $T_{MAX} = 35$
  - $t_{idle,1} = 0$
  - $t_{idle,2} = 4$
-

### 1.1.3 Altri tipi

- Prodotti diversi con lavorazioni diverse
- Divisione in reparti
- Routing tra i reparti

- Famiglie di prodotti con lavorazioni abbastanza omogenee
- Raggruppamento di gruppi di macchine in celle
- Flussi più semplici

- Simile al precedente
- Uso di trasporto automatico
- Uso di calcolatori per il controllo del processo





## 1.2 CIM

**Definizione** Il sistema di supporto alla produzione è l'insieme di attività di gestione delle informazioni riguardanti la produzione.

**Definizione** L'Enterprise Resource Planning è un insieme di applicazioni informatiche volte all'automazione di attività amministrative, logistiche, ... .

**Definizione** Il Decision Support System è un software che fornisce delle funzionalità atte a migliorare il processo decisionale.

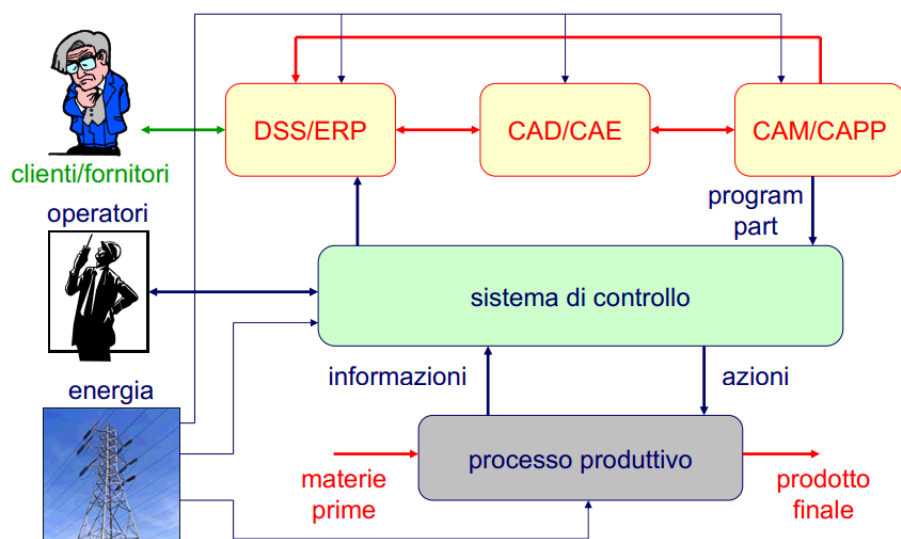
**Definizione** Il Computer Aided Design è un insieme di software che assistono i progettisti nelle attività di progettazione.

**Definizione** Il Computer Aided Engineering è un software per la verifica delle funzionalità del progetto.

**Definizione** Il Computer Aided Manufacturing è un software che permette di automatizzare le prove di fattibilità del processo produttivo.

**Definizione** Il Computer Aided Process Planning è un software che permette di automatizzare la pianificazione della produzione.

**Definizione** Il Computer Integrated Manufacturing è un modello teorico di un sistema produttivo che integra i processi produttivi con i sistemi di automazione e i sistemi informativi gestionali.



### 1.2.1 Livelli

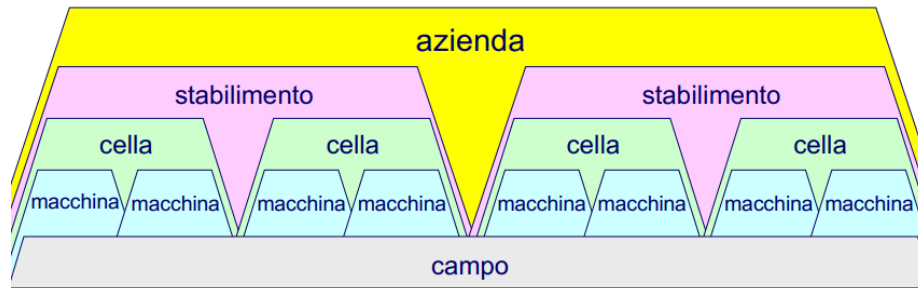


Figura 4: Gerarchia CIM

- **Campo**

Contiene i componenti hardware che eseguono le attività produttive ed il loro controllo.

- **Macchina**

Raggruppa gli elementi del livello precedente in gruppi atti a svolgere una certa funzionalità.

- **Cella**

Raggruppa gli elementi del livello precedente in celle.

- **Stabilimento**

Racchiude tutte le celle e le linee produttive facenti parte di un impianto industriale.

- **Azienda**

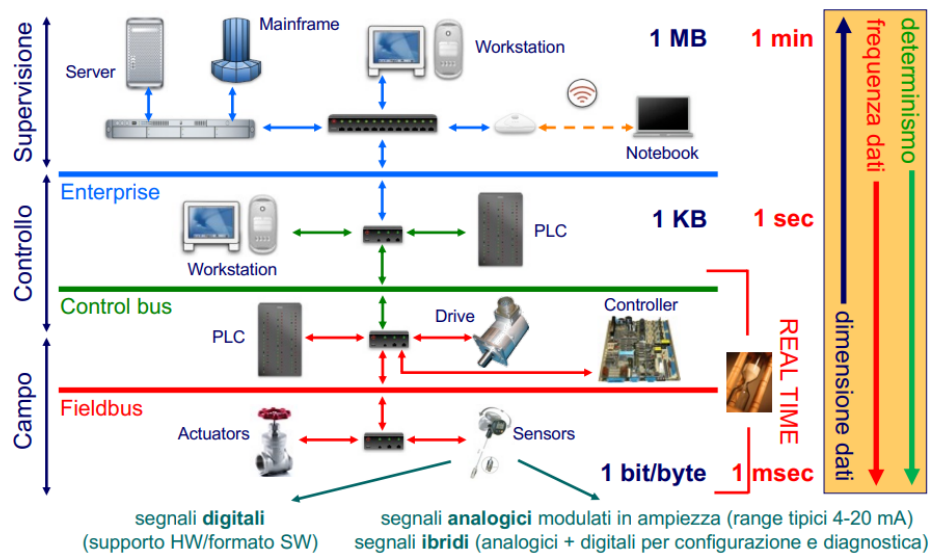
Qui avvengono i processi gestionali di supporto ai livelli inferiori.

### 1.2.2 Reti di comunicazione

Ad ogni livello della piramide :

- Si acquisiscono informazioni
- Si elaborano strategie
- Si attuano azioni correttive

Diventa quindi di vitale importanza il sistema di comunicazione.



- **Supervisione**

- Informazioni gestionali
- Client e Server standard
- Non Real-time
- Ethernet

- **Controllo/Campo**

- Client e Server non standard
- Dati piccoli ma frequenti
- Vincoli Real-time
- Serve determinismo
- Serve robustezza

Usare il modello ISO-OSI risulterebbe troppo oneroso, quindi si usa il modello *Fieldbus* che presenta solamente i livelli:

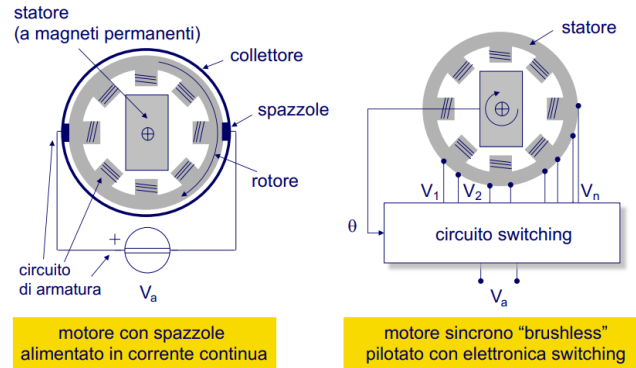
1. Fisico
2. Data Link
3. Applicazione

## 2 Attuazione e controllo del moto

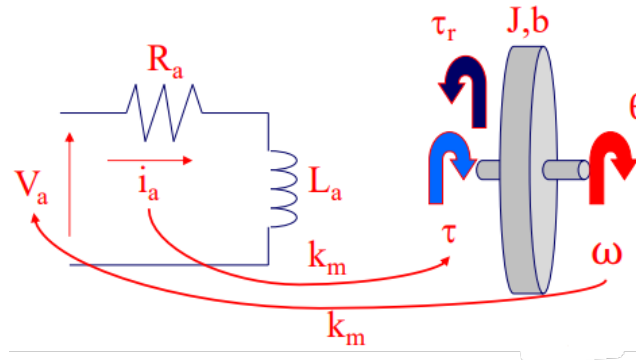
Gli azionamenti elettrici sono dispositivi che convertono in modo controllato l'energia elettrica in meccanica, sono formati da 3 componenti:

1. Amplificatore/Convertitore di potenza
2. Motore elettrico
3. Controllore

I motori elettrici hanno 2 schemi realizzativi:



In particolare quelli a corrente continua si possono rappresentare con il seguente modello:



- **Modello elettrico**  $L_a \frac{d i_a}{d t} = V_a - R_a i_a - k_m \omega$
- **Modello meccanico**  $J \frac{d \omega}{d t} = k_m i_a - b \omega - \tau_r, \quad \frac{d \theta}{d t} = \omega$

### 3 Sistemi di controllo Real-time

Un sistema di controllo può definirsi Real-time sse:

- Può elaborare le informazioni in modo da dare risposte logicamente corrette
- Può elaborare le informazioni in modo da dare risposte temporalmente corrette

**Definizione** Il Quality of Service esprime la capacità del sistema di rispettare entrambi i vincoli di correttezza logica e temporale.

**Definizione** Un sistema è Hard Real-time se rispetta **SEMPRE** i vincoli di correttezza.



**Definizione** Un task è un'unità atomica di lavoro.

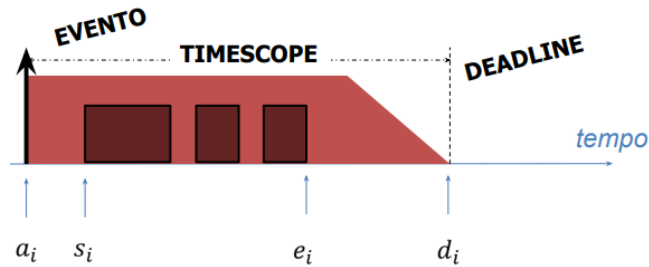


Figura 5: Task  $A_i$

Parametri caratteristici:

- |                                       |                     |
|---------------------------------------|---------------------|
| • $a_i$ , istante di attivazione      | • $C_i = e_i - s_i$ |
| • $s_i$ , istante di prima esecuzione | • $D_i = d_i - a_i$ |
| • $e_i$ , istante di fine esecuzione  | • $R_i = e_i - a_i$ |
| • $d_i$ , deadline assoluta           | • $L_i = e_i - d_i$ |

### 3.1 Scheduling

**Definizione** Un insieme di task è schedulabile se esiste un algoritmo di scheduling che permetta di rispettare tutti i vincoli.

Classificazione degli algoritmi di scheduling:

- **Guaranteed** se rispetta sempre i vincoli temporali, **Best effort** altrimenti
- **Preemptive** se può interrompere l'esecuzione di un task in favore di un altro con priorità maggiore, **Non preemptive** altrimenti
- **Offline** se lo scheduling è noto a priori, **Online** altrimenti
- **Statico** se il dispatching dipende da parametri immutabili, **Dinamico** altrimenti

Nel caso dell'automazione ha senso considerare dei task attivati periodicamente, si parla quindi di istanze dei task.

**Definizione** Il tempo di attivazione  $T_i(k)$  dell'istanza  $k$  di  $A_i$  è l'intervallo di tempo tra l'attivazione dell'istanza  $k$  e l'attivazione di  $k + 1$ .

**Definizione** Un task è detto periodico se il tempo di attivazione resta costante.

**Definizione** Il fattore di utilizzazione è pari a:

$$\sum_{i=1}^n \frac{C_i}{T_i}$$

Se è maggiore di 1 non esiste uno scheduling.

---

Esempio:

- $A_1, T_1 = 8 \text{ t.u.}, C_1 = 2 \text{ t.u.}$
- $A_2, T_2 = 12 \text{ t.u.}, C_2 = 8 \text{ t.u.}$
- Fattore =  $\frac{2}{8} + \frac{8}{12} \approx 0.917$ , potenzialmente schedulabile

---

**Definizione** Il limite superiore minimo del fattore di utilizzazione di un algoritmo di scheduling è il minimo tra i fattori calcolati per ogni possibile insieme di task periodici.

### 3.1.1 Algoritmi

In presenza di soli task periodici:

- **RMPO**

Assegna ad ogni task una priorità inversamente proporzionale al periodo di attivazione, schedulabilità garantita fino ad un fattore di 69.3%.

- **EDF**

Assegna ad ogni task una priorità inversamente proporzionale alla sua deadline assoluta, in caso di parità si guarda il numero d'istanza minore. Schedulabilità possibile fino ad un fattore  $\leq 1$ .

- **DMPO**

Assegna la priorità in modo inversamente proporzionale a  $D_i$ , schedulabilità possibile solo con fattore  $\leq n(2^{0.5} - 1)$ .

- **TS**

Divisione del tempo in *slices*, assegnazione arbitraria.

In caso siano presenti anche task aperiodici:

- **Servizio in background**

I task aperiodici vengono eseguiti negli istanti liberi.

- **Server**

Si inserisce un task periodico detto Server, i task aperiodici vengono eseguiti quando esso è in esecuzione.

- **Polling Server**

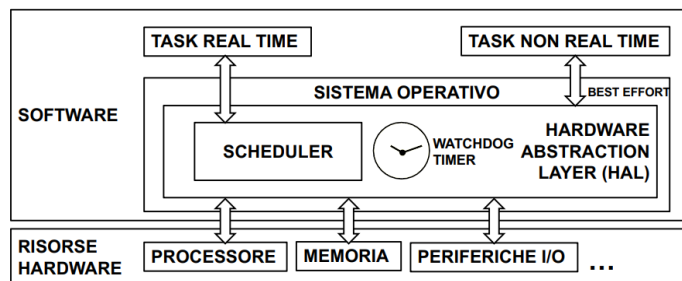
Variante del precedente in cui il tempo di computazione dell'istanza dipende dai task in attesa.

- **Deferrable Server**

Ulteriore variante in cui il tempo di computazione è sempre al massimo.

## 3.2 Implementazione

Oggigiorno qualsiasi sistema di controllo è implementato via software, per disaccopiarlo dall'hardware si usa uno strato di astrazione detto *HAL*, esso si occupa dei task non real-time in modo best effort. Quelli real-time vengono gestiti tramite lo scheduler ed è presente un timer per il controllo delle deadline.



**Definizione** Un s.o. è detto Event Driven se può schedulare un task nello stesso istante che viene attivato.

Un approccio puramente Event Driven non è realizzabile se l'unità di elaborazione è digitale, in questo caso si fa una rilevazione periodica. A livello implementativo questo metodo è migliore.

In base al tipo di caratteristiche richieste i sistemi di controllo possono essere implementati in diversi modi:

- **SCADA**
- **DCS**
- **Sistemi embedded**
- **PLC e SoftPLC**



## 4 Linguaggi per PLC

Secondo normativa esistono 5 linguaggi per i PLC:

- Grafici

1. **SFC** (quello approfondito)

2. **FBD**

3. **LD**

- Testuali

1. **ST**

2. **IL**

### 4.1 SFC

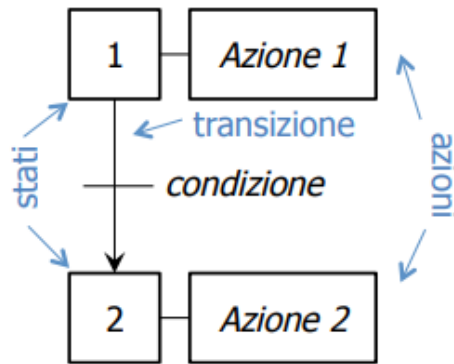


Figura 6: Diagramma SFC

Il passaggio da uno stato al successivo può avvenire sse:

- La condizione è verificata
- Lo stato precedente è attivo

Questo fa diventare lo stato successivo attivo e interrompe quello precedente.

Ogni stato ha 2 variabili associate:

1. **Marker**, indica se lo stato è attivo (nome-stato.X)
2. **Timer**, indica da quanto tempo è attivo (nome-stato.T)

All'avvio del PLC tutti i timer vengono azzerati e solamente i marker degli stati iniziali sono posti ad 1.

Ogni stato ha associata un'azione, essa ha la forma:

- $A_m$ , identificatore dell'azione

Semplici:

- $N$ , azione ripetuta ciclicamente finché lo stato è attivo
- $P$ , azione eseguita una volta finché lo stato è attivo
- $S$ , ripete l'azione finché non incontra  $R$  in uno stato successivo
- $R$
- $L$ , come  $N$  o fino al tempo fornito
- $D$ , come  $N$  ma dopo un tempo di delay

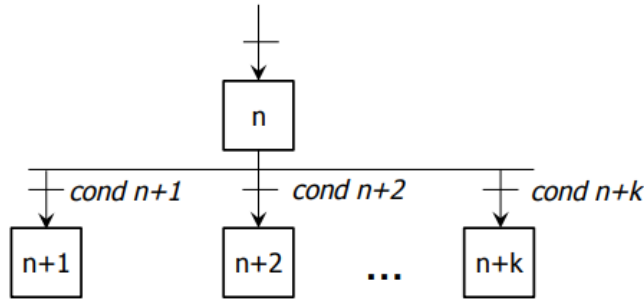
Composti:

- $SD$
- $DS$
- $SL$

- $Q_m$ , qualificatore che definisce il tipo d'azione
- $V_m$ , variabile che indica se l'azione è finita

#### 4.1.1 Strutture di collegamento

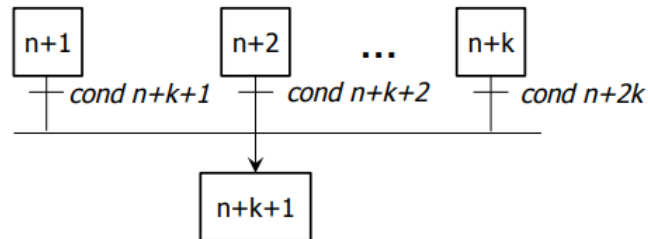
Nel caso ci siano sequenze diverse in base a diverse condizioni si usa la divergenza:



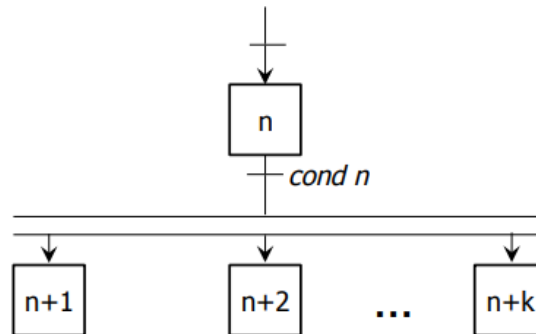
Questa struttura deve soddisfare il vincolo della mutua esclusione per funzionare correttamente, ad ogni condizione viene aggiunto implicitamente il NAND di tutte le altre condizioni:

$$cond(n+1) = cond(n+1) * \prod_{j=1}^{k-1} !cond(n+i-j) \quad \text{con } i \in [1, k-1]$$

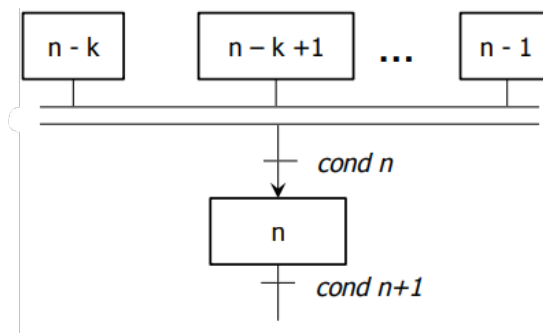
Quando si usa una divergenza bisogna sempre ritornare ad un'unica sequenza, per fare ciò si usa la convergenza:



Se invece c'è necessità di svolgere sequenze di azioni parallele si usa il parallelismo:

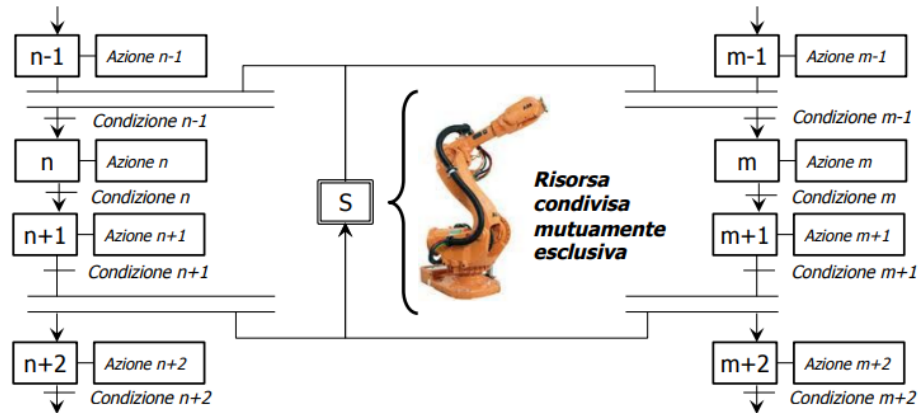


L'equivalente della convergenza in questo caso è la sincronizzazione:

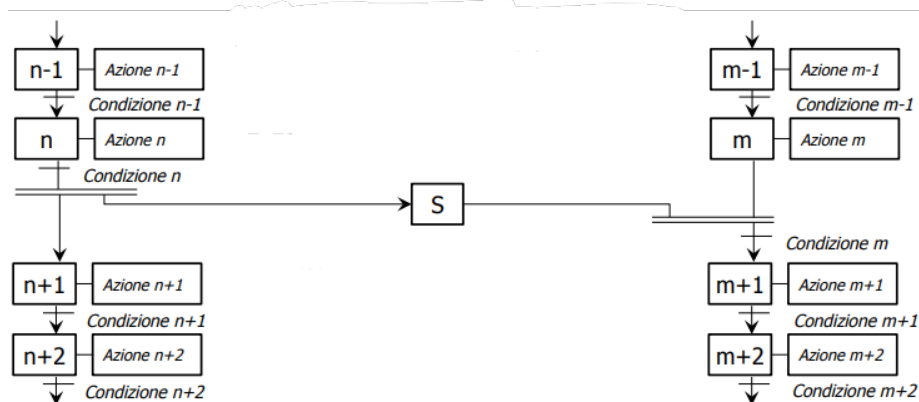


**N.B.** questa struttura "scatta" quando tutti gli stati a monte sono attivi e si verifica la condizione, in caso di necessità si possono inserire stati di attesa.

In presenza di risorse condivise si può usare un semaforo (uno stato) per garantire la mutua esclusione:



Si può anche usare per sincronizzare le sequenze parallele:



Generalmente viene impostato come stato iniziale per indicare che all'avvio la risorsa è disponibile.

## 5 DEDS

**Definizione** Un sistema è un insieme di componenti cooperanti ed interagenti che realizzano una funzionalità complessiva, si distinguono:

- Guidati dal tempo
- Guidati dagli eventi

**Definizione** Un sistema dinamico ad eventi discreti evolve in base agli eventi ed ha uno spazio degli stati discreto.

Per rappresentare questi sistemi si usano diversi modelli formali:

- Operazionali
  - Automi
  - Reti di Petri
  - SFC
- Dichiarativi
  - Basati su equazioni
  - Basati su regole

## 5.1 Reti di Petri

**Definizione** Un grafo di Petri è un grafo orientato e bipartito  $(P, T, A, w)$  in cui:

- $P$  è l'insieme dei posti (nodi)
- $T$  è l'insieme delle transizioni
- $A \subseteq (T \times P) \cup (P \times T)$  (archi)
- $w : A \rightarrow \mathbf{N} \setminus \{0\}$

Per esprimere la topologia si usano 2 matrici  $|P| \times |T|$ :

1.  $I$  contiene i pesi degli archi posti-transizioni
2.  $O$  contiene i pesi degli archi transizioni-posti

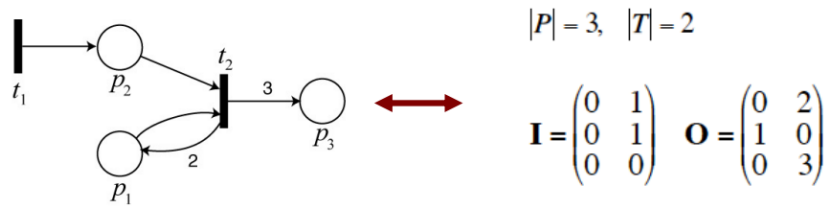


Figura 7: Esempio

**Definizione** Una rete di Petri è un grafo di Petri con una funzione di marcatura che associa ad ogni posto un numero di token  $x : P \rightarrow \mathbf{N}$ .

Per rappresentare lo stato della rete si usa il vettore di marcatura con  $|P|$  elementi, nel caso visto sopra potrebbe essere:

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \subseteq \mathbf{N}^3$$

Una transizione  $t_j$  è abilitata se:

$$\forall p_i \in I(t_j) \quad x(p_i) \geq w(p_i, t_j)$$

Se è abilitata allora può "scattare", questo comporta un cambiamento dello stato rappresentabile con la funzione  $\mathbf{N}^{|P|} \times T \rightarrow \mathbf{N}^{|P|}$ .

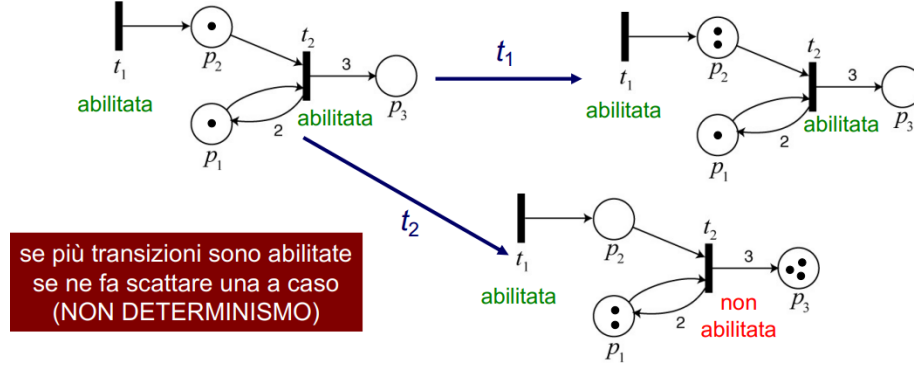


Figura 8: Esempio

**Definizione** La matrice di incidenza è  $C = O - I$ , rappresenta l'andamento della rete.

**Definizione** Una sequenza di scatti  $S$  è una sequenza di transizioni tali che:

- La prima è abilitata nella marcatura corrente
- Ogni scatto porta ad una marcatura in cui è abilitata la transizione successiva

**Definizione** Il vettore delle occorrenze di una sequenza è un vettore  $|T|$  in cui l'elemento  $k$  è pari al numero di occorrenze della transizione  $t_k$  in  $S$ .

L'evoluzione della rete data una sequenza  $S$  con vettore delle occorrenze  $s$  è  $x' = x + Cs$ .

Proprietà strutturali delle reti:

- Una marcatura  $x$  è raggiungibile dalla marcatura  $y$  se esiste una sequenza di scatti che porta dalla seconda alla prima
- Una marcatura è detta di base se è raggiungibile da tutte le altre marcature, se è anche iniziale la rete è reversibile
- Una transizione  $t$  è viva se per ogni marcatura ne esiste un'altra raggiungibile da essa che abilita  $t$
- La rete è viva se tutte le sue transizioni sono vive
- La rete è bloccante se esiste una marcatura che non abilita alcuna transizione
- Un posto è  $k$ -limitato se in ogni marcatura raggiunge un numero di token al massimo pari a  $k$ , illimitato altrimenti
- La rete è limitata se ogni suo posto è  $k$ -limitato con  $k$  finito, se  $k = 1$  è anche binaria
- La rete è illimitata se esiste un posto illimitato
- La parte conservativa della rete è un sottoinsieme di posti in cui ogni possibile evoluzione mantiene una combinazione lineare di token

Tipi di transizioni:

- **In conflitto**

Uno o più posti d'ingresso in comune ma con token non sufficienti a farle scattare tutte.

- **In concorrenza con successiva sincronizzazione**

No posti in comune, tutte abilitate e seguite da posti d'uscita che sono d'ingresso per un'altra transizione comune.

- **In alternativa**

In concorrenza tra loro ma in conflitto con altre.

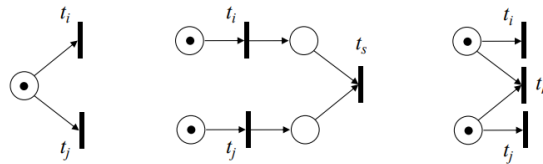


Figura 9: In ordine

Classi di reti:

- **SM**

- Ogni transizione ha un solo posto d'ingresso ed un solo posto d'uscita
- Numero di token fisso
- Viva sse esiste almeno un token ed il grafo è fortemente connesso
- Presenta solo conflitti

- **MG**

- Ogni transizione ha un solo posto d'ingresso ed un solo posto d'uscita
- Viva sse ogni ciclo ha almeno un posto marcato
- No conflitti

- **FC**

- Concorrenza e conflitti presenti ma non si influenzano tra loro
- Per ogni arco p-t o il posto è l'unico d'ingresso ad essa o essa è l'unica d'uscita a quel posto

Se 2 o più posti hanno una o più transizioni in comune si distinguono:

- **Estesa**

Hanno tutte le transizioni d'uscita in comune.

- **Asimmetrica**

Tutte le transizioni d'uscita di uno lo sono anche dell'altro.

Alcune estensioni delle reti sono:

- Reti **temporizzate**

Aggiunta di una struttura di clock ad ogni transizione, scatto dopo la raggiunta del relativo tempo.

- Reti **colorate**

Uso di diversi token, ognuno con il suo colore.

- Reti **con inibitori**

Aggiunta di archi speciali verso le transizioni, se il posto con l'arco ha almeno un token la transizione puntata non può scattare.



### 5.1.1 Analisi matriciale

(Con  $R(PN)$  si intende l'insieme delle marcature raggiungibili dalla marcatura iniziale  $x_0$ )

**Definizione** Un vettore colonna  $\gamma$  è detto P-invariante se:

$$\forall x \in R(PN) \quad \gamma^T * x = \text{costante} \quad \text{con } \gamma \in \mathbb{N}^{|P|} \wedge \gamma \neq 0$$

Si cercano tra le soluzioni del sistema lineare:

$$C^T \gamma = 0 \quad (\text{o } \gamma^T C = 0^T)$$

**Definizione** Il supporto di  $\gamma$  è l'insieme dei posti in esso non nulli.

Si distinguono:

- **Supporto minimo** se il supporto non contiene quello di nessun'altro P-invariante
- **Canonic** se il MCD dei suoi elementi non nulli è 1

$x_0 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

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

$\gamma^T \cdot C = 0^T \Rightarrow \begin{cases} -\gamma_1 - \gamma_2 + \gamma_3 = 0 \\ -\gamma_3 + \gamma_4 + \gamma_5 = 0 \\ \gamma_1 - \gamma_4 = 0 \\ \gamma_2 - \gamma_5 = 0 \end{cases} \Rightarrow \begin{cases} \gamma_1 = \gamma_4 \\ \gamma_2 = \gamma_5 \\ \gamma_3 = \gamma_1 + \gamma_2 \end{cases} \Rightarrow \begin{matrix} \gamma_A^T = [1 & 0 & 1 & 1 & 0] \\ \gamma_B^T = [0 & 1 & 1 & 0 & 1] \end{matrix}$

$\|\gamma_A\| = \{p_1, p_3, p_4\} \quad x(p_1) + x(p_3) + x(p_4) = 1$   
 $\|\gamma_B\| = \{p_2, p_3, p_5\} \quad x(p_2) + x(p_3) + x(p_5) = 1 \Rightarrow \text{rete conservativa}$

gli insiemi di supporto degli invarianti  
 non negativi ricoprono  $P = \{p_1, p_2, p_3, p_4, p_5\}$

Figura 10: Esempio

**Definizione** Un vettore delle occorrenze  $\eta$  è detto T-invariante se:

$$x_0 + C\eta = x_0$$

Si cercano tra le soluzioni del sistema lineare:

$$C\eta = 0$$

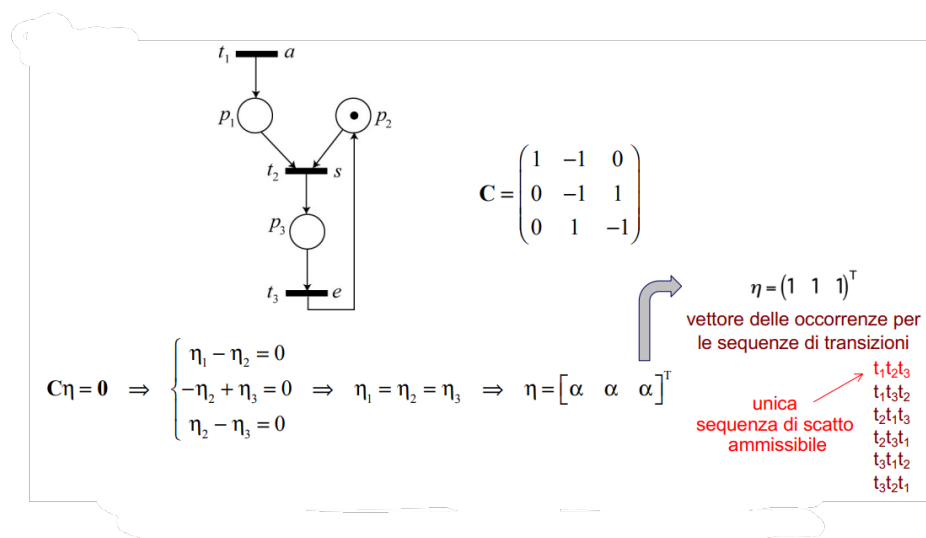


Figura 11: Esempio

In breve:

- Un P-invariante è un insieme di posti in cui il numero di token resta costante
- Un T-invariante è una sequenza di scatti che partendo dalla marcatura iniziale riporta alla stessa

### 5.1.2 Modellazione con reti

2 metodi:

1. Fisico
  - Suddivisione del sistema in sottoinsiemi elementari
  - Modellazione degli stessi con reti elementari
  - Composizione tra le reti ottenute

## 2. Funzionale

- Individuazione delle fasi logiche di funzionamento del sistema
- Identificazione delle risorse fisiche che le eseguono
- Allocazione delle fasi sulle risorse

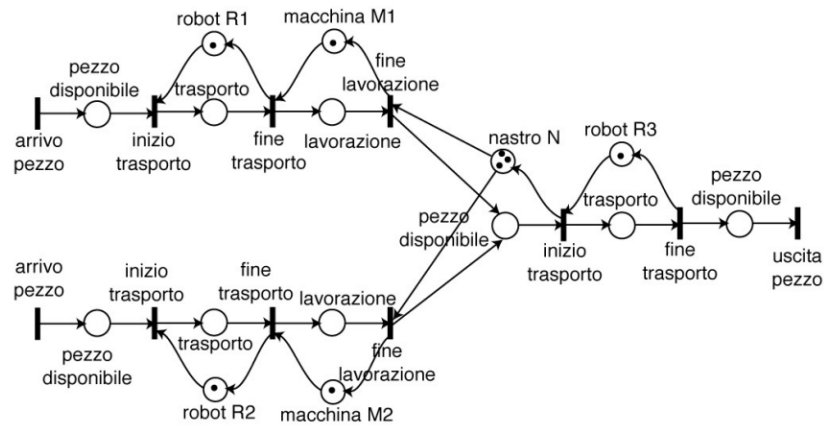


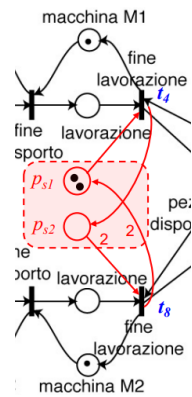
Figura 12: Esempio di modellazione

Gli archi all'indietro rappresentano la disponibilità delle macchine e il numero di posti liberi sul nastro.

Si può implementare una struttura di controllo nelle reti che permetta di modificare l'andamento in base alla marcatura:

- **Con posti di controllo opportunamente marcati**

Nell'esempio precedente si vogliono sul nastro 2 pezzi di *M1* seguiti da uno di *M2*:



- **Tramite invarianti e monitor**

Si assume che il comportamento desiderato sia descrivibile con la disequazione:

$$h^T x \leq k \quad \text{con } h \in \mathbf{Z}^{|P|}, \quad k \in \mathbf{N}$$

2 casi:

1. Transizioni tutte controllabili

Si aggiunge un posto monitor  $p_i^m$  tale che:

$$C_i^m = -h_i^T C, \quad x_0(p_i^m) = k - h_i^T x_0$$

$C_i^m$  viene aggiunto come riga a  $C$ .

2. Presenza di transizioni non controllabili

Si usa il metodo precedente ma bisogna verificare che nessun monitor vada a disabilitare una delle transizioni non controllabili:

$$x(p_i^m) \geq w(p_i^m, t_j) \quad \text{per ogni } t_j \text{ non controllabile}$$

Una condizione sufficiente è che nessuna delle transizioni non controllabili abbia dei monitor in ingresso.

si vuole evitare di fare arrivare altri pezzi ai buffer di ingresso dell'impianto quando il nastro di uscita è già saturo (con tre pezzi): si vuole quindi imporre il vincolo  $x(p_1) + x(p_6) + x(p_{12}) \leq 3$

**PN originale:** 15 posti, 11 transizioni  $\Rightarrow C: (15 \times 11)$ ; **PN supervisore con 1 posto monitor**

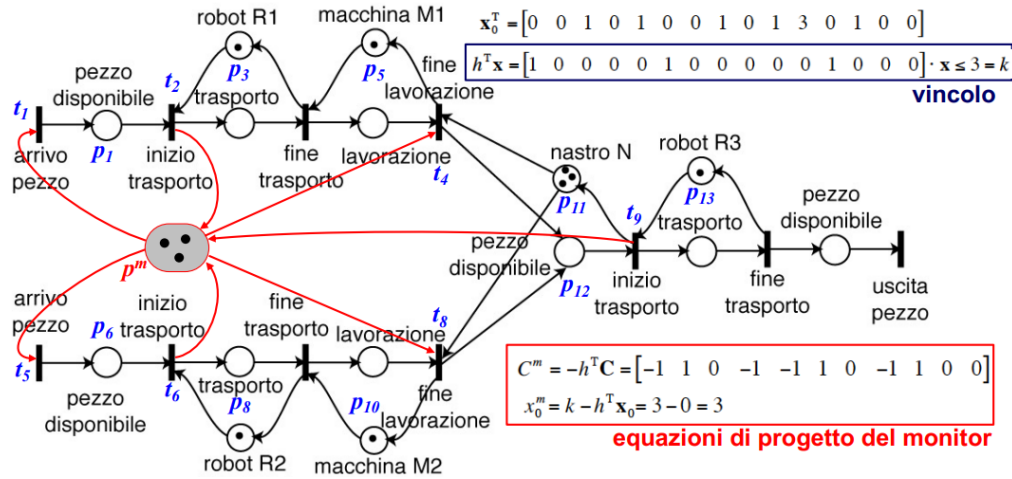


Figura 13: Esempio primo caso

se le transizioni  $t_4$  e  $t_8$  sono **non controllabili**, la precedente soluzione non è realizzabile e va modificata; scegliamo che il numero totale di pezzi e semilavorati presenti nell'impianto fino al nastro sia limitato dalla sua capacità:  $x(p_1)+x(p_2)+x(p_4)+x(p_6)+x(p_7)+x(p_9)+x(p_{12}) \leq 3$

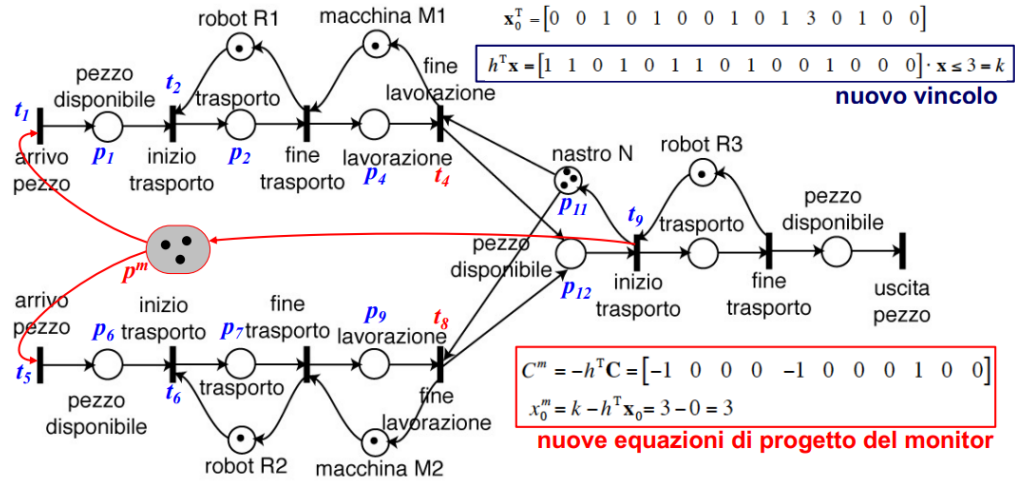


Figura 14: Esempio secondo caso