

Massimo Gessa
Matr:28048
09/05/03

MONITORDESIGN

funzione matlab da inserire
nel pacchetto di

RETI DI PETRI

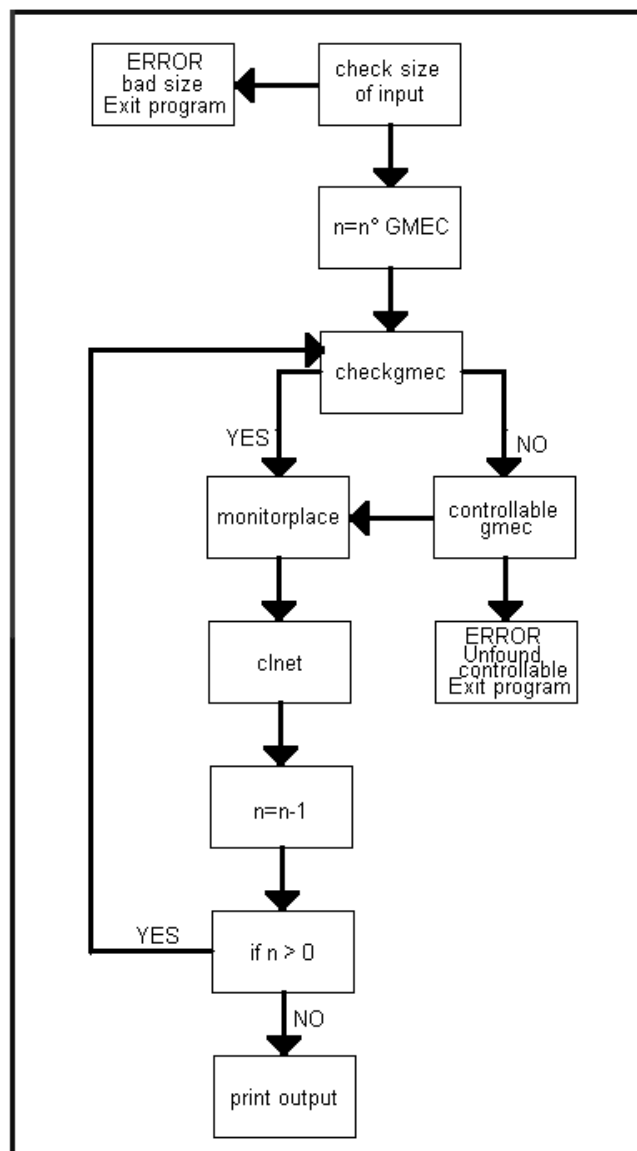
La funzione *monitordesign* è studiata per effettuare il progetto di una rete di *Petri* a ciclo chiuso partendo dalla descrizione della rete in esame e dalle specifiche di controllo.

La funzione analizza le specifiche per verificarne la controllabilità ed in caso di esito positivo, viene calcolato il posto monitor che la realizza; in caso contrario viene invece, prima calcolata una specifica controllabile (più restrittiva ed equivalente a quella data), e successivamente viene fornita la descrizione (in termini matriciali) del posto monitor che realizza la specifica stessa.

Il passo successivo consiste nel costruire le matrici *pre*, *post* e di *marcatura* della rete a ciclo chiuso.

L'algoritmo interno della funzione è schematizzato nel diagramma di flusso sotto riportato, e dalla cui analisi si evince la sequenza delle fasi dell'elaborazione.

Il primo passo consiste nel controllo delle dimensioni degli input; se il check fallisce la funzione ritorna un messaggio d'errore e termina, in caso contrario inizia la fase ricorsiva dell'algoritmo in cui il numero di cicli è pari al numero di specifiche da imporre.



L'unità di *ricorsione* è costituita da una sequenza di chiamata a funzioni esterne:

checkgmec (verifica la controllabilità della specifica) ;

controllablegmec (genera una specifica equivalente a quella data, ma meno restrittiva e controllabile);

monitorplace (calcola la matrice incidenza di un posto monitor relativo alla specifica);

clnet (fornisce le matrici pre, post, e di marcatura della rete a ciclo chiuso).

Come osservabile dal diagramma, il risultato del **check** sulla controllabilità stabilisce se la successiva chiamata di funzione sia diretta verso **monitorplace** o se prima si debba passare per **controllablegmec**.

Se durante la sua elaborazione, **controllablegmec**, verificasse che non esiste alcuna specifica equivalente e controllabile, l'algoritmo terminerebbe fornendo un messaggio d'errore.

Clnet raccoglie i risultati di **monitorplace** e costruisce, iterazione dopo iterazione, le nuove matrici pre post e di marcatura del sistema a ciclo chiuso.

L'algoritmo termina quindi dopo aver analizzato tutte le specifiche, stampando i risultati ottenuti.

Trattandosi di una funzione matlab la chiamata alla funzione è soggetta alla sintassi del linguaggio, $[OUTPUT] = NOMEFUNZ (INPUT)$, quindi nel caso specifico:

$[PRE , POST , M] = \text{monitordesign} (pre , post , m , w , k , u) ;$

Tra gli input sono presenti le matrici "pre", "post" ed "m" della rete *openloop* sufficienti a caratterizzarla; le matrici "w" e "k" rappresentano le specifiche di controllo, mentre l'ultimo input, "u" è un vettore riga che contiene indicazioni sulle transizioni non controllabili; se ad esempio le transizioni *t1*, *t3* e *t4* della rete in esame fossero non controllabili il vettore "u" sarebbe [1,3,4], se invece non ci fossero transizioni non controllabili il vettore "u" sarebbe un vettore vuoto quindi omissibile (input facoltativo).

Tra gli output troviamo le matrici "PRE", "POST" ed "M" rappresentanti la rete a ciclo chiuso; queste matrici risultano essere costituite dalle vecchie matrici pre, post ed m a cui sono aggiunte le righe relative ai posti monitor scaturiti dalla imposizione delle specifiche.

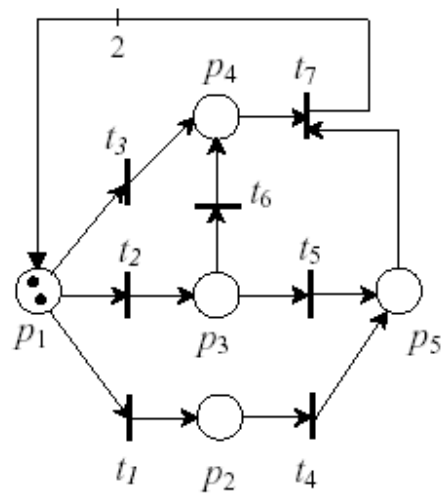
La funzione provvede, nella prima fase dell'algoritmo, al controllo della correttezza della dimensione degli input:

- **pre** e **post** devono avere le stesse dimensioni ($i \times j$)
- **m** deve essere un vettore colonna con lo stesso numero di righe di pre e post ($i \times 1$)
- **w** deve avere la stesso numero di righe di pre o post e numero di colonne pari al numero di specifiche da imporre ($i \times n$)
- **k** è un vettore colonna con numero di righe pari al numero di specifiche ($n \times 1$)
- **u** (se presente) è un vettore riga con numero di colonne pari al numero di transizioni non controllabili ($1 \times h$)

Le dimensioni degli output sono in funzione di quelle degli input, nello specifico

- **PRE** e **POST** avranno numero di colonne pari a quelle di pre (o post) e numero di righe pari a quelle di pre (o post) più il numero di colonne di w ($i+n \times j$)
- **M** è un vettore colonna con numero di righe pari a quelle di m più il numero di specifiche imposte ($i+n \times 1$)

Per testare la funzione si è scelta la rete in figura, in cui tutte le transizioni sono controllabili



Per questa rete le matrici “pre”, “post” ed “m” sono di seguito indicate

pre =

1	1	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	1	0
0	0	0	0	0	0	1
0	0	0	0	0	0	0

post =

0	0	0	0	0	0	2
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	1	0
0	0	0	1	1	0	0

m =

2
0
0
0
0

Le specifiche, in un primo esempio di funzionamento, potrebbero essere costituite dalle due GMEC $M(P2)+M(P3)+M(P5) \leq 1$ e $M(P4) \leq 1$ che, riportate in termini matematici, sono rappresentate dalle due matrici w e k

```
w =
    0    0
    1    0
    1    0
    0    1
    1    0
```

```
k =
    1
    1
```

la chiamata `[PRE, POST, M]=monitordesign(pre, post, m, w, k)` alla funzione, offre l'output mostrato in figura 1

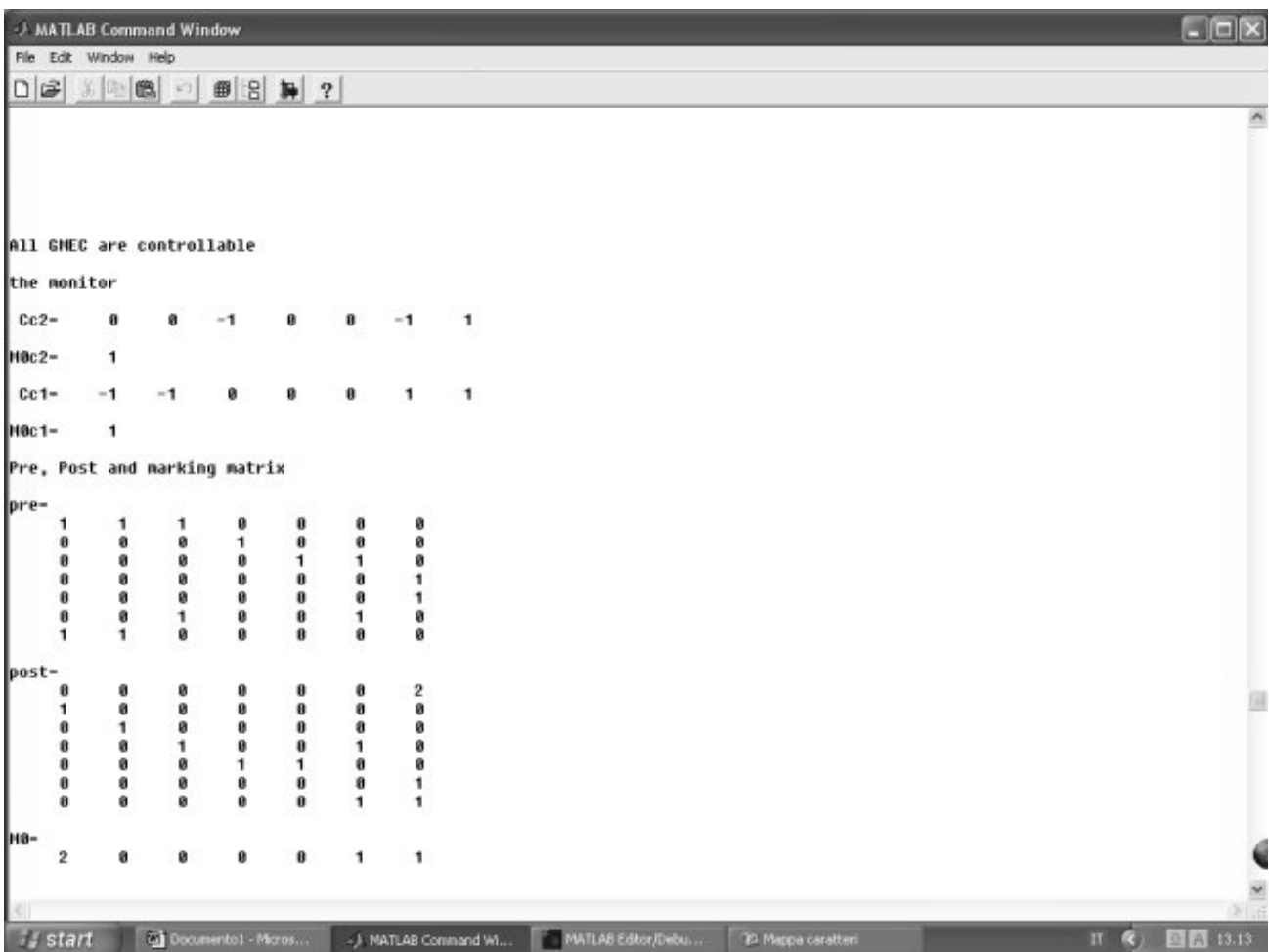


fig. 1

La prima riga ci informa che tutte le GMEC imposte sono controllabili; di seguito sono visualizzati i posti monitor calcolati e le nuove matrici pre, post e di marcatura (di quest'ultima, per ragioni di spazio, è visualizzata la trasposta).

Se per la stessa rete con le stesse specifiche avessi considerato non controllabili le transizioni 4,6,7 avrei riscontrato che la prima GMEC non sarebbe stata controllabile, quindi l'algoritmo avrebbe proceduto, da prima alla computazione di una GMEC equivalente controllabile, poi al calcolo dei monitor e al loro inserimento nelle matrici pre, post e di marcatura.

La chiamata `[PRE, POST, M]=monitordesign(pre, post, m, w, k,[4,6,7])` produce la schermata di risultati mostrata in figura 2.

```

MATLAB Command Window
File Edit Window Help
Uncontrollable GMEC:
w1=  0  0  0  1  0
k1=  1
the minimally restrictive is
w1=  0  0  1  1  0
k1=  1
the monitor
Cc2=  -1  -1  0  0  0  1  1
Hc2=  1
Cc1=  0  -1  -1  0  1  0  1
Hc1=  1
Pre, Post and marking matrix
pre=
  1  1  1  0  0  0  0
  0  0  0  1  0  0  0
  0  0  0  0  1  1  0
  0  0  0  0  0  0  1
  0  0  0  0  0  0  1
  0  1  1  0  0  0  0
  1  1  0  0  0  0  0

post=
  0  0  0  0  0  0  2
  1  0  0  0  0  0  0
  0  1  0  0  0  0  0
  0  0  1  0  0  1  0
  0  0  0  1  1  0  0
  0  0  0  0  1  0  1
  0  0  0  0  0  1  1

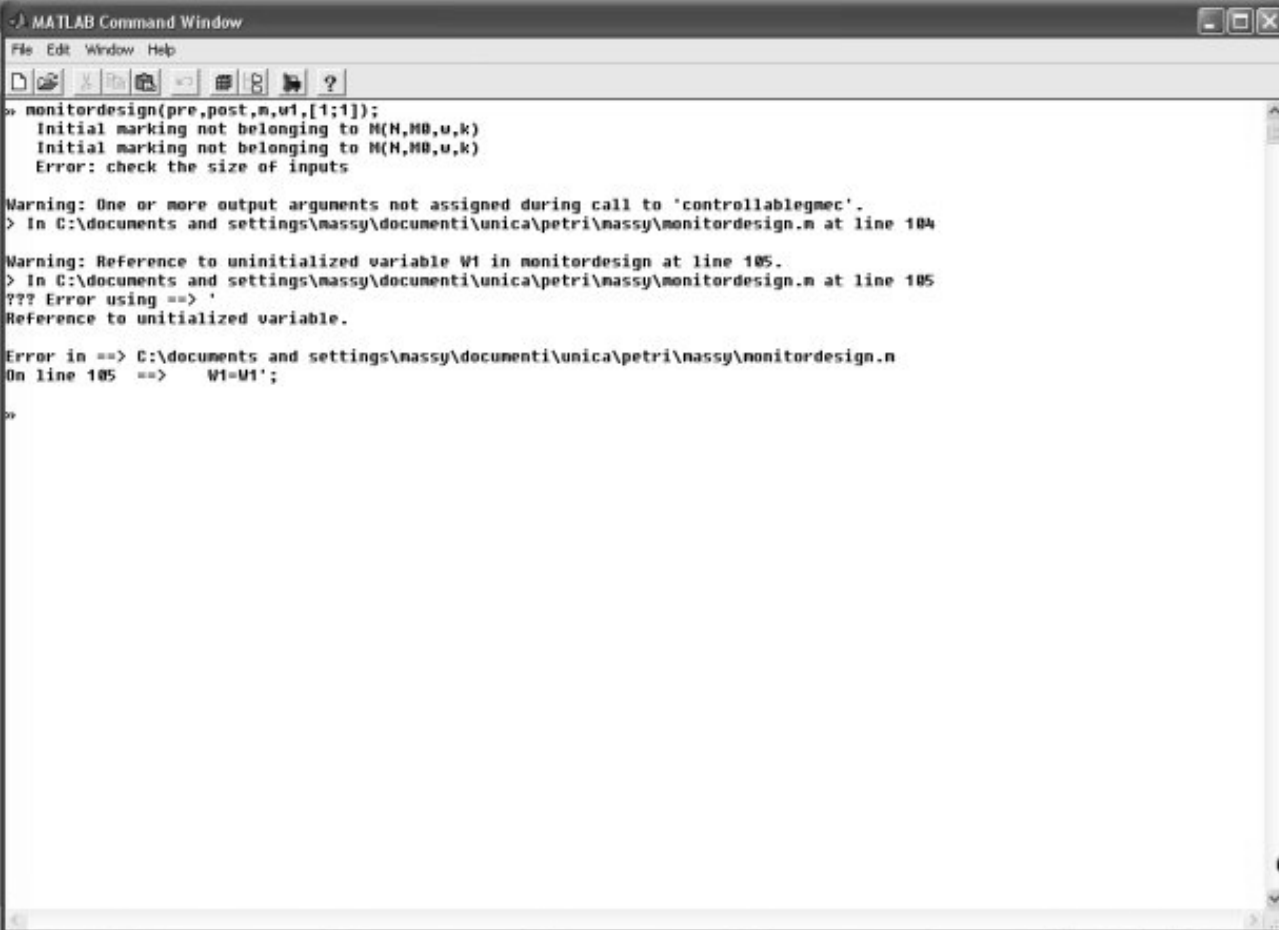
M=
  2  0  0  0  0  1  1
  
```

fig. 2

Se le GMEC fossero state $2M(P1)+M(P4) \leq 4$ e $M(P2)+M(P4) \leq 2$, considerando non controllabili le transizioni 4,6,7 entrambe le specifiche sarebbero state non controllabili, quindi, come da algoritmo, la funzione calcolerebbe le GMEC equivalenti ed i monitor e li aggiungerebbe alle matrici pre, post e di marcatura.

Differente sarebbe stato il caso in cui le GMEC fossero state $2M(P1)+M(P4) \leq 1$ e $M(P2)+M(P4) \leq 1$, in questo caso le specifiche sarebbero state entrambe non controllabili ed inoltre la marcatura iniziale non sarebbe stata sufficiente per cui il progetto non si sarebbe potuto terminare.

In un caso simile la funzione visualizza un messaggio d'errore.



```

MATLAB Command Window
File Edit Window Help
> monitordesign(pre,post,m,w1,[1;1]);
  Initial marking not belonging to M(N,MB,u,k)
  Initial marking not belonging to M(N,MB,u,k)
  Error: check the size of inputs

Warning: One or more output arguments not assigned during call to 'controllablegme'.
> In C:\documents and settings\massy\documenti\unica\petri\massy\monitordesign.m at line 104

Warning: Reference to uninitialized variable W1 in monitordesign at line 105.
> In C:\documents and settings\massy\documenti\unica\petri\massy\monitordesign.m at line 105
??? Error using ==> '
Reference to uninitialized variable.

Error in ==> C:\documents and settings\massy\documenti\unica\petri\massy\monitordesign.m
On line 105 ==> W1=U1';
>

```

fig. 3