

Simulazione di sistemi non lineari Implementazione di Automi Ibridi con Stateflow

Gianmaria De Tommasi¹

¹Università degli Studi di Napoli Federico II
detommas@unina.it

Ottobre 2012

Corsi AnsaldoBreda

Outline

1 Automi ibridi

2 Implementazione di automi ibridi con Simulink/Stateflow

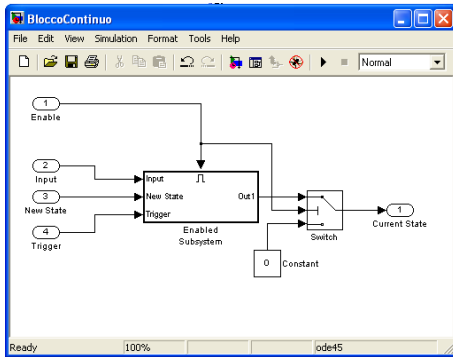
Definizione

Un *automa ibrido* H è una 8-pla $H = (Q, X, f, \text{Init}, D, E, G, R)$, dove:

- $Q = \{q_1, q_2, \dots, q_d\}$ è l'insieme degli **stati discreti**
- $X = \mathbb{R}^n$ è lo **spazio di stato continuo**
- $f(\cdot, \cdot) : Q \times X \rightarrow \mathbb{R}^n$ è la **funzione generatrice**
- $\text{Init} \subseteq Q \times X$ è l'insieme degli **stati iniziali**
- $\text{Dom}(\cdot) : Q \rightarrow 2^X$ è il **dominio** nello spazio di stato continuo per ognuno degli stati discreti
- $E \subseteq Q \times Q$ è l'insieme degli **archi**
- $G(\cdot) : E \rightarrow 2^X$ definisce le **condizioni di guardia** sugli archi
- $R(\cdot, \cdot) : E \times X \rightarrow 2^X$ è la **reset map**

Dinamiche tempo continue - 1

$\forall q \in Q$ realizzare la funzione $f(q, \cdot)$ mediante un blocco Simulink di questo tipo



Dinamiche tempo continue - 2

Il blocco che realizza la dinamica continua $f(q, \cdot)$:

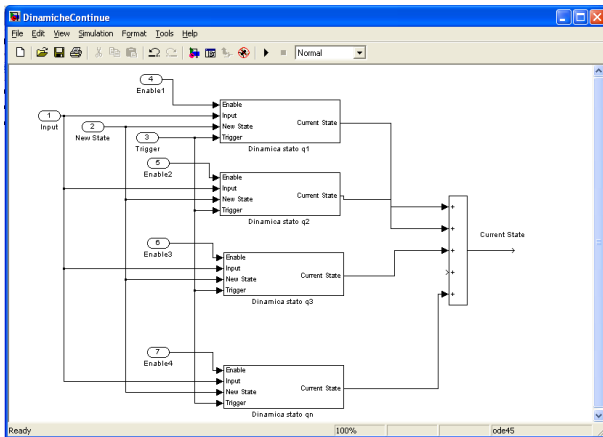
- è abilitato quando l'automa ibrido si trova nello stato discreto q mediante il segnale di `Enable = 1`
- avrà uscita pari a 0 quando l'automa ibrido **non** si trova nello stato discreto q
- in uscita restituisce il valore *corrente* dello stato continuo - `Current State`
- in ingresso ha
 - l'ingresso al sistema tempo continuo - `Input`
 - il nuovo valore dello stato tempo continuo - `New State`
 - il segnale `Trigger` che consente di *resettare* lo stato continuo

Nota:

- in generale `Input`, `Current State` e `New State` sono segnali **vettoriali**
- `New State` e `Trigger` vengono utilizzati per implementare la

Dinamiche tempo continue - 3

Nello schema complessivo si avrà



Dinamiche tempo continue - 4

Si noti che nello schema precedente

- Ogni dinamica tempo continua ha bisogno del *proprio* segnale di abilitazione `EnableN`
- I segnali `Input`, `New State` e `Trigger` sono comuni a tutti i blocchi
- Le uscite dei blocchi sono sommate tra loro (essendo attivo un solo blocco alla volta nel caso di automi ibridi **deterministici**)

Interfaccia del grafo Stateflow - 1

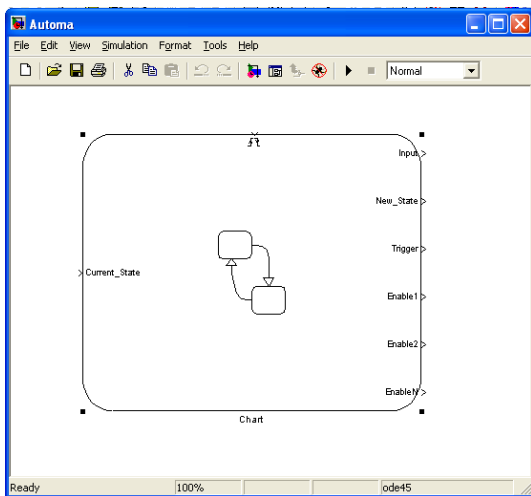
La dinamica ad eventi discreti di una automa ibrido viene descritta attraverso un grafo Stateflow

Interfaccia

L'interfaccia del grafo Stateflow deve comprendere

- il segnale d'ingresso `CurrentState`
- il segnale d'uscita (ingresso alla dinamica continua) `Input`
- il segnale d'uscita `NewState`
- il segnale d'uscita `Trigger`
- i segnali d'uscita `EnableN` per l'abilitazione delle dinamiche continue
- l'evento d'ingresso `CLOCK` per richiamare forzare l'esecuzione del grafo

Interfaccia del grafo Stateflow - 2



Definizione dello spazio di stato discreto

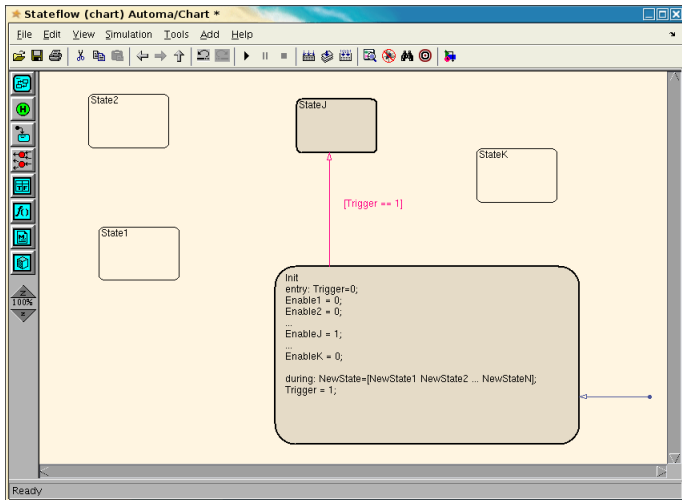
Se l'automa H possiede k stati discreti $Q = \{q_1, q_2, \dots, q_k\}$, il grafo Stateflow deve prevedere $k + 1$ *OR-state*
 $\tilde{Q} = \{Init, State1, \dots, StateK\}$

Lo stato *Init*

Lo stato *Init*

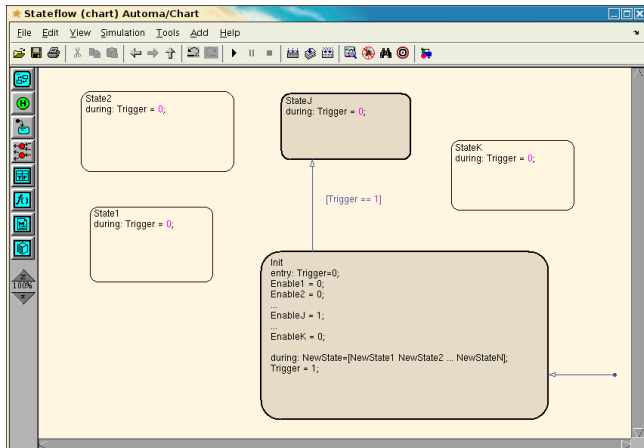
- è lo stato iniziale del grafo Stateflow
- effettua l'inizializzazione dell'automa ibrido, abilitando la dinamica iniziale j ($Enable_j = 1$) ed inizializzando lo stato continuo
- una volta effettuata l'inizializzazione dello stato continuo, transita nello stato discreto iniziale $StateJ$ corrispondente a q_j

Lo stato *Init*



Definizione delle azioni associate agli stati

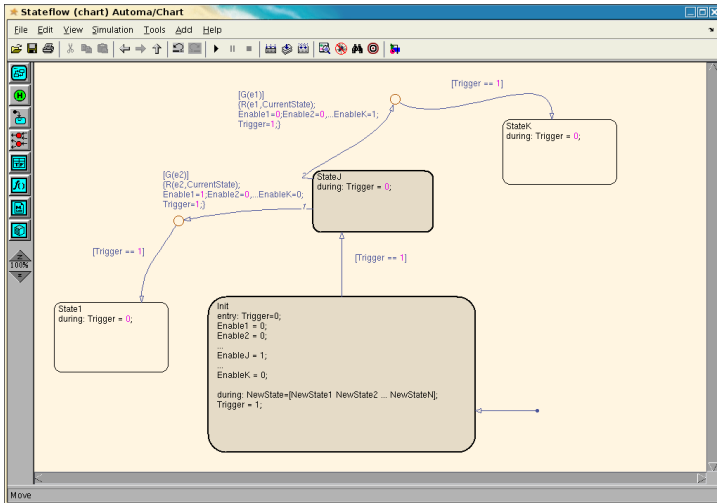
All'interno di ogni stato discreto $State1$, $State2$, \dots , $StateK$ va definita la sola azione $Trigger = 0$ di tipo *during*.



Definizione delle transizioni

- Ad ogni arco $e = (q_m, q_n) \in E$ deve essere associata una coppia di transizioni. La prima transizione sarà uscente dallo stato *StateM*, mentre la seconda sarà entrante nello stato *StateN*
- La condizione associata alla transizione uscente da *StateM* sarà la **condizione di guardia** $G(e)$ - $[G(e)]$
- Le azioni associate alla transizione uscente da *StateM* saranno:
 - la **reset map** $R(e, CurrentState)$
 - l'abilitazione della dinamica continua *N*-ma - $Enable1 = 0; Enable2 = 0; \dots; EnableN = 1; \dots; EnableK=0;$
 - la generazione dell'evento di reset $Trigger = 1;$
- La condizione associata alla transizione entrante in *StateN* sarà $[Trigger == 1]$

Transizioni del grafo Stateflow



Schema complessivo

