

Simulazione di sistemi ibridi manifatturieri

Osvaldo Barbarisi, Carmen Del Vecchio, Luigi Glielmo, Francesco Vasca

Viene proposto un modello ibrido per processi manifatturieri. I lavori (job) sono processati attraverso una rete di stazioni di lavorazione (macchine) e buffer. Ciascuna delle macchine è modellata come un sistema a stati finiti. In ogni stato di funzionamento le condizioni operative sono modellate mediante un opportuno sistema dinamico. I collegamenti tra le macchine sono realizzati per mezzo dei buffer che sono modellati a loro volta come sistemi a stati finiti. Il modello è implementato sia in Matlab sia in Witness, per confrontarne le prestazioni per la simulazione di modelli ibridi di sistemi manifatturieri.

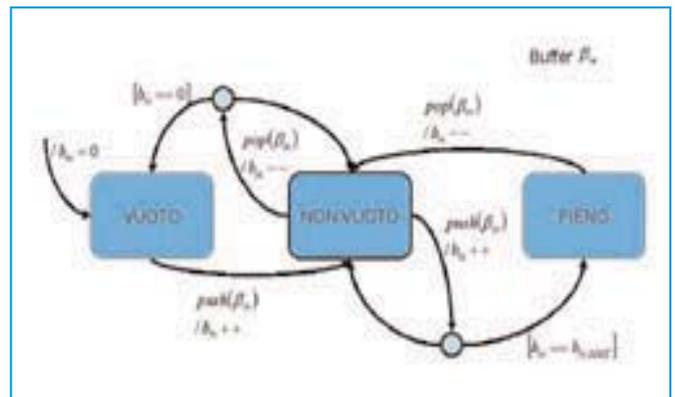


Figura 1 - Automa del buffer

La teoria dei sistemi ibridi è stata recentemente applicata anche ai sistemi manifatturieri [1, 2, 3, 4]; essa permette di modellare efficacemente le dinamiche continue e quelle a eventi delle macchine di produzione. La dinamica continua riguarda la lavorazione in senso stretto e comprende l'operazione svolta e il controllo del processo produttivo.

La dinamica a eventi descrive la macchina come sistema a stati finiti; lo stato evolve quando avviene un particolare evento che può dipendere dall'interazione con le altre macchine dell'impianto o da condizioni verificate dalle variabili descrittive della dinamica continua. Se da un lato la teoria dei sistemi ibridi si presta a modellare gli impianti manifatturieri, dall'altro i software presenti in commercio non supportano soddisfacentemente la loro simulazione.

I software specifici per la simulazione di impianti (quali Witness, Arena ecc.) permettono di rappresentare la dinamica a eventi, ma non quella continua a meno di non ricorrere ad alcuni artifici. D'altro canto, i software utilizzati per simulare le dinamiche tempo continuo o tempo discreto (quali Matlab) non sempre offrono strumenti specifici per la simulazione ad eventi, e qualora questi siano presenti (ad esempio il toolbox Stateflow, nel caso di Matlab) il loro adattamento alle caratteristiche dei sistemi manifatturieri è laborioso e il risultato finale non sempre utilizzabile da un utente non esperto di programmazione. In questo lavoro si presenta

un confronto tra due software per la simulazione di sistemi manifatturieri modellati come sistemi ibridi. Witness, prodotto dalla Lanner Group [5], è un software dedicato alla simulazione di impianti. Esso offre moduli predefiniti per la rappresentazione degli elementi di una linea di produzione (macchine, conveyor, magazzini, parti da lavorare ecc.), misure e statistiche sull'efficienza delle singole macchine e della linea, animazioni che riproducono il flusso dei pezzi nell'impianto reale, ma non moduli specifici per l'integrazione delle equazioni differenziali. Simulink di Matlab prodotto dalla Mathworks [6], presenta librerie di programmi orientati alla simulazione di sistemi tempo continuo e discreto ed è corredato di un toolbox, Stateflow [7], dedicato alla simulazione di macchine a stati finiti. Le caratteristiche delle macchine manifatturiere devono essere costruite ad hoc.

Un modello per la simulazione di un job-shop con un controllo tempo continuo sulla lavorazione delle macchine è stato realizzato utilizzando entrambi i software. Essi sono stati confrontati secondo criteri quantitativi quali numero di componenti elementari del simulatore e i tempi di simulazione e criteri qualitativi quali rappresentazione dei risultati, facilità di realizzazione del modello, modularità, manutenibilità, flessibilità.

Modello ibrido

Il job-shop è una struttura di un processo produttivo manifatturiero caratterizzato da un insieme di differenti macchine e processi che eseguono le operazioni di alcuni lavori (job). Si consideri un job caratterizzato da un insieme di L lavori

$$J = \{J_1, J_2, \dots, J_j, \dots, J_L\}$$

Ing. O. Barbarisi, dottorando di ricerca, barbarisi@unisannio.it, dott. C. Del Vecchio, dottore di ricerca - assegnista di ricerca, c.delvecchio@unisannio.it, prof. L. Glielmo, professore ordinario, direttore del Dipartimento di Ingegneria, glielmo@unisannio.it, prof. F. Vasca, professore associato, vasca@unisannio.it; Dipartimento di Ingegneria, Università del Sannio, Benevento, www.ing.unisannio.it/grace.

dall'insieme di macchine

$$M = \{M_1, M_2, \dots, M_m, \dots, M_M\} \cup \{M_D, M_S, M_F\}$$

costituito da M macchine operatrici e da tre macchine fittizie utili alla descrizione del problema, e dall'insieme di buffer

$$B = \{\beta_1, \beta_2, \dots, \beta_m, \dots, \beta_M\}$$

associati alle M macchine operatrici. Ogni job deve essere lavorato da alcune o tutte le macchine operatrici. Esso è inizialmente processato nella macchina M_D e inviato alla prima delle macchine operatrici previste dalla sua sequenza di lavorazione.

Il percorso seguito dal job tra le macchine operatrici non è deterministico: se non rientra nei parametri standard della produzione esso può essere reindirizzato (corrispondente alla più probabile sequenza di lavorazione) anche verso una delle macchine che lo hanno già lavorato oppure scartato. I job ultimati sono inviati alla macchina M_F , quelli scartati alla macchina M_S . Le probabilità di transizione tra una macchina e un'altra di ogni job sono assegnate in L matrici (una per ogni job) $T_j = R^{(M+3) \times (M+3)}$.

$$T_j = \begin{pmatrix} t_{11}^j & t_{12}^j & \dots & t_{1k}^j & \dots & t_{1M}^j & 0 & t_{1S}^j & t_{1F}^j \\ t_{21}^j & t_{22}^j & \dots & t_{2k}^j & \dots & t_{2M}^j & 0 & t_{2S}^j & t_{2F}^j \\ \dots & \dots \\ t_{m1}^j & t_{m2}^j & \dots & t_{mk}^j & \dots & t_{mM}^j & 0 & t_{mS}^j & t_{mF}^j \\ \dots & \dots \\ t_{M1}^j & t_{M2}^j & \dots & t_{Mk}^j & \dots & t_{MM}^j & 0 & t_{MS}^j & t_{MF}^j \\ \dots & \dots \\ t_{D1}^j & t_{D2}^j & \dots & t_{Dk}^j & \dots & t_{DM}^j & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 & 0 & 0 & 1 \end{pmatrix} \quad j = 1, \dots, L \quad (1)$$

dove l'elemento t_{mk}^j indica la probabilità di transizione del lavoro J_j dalla macchina M_m alla macchina M_k . A ciascuna macchina operatrice M_m è associato un buffer d'ingresso β_m contenente i job in attesa di essere lavorati. Il buffer è gestito con logica First In-First Out ed il numero di job contenuti nel buffer è indicato dalla variabile intera

$$b_m \in [0, b_{m,MAX}]$$

Ciascun buffer β_m è modellato mediante una macchina a stati finiti (Figura 1). Il buffer può trovarsi nello stato vuoto, non-vuoto o pieno a seconda del numero dei pezzi in esso contenuti. Lo stato iniziale dell'automa è vuoto.

L'arrivo di un nuovo semi-lavorato è individuato dal segnale esterno *push*(β_m). Il buffer, quindi, va nello stato *non vuoto* e la variabile b_m è incrementata. All'occorrenza di altri segnali *push* e *pop*, il valore di b_m viene incrementato e decrementato rispettivamente. Se b_m raggiunge il massimo valore $b_{m,MAX}$ allora il buffer va nello stato *pieno*. Se b_m assume va-

lore nullo, allora il buffer va nello stato *vuoto*. Le macchine operatrici M_m svolgono tutte la stessa sequenza di operazioni: prelevano i semilavorati che arrivano nel proprio buffer, identificano il tipo di job, lo processano, e, una volta ultimata l'operazione, inseriscono il semilavorato nel buffer d'ingresso della macchina successiva. Ciascuna macchina è modellata

come un sistema ibrido, formato da un automa a stati finiti che descrive la condizione operativa della macchina e da una parte dinamica che modella la lavorazione eseguita sul job. La macchina può trovarsi in uno dei seguenti stati:

- *libera*: il buffer di ingresso della macchina è vuoto e quindi essa non può operare;
- *occupata*: la macchina è impegnata a operare su un semilavorato;
- *attesa*: la macchina ha terminato di operare su un lavoro ma il buffer in

ingresso della macchina successiva è pieno.

La dinamica tempo continuo della macchina M_m varia a seconda dello stato in cui essa si trova. Indicando con *l*, *o*, *a*, rispettivamente gli stati *libera*, *occupata* e *attesa*, la dinamica della macchina può essere sinteticamente rappresentata dall'equazione

$$\dot{x}_m = f_j^i(x_m, u_{mj}^i), \quad i \in \{l, o, a\} \quad \text{ed} \quad j = 1, \dots, L \quad (2)$$

dove x_m è il vettore di stato della macchina, u_{mj}^i è il vettore degli ingressi che dipende dallo stato in cui la macchina si trova e dal job. Al modello (2) è associata una diversa condizione iniziale ogni qual volta lo stato dell'automa cambia. Il sistema ibrido che descrive la macchina è mostrato in Figura 2.

I simulatori

Il flusso produttivo del job-shop oggetto di studio è stato simulato con Matlab/Simulink/Stateflow versione 5.0 (in seguito chiamato semplicemente Stateflow) e con Witness versione 1.0. In entrambi i casi il simulatore è stato realizzato rispettando i criteri di modularità, manutenibilità, flessibilità e facilità di utilizzo per un utente non esperto.

Ogni macchina è stata implementata usando uno schema a blocchi comune creato in modo tale che l'aggiunta di nuove macchine o di nuove funzionalità per la macchina nel job-shop fosse facilmente realizzabile. I simulatori forniscono in output le statistiche comunemente utilizzate per la valutazione di un processo produttivo: il numero di pezzi realizzati e scartati per ogni tipologia di job, i tempi di attraversamento del job, la percentuale di utilizzo delle macchine (ossia il rapporto tra il tempo di occupazione ed il tempo totale di simulazione). L'aspetto di maggiore interesse del simulatore realizzato con Stateflow è l'implementazione del passaggio dei pezzi da una macchina all'altra; esso è stato ottenuto utilizzando variabili di tipo globale, cioè aggiornabili da ogni macchina, in cui sono memorizzate le informazioni sui semilavorati contenuti nei buffer. La variabili sono modificate quando una macchina preleva un semilavorato dal proprio

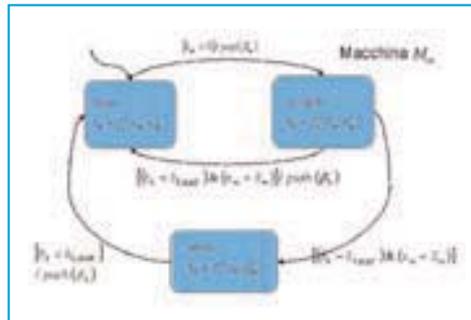


Figura 2 - Descrizione della macchina

buffer e quando ultimata la lavorazione, un job viene depositato nel buffer della macchina successiva. La realizzazione della parte dinamica della macchina è l'aspetto più interessante del simulatore realizzato con Witness. L'implementazione del sistema (2) comporta la definizione di variabili di stato continue e l'utilizzo della funzione di integrazione; questi elementi non sono predefiniti in Witness.

Per aggirare l'ostacolo sono stati utilizzati i blocchi per la modellizzazione del flusso (fluidi, serbatoi e condotte) presenti nella libreria di Witness. Il serbatoio simula la funzione integrale. Il livello di fluido in esso contenuto è la variabile di stato continua. La differenza dei flussi in ingresso ed in uscita al serbatoio rappresenta la variazione della variabile di stato nel tempo. Quindi in Witness ogni variabile di stato del sistema (2) è stato associato un serbatoio.

I tempi di calcolo del simulatore in Witness sono decisamente più lunghi di quello in Stateflow. Witness non ha implementati algoritmi di integrazione ad hoc, quindi le equazioni differenziali sono risolte utilizzando un passo di integrazione piccolo e costante. Stateflow è corredato di algoritmi di integrazione a passo variabile e tra questi è possibile scegliere quello

$$T_1 = \begin{pmatrix} 0 & 0.95 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0.95 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0.9 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$T_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0.8 & 0.1 & 0 & 0 & 0.1 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$T_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0.3 & 0.1 & 0.5 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0.9 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

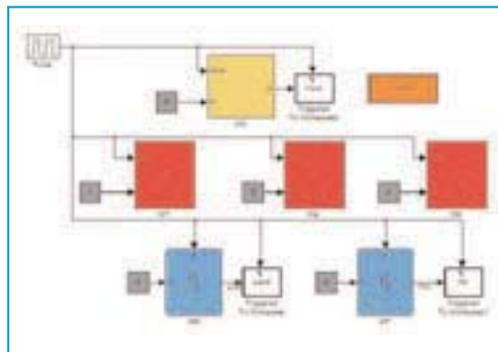
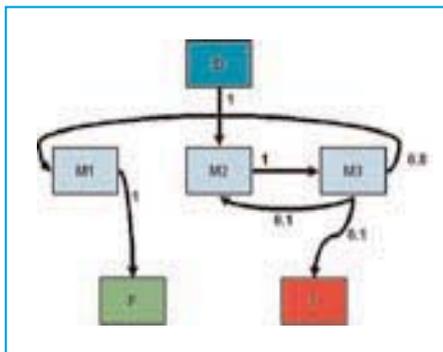


Figura 3 - Sequenza di lavorazioni per il job J_2 Figura 4 - Schema Stateflow del caso di studio

che integra meglio le equazioni differenziali del problema. Per quanto concerne lo sviluppo delle statistiche richieste in uscita al simulatore, nel modello realizzato con Stateflow è stato necessario aggiungere variabili specifiche per la registrazione dei tempi di lavoro e di attesa delle macchine e i tempi di ingresso ed uscita dei job. Queste informazioni sono invece automaticamente registrate e fornite in output da Witness. I due simulatori sono confrontabili per flessibilità e manutenibilità; l'aggiunta di nuovi moduli o la variazioni di quelli già esistenti è in parte dipendente dalla esperienza del programmatore. Tuttavia l'utilizzo di Stateflow richiede una conoscenza elementare di linguaggi di programmazione.

Risultati di simulazione

Per confrontare i due simulatori è stato realizzato un job-shop costituito da tre macchine riempitrici $M = \{M_1, M_2, M_3\} \cup \{M_D, M_S, M_F\}$ e da tre job $J = \{J_1, J_2, J_3\}$ rappresentativi di tre diversi tipi di serbatoi. Il processo produttivo consiste nel riempire i tre serbatoi con i differenti tipi di liquido versati dalle macchine operatrici. Supponiamo che le matrici di transizioni siano così definite:

In Figura 3 è rappresentato, per esempio, il percorso del lavoro J_2 . Su ogni arco è indicata la probabilità che il lavoro passi da una macchina a un'altra. La dinamica della macchina M_m nello stato occupata è composta dal processo in senso stretto e dal sistema di controllo. Il processo è descritto dalle seguenti equazioni

$$\dot{x}_{m1} = \alpha_1 x_{m2} v_m \quad (6a)$$

$$\dot{x}_{m2} = -\alpha_2 x_{m1} + \alpha_2 x_{m2} + \alpha_3 \quad (6b)$$

dove x_{m1} è l'altezza del liquido nel contenitore, v_m è il flusso di liquido versato nel contenitore, α_1 è l'inverso della capacità del contenitore, x_{m2} è un parametro di impianto che varia molto lentamente (α_1 molto piccolo) e la sua evoluzione dipende dall'ingresso dell'impianto. Nel caso in studio x_{m2} rappresenta il termine di correzione del flusso di ingresso del liquido. Il controllo del livello è ottenuto con una retroazione di tipo proporzionale

$$v_m = k(r_m^j - x_{m1}) \quad (7)$$



Figura 5 - Modello Witness che implementa il caso di studio

dove r_m^j è il livello di riferimento, e k è la costante di proporzionalità del controllore. Sostituendo la (7) nella (6), si ottiene il modello (2) relativo allo stato occupato. Quando la macchina si trova nello stato *attesa* o *libera* il controllo è disattivato, ovvero l'equazione (7) è sostituita dalla seguente

$$v_m = 0 \quad (8)$$

e lo stato x_1 è settato a zero. In Figura 4 è illustrato lo schema Stateflow che implementa il caso di studio. In Figura 5 è illustrato il modello realizzato in Witness.

Conclusione

In questo lavoro è stato presentato un esempio di simulazione di un job-shop modellato come sistema ibrido utilizzando due software commerciali: Matlab e Witness. Entrambi presentano alcuni limiti di implementazione: Witness consente di simulare efficacemente la dinamica ad eventi, Matlab quella tempo continuo. Il caso di studio implementato non è esaustivo delle caratteristiche di un sistema manifatturiero (mancano, ad esempio, i sistemi di trasporto dei job, i guasti delle macchine ecc.), ma ben si presta a rappresentare un modello di impianto ibrido poiché presenta sia la dinamica tempo continuo utilizzata per modellare il processo di lavorazione sia la dinamica ad eventi che regola il flusso dei job e lo stato delle macchine. Dal confronto tra i due software Matlab sembra essere più idoneo alla simulazione di sistemi manifatturieri ibridi in quanto consente di scrivere istruzioni ad hoc per realizzare le funzionalità necessarie all'implementazione della dinamica a eventi; il suo utilizzo tuttavia richiede una conoscenza, seppur di base, dei linguaggi di programmazione. È in fase di studio l'utilità e la possibilità di far comunicare i due software in modo tale che la parte dinamica della simulazione sia svolta utilizzando Matlab e quella ad eventi usando Witness.

Bibliografia

- [1] J. S. Smith, I. Castillo, *Formal modelling methodologies for control of manufacturing cell: Survey and comparison*, Journal of Manufacturing System, 21(1):40-58, 2002.
- [2] Jihui Zhang, Likuan Zhao, Wook Hyun Kwon, *Scheduling and optimization for a class of single-stage hybrid manufacturing systems*, Ieee Proceeding of the Inter. Conf. on Robotics and Automation, pag. 3115-3120, 2001.
- [3] D. L. Pepyne, C. G. Cassandras, *Optimal control of hybrid systems in manufacturing*, Proceedings of Ieee, 88(7):1108-1123, July 2000.
- [4] F. Balduzzi, G. Menga, *Open-loop feedback control for hybrid manufacturing systems*, Ieee Proc. of the 39th Conf. on Decision and Control, pag. 3144-3150, December 2000.
- [5] Lanner Group, Inc, Houston, Texas, Learning Witness, 1998.
- [6] A. Cavallo, F. Setola, F. Vasca, *La nuova guida a Matlab Simulink e Control Toolbox*, 2002.
- [7] The MathWorks, Inc., Natick Massachusetts, Stateflow and Stateflow Coder User's Guide, 5th edition, 2002.