Hybrid Analysis of Automated Manufacturing Systems Using Discrete Linear Inclusions*

†Fabio Balduzzi, ‡Alessandro Giua, †Giuseppe Menga

† Dipartimento di Automatica ed Informatica, Politecnico di Torino

Corso Duca degli Abruzzi 24 , 10129 Torino, Italy

(balduzzi, menga)@polito.it

‡ Dipartimento di Ingegneria Elettrica ed Elettronica, Università di Cagliari

Piazza d'Armi, 09123 Cagliari

giua@diee.unica.it

Abstract

In this paper we present an hybrid formulation for the modeling and control of automated manufacturing systems. This original approach leads to a linear time–varying discrete–time state variable model which allows fast and direct design of the system configuration. The problem is addressed by splitting the discrete event dynamic process into two hierarchical layers and defining what we call macro–events and macro–states. Then the stability of the system is considered in terms of the stability of discrete linear inclusions which evolution in time is described by language theory and finite automaton methods. We derive structural properties of such model and we make evidence of an intriguing relationship existing between idempotent matrices and this hybrid formulation of the discrete event dynamic processes.

1 Introduction

In this paper we develop an hybrid model of discrete event dynamic processes that combines aspects of automata and linear systems theory, thus exploring potential interconnections between those classes of systems. We considered automated manufacturing systems with unreliable machines, buffers of finite capacity, arbitrary service time distributions and routing policies, where several parts of a single class of products are circulated and processed. The main contribution of this work is to develop a mathematical descriptions of the process which specifies a mechanism for determining when a limited number of events do occur. Precisely we adopt the *fluid approximation theory* to derive a discrete-time linear state variable model which can be used to evaluate performance measures and make gradient estimation. This is achieved by splitting the discrete event process into two hierarchical layers and defining what we call *macro–events* and

*Published as: F. Balduzzi, G. Menga, A. Giua, "Hybrid Analysis of Automated Manufacturing Systems Using Discrete Linear Inclusions," *Proc. IEEE 37th Int. Conf. on Decision and Control* (Tampa, Florida), pp. 1710-1715, December, 1998.

macro–states.

The evolution in time of an automated manufacturing systems is discussed within a framework that distinguishes two levels of aggregation. At the lower level, the microscopic behavior of arrivals and departures of parts to(from) each machine is modeled using first order fluid approximations. At the higher level a discrete event model will represent the transitions through a sequence of *macro–states* at the occurrence of a limited number of events, the macro–events. Both levels of the process are represented by a discrete–time linear state variable model which describes the temporal behavior of a timed event graph that represents the evolution of a *discrete linear inclusion* (DLI) through a sequence of admissible macro–states.

Specifically the stability of the system is considered in terms of the stability of the associated DLI and we show that steady regimes are defined by sets of idempotent matrices all infinite product of which do converge. This result highlights an intriguing new relationships that have emerged with automata theory, discrete event systems and *idempotent analysis* as recently mentioned in [6]. This original formulation also has implications for efficient simulations of discrete event dynamic systems (DEDS) as the macro–event simulators can take into account the underlying algebraic structure for performance modeling and evaluation.

1.1 Previous Work

A substantial body of literature about DEDS is concerned with the analysis of automated manufacturing systems. Even so analytic results are available only for simple systems. Then discrete event simulation models have been extensively used for analysis and design issues. Combining simulation and elementary queueing analysis Phillis and Kouikoglou [10] proposed a discrete event continuous flow model more efficient than conventional simulators, to predict the system evolution. Continuous flow models for transfer lines have also been studied by Suri in [5]. In a recent work Balduzzi and Menga [1] developed a discrete–time linear stochastic state variable model for the fluid approximation of flexible manufacturing systems. Then, by using perturbation analysis techniques [8] they obtained average values and variances of both performance measures and of their gradients with respect to the system parameters to perform optimal design of the system configuration.

2 Description of the Model

The queueing network considered in this work consists of a set of n single-server stations, denoted by M_i, for $i = 1, \ldots, n$, serving a single class of product. Parts move generically from machine M_i to M_j according to the their production cycle and are queued in buffers, one for each machine, of finite capacity C_i. Since machines are unreliable we consider failures as *operation–dependent failures* and we define for each machine the *production volume before breaking*, denoted by w_i, and the *repairing time*, denoted by d_i, both assumed as independent identically distributed random variables. Machine service times are also assumed independent random variables with identical distribution with mean τ_i. The maximum average machine production rates are indicated with $v_{MAX,i}$, i.e. $v_{MAX,i} = \frac{1}{\tau_i}$, and we denote by $\tilde{w}_i$ and $\tilde{d}_i$ the samples of the random variables w_i and d_i drawn during a simulation run.

2.1 The microscopic level: *Fluid Approximations*

We consider fluid approximations to model the microscopic behavior of an automated manufacturing systems under conditions of balanced heavy loading which involves smoothing of discrete processes [3]. We denote with $\Delta_k = [t_k, t_{k+1}]$ the interval of time between the occurrence of consecutive macro–events at times t_k, for $k = 0, 1, 2, \ldots$, called *macro–periods*. We assume that within each macro–period the stochastic processes which describe the queueing system are stationary. Then the global behavior of the system will be piecewise stationary.

We now derive a first order fluid approximation of the stochastic counting processes accounting for the number of departures of parts from machine M_i to M_j. These processes are assumed independent among different sources and destinations and are approximated by continuous processes denoted with $\{N_{i,j}(t),\, t \geq 0\}$. The average values $E[N_{i,j}(t)]$ correspond to the first order fluid quantities, the flow rates $v_{i,j}(t)$, which are piecewise constant over the entire time horizon. We find that [3], [9]

$$N_{out,i}(t_{k+1}) = N_{out,i}(t_k) + v_{out,i}(t_k) \cdot (t_{k+1} - t_k)$$
$$N_{in,i}(t_{k+1}) = N_{in,i}(t_k) + v_{in,i}(t_k) \cdot (t_{k+1} - t_k)$$

where

$$N_{in,i}(t_k) = \sum_h N_{h,i}(t_k), \quad N_{out,i}(t_k) = \sum_j N_{i,j}(t_k)$$

are the input and output fluid processes at the beginning of each macro–period,

$$v_{in,i}(t_k) = \sum_h v_{h,i}(t_k), \quad v_{out,i}(t_k) = \sum_j v_{i,j}(t_k)$$

are the inflow and outflow rates of parts which are constant values within the macro–periods. Then the buffer levels, denoted by $x_i(t)$, are simply obtained as

$$x_i(t_{k+1}) = x_i(t_k) + [v_{in,i}(t_k) - v_{out,i}(t_k)](t_{k+1} - t_k)$$

2.2 The macroscopic level: *Macro–States* and *Macro–Events*

The system evolves through a sequence of *macro–states*, characterized by the functional status of the physical components of all services: the machines, *operational* or *down* and the buffers, *full*, *not full–not empty*, *empty*. We denote the set of macro–event types by $\mathcal{E} = \{F_i, R_i, BE_i, BF_i, RC_i\}$ which elements are defined as follows:

- $\mathrm{F_i}$ (*Failure of machine* M_i). After a machine is repaired, failures will occur after the production volume $\tilde{w}_i$.
- $\mathrm{R_i}$ (*Repair of machine* M_i). When a machine fails, it will be repaired after $\tilde{d}_i$ time units.
- $\mathrm{BF_i}$ (*Buffer Full at machine* M_i). The buffer level reaches its capacity C_i while $v_{in,i}(t) \geq v_{out,i}(t)$.
- $\mathrm{BE_i}$ (*Buffer Empty at machine* M_i). The buffer level reaches 0 while $v_{in,i}(t) \leq v_{out,i}(t)$.

- $\mathrm{RC_i}$ (*outflow Rate Changed at machine* M_i). When the machine is operational, upon the occurrence of exogenous macro–events, changes do occur in the inflow and outflow rates of parts of machine M_i.

Let $\mathcal{M} = \{\mathbf{M_i^o}, \mathbf{M_i^b}\}$ and $\mathcal{B} = \{\mathbf{B_i^f}, \mathbf{B_i^e}, \mathbf{B_i^n}\}$ be the machine status and the buffer status set types respectively. The set of admissible macro–states is denoted by $Q = \{(m_1, b_1, \ldots, m_n, b_n) \mid m_i \in \mathcal{M}, b_i \in \mathcal{B}\}$. Therefore the macro–states of the system, $q_i \in Q$, combining the functional status of all services, may only change due to the occurrence of the macro–events.

The machine status $m_i \in \mathcal{M}$ may take the following symbolic values: $\mathbf{M_i^o}$ (*Machine Operational*) and $\mathbf{M_i^b}$ (*Machine Broken*). Note that when a machine breaks down then its production rate is $v_{out,i}(t_k) = 0$. The buffer status $b_i \in \mathcal{B}$ may take the following symbolic values: $\mathbf{B_i^f}$ (*Buffer Full*), $\mathbf{B_i^e}$ (*Buffer Empty*) and $\mathbf{B_i^n}$ (*Buffer not Full–not Empty*).

These specifications allow us to describe the dynamics of the system in a form that enables GSMP representation. Each macro–event is associated with a clock representing its residual lifetime and each clock has a rate (the clock speed) at which it runs down. When the clock associated with a macro–event goes to 0, then that macro–event does occur. Note that upon the occurrence of a macro–event changes may occur to the clock speeds and the macro–state of the system.

3 A Discrete Time Linear State Variable Model

In this section we develop a discrete–time linear state variable model which represents both layers of the discrete event process. We adopt a dynamic control policy to myopically generate an optimal solution for the average machine production rates, that is obtained by solving a sequence of linear programming problems, one for each macro–period. Each solution represents the average flow rates $v_{i,j}(t_k)$ to be expected for the macro–period Δ_k. Precisely when the system enters a new macro–state those rates, as reference values for the next macro–period, are determined as the solution of the following LP problem:

$$J = \max \sum_i v_{out,i}(t_k), \qquad \textit{subject to:}$$

$$\begin{cases} 0 \le \sum_h v_{i,h}(t_k) \le v_{MAX,i} & \\ \sum_j v_{j,i}(t_k) \le \sum_h v_{i,h}(t_k), & \forall\, M_i \mid b_i = \mathbf{B_i^f} \\ \sum_h v_{i,h}(t_k) \le \sum_j v_{i,j}(t_k), & \forall\, M_i \mid b_i = \mathbf{B_i^e} \\ \sum_j v_{i,j}(t_k) = 0, & \forall\, M_i \mid m_i = \mathbf{M_i^b} \end{cases}$$

where the objective function J has been conceived in order to maximize machines utilization.

We now describe a discrete–time linear state variable model which samples are the occurrence of macro–events at time t_k, for $k = 0, 1, 2, \ldots$. Let us define the input signal $u(k)$ as $t_{k+1} = t_k + u(k)$ and for each machine we arrange the following set of equations:

$$x_i(t_{k+1}) = x_i(t_k) + [v_{in,i}(t_k) - v_{out,i}(t_k)]u(k) \tag{1}$$

$$\chi_i(t_{k+1}) = \alpha_i(k)\chi_i(t_k) + v_{out,i}(t_k)u(k) \tag{2}$$

$$s_i(t_{k+1}) = \beta_i(k)s_i(t_k) + \beta_i(k)u(k) \tag{3}$$

where $\alpha_i(k)$, $\beta_i(k) \in \{0, 1\}$. For simplicity of presentation we consider in this work production systems with a single class of part flowing through and in what follows we do not take into account fluctuations in the average values of state variables x_i, χ_i and s_i due to the approximations introduced by the fluid model. The complete stochastic state variable model can be found in [1]. At the occurrence of the macro–events equation (1) accounts for the current buffer levels. Equations (2) and (3) represent the partial production volume currently processed by the machine next to fail since the last repair, and the time spent by the machine under repairing since the last failure, respectively. Indicators α_i and β_i are switched from 0 to 1 at the occurrence of the proper macro–events. That is when a machine breaks down, say machine M_i, its current production rate will be $v_{out,i}(t) = 0$, then $\alpha_i = 0$ and $\beta_i = 1$. When M_i is repaired then $v_{out,i}(t) = v_{MAX,i}$, then $\alpha_i = 1$ and $\beta_i = 0$.

The state of the system is represented by the vector $\mathbf{x}(k)$ defined as:

$$\mathbf{x}(k) = [t_k, x_1(t_k), \chi_1(t_k), s_1(t_k), \ldots \\ \ldots, x_n(t_k), \chi_n(t_k), s_n(t_k), x_{n+1}(t_k)]^T$$

which dimension is $m = dim[\mathbf{x}(k)] = 3n + 2$. In fact there are $3n$ equations as defined through Eq. (1)–(3) plus the two extra equations:

$$t_{k+1} = t_k + u(k) \tag{4}$$

$$x_{n+1}(t_{k+1}) = x_{n+1}(t_k) + v_{out,n}(t_k)u(k) \tag{5}$$

where state variables t_k and $x_{n+1}(t_k)$ occupy the first and the last position within the state vector $\mathbf{x}(k)$, respectively. These equations are accounting for the current time of the occurring macro–event and for the final storage area (total production).

We can now formally describe the behavior of this hybrid system with a linear time–varying discrete–time state variable model which can be expressed in matrix notation as:

$$\mathbf{x}(k+1) = \mathbf{D}(k)\mathbf{x}(k) + \mathbf{b}(k)u(k) \tag{6}$$

Precisely Eq. (6) describes the temporal behavior of a timed event graph that represents the evolution of an hybrid systems through the sequence of admissible macro–states. That is, if $\mathbf{x}(k)$ denotes the macro–state variable vector at the k^{th} occurrence times of the different macro–events, then there exist a finite set of matrices $\Sigma_D = \{\mathbf{D}_1, \cdots, \mathbf{D}_p\}$, $\mathbf{D}_i \in \mathbb{R}^{m \times m}$, for $i = 1, \ldots, p$ such that

$$\mathbf{x}(k) = \mathbf{\Phi}(k, 0)\mathbf{x}(0) + \sum_{i=0}^{k-1} \mathbf{\Phi}(k, i+1)\mathbf{b}(i)u(i)$$

is the solution for the macro–state Equation (6) in terms of the initial state $\mathbf{x}(0)$ and the input signal $u(k)$ at step 0 and beyond, and where $\mathbf{\Phi}(k, h)$, $k > h$, is the state transition matrix

$$\mathbf{\Phi}(k, h) = \mathbf{D}(k-1)\mathbf{D}(k-2)\cdots\mathbf{D}(h-1)\mathbf{D}(h)$$

where each macro–state coupling matrix is in the finite set Σ_D. Particularly each $\mathbf{D}(k) \in \Sigma_D$ is diagonal with entries 0 and 1 and each vector $\mathbf{b}(k)$ has elements that depend on the average machine outflow rates.

With the GSMP formulation any macro–event does occur when its associated clock runs down to 0; with this model any macro–event happens when a certain state variable reaches a specified

value. Precisely when a machine breaks down or gets repaired then it must result $\chi_i(t_{k+1}) = \tilde{w}_i$ or $s_i(t_{k+1}) = \tilde{d}_i$. When a buffer gets full or empty then the condition $x_i(t_{k+1}) = C_i$ or $x_i(t_{k+1}) = 0$ will be satisfied. We indicate with $r(k+1)$ these values reached by the appropriate state variables at the occurrence of the macro–events. Then

$$r(k+1) = \mathbf{e}_j^T(k)\mathbf{x}(k+1) = \mathbf{h}(k)\mathbf{x}(k) + b(k)u(k) \tag{7}$$

$$u(k) = K(k)[r(k+1) - \mathbf{h}(k)\mathbf{x}(k)] \tag{8}$$

where $\mathbf{e}_j^T(k)$ is a vector with entries 0 and 1 which selects the state variable within vector $\mathbf{x}(k)$ that is leading to the new transition, and

$$\mathbf{h}(k) = \mathbf{e}_j^T(k)\mathbf{D}(k),\ b(k) = \mathbf{e}_j^T(k)\mathbf{b}(k),\ K(k) = \frac{1}{b(k)}$$

Equation (6) can be straightforward rewritten as a closed-loop system:

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + K(k)\mathbf{b}(k)r(k+1) \tag{9}$$

where

$$\mathbf{A}(k) = \mathbf{D}(k) - K(k)\mathbf{b}(k)\mathbf{h}(k) \tag{10}$$

represents the closed–loop system matrix.

3.1 The Macro–Behavior of the System

At the macroscopic level the evolution in time of the system can be represented by a finite automata (FA) with the output associated with the transitions through the macro–states. It is a six–tuple $\mathrm{Z} = (\mathrm{Q}, \mathrm{E}, \Sigma, \delta, \lambda, \mathrm{q}_0)$ where Q is the finite set of admissible macro–states as defined in Section 2.2, E is the input alphabet, $E = \{\ldots, F_i, R_i, BE_i, BF_i, RC_i, \ldots\}$, δ is the transition function, $\delta : Q \times E \to Q$, $q_0 \in Q$ is the initial state, Σ is the output alphabet, $\Sigma = \{\mathbf{A}_1, \ldots, \mathbf{A}_r\}$, $\mathbf{A}_i \in \mathbb{R}^{m\times m}$ for $i = 1, \ldots, r$ and λ is the map $\lambda : Q \times E \to \Sigma$. That is $\lambda(q, a)$ gives the output associated with the transition from state q on input a. The output of Z in response to input $a_1, a_2, \ldots, a_n$ is $\lambda(q_0, a_1)\lambda(q1, a_2)\cdots\lambda(q_{n-1}, a_n)$, where $q_0, q_1, \ldots, q_n$ is the sequence of macro–states such that $\delta(q_{i-1}, a_i) = q_i$ for $i \le i \le n$.

As already pointed out, we know that this hybrid system evolves through admissible macro–state sequences. Therefore, given a system configuration, the number of different macro–states is finite, hence the number of different transitions through the macro-states must be finite. This observation simply states that the transition matrices $\mathbf{\Phi}(k, h) = \mathbf{A}(k-1)\cdots\mathbf{A}(h)$, for $k > h$, associated with the closed–loop system (9), are generated by the underlying finite automaton transition structure which evolution in time is governed by the stochastic sequence of the macro–events.

Furthermore the input sequence $a_1, a_2, \ldots, a_i, \ldots$, $a_i \in E$, which corresponds to the sequence of the occurring macro–events, depends on the dynamics of the discrete event process. Particularly this sequence is not completely stochastic, that is, if all machines are down, then the next macro–event must be a *repair* event on a certain machine. Finally we can recursively derive an equivalent regular expression R from the finite automata Z, such that $\mathrm{Z} = \mathrm{L}_\Sigma(\mathrm{R})$, where $L_\Sigma(R)$ is the language described by the regular expression R made of symbols $\mathbf{A}_i \in \Sigma$. Then the transition matrix $\mathbf{\Phi}$ can be fully characterize by the associated regular expression R.

Let $q_i \in Q$ be the any admissible macro–state. We define an *elementary cycle* as the minimum path $p(q_i)$ from q_i to q_i, with $\mid p(q_i) \mid \geq 2$. Then given the finite automata Z it is possible to show that there exists a finite number of elementary cycles by deriving the equivalent regular expression R denoting the language $L_\Sigma(Z)$ that is accepted by Z. Our main goal is to derive structural properties of $\mathbf{\Phi}$ which are related to the stability of this hybrid system, by means of language theory and automaton methods. This approach lends naturally itself to the analysis of discrete linear inclusion (DLI).

4 Structural Properties of the Closed–Loop Matrices

We now present some relevant structural properties of the closed–loop matrices $\mathbf{A}_i \in \Sigma$. To give evidence of the situations we are dealing with, we introduce an example.

Example 4.1. *Let us consider a simple transfer line with two machines M_1 and M_2 coupled with an intermediate buffer B_2 of finite capacity C_2. Let $v_{MAX,1}$ and $v_{MAX,2}$ be the maximum machine production rates and assume, without loss of generality, $v_{MAX,1} > v_{MAX,2}$. We build matrices $\mathbf{D}(k)$ and $\mathbf{b}(k)$ assuming the system in the macro–state $q_k = (\mathbf{M}_1^o, \mathbf{M}_2^o, \mathbf{B}_2^n\}$ at time t_k. Let $v_1(t_k) = v_{MAX,1}$ and $v_2(t_k) = v_{MAX,2}$, then*

$$\mathbf{D}(k) = diag(1, 1, \alpha_1, \beta_1, 1, \alpha_2, \beta_2, 1)$$
$$\mathbf{b}(k) = [1, -v_1, v_1, \beta_1, v_1 - v_2, v_2, \beta_2, v_2]^T$$

where $\alpha_1 = 1$, $\alpha_2 = 1$, $\beta_1 = 0$ and $\beta_2 = 0$. Now suppose that at time t_{k+1} the macro–event BF_2 *does occur. Then*

$$\mathbf{A}(k+1) = \mathbf{D}(k) - K(k)\mathbf{b}(k)\mathbf{h}(k)$$

where

$$K(k) = \frac{1}{v_1 - v_2}, \quad \mathbf{h}(k) = [0, 0, 0, 0, 1, 0, 0, 0]$$

In general each $\mathbf{A}_i \in \Sigma = \{\mathbf{A}_1, \ldots, \mathbf{A}_r\}$ is block triangular of the form

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{U} & \mathbf{N} \\ \mathbf{0} & \mathbf{L} \end{bmatrix}$$

where $\mathbf{U}$ is diagonal, $\mathbf{L}$ and $\mathbf{N}$ are lower triangular matrices of appropriate dimensions, which diagonal entries are 0 and 1. Hence $\mathbf{A}_i$ has repeated eigenvalues 0 and 1, then it is not asymptotically stable. Nevertheless this does not necessarily means that $\mathbf{A}_i$ does not have a full complement of eigenvectors as we shall prove that each $\mathbf{A}_i$ is idempotent, i.e., it is stable.

Let us first observe the following.

Fact 4.2. Each $\mathbf{A}_i \in \Sigma$ can be written as $\mathbf{A}_i = \mathbf{D}_i - \mathbf{C}_i$, where:

1. $\mathbf{D}_i$ is a diagonal matrix of 0's and 1's.
2. $\mathbf{C}_i = [\mathbf{0}, \cdots \mathbf{0}, \mathbf{c}_{j(i)}, \mathbf{0}, \cdots \mathbf{0}]$, i.e., each $\mathbf{C}_i$ has only a column of index $j(i)$ different from zero. Index $j(i)$ corresponds to the index of the element $x_j(k)$ within the state vector which is causing the transition to the new macro–state.
3. $\mathbf{D}_i(l, l) = 1$ if $\mathbf{C}_i(l, j(i)) \neq 0$.

4. $\mathbf{D}_i(j(i), j(i)) = \mathbf{C}_i(j(i), j(i)) = 1$

We can now prove the following proposition.

Proposition 4.3. All matrices $\mathbf{A}_i \in \Sigma$ are idempotent, i.e., $\mathbf{A}_i^2 = \mathbf{A}_i$.

Proof: It is enough to show that $\mathbf{A}_i(\mathbf{A}_i - \mathbf{I}) = \mathbf{0}$. By Fact 4.2 $\mathbf{A}_i = \mathbf{D}_i - \mathbf{C}_i$, hence $\mathbf{A}_i(\mathbf{A}_i - \mathbf{I}) = \mathbf{D}_i^2 - \mathbf{D}_i\mathbf{C}_i - \mathbf{D}_i - \mathbf{C}_i\mathbf{D}_i + \mathbf{C}_i^2 + \mathbf{C}_i$. Now: $\mathbf{D}_i^2 = \mathbf{D}_i$ (by Fact 4.2.1). Also Fact 4.2 implies that $\mathbf{D}_i\mathbf{C}_i = \mathbf{C}_i\mathbf{D}_i = \mathbf{C}_i^2 = \mathbf{C}_i$. This completes the proof. ■

Let $\mathbf{A}_i \in \Sigma$ be the closed-loop matrix of the linear time invariant discrete time system $\mathbf{x}(k+1) = \mathbf{A}_i\mathbf{x}(k) + \mathbf{b}(k)u(k)$. Then Proposition 4.3 guarantees the stability of the system. Nevertheless such property do not hold for time-varying systems. In this case it is required that all infinite cycles, even those not elementary, obtained as $\prod_i \mathbf{A}_i$ for any $q_i \in Q$, are idempotent. We conjecture that this statement holds for the system expressed by Eq. (9) and in the next section we provide an algorithm to determine the finiteness of the set of all cycles. First we show that the elementary cycles Fail/Repair on machine M_i are idempotent.

Suppose that at time t_k the macro–state of the system is $q_k = (\ldots, \mathbf{M}_i^o, \mathbf{B}_i^n, \ldots)$ and the sequence of macro–events denoted by $\mathcal{C} = \{\mathrm{F}_i, \mathrm{R}_i\}$ does occur on machine M_i. Obviously after $\mathcal{C}$ the macro–state of the system will not change. Let $\mathbf{A}_{\mathrm{F_i}}$ and $\mathbf{A}_{\mathrm{R_i}}$ be the transition matrices associated with the macro–events F_i and R_i respectively. Now let us partitioned matrix $\mathbf{D}_i$ as $\mathbf{D}_e = diag(\mathbf{D}_U, \mathbf{D}_{V,e}, \mathbf{D}_L)$, where $e \in \mathcal{C}$ and the two diagonal elements in $\mathbf{D}_{V,e} \in \mathbb{R}^{2\times 2}$ are the only entries in $\mathbf{D}_e$ which are affected by the sequence $\mathcal{C}$, that is

$$\begin{matrix} \mathbf{D}_{V,\mathrm{F_i}}(1,1) = \alpha_i = 1 \\ \mathbf{D}_{V,\mathrm{F_i}}(2,2) = \beta_i = 0 \end{matrix} , \quad \begin{matrix} \mathbf{D}_{V,\mathrm{R_i}}(1,1) = \alpha_i = 0 \\ \mathbf{D}_{V,\mathrm{R_i}}(2,2) = \beta_i = 1 \end{matrix}$$

Vectors $\mathbf{b}_i$ can also be partitioned as $\mathbf{b}_e = [\mathbf{b}_U^T, \mathbf{b}_{V,e}^T(k), \mathbf{b}_L^T]^T$ where

$$\begin{matrix} \mathbf{b}_{V,\mathrm{F_i}}(1,1) = v_{out,i} > 0 \\ \mathbf{b}_{V,\mathrm{F_i}}(2,1) = \beta_i = 0 \end{matrix} , \quad \begin{matrix} \mathbf{b}_{V,\mathrm{R_i}}(1,1) = 0 \\ \mathbf{b}_{V,\mathrm{R_i}}(2,1) = \beta_i = 1 \end{matrix}$$

Proposition 4.4. *Let $q_k = (\ldots, \mathbf{M}_i^o, \mathbf{B}_i^n, \ldots)$ be the macro–state of the system at time t_k and let $\mathcal{C} = \{\mathrm{F}_i, \mathrm{R}_i\}$ be the sequence of macro–events occurring on machine M_i. Then $\mathbf{\Pi} = \mathbf{A}_{\mathrm{F_i}}\mathbf{A}_{\mathrm{R_i}}$ is* idempotent *and $\mathbf{\Pi} = \mathbf{A}_{\mathrm{F_i}}\mathbf{A}_{\mathrm{R_i}} = \mathbf{A}_{\mathrm{R_i}}$. Moreover if the macro–state of the system is $\bar{q}_k = (\ldots, \mathbf{M}_i^b, \mathbf{B}_i^n, \ldots)$ and the sequence of the occurring macro–events is $\bar{\mathcal{C}} = \{\mathrm{R}_i, \mathrm{F}_i\}$, then $\mathbf{\Pi} = \mathbf{A}_{\mathrm{R_i}}\mathbf{A}_{\mathrm{F_i}} = \mathbf{A}_{\mathrm{F_i}}$.*

Proof. By direct computation we obtain $\mathbf{\Pi} = \mathbf{A}_{\mathrm{F_i}}\mathbf{A}_{\mathrm{R_i}} = \mathbf{D}_{\mathrm{F_i}}\mathbf{A}_{\mathrm{R_i}} = \mathbf{A}_{\mathrm{R_i}}$, since $\mathbf{h}_{\mathrm{F_i}}\mathbf{D}_{\mathrm{R_i}} = \mathbf{0}$ and $\mathbf{h}_{\mathrm{F_i}}\mathbf{b}_{\mathrm{R_i}} = 0$. Therefore it must result

$$\mathbf{A}_e = \begin{bmatrix} \mathbf{D}_U & \bullet & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bullet & \mathbf{D}_L \end{bmatrix}$$

for $e \in \mathcal{C}$, where $(\bullet)$ represent vertical band matrices of appropriate dimensions. Hence recalling that $\mathbf{D}_e = diag(\mathbf{D}_U, \mathbf{D}_{V,e}, \mathbf{D}_L)$ we can easily state that $\mathbf{D}_{\mathrm{F_i}}\mathbf{A}_{\mathrm{R_i}} = \mathbf{A}_{\mathrm{R_i}}$. This observation complete the proof. ■

Based on the previous propositions we make the following conjecture.

Conjecture 4.5. The set $\{\mathbf{\Phi}_k = \mathbf{A}(k-1)\cdots\mathbf{A}(0) \mid k > 0\}$ is finite.

Conjecture (4.5), that can be proved, allows us to infer the stability of our model in the sense that the state $\mathbf{x}(k)$ is always bounded. In the next section we introduce the concept of discrete linear inclusion (DLI) for the analysis of the stability of the system and show how this conjecture can be proved.

5 Discrete Linear Inclusions

Let $\Sigma = \{\mathbf{A}_i : 1 \leq i \leq r\}$ be a finite set of constant matrices in $\mathbb{R}^{m\times m}$ and consider a system whose dynamical evolution is given by:

$$\mathbf{x}(k+1) \in \{\mathbf{A}(k)\mathbf{x}(k) \mid \mathbf{A}(k) \in \Sigma\} \tag{11}$$

Thus, Eq. (11) can be seen as the equivalent of a differential inclusion for a linear discrete time system. Let $\mathbf{x}(0)$ be the initial state. Then for all $k > 0$ it holds $\mathbf{x}(k) \in \{\mathbf{\Phi}_k\mathbf{x}(0) \mid \mathbf{\Phi}_k \in \Sigma^k\}$ where $\mathbf{\Phi}_k = \mathbf{A}(k-1)\mathbf{A}(k-2)\cdots\mathbf{A}(0)$, for $k > 0$ and $\mathbf{\Phi}_0 = \mathbf{I}$. With a notation from language theory, we have written

$$\mathbf{\Phi}_k \in \Sigma^k \equiv \underbrace{\Sigma \circ \ldots \circ \Sigma}_{k}$$

where we have considered the concatenation of two symbols $\mathbf{A}_i \circ \mathbf{A}_j$ as the matrix product $\mathbf{A}_j \cdot \mathbf{A}_i$. Let Σ^* be the set of all possible $\mathbf{\Phi}_k$, for $k \geq 0$. Assume that not all $\mathbf{\Phi}_k \in \Sigma^*$ correspond to possible evolutions. Then we consider the case in which the set of all possible $\mathbf{\Phi}_k$ is a language over Σ, denoted as $L \subseteq \Sigma^*$. Hence we can write that:

$$\mathbf{x}(k) \in \{\mathbf{\Phi}_k\mathbf{x}(0) \mid \mathbf{\Phi}_k \in L \subseteq \Sigma^*\}$$

The set L is called a *discrete linear inclusion* (DLI). In particular Σ^* is called a *free monoid* DLI. Note that we are slightly extending the usual definition of DLI as proposed in [7] where only free monoid DLI are considered. Since a DLI is a language we can use any language generator to represent it. In particular if L is a regular language then it can be represented as a finite automata where each edge is labelled by a matrix $\mathbf{A}_i \in \Sigma$.

Example 5.1. *The DLI corresponding to the discrete time state variable model of the two machines-one buffer system discussed in Example (4.1) may be represented as a finite automata Z. Each state is represented by a triple $q_{t_k} = (m_1, m_2, b_2)$ with the notation introduced in section 2.2 and each arc is labelled by a matrix $\mathbf{A}_e, i$ where $e \in \mathcal{E}$ and $i = 1, 2, \ldots$ indexes the different matrices associated to the same macro–event. Note that due to the particular choice of the system parameters (i.e., $V_1 > V_2$) certain states may result "vanishing" because as soon as the system enters one of these states then it leaves immediately. For instance a repair event from state $(\mathbf{M}_1^b, \mathbf{M}_2^o, \mathbf{B}_2^e)$ causes the system to enter the state $(\mathbf{M}_1^o, \mathbf{M}_2^o, \mathbf{B}_2^e)$ and to jump immediately to the state $(\mathbf{M}_1^b, \mathbf{M}_2^o, \mathbf{B}_2^n)$ since $V_1 > V_2$. Thus we will have some dotted edges that we assume to be labelled with the identity matrix (that takes the place of the null string for the matrix product operator).* ■

Definition 5.2. A DLI L is *stable* if and only if for all $\mathbf{\Phi} \in L$, $\mathbf{\Phi}$ is element–wise bounded, i.e., there exists a constant matrix $\mathbf{B}$ such that for all $\mathbf{\Phi} \in L$ $-\mathbf{B} \leq \mathbf{\Phi} \leq \mathbf{B}$. A DLI L is *asymptotically stable* if for all $\mathbf{\Phi}_k \in \Sigma^k \cap L$

$$\lim_{k\to\infty} \mathbf{\Phi}_k = \mathbf{0}.$$

The basic issue for any control system concerns its stability. Stability analysis of DLIs is a complex issue (readers are referred to [7], [4] and [2]). In general the DLI considered in the literature are of the form of a free monoid Σ^*. We make the following trivial observation: if $L' \subset L$ and L is (asymptotically) stable then L' is (asymptotically) stable, i.e. we may be able to apply some results taken from the literature to study DLIs that are not free monoids (norm conditions, spectral analysis). We consider a simple condition for stability that consists in the finiteness of L.

Remark 5.3. A DLI L is called *finite* if L is a finite subset of $\mathbb{R}^{m\times m}$. If L is finite then it is clearly stable.

In the case of (non-commutative) languages $\Sigma^k \cap \Sigma^j = \emptyset$ for $k \neq j$, since strings of different length are necessarily different. In the case of DLIs, however, $\Sigma^k \subseteq \mathbb{R}^{m\times m}$ for all k, and it may well be the case that L is finite even if it contains infinite products of unbounded length.

Example 5.4. Let $\mathbf{A}$ be an idempotent matrix. Then $\mathbf{A}^* = \cup_{k\geq 0}\mathbf{A}^k = \{\mathbf{I}, \mathbf{A}\}$.

In the case of regular DLIs, the procedure below may be used to check whether L is finite.

Algorithm 5.5. Let Σ be a finite set of matrices in $\mathbb{R}^{m\times m}$ and $L \subseteq \Sigma^*$ a regular DLI. Let the finite state machine associated to L be $M = (Q, \Sigma, \delta, q_0)$, where Q is the set of all states, δ is the transition relation, and q_0 is the initial state.

begin
$n_0 := (q_0, \mathbf{I})$;
NEW $:= n_0$;
$OLD := \emptyset$;
while NEW $\neq \emptyset$ **do**
  begin
  choose $n = (q, \mathbf{\Phi}) \in$ NEW ;
  forall $q_i \in Q$ **such that** $q_i = \delta(q, \mathbf{A}_i)$ **do**
    begin
    $n_i := (q_i, \mathbf{A}_i \cdot \mathbf{\Phi})$;
    if $n_i \notin$ (NEW $\cup$ OLD) **then**
      NEW $:=$ NEW $\cup \{n_i\}$;
    end;
  NEW $:=$ NEW $\setminus \{n\}$;
  OLD $:=$ OLD $\cup \{n\}$;
  end;
end.

We now prove that this procedure computes all elements of a finite L.

Proposition 5.6. Let $L \subseteq \Sigma^*$ be a regular DLI. Algorithm 5.5 terminates if and only if L is finite. Furthermore, when it terminates $L = \{\mathbf{\Phi} \mid (q, \mathbf{\Phi}) \in$ OLD $\}$. **Proof:** Let the *index* a pair $(q, \mathbf{\Phi}) \in Q\times L$ be the smallest i such that $q = \delta(q_0, \mathbf{A}_{\alpha_i}\mathbf{A}_{\alpha_{i-1}}\cdots\mathbf{A}_{\alpha_1})$ and $\Phi = \prod_{k=i}^{1}\mathbf{A}_{\alpha_k}$; if such an i does not exist the index of a pair is undefined. Let $i_{\max} = \max\{\text{index}(q, \mathbf{\Phi}) \mid (q, \mathbf{\Phi}) \in Q\times L\}$. Clearly L is finite $\iff$ the set $Q\times L$ is finite $\iff i_{\max} < \infty$. (If) Each time the **while** instruction is executed, one new element is added to the set OLD $\subseteq Q \times L$, hence if this set if finite the **while** loop can only be repeated a finite number of times and the algorithm terminates. (Only if) Assume that the algorithm terminates. We will prove by induction that for all $i > 0$, all pairs $(q, \mathbf{\Phi})$ of index i are contained in OLD, hence $i_{\max} < \infty$.

- *Base step* When the algorithm terminates, all pairs of index 1 are contained in OLD. In

fact all these pairs are added to NEW the first time the **while** loop is executed and will all be added to OLD before termination.

- *Induction* Assume that when the algorithm terminates all pairs of index i are contained in OLD. Clearly, for any pair $(q, \mathbf{\Phi})$ of index $i+1$ there exist a pair $(q', \mathbf{\Phi}')$ of index i such that $q = \delta(q', \mathbf{A})$, $\mathbf{\Phi} = \mathbf{A} \cdot \mathbf{\Phi}'$. During the same **while** loop in which $(q', \mathbf{\Phi}')$ is added to OLD, $(q, \mathbf{\Phi})$ is added to NEW and will all be added to OLD before termination.

This also proves that when the algorithm terminates $L = \{\mathbf{\Phi} \mid (q, \mathbf{\Phi}) \in \text{ OLD }\}$. ■

6 Conclusion

The main contribution of our work has been to represent the dynamic behavior of an automated manufacturing system with a discrete-time linear state variable model that can be used to evaluate performance measures and make gradient estimation [1]. The proposed formulation leads to a hybrid model that allows a neat representation of the process in terms of discrete linear inclusions. In particular we have presented an intriguing relationship that have emerged between idempotent matrices and the stability of DLI. Our main future goal will be to explore potential application fields of this hybrid model for the optimal design and control of discrete event processes.

References

[1] F. Balduzzi, G. Menga, "A State Variable Model for the Fluid Approximation of Flexible Manufacturing Systems" *IEEE International Conference on Robotics and Automation* (Leuven, Belgium), May 1998, to appear.

[2] V.D. Blondel, J.N. Tsitsiklis, "The Lyapunov exponent and joint spectral radius of pairs of matrices are hard - when not impossible - to compute and to approximate", *IEEE Conference on Decision and Control*, (S.Diego, USA), December 1997.

[3] H. Chen, A. Mandelbaum, "Discrete Flow Networks: Bottleneck Analysis and Fluid Approximations" *Math. Opns. Res.*, Vol. 16, pp. 408–446, 1991.

[4] I. Daubechies, J.C. Lagarias, "Sets of Matrices All Infinite Products of Which Converge", *Linear Algebra and its Applications*, No. 161, pp. 227–263, 1992.

[5] B.R. Fu, R. Suri, "On Using Continuous Flow Lines to Model Discrete Production Lines", *Discrete Event Dynamic Systems*, No. 4, pp. 129–169, 1994.

[6] J. Gunawardena, *Idempotency*, Publications of the Isaac Newton Institute, Cambridge University Press, 1997.

[7] L. Gurvits, "Stability of Discrete Linear Inclusion", *Linear Algebra and its Applications*, No. 161, pp. 47–85, 1995.

[8] Y.C. Ho, "Performance Evaluation and Perturbation Analysis of Discrete Event Dynamic Systems", *IEEE Trans. Automatic Control*, Vol. 32, No. 7, pp. 563–572, 1987.

[9] L. Kleinrock, *Queueing System Volume II: Computer Applications*, John Wiley and Sons, 1976.

[10] V.S. Kouikoglou, Y.A. Phillis, "Discrete Event Modeling and Optimization of Unreliable Production Lines with Random Rates", *IEEE Trans. Robotics and Automation*, Vol. 10, No. 2, pp. 153–159, 1994.