A State Equations Approach to Discrete-Event Dynamic System Representation

F. DONATI(1), G. CAPORALETTI(2)

(1)Dipartimento di Automatica e Informatica, Politecnico di Torino,
Corso Duca degli Abruzzi 24, I 10129 Torino
ITALY
(2)Eicas Automazione S.p.A,
Via Vincenzo Vela 27, I 10128 Torino
ITALY

Abstract: - The paper is in the line of the theoretical approach to discrete-events dynamic systems modelling developed in the frame of the *manufacturing algebra* [1-3]. A discrete-event dynamic system representation by state equation consistent with the Kalman classical definition of dynamic systems is introduced. The time is assumed to be a real variable, the system inputs and outputs are asynchronous event sequences, that is sequences of events which occur at a countable time instant set not a priori defined. The system state is described by state variables which are functions of the time real variable. Then, the state is defined at any time instant, while inputs and outputs are events defined over input and output countable time sets.

Hybrid systems are introduced, where inputs and outputs are both event sequences and time functions.

The proposed approach results appropriate for modelling at any level of detail the discrete manufacturing systems, allowing to give a description of the production processes in terms of state equatiions.

Key-Words:-Discrete Event System, Hybrid System, Dynamic System, State Equations, Manufacturing Algebra *IMACS/IEEE CSCC'99 Proceedings*, Pages:2121−2126

1 Introduction

A new approach to discrete-events dynamic systems has been recently developed in the frame of the Esprit project HIMAC devoted to discrete manufacturing plant modelling and control as part of the *manufacturing algebra* [1-3].

Here a synthesis of the previous results is presented and the approach is extended with some refinements to the hybrid systems (continuous-time and discrete-event dynamic systems).

The starting point of the approach followed is the dynamic system concept, originally developed by Kalman [5], based on the input, state and output variables and their dynamic relations. According to the above classic concept inputs $\boldsymbol{u}(t)$, states $\boldsymbol{x}(t)$ and outputs $\boldsymbol{y}(t)$ are multidimensional real variables which are functions of the time t, which can be real variable (continuous-time systems) or integer variable (discrete-time systems). A dynamic (strictly causal) system is then described by the following state equations:

Continuous-time system:

$$\dot{\mathbf{x}}(t) = F(\mathbf{x}(t), \mathbf{u}(t), t)$$
$$\mathbf{y}(t) = G(\mathbf{x}(t), t)$$

Discrete-time system:

$$\mathbf{x}(i+1) = F(\mathbf{x}(i), \mathbf{u}(i), i)$$

$$\boldsymbol{y}(i) \quad = G(\boldsymbol{x}(i), i)$$

The above definition of dynamic system can not be immediately extended to discrete-event dynamic systems, but for the particular case in which the word "event" indicates the presence of a pulse sequence in the admissible input functions, considered as real functions of the real time variable. This fact justifies the wide literature concerning discrete-event systems, with sometimes very different definitions and approaches.

In spite of such a variety of methodologies going from the very common Petri-Nets and automata to the perturbation methods or to the max-plus algebra, we have found that the already existing methodologies are not well suited to model those physical systems (the manufacturing discrete systems) for which we had to develop a general control methodology [4]. So we have formulated the new approach presented in this paper.

In order to explain what we mean by "event" and by "discrete-event dynamic system", let us consider the following simple example, which is very similar to some production units used in discrete-manufacturing plants: a drink slot-machine equipped with memory. The machine is commanded by means of a push-button panel, where each button corresponds to a drink: coffee, tea, coca cola etc. Whenever a button is pushed (input event to the system or cause event) the machine dispenses after a certain working time the requested drink (output event from the system or effect event). If another button is pushed when the machine is still working, the machine accepts the new order and puts it into a waiting-list served later according to some criterion (e.g. FIFO). When a product has run out, the next one in the list is dispensed until the waiting-list is cleared out.

The machine's input is an event sequence (independent and unpredictable), each event being described by a *fact* (drink requested) and by a *time* (time instant at which the event occurs). The output system is also an event sequence, each event being described by a *fact* (drink dispensed) and by a *time* (time instant at which the event occurs). The input and output events occur in a countable but *not a priori* defined set of instants. Therefore the system can not be treated as a discrete-time system: in effect the time set is not countable but continuous, because the input and output events can occur at any instant not a priori defined. The system state variables should be continuous-time function, while inputs and outputs are discrete-event sequences. Such a system is denoted here *discrete-event dynamic system.*

2 Time, Events And Event Sequences

The time set. Let us introduce the real variable t named "*time"*. The generic finite or infinite time interval $\mathcal{T} = [t_1, t_2]$ is named *time set*.

The fact set. Let Ξ be a set of elements $\xi \in \Xi$ called *facts*.

The event set. Given a time set $\mathcal{T}$ and a fact set Ξ, the following Cartesian product is defined $\mathcal{E} = \mathcal{T} \times \Xi$. The elements $e = (t, \xi) \in \mathcal{E}$ are called *events.* Any event is therefore described by the pair t = *"the occurrence time instant of the event"*, ξ = *"the fact associated with the event"*.

Event sequences or event lists. An event sequence (or event list) σ is a countable (finite or infinite) set of events $\{e(i)=(t(i),\xi(i))\}\ i \in [1,m]$ belonging to the same event set $\mathcal{E}$, which is put in correspondence one-to-one with a segment of natural numbers $[1\text{-}m]\ m \in \mathcal{N}$, so to result strictly ordered with respect to the variable time, i.e. with the constraint $t(i) > t(j)$ for $i > j$.

In the following an event sequence (or event list) will be denoted by σ, when the sequence is considered globally, that is *:*

$$\sigma = \{e(i)=(t(i),\xi(i))\}\ i \in [1,m]. \tag{1}$$

The *i-th* element of the sequence (or list) is denoted $e(i)$ or $(t(i),\xi(i))$, being $e(i)=(t(i),\xi(i))$. Of course, when i is varying over the sequence (or list) definition segment [1-m], the single event $e(i)$ may be used also to denote the global sequence (or lists) to which it belongs.

In the following the term *event sequence* is always used when $t(i) \in \mathcal{T}$ is the current time. On the contrary when the above real variable *t(i)* is not the current time (but for instance a date) and the above event set is taken as a whole and considered as function of the current time, the term *event list* is normally used. Of course, event sequences or lists are the same mathematical object with the same properties.

Event unicity. By assumption, it is excluded that two different events can occur at the same time (simultaneous events). But it is accepted that more than one fact may be associated with the same event.

When a plurality of facts can be associated with the same event, the above plurality is defined to be a fact belonging to the set Ξ . Then:

- an event sequence can not have simultaneous events;
- the facts associated with an event, belonging to an event sequence, can always be described by a single element of the fact set Ξ.

Properties of the fact set. In the following two different kinds of event sets are considered, which are equipped with different properties, as follows.

- The fact set Ξ is a *linear space* over $\mathcal{R}$ (set of real numbers), i.e. two operations, the *addition* and the *scalar multiplication*, are defined over the set Ξ, with their well known properties.
  The addition of facts defines the fact equivalent what means that, whenever different facts are associated with a same event, the equivalent fact is given by their addition.
  Example. In manufacturing algebra events are considered, where the facts are object drawing/delivery operations. The fact set is then the quantity vector linear space $\mathcal{Q}$ [1].
- The fact set Ξ is a *finite set* which does not have other specific properties. Different facts are not mutually compatible, that is they can not be associated with the same event.
  Example. In manufacturing algebra [10] events are considered, where the facts are commands applied to a production unit in order to select the specific manufacturing operation to be performed. The fact set is the finite set of the manufacturing operation, which can be performed by the production unit considered. Only one manufacturing operation at a time can be performed and then commanded whenever the production unit is memory-less.

The event sequence (or list) set. The set of all countable (finite and infinite) sequences_(or lists), which can be constructed over the event set $\mathcal{E}=\mathcal{T}\times\Xi$, will be denoted by $\Sigma(\mathcal{E})$.

Over the set $\Sigma(\mathcal{E})$ of the event sequences, the following operations are defined.

Restriction of an event sequence. The restriction σ' of an event sequence σ to a time interval $t(i)>t$ is a sub-sequence $\sigma'\in\Sigma(\mathcal{E})$ of σ which includes only the events *e(i)* occurring at times $t(i)>t$.

The restriction operation will be denoted by $\sigma'=\sigma(t(i)>t)$.

Addition of event sequences. Two cases are considered:

1) The fact set Ξ is a linear space. Given two sequences $\sigma_1,\sigma_2\in\Sigma(\mathcal{E})$, let us consider the union set $\sigma_1\cup\sigma_2$ of all the events of the two sequences. Simultaneous events are assumed to be the same event and their facts are added up. The resulting set is ordered with respect to the time variable t , giving the addition $\sigma_3=\sigma_1+\sigma_2$.
2) The fact set Ξ is a generic finite set. Two sequences $\sigma_1,\sigma_2\in\Sigma(\mathcal{E})$ are said to be *summable* if and only if one has not events simultaneous to the events of the other one. When two event sequences are summable, their addition is obtained as described above.

The addition of event sequences defined in $\Sigma(\mathcal{T}\text{x}\Xi)$ has the commutative and associative properties.

Event sequence linear space. If the fact set Ξ is a linear space, the corresponding event sequence set $\Sigma(\mathcal{T}\text{x}\Xi)$ is a linear space where the addition is defined as above and the scalar multiplication and the addition properties are induced by the properties of the linear space Ξ.

3 Discrete-Event Dynamic Systems

A *discrete-event dynamic system* is a system whose inputs and outputs are event sequences. In other words it is an operator which maps a set of admissible input event sequences into a set of output event sequences. In the following the class is considered of the *discrete-event dynamic (causal) systems* described by state equations.

With reference to the basic definitions introduced in the previous paragraph, the following notations are used.

Let $\mathcal{T}$ be the time set and t be the current time.

Let us denote by $\sigma_u=\{e_u(j)\}$ the input event sequence, being

- $\boldsymbol{U}$ the set of facts u,
- $\mathcal{E}_{\mathbf{u}}=\mathcal{T}\mathrm{x}\boldsymbol{U}$ the set of the input events $e_u(j)=(t_u(j),u(j))$,
- $\Omega\subset\Sigma(\mathcal{E}_{\boldsymbol{u}})$ the set of the admissible input sequences.

The concatenation property holds in the set Ω of the admissible input sequences.

Let us denote by $\sigma_y=\{e_y(j)\}$ the output event sequence, being

- $\boldsymbol{Y}$ the set of facts y,
- $\mathcal{E}_{\boldsymbol{y}}=\mathcal{T}\mathrm{x}\boldsymbol{Y}$ the set of the output events $e_y(i)=(t_y(i),y(i))$,
- $\Sigma(\mathcal{E}_{\boldsymbol{y}})$ the set of the output sequences.

Input and output events are not synchronous and generally they do not have the same periodicity. To describe the input-output relation two different sets of system state variables are introduced, as follows.

- The real variable $t_c\in\mathcal{T}$ denoting the system clock time.
- The finite event list $\sigma_x(t)=\{e_x(i)=(t_x(i),\xi(i))\}$ $i\in[1,m]$, where $t_x(i)\geq t_c$

The time evolution of the real variable t_c is not depending on the input event sequences, being described by the differential equation:

$$\dot{t}_c(t) = 1 \tag{2}$$

The finite event list $\sigma_x(t)$ is subject both to a free evolution not depending on the system inputs and to a forced evolution caused by the input event occurrence. The free evolution occurs when the system clock time t_c is equal to the occurrence time $t_x(1)$ of the first event of the state event list $\sigma_x(t)$, that is when the following logic expression is *true:*

$$t_c(t) = t_x(1) \tag{3}$$

The state event list evolution is then described by the relation:

$$\sigma_x(t_+) = F_\sigma(\sigma_x(t),t) \tag{4}$$

The state event list forced evolution caused by the input event $e_u(j)=(t_u(j),u(j))$ occurs at the input event time $t=t_u(j)$ and is described by the relation:

$$\sigma_x(t_+) = F_u(\sigma_x(t),e_u(j)) \tag{5}$$

Considering a strictly causal system, output events are related only to the free evolution of the state event list $\sigma_x(t)$. Then, when the logic expression (3) is true, the *i-th* output event $e_y(i)=(t_y(i),y(i))$ occurs described by the relation:

$$e_y(i)=(t_x(1),g_y(\xi(1),t)) \tag{6}$$

where, of course, $e_x(1)=(t_x(1),\xi(1))$ is the first event of the list $\sigma_x(t)$ at the time t at which the logic expression $t_c(t) = t_x(1)$ is true, that is $e_x(1)$ is the occurring event of the state event list σ. The output event sequence is then synchronous with the occurrence of the events of the state event list σ_x.

4 Two Simple Examples

In order to clarify the above stated concept of discrete-event dynamic system, two simple examples are now given. The first one corresponds to the mathematical model of the drink slot machine already described in the introduction, the second one is the mathematical model of an unstable *flip-flop.*

4.1 The Drink Slot-Machine Model

Let Ξ be the set of the available drinks and let us denote by $u\in\Xi$ an ordered drink, by $\xi\in\Xi$ a scheduled drink and by $y\in\Xi$ a produced drink. Then it results:

- $e_u(j)=(t_u(j),u(j))$ is the input event sequence, being $t_u(j)$ the time at which the *j-th* drink $u(j)$ is ordered;
- $\sigma_x(t)=\{e_x(k)=(t_x(k),\xi(k))\}$ $k\in[1,m]$, is the state

event list at the time t corresponding to the ordered drinks $\xi(k)$ not yet produced with their production scheduled time $t_x(k)$;

- $e_y(i)=(t_y(i),y(i))$ is the output event sequence, being $t_y(i)$ the time at which the *i-th* drink $y(i)$ is produced.

Each time an input event $e_u(j)=(t_u(j),u(j))$ occurs, the scheduled time $t_{us}(j)$ for the ordered drink production is computed and the drink production event is scheduled by adding the event $(t_x(m)=t_{us}(j)\ ,\ \xi(m)=u(j))$ in the last position (*m-th*) of the state event list. The above state event list updating represents the forced state evolution generally described by Eq. (5).

In order to describe the system free evolution the system clock time t_c must be introduced, being t_c the time respect to which the production scheduling is performed. In an ideal condition the machine time t_c increases as the *true time t* and it results:

$$\dot{t}_c(t) = 1$$

but in a more general condition it may be:

$$\dot{t}_c(t) = 1 + \varepsilon(t)$$

where $\varepsilon(t)$ models a perturbation in the machine production time. When such a perturbation is introduced either as an input system stochastic disturbance or by a continuous-time dynamic model, an hybrid (discrete-event and continuous-time) system is obtained as presented in the next section.

When the logic expression (3) is true, that is when the system clock time $t_c(t)$ is equal to the occurrence time $t_x(1)$ of the first event of the state event list $\sigma_x(t)$, the scheduled drink $\xi(1)$ is produced giving a new output event, the *i-th* event of the output event sequence:

$$e_y(i)=(t_y(i)=t,\ y(i)=\xi(1))$$

At the same time t for which it results $t_c(t)=t_x(1)$ the state event list $\sigma_x(t)$ is updated: the first event of the list is removed and the list is reordered by forward shifting the list events.

4.2 The Unstable Flip-Flop

The input event facts belong to the following finite set $\boldsymbol{U}=\{start,\ stop,\ set\ flip,\ set\ flop\}$. Then the input event sequence is given by $e_u(j)=(t_u(j),u(j))$, where $u(j)\in \boldsymbol{U}$ and $t_u(j)$ is any ordered time instant sequence.

The output event facts belong to the finite set $\boldsymbol{Y}=\{flip,flop\}$.

The state event list is always composed by only one event $\sigma_x=\{(t_x,\xi)\}$, being $\xi\in \boldsymbol{Y}$.

The system clock time t_c increases as the *true time t*, being: $\dot{t}_c(t) = 1$. At the time t for which the logic expression $t_c(t)=t_x$ is true the output event $e_y=(t,y=\xi)$ is produced and the state event list is updated according to the following transition table:

State event $(t_x(t),\xi(t))$ at time t for which $t_c(t)=t_x$	State event $(t_x(t_+),\xi(t_+))$ at time t_+
$(t_x(t)=t_c\ ,\ \xi(t)=flip)$	$(t_x(t_+)=t_c+T\ ,\ \xi(t_+)=flop)$
$(t_x(t)=t_c\ ,\ \xi(t)=flop)$	$(t_x(t_+)=t_c+T\ ,\ \xi(t_+)=flip)$

When an input event occurs, the state event is updated according to the following transition table:

Input event (t_u,u)	State event $(t_x(t_{u+}),\xi(t_{u+}))$ at time t_+
$(t_u\ ,\ start\ )$	$(t_x(t_{u+})=t_{u+}\ ,\ \xi(t_{u+}) = \xi(t_u))$
$(t_u\ ,\ stop\ )$	$(t_x(t_{u+})=\infty\ ,\ \xi(t_{u+}) = \xi(t_u))$
$(t_u\ ,\ set\ flip\ )$	$(t_x(t_{u+})=t_{u+}\ ,\ \xi(t_{u+}) = flip)$
$(t_u\ ,\ set\ flop\ )$	$(t_x(t_{u+})=t_{u+}\ ,\ \xi(t_{u+}) = flop)$

5 Hybrid Systems

Hybrid systems are composed by a discrete-event dynamic sub-system and a continuous-time dynamic sub-system mutually interacting.

The continuous-time sub-system is described by the usual input $u(t)$, state $x(t)$, output $y(t)$ variables among which the classic relations hold:

$$\dot{\boldsymbol{x}}(t) = F(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \qquad (7)$$

$$\mathbf{y}(t) = G(\mathbf{x}(t), t) \qquad (8)$$

The discrete-event sub-system is described by the input event sequence $e_u(j)=(t_u(j),u(j))$, the output event sequence $e_y(i)=(t_y(i),y(i))$ and the state event list $\sigma_x(t)=\{e_x(k)=(t_x(k),\xi(k))\}\ k\in[1,m]$.

The real state variable t_c, which represents the clock time in discrete-event dynamic systems, is substituted in hybrid systems by the more general continuous-time dynamic sub-system described by Eq.s (7,8). Similarly, the logic expression (3) is now substituted by a more general one:

$$L(x(t), e_x(k), t) = true \qquad (9)$$

The above logic expression involves the state $x(t)$ of the continuous-time subsystem and the events $e_x(k)=(t_x(k), \xi(k))$ of the state event list $\sigma_x(t)$. When, at time t, the logic expression (9) is *true*, then the state event $(t, \xi(k))$ occurs, causing the free evolution of the hybrid system as follows:

- The state event list is updated:
  $\sigma_x(t_+) = F_\sigma(\sigma_x(t), e_x(k), t)$ (10)
- The continuous-time state $x(t)$ is updated:
  $x(t_+) = F_x(x(t), e_x(k), t)$ (11)
- The new *i-th* output event is produced:
  $e_y(i)=(t_y(i)=t, y(i)=g_\sigma(\mathbf{x}(t), e_x(k), t))$ (12)

Similarly the occurrence of an input event $e_u(j)$ causes an updating of all the system state variables $x(t)$ and $\sigma_x(t)$, as follows:

- The state event list is updated:
  $\sigma_x(t_+) = F_{u\sigma}(\sigma_x(t), e_u(j), t)$ (13)
- The continuous-time state $x(t)$ is updated:
  $x(t_+) = F_{ux}(x(t), e_u(j), t)$ (14)

In conclusion hybrid systems are a complex mathematical concept defined by the set of equations from (7) to (14) together with the definition of the input variables (continuous-time $u(t)$ and event sequence $e_u(j)$), of the output variables (continuous-time $y(t)$ and event sequence $e_y(i)$) and of the state variables (continuous-time $x(t)$ and event list $\sigma_x(t)$).

6 Conclusions

A new theoretical approach to modelling both discrete-event dynamic systems and hybrid systems has been presented, which is consistent with the classical definition of dynamic systems. Such a new approach has been applied and appears well suited to model discrete manufacturing processes with the aim of real time control design.

7 Acknowledgments

The concepts and results of the discrete-event dynamic system theory presented in this paper are part of the *Manufacturing Algebra* developed within the ESPRIT Basic Research Project 8141-HIMAC, coordinated by EICAS Automazione S.p.A.

8 References

[1] Donati F., Canuto E., Vallauri M.: *Manufacturing Algebra. An overview* Proceed. 2nd HIMAC Workshop: A new math. approach to manufacturing eng., (M. Vallauri ed.), CELID, Torino 1997.

[2] Donati F., Caporaletti G.: *Discrete-event dynamic system state equations.* 2nd HIMAC Workshop: A new math. approach to manufacturing eng., (M. Vallauri ed.), CELID, Torino 1997.

[3] Donati F., Canuto E., Vallauri M.: *Advances in Manufacturing Algebra: discrete-event dynamic models of production processes.*
Preprints Ist IFAC Workshop Man. Systems: MIM’97, p. 461-467, Vienna 1997.

[4] Donati F., Canuto E., Vallauri M.: *A new approach to Discrete-Event Dynamic System Theory.*
2nd Workshop on Trends in Theoretical Informatics, Budapest March 1997.

[5] Kalman R. L., Falb. P.L., Arbib M.A.: Topics in Mathematical System Theory. McGraw-Hill, New York 1969.