synthesize whatever global structure is needed as a sum of their reduced local counterparts.

REFERENCES


The Predictability of Discrete Event Systems

XI-REN CAO

Abstract—Perturbation analysis and the automaton and language model are two recently developed approaches for discrete event systems (DES’s). The prediction of a trajectory of a new system is the essential idea of perturbation analysis. The automaton theory models a trajectory of a DES by a string in a particular language. In this note, we formulate the trajectory prediction as a projection of a string to a language. A sufficient condition is found for one language to be predictable from another language. Examples are given to show the application of this concept.

I. INTRODUCTION

Recently, discrete event systems have been increasingly attracting interest. Different approaches have been developed. Among them are the queueing network model [5], the generalized semi-Markov process approach and discrete event simulation method, Petri nets [8], the min-max algebraic method [1], automaton and language theory [9], and perturbation analysis [3].

The last two approaches, i.e., the method based on automaton and language theory and perturbation analysis, are particularly relevant to this

Manuscript received April 12, 1988; revised June 30, 1988 and September 27, 1988.

The author is with Digital Equipment Corporation, Marlboro, MA 01752.

IEEE Log Number 8930295.
Thus, the extended transition function is defined on set $\Sigma$. To distinguish the original domain $\Sigma(q)$ and the extended domain $\Sigma$, we say that $\delta(q, a)$ is properly defined if $a \in \Sigma(q)$. Note that $\delta(q, a) = q$ may hold for some $a \in \Sigma(q)$. Furthermore, for any $s \in \Sigma^*$, we say that $\delta(s, q_0)$ is properly defined if it is defined for any suffix $u$, prefix $v$, and event $a$ satisfying $s = uav$, $\delta(u, a)$ is properly defined and $a \in \Sigma(q')$, $q' = \delta(s, q_0)$. We use $\delta(s, q_0)$ to denote the fact that $\delta(q_0, s)$ is properly defined. The language generated by $G$ is defined as

$$L(G) = \{ s \in \Sigma^* | \delta(s, q_0) \}.$$

By this definition, $L(G)$ is always closed. Note that because of the extension of the transition function $\delta(s, q_0)$, for any $s \in \Sigma^*$, $\delta(s, q_0)$ has a value. However, $\delta(s, q_0)$ may not be properly defined since there may exist some $u, a, v$, satisfying $s = uav$, for which $a \notin \Sigma(q'_0)$, $q'_0 = \delta(s, q_0)$.

If $X$ is a set and $\theta$ is an equivalence relation on $X$, then the quotient is denoted by $X/\theta$ and the quotient set is denoted by $X/\theta$. The fact that the two elements $x_1$ and $x_2$ are equivalent is denoted by $x_1 \equiv x_2$.

### III. Predictability of DES’s

Before we introduce the concept of predictability of DES’s, we shall first define some new notions. Let $\Pi \subseteq \Sigma^*$ be a closed language.

**Definition 1:** A projection from $\Sigma^*$ onto $\Pi$ is defined by $P_{\Pi}(1) = 1$

$$P_{\Pi}(s) = \begin{cases} a & \text{if } a \in \Pi, \\ 1 & \text{if } a \notin \Pi \end{cases}$$

and for any $s \in \Sigma^*$ and $a \in \Sigma$

$$P_{\Pi}(as) = \begin{cases} P_{\Pi}(s) & \text{if } P_{\Pi}(s) \in \Pi, \\ P_{\Pi}(t) & \text{if } P_{\Pi}(s) \notin \Pi. \end{cases}$$

Roughly speaking, the projection of $s$ on $\Pi$ is a string in $\Pi$ which can be constructed from $s$ by sequentially removing the symbols which make the prefixes of $s$ outside of $\Pi$. For instance, let $\Pi = \{ 1, \alpha, \alpha \alpha, \alpha \alpha \beta, \alpha \gamma, \alpha \gamma \alpha, \alpha \gamma \alpha \beta \}$ and $s = \alpha \beta \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \alpha \gamma \alpha \beta \gamma \alpha \beta \}$

Thus, the extended transition function is defined on set $\Sigma$. To distinguish the original domain $\Sigma(q)$ and the extended domain $\Sigma$, we say that $\delta(q, a)$ is properly defined if $a \in \Sigma(q)$. Note that $\delta(q, a) = q$ may hold for some $a \in \Sigma(q)$. Furthermore, for any $s \in \Sigma^*$, we say that $\delta(s, q_0)$ is properly defined if it is defined for any suffix $u$, prefix $v$, and event $a$ satisfying $s = uav$, $\delta(u, a)$ is properly defined and $a \in \Sigma(q')$, $q' = \delta(s, q_0)$. We use $\delta(s, q_0)$ to denote the fact that $\delta(q_0, s)$ is properly defined. The language generated by $G$ is defined as

$$L(G) = \{ s \in \Sigma^* \mid \delta(s, q_0) \}.$$
any prefix $u$ of $s \in L(G)$, we can check whether the transition function $\chi(u, r)$ is properly defined and then determine the string $t \in L(H)$. In other words, $t$ can be constructed from $s$, or a trajectory of the discrete event system $H$ can be obtained by analyzing a trajectory of the discrete event system $G$. This is exactly the essential idea of perturbation analysis.

**Definition 3:** The disjunction of two DES’s $G$ and $H$ is defined by

$$HVG - (Q \times R, \delta \cdot V_X, (q_0, r_0))$$

where $Q \times R$ is the set of states of the disjunction DES, $(q_0, r_0)$ is the initial state of the DES, and $\delta \cdot V_X$ is the transition function of HVG, which is the disjunction of the two transition functions $\delta$ and $\chi$ defined as follows:

$$\delta \cdot V_X(s, (q, r)) =
\begin{cases}
    (\delta_1(s), q, \chi_1(s, r)) & \text{if } s \in \Sigma_1 \land \chi_1 \in \Sigma_1(r), \\
    (\delta_1(s), q, \chi_1(s, r)) & \text{if } s \in \Sigma_2 \land \chi_1 \in \Sigma_2(r), \\
    \text{undefined} & \text{if } s \in \Sigma_3 \land \chi_1 \in \Sigma_3(r).
\end{cases}$$

By the extension of the transition functions $\delta$ and $\chi$, this equation can be rewritten as

$$\delta \cdot V_X(s, (q, r)) =
\begin{cases}
    (\delta_1(s), q, \chi_1(s, r)) & \text{if } s \in \Sigma_1 \land \chi_1 \in \Sigma_1(r), \\
    (\delta_1(s), q, \chi_1(s, r)) & \text{if } s \in \Sigma_2 \land \chi_1 \in \Sigma_2(r), \\
    \text{undefined} & \text{if } s \in \Sigma_3 \land \chi_1 \in \Sigma_3(r).
\end{cases}$$

The domain of $(\delta \cdot V_X)$ is $s \in (Q_0 \cup \Sigma_1(r)$. As mentioned above, we extend the definition of $(\delta \cdot V_X)$ by defining $(\delta \cdot V_X)(s, (q, r)) = (q, r)$ for $s \in \Sigma_3 \land \chi_1 \in \Sigma_3(r)$. The definition implies that $(\delta \cdot V_X)$ is properly defined if and only if either $\delta$ or $\chi$ is properly defined. $(\delta \cdot V_X)$ can also be defined for any string $s \in \Sigma^*$. The language generated by HVG, $L(HVG)$, is called the disjunction of the two languages generated by $H$ and $G$, i.e., $L(H) \cup L(G)$.

If $s \in L(G)$, then $(\delta, s)$ is properly defined. By the definition of the disjunction of two transition functions, this implies that $(\delta \cdot V_X)(s, (q, r))$ is properly defined. Therefore, $s \in L(HVG)$.

Similarly,

$$L(H) \subseteq L(HVG).$$

Now we shall study the property of the projection. Consider the two projections $P_{L(H)}$ and $P_{L(G)}$. Projection $P_{L(H)}$ specifies an equivalence relation $\theta_H$ defined by $s \equiv s'$, $s', s' \in \Sigma^*$, if and only if $P_{L(H)}(s) = P_{L(H)}(s')$. Similarly, projection $P_{L(G)}$ defines another equivalence relation $\theta_G$. The quotient sets of $L(HVG)$ and these two equivalence relations are $L(HVG)/\theta_H$ and $L(HVG)/\theta_G$. For any $s \in L(H)$, $\theta_H(s)$ is the equivalence class of $s$ under $\theta_H$. $\theta_G(s)$ is the coset whose elements have the same projection $s$ under $\theta_G$. $\theta_H(s)$ has a similar meaning. We have the following lemma.

**Lemma 1:** For any $t \in L(H)$, $t \neq t'$, there always exists an $s \in L(G)$, $s \neq s'$, and $t' \in L(H)$ such that $\theta_H(t') \cap \theta_G(s) = \emptyset$, and $t$ is a prefix of $t'$.

**Proof:** If $P_{L(G)}(t) \neq 1$, let $s = P_{L(G)}(t)$. Then $t \in \theta_G(s)$. By definition, $t' \in \theta_H(t)$. Thus, $t' \in \theta_H(t) \cap \theta_G(s)$, i.e., the lemma holds in this case. If $P_{L(G)}(t) = 1$, then all events contained in $t$ are not permitted for $q_0$ in $G$. This means $q_0 = \delta(t, e)$. In this case, we choose any $u \in L(G), u \neq 1$. Let $t' = P_{L(H)}(u)$. Then $u \in \theta_H(t') \cap \theta_G(s)$, i.e., $t$ is a prefix of $t'$. $t'$ is a prefix of $t'$.

Using this lemma, we can prove the following theorem.

**Theorem 1:** If $P_{L(H)} = P_{L(G)}$, then $L(L(H)) \rightarrow L(H)$.

**Proof:** Let $s \in L(G)$. From Lemma 1, there is an $s \in L(G)$ and a $t' \in L(H)$ such that $\theta_H(t') \cap \theta_G(s) = \emptyset$, and $t'$ is a prefix of $t'$. Let $u \in \theta_H(t') \cap \theta_G(s)$. Then $P_{L(G)}(u) = s$, and $P_{L(H)}(u) = t'$. Therefore, we obtain $P_{L(H)}(u) = P_{L(G)}(u) = P_{L(H)}(s)$. Since $t'$ is a prefix of $t'$, there is a word $s' \in \Sigma^*$ and $s' \in L(G)$ such that $s = s'$. By the definition of projection, we can choose $s'$ such that $t = P_{L(H)}(s')$. The proof is thus completed.
because $L(H) \subseteq L(G)$. This relation can be easily checked by comparing the directed graphs of the two automata. Here is a formal proof. From the definitions of the transition functions $\delta$ and $\chi$, we can verify that, for any $u \in \Sigma^*$, if both $\delta(u_1)$ and $\chi(u_1)$ are properly defined, then $\delta(u_1) = \chi(u_1)$. Now let $s = uav \in L(H)$, where $u$ and $v$ are two substrings and $a = \alpha$ or $\beta$ is an event. Suppose that $u \in L(G)$. Then $q(u) = \delta(u_1) = r(u) = \chi(u_1)$. Note that the only case in which $\delta(s, q)$ is not properly defined is $q(s) = 0$ and $e = \beta$. Thus, $\delta(s, q(u))$ is always properly defined since $q(u) = r(u)$ and $\chi(s, r(u))$ is properly defined. Therefore, $\delta(u_1)$ is properly defined and $u \in L(G)$. By induction, $s \in L(G)$.

In the second example, we shall consider a simple queueing system consisting of one server and two classes of customers. The customers feed back to the queue immediately after leaving the server. Let $G$ be such a system with one class 1 customer and two class 2 customers, and let $H$ be a similar system but with one class 1 customer and one class 2 customer. The set of events is $\Sigma = \{\alpha, \beta\}$, where $\alpha$ denotes the departure of a class 1 customer and $\beta$ denotes a departure of a class 2 customer. A string of $L(G)$ has the form $a\alpha\beta a\beta a\beta \cdots$, and a string of $L(H)$ has the form $a\alpha\beta a\beta a\beta \cdots$. Any string of $L(H)$ is a projection of a string of $L(G)$, e.g., $a\alpha\beta = P_1, \alpha(\alpha\beta a\beta)$. Thus, $L(G) \supseteq L(H)$. Note that $L(G) \cap L(H) = \emptyset$. However, Theorem 1 does not hold. For example, for $u = a\alpha\beta a\beta a\beta \in L(GH)$, $P_1, \alpha(\alpha\beta) = a\alpha\beta$, but $P_1, \alpha(\alpha\beta a\beta a\beta a\beta) = a\alpha\beta$. This shows that the condition in Theorem 1 is not a necessary one.

V. DISCUSSION

The trajectories in the two examples can be considered as those of the embedded Markov chains of the queueing systems. Thus, the examples show that, by analyzing a trajectory of the embedded Markov chain of a queueing system, we can obtain a trajectory of the embedded Markov chain of another queueing system. However, the current automaton model of DES does not consider the stochastic feature of a system. Thus, problems related to stochastic properties are left for further research. Such problems are important for performance studies. For instance, to obtain steady-state performance one needs an ergodic trajectory. The question arises as to whether or not a trajectory of a system $H$ predicted from a trajectory of another system $G$ is ergodic. This kind of problem shows that it is necessary to introduce the stochastic feature into the automaton model.

Recently, Cassandra and Strickland [2] studied the performance sensitivity estimation of discrete event systems modeled as Markov chains. They introduced an augmented chain and investigated the observability of the augmented chain. They also provided a method of constructing the augmented chain which is "stochastically similar" to the perturbed chain. It is interesting to note that the augmented chain corresponding to two Markov chains can be expressed in terms of the disjunction of two languages defined in this note. The observability is similar to the predictability of a language from another language. However, the notion of stochastic similarity does not have a counterpart in the automaton framework, since the latter does not consider the stochastic feature. A more detailed study of the connections between these two approaches is in progress.

Perturbation analysis and the automaton model are two recently developed theories for discrete event systems. This note shows that perturbation analysis can be described by the automaton theory. The predictability concept proposed in this note may have other applications in the area of discrete event systems. We hope that the results in this note will provide some new insight into the problems in this area.

ACKNOWLEDGMENT

The author wishes to thank Dr. F. Lin and Prof. Cassandra for their suggestions for and corrections to an earlier draft of this note, and he is indebted to the reviewers for their helpful comments.

REFERENCES


Output Residence Time Control

S. M. MEERKOV and T. RUNOLFSSON

Abstract—The problem of residence time control, introduced in [1], is extended to systems with outputs. Necessary and sufficient conditions for output residence time controllability in linear systems with small, additive noise are derived. State feedback controller design techniques are developed and applied to a robotics control problem. The approach is based on an extension of the asymptotic first passage time theory to output processes.

I. INTRODUCTION

Given a controlled dynamical system with states $x(t) \in \mathbb{R}^n$, control $u(t) \in \mathbb{R}^m$, output $y(t) \in \mathbb{R}^p$, and disturbances $\xi(t) \in \mathbb{R}^q$, assume its desired behavior is specified by a pair $\{Q, \Gamma\}$, where $\Gamma \subseteq \mathbb{R}^q$ is the domain to which the outputs $y(t)$ should be confined and $\Gamma$ is the period of the confinement, i.e., $y(t) \in \mathbb{Y}$, $\forall t \in \{t_0, t_1, t_2 + \tau\}$, $t_0 \in \mathbb{R}$. Problem formulation of this form arises in numerous applications. For instance, in the problem of telescope pointing, the domain $\mathbb{Y}$ is defined by the field of view and $\tau$ is the exposure time (see [1] for additional examples).

For a given pair $\{\mathbb{Y}, \tau\}$, the problem of output residence time control is formulated as the problem of choosing a feedback control law, so as to force $y(t)$ to remain, at least on average, in $\mathbb{Y}$ during period $\tau$, in spite of the disturbances $\xi(t)$ that are acting on the system.

The purpose of the present note is to analyze the fundamental capabilities and limitations of output residence time control for linear systems with small additive perturbations. The approach is based on an extension of the asymptotic first passage time theory to output processes.

The structure of the note is as follows: In Section II the notion of an output residence time is introduced; in Section III output residence time controllability is defined and analyzed; in Section IV output residence time controller techniques are given; and in Section V an example is considered. The proofs are given in the Appendix.

II. OUTPUT RESIDENCE TIME

Consider a linear stochastic system

$$\dot{x} = Ax + Bu + Cd\eta$$
$$y = Dx$$

(2.1)

Manuscript received January 12, 1988; revised December 22, 1988. Paper recommended by Associate Editor, A. Beurenstein. This work was supported by the Air Force Office of Scientific Research under Contract F49620-87-C-0079.

S. M. Meerkov is with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Ml 48109-2122.
T. Runolfsson was with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Ml 48109-2122. He is now with the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218.
IEEE Log Number 8902089.

0018-9225/89/1100-1171$1.00 © 1989 IEEE