Proceedings of the IEEE Conference on Decision and Control, December 1994, pp. 4098-4103, Lake Buena Vista, Florida

ON SOME CONTROL PROBLEMS IN MOLECULAR BIOLOGY

Jacob Hammer
Department of Electrical Engineering
University of Florida
Gainesville, FL 32611, USA

Abstract

The paper deals with the control of a sequential machine whose model is not precisely known. The objective is to design a controller that assigns to the machine a prescribed steady-state behavior. The results include necessary and sufficient conditions for the existence of a controller, as well as algorithms for its design. This work is motivated by potential applications in biology. It is an abridged version of HAMMER [1994].

1. Introduction

Many of the basic processes that govern the operation of (biological) cells have a natural sequential structure. Some examples are the Krebs cycle; the transcription of DNA into RNA; the translation of RNA into protein; and others. The modeling of various biological phenomena within the general framework of automata theory has been documented in the literature for quite some time (e.g., RASHEVSKY [1948], M. SUGITA [1963], von NEUMANN [1966], LINDENMAYER [1968], KAUFFMAN [1969], ROSENBERG and SALOMAA [1975], IEEE [1974], the references cited in these works, and many others).

The present paper deals with the development of methods for the control of sequential machines that are incompletely described. The basic motivation is to formulate controllers that correct impaired function of biological cells. Control techniques offer the prospect of providing new insight into the regulation of unacceptable behavior of cells, such as the unrestrained division associated with pre-cancerous or cancerous transformations, or other malfunctions of the genetic system. Sequential models of cells are empirically derived input/output models of cell function; they may include discrete-event approximations of continuous models, to facilitate simulation or control via digital computers.

The sequential machines we consider are defined so as to be suitable for modeling chemical or biochemical reaction systems within a medium. Here, each reactant molecule (or molecular complex) in the medium is regarded as a variable, called a *word*. The sequential machine implements the rules of chemistry that govern the reaction steps, changing the molecular population within the medium as the reaction evolves. Molecules that are externally injected into the medium are regarded as *input words*, whereas molecules (or complexes) present within the medium at the end of a reaction step are regarded as *output words*. The number of molecules within each category varies with reaction, step, and circumstance, and may, of course, be quite large. However, in many systems of interest in molecular biology, the number of significant kinds of molecules seems manageable. For instance, the functioning of an E-coli bacterium probably involves no more than a few thousand kinds of significant molecules; and the operation of a mammal cell probably involves no more than a few hundred thousand kinds of significant molecules (e.g., ALBERTS, BRAY, LEWIS, RAFF, ROBERTS, and WATSON [1994]).

An important consideration in molecular biology is the fact that detailed models of cell function are not available. The lack of exact models originates from the lack of complete data, as well as from the differences between individual specimens. Thus, it is critical to discuss the control of a sequential machine Σ whose model is only partly known. The information available about Σ is given in the form of a family M of potential models; the real model of Σ is one of the members of M.

An important concern is to reduce the amount of data that needs to be collected about the sequential machine Σ. This data can be divided into two broad categories: *a-priori data*, which determines the class M of potential models; and *real-time data*, which consists of data the controller requires during its operation for feedback. It is particularly important to reduce the real-time data requirements, so as to avoid complex measurements in real time. In molecular biology, one must strive to reduce the number of real-time chemical tests, and entirely avoid lengthy chemical tests.

We consider sequential machines Σ that can be described by models of the form

$$\begin{aligned} \sigma_{k+1} &= \varphi(\sigma_k, u_k), \\ y_k &= h_o(\sigma_k, u_k), \; k = 0, 1, 2, \ldots \end{aligned} \tag{1.1}$$

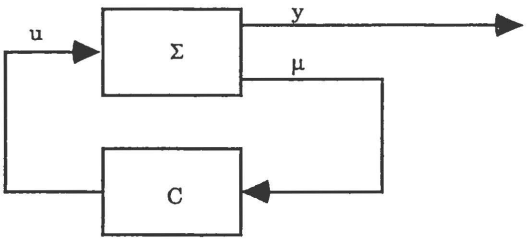
Here, φ and h_o are functions; σ_k is the state; u_k is the input value, and y_k is the output value. The initial condition σ_0 is given. The class M of potential models of Σ consists of a finite family $\{\varphi_i, h_{io}, \sigma_{i0}\}_{i=1, 2, \ldots}$ of functions and initial conditions.

The output value y_k of Σ signifies the 'outcome' of the machine's operation, and is not necessarily available for feedback use. The real-time feedback data for the controller is created by a *monitoring function* h_m that generates a *monitoring sequence* μ given by

$$\mu_k = h_m(\sigma_k, u_k), k = 0, 1, 2, ... \tag{1.2}$$

The function h_m indicates measurements that need to be performed in real-time to generate a feedback signal for the controller. It is designed together with the controller, so as to create all the feedback data necessary for the closed loop.

(1.3)



Here, the controller C uses the feedback signal μ to create the input signal u of Σ, and no external reference signal is used. Such a controller is called an *autonomous controller.*

In biological applications, where the monitoring process requires the detection of chemical compounds, it is of great importance to design h_m so as to reduce the complexity of measurements required in real time. In general, it is not possible to characterize mathematically a monitoring function that calls for a 'minimal' measurement complexity. For instance, the presence of certain compounds may be relatively easy to determine, whereas the presence of other compounds may be difficult, or even impractical, to establish. The monitoring complexity is not necessarily determined by the number of measurements, but rather by the nature of each individual measurement. Consequently, it seems preferable to characterize all possible pairs (h_m, C) of monitoring functions and controllers that achieve the control objective; This facilitates the selection of a simple pair, and, in particular, of a simple monitoring function h_m. In our framework, h_m can be selected prior to the selection of C, allowing reduction of measurement complexity (see section 4).

The controllers considered in the paper comprise all possible controllers, including controllers that exhibit "adaptive" or "learning" characteristics.

The present paper is an abridged version of HAMMER [1994] and [1993], which contain detailed proofs of all statements. Background on topics related to automata theory and discrete-event systems can be gained from GINSBURG [1962 and 1966], EILENBERG [1974], HOARE [1976], MILNER [1980], ARNOLD and NIVAT [1980], RAMADGE and WONHAM [1987], and the references listed in HAMMER [1993].

2. Basic notions and properties

2.1. Interpreters and controllers.

Let A be a non empty alphabet, and let A^* be the set of all words over A. A *sentence* over A is any (possibly empty) collection of words from A^*, which may include multiple copies of words. For applications in molecular biology, multiple copies of words represent multiple copies of molecules. The set of all sentences over the alphabet A is denoted by S_A. The *cardinality* #s of a sentence s is the total number of words in s, counting each word according to its multiplicity. When combining two sentences $s_1, s_2 \in S_A$ into a union, all copies of similar words need to be preserved. This is accomplished by the use of the disjoint union $s_1 \dot\cup s_2$, which includes all copies of all words contained in s_1 or in s_2.

Let $S(S_A)$ be the set of all sequences of sentences $s_0, s_1, s_2, ...$, where $s_i \in S_A$ for all integers $i \geq 0$. For a sequence $s \in S(S_A)$, we denote by s_i the i-th element of the sequence, i = 0, 1, ... The index i serves as a step counter; a step may or may not be linked to a specific time duration. It is convenient to use the notation s_i^j for the list of sentences $s_i, s_{i+1}, ..., s_j, j \geq i$.

The machines considered in the paper are defined as follows. Let $D \subset S(S_A)$ be a subset. An *interpreter* is a map $\Sigma : D \to S(S_A) \times S(S_A)$; It accepts input sequences $u \in D$, and generates pairs of output sequences $(y,\mu) \in S(S_A) \times S(S_A)$. The sequence y is the *output sequence*, and μ is the *monitored sequence* of Σ. The monitored sequence describes quantities that are measured at each step, and are available for feedback use in (1.3). For brevity, we set $SS_A := S(S_A) \times S(S_A)$.

In order to control a causal interpreter $\Sigma : D \to SS_A$, we combine it with a controller $C : \mathrm{Im}\, \Sigma_m \to D$, using the scheme

$$\begin{aligned} (y,\mu) &= \Sigma u, \\ u &= C\mu, \end{aligned} \tag{2.1.1}$$

depicted in (1.3). We denote the resulting input/output map by Σ_c. The system Σ_c has no input sequence, and is therefore called an *autonomous interpreter* $\Sigma_c : \emptyset \to SS_A$. We require the controller C to be strictly causal. This simplifies the discussion as well as the implementation.

2.2. Recursive models of interpreters.

The interpreters we consider are described by recursive models

$$(s_{k+1},x_{k+1}) = f[(s_k \cup u_k),x_k],$$
$$y_k = h_o(s_k \cup u_k), \quad (2.2.1)$$
$$\mu_k = h_m(s_k \cup u_k), k = 0, 1, 2, ...$$

Here, $f : S_A \times X \to S_A \times X$ is the *recursion function,* $h_o : S_A \to S_A$ is the *output function,* and $h_m : S_A \to \Delta$ is the *monitoring function;* The sets X and Δ are finite and non-empty. The recursion is started from a given initial condition $\sigma_0 := (s_0,x_0) \in S_A \times X$, and induces an interpreter $\Sigma : D \to SS_A$. An interpreter so induced is said to be a *recursive interpreter.*

The set X consists of the *states* of Σ, and the pair (s_k,x_k) constitutes the *status* of Σ at the step k, and s_k is the *internal value;* The *medium value* at the step k is $s_k \cup u_k$. For applications in molecular biology, s_k describes molecules present in the medium as a result of step k-1, and u_k describes the molecules externally injected into the medium at step k (the input). The medium value describes the set of all molecules in the medium at step k. We denote by $\Pi_s : S_A \times X \to S_A : (s,x) \mapsto s$ the standard projection onto the internal value.

We consider an interpreter Σ having a family $M = \{(f_1,h_{1o},\sigma_{10}), (f_2,h_{2o},\sigma_{20}), ..., (f_q,h_{qo},\sigma_{q0})\}$ of potential models; Here, $\{f_i\}$ are potential recursion functions, $\{h_{io}\}$ are potential output functions, and $\{\sigma_{i0}\}$ are potential initial conditions. The monitoring function for Σ is computed as part of the controller design, and is the same for all potential models. In operation, only one of the potential models M of Σ is present; We refer to this model of Σ as the *active model.* The identity of the active model is not known in advance.

Our objective is to develop techniques for the design of controllers that assign a prescribed steady-state response to a family of machines, so we need to discuss what constitutes a 'steady-state response'.

The *complete tail set* $T(S)$ of a set of sequences $S \subset S(S_A)$ is

$$T(S) := \cup_{s \in S} \cup_{k \geq 0} s_k^{\infty}.$$

For the autonomous interpreter $\Sigma_c : \varnothing \to S(S_A)$, denote by $T(\Sigma_c)$ the complete tail set of its (single) output sequence.

The intersection $T_1 \cap T_2$ of two complete tail sets T_1 and T_2 consists of all sequences $z := z_0, z_1, ...$ that satisfy the following property: there is a pair of integers $t, \tau \geq 0$ such that the sequence $v_t := z_0, v_{t+1} := z_1, v_{t+2} := z_2, ...$ belongs to T_1, and the sequence $w_\tau := z_0, w_{\tau+1} := z_1, w_{\tau+2} := z_2, ...$ belongs to T_2. In other words, shifted versions of the sequence z are found in T_1 and in T_2.

Finally, a *periodic tail set* is a complete tail set of a finite set of periodic sequences.

2.3. Statement of the problem.

Consider again the interpreter Σ having the family $M = \{(f_1,h_{1o},\sigma_{1o}), ..., (f_q,h_{qo},\sigma_{q0})\}$ of potential models; denote by Σ_i the interpreter having the model (f_i,h_{io},σ_{i0}), $i = 1, ..., q$. With the potential model Σ_i we associate a periodic tail set T_i that forms the prescribed steady state behavior of Σ in case model number i is the active model. We shall refer to T_i as the *target tail set* of the potential model i. We denote by

$$T := T_1 \times T_2 \times ... \times T_q$$

the cross product of all individual target tail sets, and refer to it as the *target tail set* of the interpreter Σ. The main question discussed in this paper is as follows.

(2.3.1)*Autonomous control of the interpreter* Σ *having the family* M *of potential models:* Find a monitoring function h_m and a strictly causal autonomous controller C such that the closed loop system Σ_{ic} satisfies $T(\Sigma_{ic}) \cap T_i \neq \varnothing$ for all $i = 1, ..., q$.♦

A controller C satisfying (2.3.1) is said to *steer* Σ to the target tail set T; It achieves the control objective without regard as to which potential model of Σ is active.

A set of sequences $D \subset S(S_A)$ is a *uniform set* if there is a set of sentences $V(D) \subset S_A$ such that D is the set of all sequences of elements of $V(D)$. We call then $V(D)$ the *value set* of D (see HAMMER [1993] for details).

We consider only recursive interpreters $\Sigma : D \to S(S_A)$ that satisfy the following requirements: (i) D is a uniform set with a finite value set $V(D)$; (ii) All potential recursion functions of Σ have a finite image; and (iii) The target tail set of Σ consists of a finite number of periodic sequences. Interpreters that satisfy these conditions are called *bounded interpreters.*

3. Existence of monitoring functions and controllers

3.1. Target sets in status space.

Let $M = \{(f_1,h_{1o},\sigma_{10}), ..., (f_q,h_{qo},\sigma_{q0})\}$ be the family of potential models of the bounded interpreter $\Sigma : D \to S(S_A)$, and let $V(D)$ be the value set of D. Recall that with each potential model (f_i,h_{io},σ_{i0}), there is associated a periodic target tail set T_i. We now translate the information contained in the target tail sets $T_1, ..., T_q$ into quantities in status space. For this purpose it will be convenient to regard the functions f_i and h_{io} as functions over the domain $S_A \times V(D) \times X$ (rather than $S_A \times X$ or S_A only), and we shall write $f_i(s_k,x_k,u_k)$ rather than $f_i[(s_k \cup u_k),x_k]$, and $h_{io}(s_k,u_k,x_k)$ rather than $h_{io}(s_k \cup u_k)$.

We have then $h_{io} : S_A \times V(D) \times X \to S_A$, and we denote by $P(S_A)$ the set of all subsets of S_A. Let $h_{io}^{-1} : P(S_A) \to S_A \times V(D) \times X$ be the inverse-set function of the output function h_{io}. Given a set of lists $S \subset S(S_A)$, we denote by $h_{io}^{-1}(S)$ the set of all lists $(b_0, b_1, ...)$ for which $(h_{io}(b_0), h_{io}(b_1), ...) \in S$. We construct the sets of sequences

(3.1.1) $\Theta_i := h_{io}^{-1}[T_i], i = 1, ..., q.$

The set Θ_i is called the *internal target set* of the model $(f_i, h_{io}, \sigma_{i0})$. Let $\Pi_u : S_A \times V(D) \times X \to V(D) : (s,u,x) \mapsto u$ be the standard projection onto the input value. For a sequence $\theta = (s_k,u_k,x_k), (s_{k+1},u_{k+1},x_{k+1}), ...$, we denote by $\Pi_u\theta$ the sequence $u_k, u_{k+1}, ...$.

Given r internal target sets $\Theta_{i(1)}, ..., \Theta_{i(r)}$, $r \in \{1, ..., q\}$, we denote by $(\Theta_{i(1)}, ..., \Theta_{i(r)})_u$ the set of all lists of sequences $(\theta_1, ..., \theta_r) \in \Theta_{i(1)} \times ... \times \Theta_{i(r)}$ for which $\Pi_u\theta_1 = \Pi_u\theta_2 = ... = \Pi_u\theta_r$, i.e., the set of all r-tuples of sequences that share a common input sequence; for $r = 1$, set $(\Theta_{i(1)})_u := \Theta_{i(1)}$. We call $(\Theta_{i(1)}, ..., \Theta_{i(r)})_u$ the *joint target tail* of the class $c := \{(f_{i(1)}, h_{i(1)o}, \sigma_{i(1)0}), ..., (f_{i(r)}, h_{i(r)o}, \sigma_{i(r)0})\}$ of potential models, and denote it by $\Theta_u(c)$. An element $(\theta_1, ..., \theta_r) \in (\Theta_{i(1)}, ..., \Theta_{i(r)})_u$ is *periodic* if the sequences $\theta_1, ..., \theta_r$ are all periodic. A non empty joint target tail always contains a periodic element (HAMMER [1994]).

Next, for an integer $r \in \{1, ..., q\}$, let N_r be the class of all subsets $\{(f_{i(1)}, h_{i(1)o}, \sigma_{i(1)0}), ..., (f_{i(r)}, h_{i(r)o}, \sigma_{i(r)0})\} \subset M$ of r models for which $(\Theta_{i(1)}, ..., \Theta_{i(r)})_u \neq \emptyset$. Each element of N_r consists of r models that share a common input sequence along certain paths within their internal target sets. Finally, define the *target compatibility class* $\mathcal{M}$ of M by

$$\mathcal{M} := \bigcup_{i=1,...,q} N_i.$$

Let $\phi = \{(f_{i(1)}, h_{i(1)o}, \sigma_{i(1)0}), ..., (f_{i(r)}, h_{i(r)o}, \sigma_{i(r)0})\} \in \mathcal{M}$ be a family of models, and consider an element $\theta := (\theta_1, ..., \theta_r) \in (\Theta_{i(1)}, ..., \Theta_{i(r)})_u$. Note that each θ_i, $i = 1, ..., r$, is a sequence $\theta_i = (s_k(i), x_k(i), u_k(i)), (s_{k+1}(i), x_{k+1}(i), u_{k+1}(i)), (s_{k+2}(i), x_{k+2}(i), u_{k+2}(i)), ...$ that starts at a step $k \geq 0$, and that $u_k(1) = u_k(2) = ... = u_k(r) =: u(\theta)$. Let

$$(s(\theta_i), x(\theta_i), u(\theta)) := (s_k(i), x_k(i), u_k(i))$$

be the first element of the sequence θ_i. Construct the set of vectors

$T(\phi) :=$

$$\bigcup_{\theta \in (\Theta_{i(1)}, ..., \Theta_{i(r)})_u} (s(\theta_1), x(\theta_1), s(\theta_2), ..., s(\theta_r), x(\theta_r), u(\theta)),$$

and set $T(\phi) := \emptyset$ if $\phi \notin \mathcal{M}$. The set $T(\phi)$ is called the *point target set* of the family ϕ, and it is a subset of $(S_A \times X)^r \times V(D)$. Once the members of the family ϕ are brought to a point belonging to $T(\phi)$, they can all be kept within their respective target tail sets by the same common input sequence.

3.2. Jointly reachable sets.

Let $M = \{(f_1, h_{1o}, \sigma_{10}), ..., (f_q, h_{qo}, \sigma_{q0})\}$ be the family of potential models of the bounded interpreter Σ. Denote by $\sigma_0 := (\sigma_{10}, ..., \sigma_{q0})$ the *initial status vector* of the family M. For a point $\rho := (\rho_1, ..., \rho_q, u) \in (S_A \times X)^q \times V(D)$, let $(f_1, ..., f_q)\rho$ be the point $(f_1(\rho_1,u), ..., f_q(\rho_q,u)) \in (S_A \times X)^q$, i.e., the result of applying the recursion function vector to the point ρ.

Now, let $\omega \in (S_A \times X)^q$ be a fixed point. We construct recursively a sequence of subsets $R_0(M,\omega)$, $R_1(M,\omega)$, ... of the space $(S_A \times X)^q \times V(D)$ as follows.

(i) $R_0(M,\omega) := \omega \times V(D)$, i.e., the set of all vectors of the form (ω,u), $u \in V(D)$.

(ii) Assume that $R_j(M,\omega)$ has been constructed for some integer $j \geq 0$; The set $R_{j+1}(M,\omega)$ is then given by

(3.2.1) $R_{j+1}(M,\omega) := \{\bigcup_{\rho \in R_j(M,\omega)} (f_1, ..., f_q)\rho\} \times V(D)$.

(iii) The *jointly reachable set* $R(M,\omega)$ of the family M *at the point* ω is defined by

(3.2.2) $R(M,\omega) := \bigcup_{j \geq 0} R_j(M,\omega)$.

The jointly reachable set $R(M,\omega)$ consists of all points in $(S_A \times X)^q \times V(D)$ that can be reached by applying common input sequences to all potential models of Σ, starting from the status vector ω. When $\omega = \sigma_0$, the initial status vector, we simply write $R(M) := R(M,\sigma_0)$, and refer to R(M) as the *jointly reachable set* of the family M. The jointly reachable set is computable (see HAMMER [1994]).

Given a subfamily $\phi = \{(f_{i(1)}, h_{i(1)o}, \sigma_{i(1)0}), ..., (f_{i(r)}, h_{i(r)o}, \sigma_{i(r)0})\} \subset M$ of potential models of Σ, let $\Pi_\phi : (S_A \times X)^q \times V(D) \to (S_A \times X)^i : (\rho_1, \rho_2, ..., \rho_q, u) \mapsto (\rho_{i(1)}, \rho_{i(2)}, ..., \rho_{i(r)})$ be the standard projection. Then,

$R(\phi,\omega) = [\Pi_\phi R(M,\omega)] \times V(D)$, $R(\phi) = [\Pi_\phi R(M)] \times V(D)$.

We induce now a partial order on the jointly reachable set R(M). For two points $r_1, r_2 \in R(M)$, we say that r_1 is a *predecessor* of r_2 (written $r_1 < r_2$) if $r_2 \in R(M,r_1)$, i.e., if there is an input list that leads the entire family M from r_1 to r_2.

3.3. Feedback control.

Let $\phi \subset M$ be a subset of potential models of Σ. A *partition* p of ϕ is a family $p = \{c_1, ..., c_k\}$ of disjoint subclasses of ϕ whose union is ϕ. A partition P of the partition $p = \{c_1, ..., c_k\}$ of ϕ is a set of partitions $P = \{p_1, ..., p_k\}$, where p_i is a partition of the class c_i, $i = 1, ..., k$. The combined partition Pp consists of the classes $Pp = \{\{p_1\}, \{p_2\}, ..., \{p_k\}\}$.

Given two partitions p and q of ϕ, the partition q is *finer* than the partition p (written $p \leq q$) if there is a partition P of the partition p such that $q = Pp$. A *partition chain* $\mathcal{P}(\phi)$ of ϕ is simply an ordered list of partitions $p_0 \leq p_1 \leq ... \leq p_m$ of ϕ, with $p_0 := \phi$ being the identity partition.

Let $\mathcal{P}(\phi) = \{p_0 \leq p_1 \leq ... \leq p_m\}$ be a partition chain, and let $c \in p_i$ be a member of the partition p_i. We denote by $p_{i+1}(c)$ the partition of the class c induced by the partition p_{i+1}; i.e., letting $p_{i+1} = \{c_{i+1,1}, ..., c_{i+1,k}\}$, the partition $p_{i+1}(c)$ consists of all non empty intersections $c \cap c_{i+1,j}$, $j = 1, ..., k$. A *path* of a partition chain $\phi \leq p_1 \leq ... \leq p_m$ of ϕ is an ordered list $\{\phi, c_1, c_2, ..., c_m\}$ of subsets of ϕ, where $c_i \in p_i(c_{i-1})$, $i = 1, ..., m$, and $c_0 := \phi$.

Let $c = \{(f_{i(1)}, h_{i(1)o}, \sigma_{i(1)0}), ..., (f_{i(r)}, h_{i(r)o}, \sigma_{i(r)0})\}$, $1 \le r \le q$, be a subset of the family M of potential models of Σ. For a point $\rho = (\rho_1, ..., \rho_q, u) \in R(M)$, denote

(3.3.1) $\Gamma(c)\rho := \bigcup_{j=i,...,r}\{[\Pi_s \rho_{i(j)}]\cup u\}$,

i.e., the set of all medium values corresponding to the members of c at the point ρ.

Now, let $\phi \subset M$ be a subfamily of potential models, let $P = \{c_1, ..., c_m\}$ be a partition of ϕ, and let $h : S_A \to \Delta$ be a function. We say that h is *compatible with the partition* P *of* ϕ *at the point* $\rho \in R(M)$ if the following holds for all i, j = 1, ..., m.

(3.3.2) $h[\Gamma(c_i)\rho] \cap h[\Gamma(c_j)\rho] = \varnothing$ whenever $i \neq j$,

i.e., the function h assumes a distinct set of values over each one of the classes $c_1, ..., c_m$.

The following statement, taken from HAMMER [1994], provides necessary and sufficient conditions for the existence of a controller that steers the bounded interpreter Σ to its target tail set. It is one of the main results of the paper, and we discuss its intuitive meaning immediately.

(3.3.3) THEOREM. Let $\Sigma : D \to S(S_A)$ be a bounded interpreter having the family M of potential models, and let T be the target tail set of Σ. Denote by T(c) the point target set of a subfamily $c \subset M$. Then, (i) and (ii) are equivalent.

(i) There is a strictly causal autonomous controller C and a monitoring function $h_m : S_A \to \Delta$ that steer Σ to its target tail set.

(ii) There is a partition chain $\mathcal{P}(M) = \{M \le p_1 \le ... \le p_m\}$ of M, every path $\{c_0, c_1, ..., c_m\}$ of which satisfies the following.

(iia) There are points $\rho_1 < ... < \rho_m$ of R(M) and a function $h : S_A \to \Delta$ such that h is compatible with the partition $p_i(c_{i-1})$ at the point ρ_i, i = 1, ..., m, and

(iib) $T(c_m) \cap \Pi(c_m)R(M,\rho_m) \neq \varnothing$ when $m \ge 1$, or

(iic) $T(M) \cap \Pi(M)R(M) \neq \varnothing$ when $m = 0$.

In qualitative terms, the Theorem suggests a controller C that acts through a hierarchical identification scheme, using the function $h_m : S_A \to \Delta$ as the monitoring function. At each step, the controller attempts to narrow the set of possible models of Σ by checking the values submitted by the monitoring function through the feedback channel. The controller starts by providing Σ with an input list that takes the family M from the initial condition to the point $\rho_1 \in R(M)$. At ρ_1 condition (iia) means that the controller can identify the class $c_1 \in p_1(M)$ to which the active model belongs. The controller then supplies a continuation of the input list that leads from ρ_1 to ρ_2. At this point, condition (iia) shows that the controller can identify the class $c_2 \in p_2(c_1)$ to which the active model belongs. And so on, up to the point ρ_m at which the class $c_m \subset M$ to which the active model belongs is identified. In view of (iib), more detailed identification of the active model is not necessary, since all members of c_m can be kept within their respective target tail sets by the same continuation of the input sequence. Through this technique, Theorem (3.3.3) induces an algorithm for the design of all controllers that steer Σ to its control objective. The complete algorithm, called the *controller design algorithm*, is provided in HAMMER [1994].

The controller C performs inter-related control and identification processes. It steers the system along a list of points $\rho_1, \rho_2, ..., \rho_m$, so that at each successive point finer and finer identification of the active model is possible. An algorithm that yields the points $\rho_1, ..., \rho_m \in R(M)$ is described in HAMMER [1994]. The selection of a monitoring function is discussed in section 4 below.

4. Common divisors and the search algorithm.

Consider a sequence $u \in S(S_A)$ and a (finite or infinite) non empty list $\lambda = (\lambda_0, ..., \lambda_n) \in (S_A)^{n+1}$. We say that λ is a *left divisor* of u if $u_0^n = \lambda$. The *length* $|\lambda|$ of λ is the number of elements in its list, i.e., $|\lambda| = n+1$ here. Also, λ is a *common left divisor* of a set $S \subset S(S_A)$ whenever λ is a left divisor of every element of S. The list λ is a *longest common left divisor* of the set S if λ is a left divisor of every common left divisor of S. We induce an equivalence relation L on the set $S(S_A)$ by writing uLv whenever the two sequences $u, v \in S(S_A)$ have a common left divisor.

Next, let $S(1), S(2), ..., S(q) \subset S(S_A)$ be a family of non empty subsets of sequences. A *comb* of the family $\{S(i)\}_{i=1}^q$ is any set χ of sequences that contains exactly one sequence from each one of the sets S(1), ..., S(q).

Consider now a bounded interpreter $\Sigma : D \to S(S_A)$ having the family $M = \{\Sigma_1, ..., \Sigma_q\}$ of potential models, and let T_i be the target tail set associated with the model Σ_i. For a sequence $u \in D$, let $T(\Sigma_i u)$ be the complete tail set of the output sequence $\Sigma_i u$. The set $I(\Sigma_i)$ of *successful input sequences* of the potential model Σ_i is the set of all ultimately periodic input sequences $u \in D$ for which $T(\Sigma_i u) \cap T_i \neq \varnothing$; i.e., all ultimately periodic input sequences that steer Σ_i to its target tail set. We shall assume throughout that $I(\Sigma_i) \neq \varnothing$ for all i = 1, ..., q, since otherwise the control objective is not achievable. For bounded interpreters, the sets $I(\Sigma_i)$ can all be computed in a finite number of steps (HAMMER [1994]).

We now describe in qualitative terms a computational algorithm that determines whether or not there is an autonomous controller that achieves the desired control objective (HAMMER [1994]).

First, select a comb χ of the class $\{I(\Sigma_i)\}_{i=1}^q$. Compare the initial segments of the sequences of χ el-

ement by element to determine whether they have a common left divisor. If there is no common left divisor, choose another comb; if none of the combs of $\{I(\Sigma_i)\}_{i=1}^{q}$ has a common left divisor, then no appropriate controller exists. This is due to the strict causality of the controller, which implies that all sequences generated by it have the same first element.

If the sequences of χ have a common left divisor, let $\alpha_{1,1}$ be their longest common left divisor. Assign $\alpha_{1,1}$ as the output list of the controller C for the steps $0, 1, ..., |\alpha_{1,1}|$; This list is appropriate for all models of the family M, irrespective of which one is active. If $|\alpha_{1,1}| = \infty$, we obtain an open loop controller C, completing the process. Otherwise, if $|\alpha_{1,1}| \neq \infty$, proceed as follows.

Delete the initial segment $\alpha_{1,1}$ from all sequences of χ, and denote by χ_1 the resulting family of sequences. Using the equivalence relation L on χ_1, induce a partition P_1 of the family M of potential models of Σ, by grouping into each class of P_1 all models whose corresponding sequences in χ_1 have a common left divisor. Then, all potential models that belong to the same class of P_1 require the same input value at the step $|\alpha_{1,1}|+1$; And potential models from different classes of P_1 require different input values at this step. Since these input values are generated by the controller C, the strict causality of C implies that it must be possible by the step $|\alpha_{1,1}|$ to determine to which class of P_1 the active model belongs. This determination is made based on the monitored values for the steps $0, ..., |\alpha_{1,1}|$, and the monitored values are determined by the medium values through the monitoring function. Thus, it is possible to select a monitoring function that facilitates such determination if and only if the determination can be made from the medium values for these steps.

Let $\mu_0(i), \mu_1(i), ..., \mu_{|\alpha_{1,1}|}(i)$ be the medium list for steps $0, ..., |\alpha_{1,1}|$ generated when model number i is driven by the input list $\alpha_{1,1}$ from its initial status $\sigma_{i0}, i = 1, ..., q$. Use the equivalence relation $\mu_0^{|\alpha_{1,1}|}(i) = \mu_0^{|\alpha_{1,1}|}(j), i, j \in \{1, ..., q\}$, to induce a partition $P_{\mu,1}$ of the family M of potential models: Each class of $P_{\mu,1}$ consists of all potential models whose medium lists $\mu_0^{|\alpha_{1,1}|}(\bullet)$ are identical. A slight reflection shows then that the class of P_1 to which the active model belongs can be identified by a strictly causal controller if and only if $P_{\mu,1} \geq P_1$. Using analogous steps, one then continues in this manner until the active model is identified to within a class of potential models that has a non empty target point set. A controller exists if and only if this process is successful for at least one comb of $\{I(\Sigma_i)\}_{i=1}^{q}$.

The complete algorithm, called the *search algorithm*, is given in HAMMER [1994]. It fulfills the following objectives.

(i) It determines whether or not the control objective can be achieved for the family M of potential models of Σ.

(ii) In case the control objective is achievable, the algorithm characterizes all possible monitoring functions.

An important aspect of the algorithm is that it does not depend on the controller choice. The design process is started with the search algorithm to determine whether or not the control objective is achievable; if it is, an appropriate monitoring function is selected. Once the monitoring function has been selected, the controller design algorithm derives a design for an appropriate controller.

5. References

B. ALBERTS, D. BRAY, J. LEWIS, M. RAFF, K. ROBERTS, and J.D. WATSON, [1994], "Molecular biology of the cell", Third Edition, Garland Pub. Inc., New York.

A. ARNOLD and M. NIVAT, [1980], "Controlling behaviors of systems: some basic concepts and some applications", in Mathematical foundations of computer science, 1980 : proceedings of the 9th symposium held in Rydzyna, Poland, September 1-5, 1980 / edited by P. Dembinski. Lecture notes in computer science vol. 88 Berlin ; New York : Springer-Verlag.

S. EILENBERG, [1974], "Automata, languages, and machines", Vol. A, Academic Press, NY.

S. GINSBURG, [1962], "An introduction to mathematical machine theory", Addison-Wesley publishing Co., Reading, MA; [1966], "The mathematical theory of context free languages", McGraw-Hill book Co., NY.

J. HAMMER, [1993], "On corrective control of sequential machines", submitted for publication; [1994], "On the control of incompletely specified sequential machines", submitted for publication.

C.A.R. HOARE, [1976], "Communicating sequential processes", Prentice-Hall, NJ.

IEEE Computer Systems Society (Conference publication, [1974], "Proceedings of the 1974 conference on biologically motivated automata theory", McLean, VA.

S.A. KAUFFMAN,[1969], "Metabolic stability and epigenesis in randomly constructed genetic nets", J. Theor. Biol., Vol. 22, 437-467.

A. LINDENMAYER, [1968], "Mathematical models for cellular interactions in development", Parts I and II, Journal of Theoretical Biology, v. 18, pp. 280-315.

R. MILNER, [1980], "A calculus of communicating systems", Lecture Notes in Computer Science, Springer Verlag, Berlin.

J. von NEUMANN, [1966], "The theory of self-reproducing automata", Edited and completed by Arthur W. Burks, University of Illinois Press, Urbana.

P.J. RAMADGE and W.M. WONHAM, [1987], "Supervisory control of a class of discrete event processes", SIAM J. Control and Optimization, vol. 25, pp. 206-230.

N. RASHEVSKY, [1948], "Mathematical biophysics", The University of Chicago Press.

G. ROZENBERG and A. SALOMAA, [1975], "L Systems", Lecture Notes in Computer Science, Vol. 15, Springer Verlag, Berlin.

M. SUGITA, [1963], "Functional analysis of chemical systems in vivo using a logical circuit equivalent. II. The idea of a molecular automaton", J. Theor. Biol., Vol. 4, pp. 179 - 189.