Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Dealing with biological constraints in the synthesis of controllers for gene regulatory networks

Fabio L. Baldissera * **José E. R. Cury** *

* *Universidade Federal de Santa Catarina, Florianópolis, SC 88040-970 Brazil (e-mail: jose.cury@ufsc.br).*

Abstract: Controlling the behavior of intracellular networks has many important applications in Biotechnology and Medicine. In a previous work, we showed how the ideas from Supervisory Control Theory could be used to solve the so-called state attraction problem of biological cells, i.e. guiding a cell from an initial state to some target state. The devised supervisor was then implemented by means of synthetic genes inserted in the cell. In this paper, we present a way to decrease the design complexity of the synthetic genes obtained by this procedure. This is an important step if one wants to apply our approach to real biological control problems.

1. INTRODUCTION

The use of mathematical models to understand the behavior of biological systems is not a new endeavor in science, as can be attested by the early efforts of the systems' theory community (see the work of Wolkenhauer [2001] and the references therein). Nevertheless, the recent progresses in Molecular Biology have sped up the pace at which such models are constructed, culminating in the impressive number of publications that are categorized under the field of Systems Biology (Machado et al. [2011], Kitano [2002]).

One purpose of such mathematical models is to control the dynamics of the system under study. Intervening in biological systems to alter their behavior might be applied, for instance, to enhance the bioproduction of metabolites, to engineer tissues or to treat and cure diseases (Benenson [2012], Menolascina et al. [2011], Datta et al. [2007], Bagh et al. [2011], van Schuppen [2005]).

The control of cellular processes has been subject of increasing interest in the scientific community (see a review directed to Control Engineers in the work of Cury and Baldissera [2013]). These researches approach the problem from distinct intervention strategies (e.g. with external devices playing the roles of sensors, actuators and controllers *or* with synthetic biomolecules performing the main tasks of a control system) and with the help of different mathematical formalisms (as can be seen in the works of Datta et al. [2007], Menolascina et al. [2011], Milias-Argeitis et al. [2011], Chaves and Gouzé [2011], Bagh et al. [2011]).

In a previous work, we showed qualitatively how the ideas from Supervisory Control Theory (SCT) could be applied to solve the state attraction problem, i.e guiding a cell from an initial state to some prespecified target state. In our approach: (a) the plant consists of a gene regulatory network that can be abstracted by a discrete-event model; and (b) the controller is implemented as a set of synthetic genes that are inserted in the cell to close the loop (Cury and Baldissera [2013], Baldissera and Cury [2012]).

The problem of state attraction in general has already been treated in the field of SCT. In the work of Brave and Heymann [1989], for example, the authors derived necessary and sufficient conditions for the existence of a controller that can drive a finite automaton from an initial state to some target state, keeping it there indefinitely, a problem they called stabilization of discrete-event processes. The same researchers tackled the issue of optimality in the state attraction problem, where a set of target states is reached through a path of minimum cost (Brave and Heymann [1993]). Kumar et al. [1993], on the other side, extended the ideas of Brave and Heymann [1989] to the stability and stabilizability of the *language* generated by finite state machines.

Our previous works, which form the basis for the present paper, differed from the ones cited in the sense that (a) the event set was qualitatively richer than that adopted by the other authors; it included, besides the traditional partition of events into controllable and uncontrollable ones, preemptability features, and (b) the development of our ideas was heavily application-oriented, that is, it was based on a concrete biological problem.

From the biological point of view, as already mentioned, our approach to control gene networks relies on the possibility to realize the controller as a set of synthetic genes to be inserted in the cell. As a result, the synthetic genes must be programmed to compute boolean functions written in the disjunctive normal form. Nevertheless, programming genes is not an easy task. Even with the recent advances brought by the field of Synthetic Biology (Andrianantoandro et al. [2006]), that aims at synthesizing new molecules and biological circuits that are able to compute, there are still many restrictions regarding the functions that can be implemented in the hardware offered by nature (i.e. in the case of synthetic genes, this hardware includes sequences of DNA nucleotides). The more logical operations and variables the boolean functions to be computed by synthetic genes require, for example, the more difficult it is to implement them (Buchler et al. [2003]). In this paper, our main contribution is to show, given a supervisor S that achieves a set of specifications, how one can build a new supervisor S^* that is realized by means of synthetic genes that can be more easily constructed.

The paper starts reviewing some basic concepts in Section 2, mainly those of Gene Regulatory Networks, Boolean Networks and Supervisory Control Theory. In Section 3, we lay out the principles of the formal model that we use for control synthesis. In Section 4, we pose the problem to be solved and present a general algorithm to tackle it. Section 5 brings the conclusions and final remarks.

2. BASIC CONCEPTS

2.1 Gene Regulatory Networks

Gene regulatory networks are intracellular biological networks that coordinate important cellular processes (Karlebach and Shamir [2008]), such as cell differentiation, cell cycle and adaptation to different environmental conditions (variations in temperature, availability of nutrients, etc.).

The nodes of gene networks are genes, which assume different *states*. A gene state can be seen as a measure of how active this gene is and, therefore, the rate at which the cellular machinery is reading information from it in order to generate proteins. The reading process (also called gene expression) is composed of two steps, (a) transcription, where the information contained in the genes is transcribed to intermediate molecules called messenger RNAs (or simply mRNAs) and (b) translation, where the information in the mRNAs is converted into specific proteins (Alberts et al. [2008]). Proteins themselves can influence the process of transcription, by either enhancing or repressing it.

An arc directed from gene v_i to v_j denotes thus the effect of the protein produced by v_i, which will be denoted u_i, onto the activation or inhibition of gene v_j. Such regulatory proteins are termed *transcription factors*. A node might impact on multiple nodes, including itself. Each node might also be affected by multiple nodes. In this case, the regulatory effect of all transcription factors over this gene is exerted in a cooperative fashion. Hence, not only are genes composed of a region that codes for a specific protein (coding region), but also by a region that regulates its expression (regulatory region), by means of protein-DNA interactions.

2.2 Boolean Networks

A boolean network is a mathematical structure completely characterized by a pair (V, F), where $V = \{v_1, \ldots, v_N\}$ is a set of boolean variables $v_i \in \{0, 1\}$ and $F = \{f_1, \ldots, f_N\}$ is the set of boolean functions $f_i : \{0, 1\}^N \to \{0, 1\}$ used to update the state of v_i, based on the present value of $\mathbf{v} = [v_1 \ldots v_N]$. This update may be synchronous (i.e. the states of all $v_i \in V$ are updated at the same time) or asynchronous.

Boolean networks may be seen as abstractions of real, and more complicated, gene networks (see Bornholdt [2005], Machado et al. [2011] and the references therein). In this case, the boolean variables v_i represent the state of each gene in the network (i.e. 1 for a gene that is being expressed, and 0 for a silent gene or one expressed at a basal level), whereas the regulatory interactions between genes, which are performed by transcription factors, are captured by the boolean functions f_i. If a gene v_i, for example, is repressed by the transcription factor produced by gene v_j, then one possible expression for f_i is $f_i = \overline{v_j}$, where $\overline{v_j}$ denotes the logical negation of v_j.

2.3 Supervisory Control Theory (SCT)

Discrete-Event Systems (DES) are dynamical systems that have two basic properties (Cassandras and Lafortune [2008]):

- Their state space is a discrete set;
- The transition between states is not time-driven (as is the case in ordinary differential equations where time is the independent variable), but event-driven.

Event in this framework is a primitive concept and could, in biological systems, represent the activation of a gene, the phosphorylation of a protein (the addition of a phosphate group to one of its amino acids) or the raise in the concentration of a given intracellular metabolite above a certain threshold, for example.

SCT is a formalism devoted to modify the behavior of discrete-event systems by means of feedback control, so that the closed-loop system achieves a given set of specifications (Ramadge and Wonham [1989]). Generally, both the open-loop system and the specifications are modeled by finite automata.

The controller (or supervisor) S within this framework interacts with the system to be controlled, i.e. the *plant* P, according to the usual feedback control concept (Cassandras and Lafortune [2008]): S observes some of the events that P executes (others may be unobservable), then S computes which events of P are allowed next, so that a set of *specifications* E is met. The controller can enable or disable plant events. Not all events can be disabled (this is a feature that depends on how the controller interacts with the plant), therefore, not all specifications are *controllable*, that is, there may be cases where no supervisor can accomplish the specifications.

Later extensions of SCT included the concept of events that can be *forced* by the supervisor, in a way to preempt the occurrence of all other plant events (Golaszewski and Ramadge [1987]) or of just a subset of them (Brandin and Wonham [1994]).

3. MODELING AND CONTROL OF GENE NETWORKS

3.1 Native Gene Network Model

In previous works, we showed qualitatively how one can model and control gene regulatory networks whose dynamics are amenable to a discrete-event representation (Baldissera and Cury [2012]). A more formal approach to this problem will be soon submitted for peer-review. There, we assumed that (a) genes have boolean states, that is, they can be either "on" (i.e. being transcribed) or "off" (i.e. not being transcribed); and (b) it is possible to "program" synthetic genes and insert them in the cell to be controlled, in a way that these genes realize the control

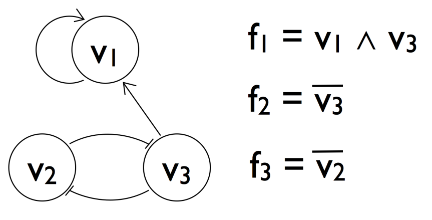


Fig. 1. Native gene network to be controlled, composed of three genes that interact by means of their transcription factors according to the updating functions f_i. Hammerhead arcs are used for repressive interactions; arrows, for activating ones.

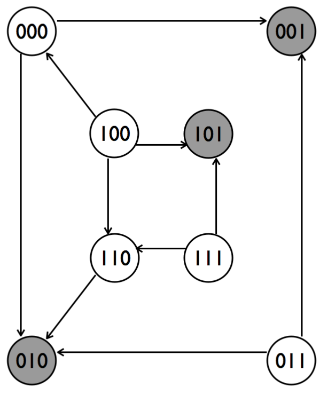


Fig. 2. Representation of the state transition diagram for the gene network in figure 1. The shaded states correspond to open-loop attractors.

actions prescribed by the mathematical formulation of the supervisor.

We shall briefly review these ideas in the context of an example that will be elaborated throughout this paper. Consider the gene regulatory network depicted in figure 1. This network is composed of three genes v_i that interact by means of their transcription factors u_i according to the updating functions $f_i : \{0,1\}^3 \to \{0,1\}$, for $i = \{1,2,3\}$. For the sake of simplicity, we assume that a protein u_i is present in the cell if and only if gene v_i is being transcribed (i.e. it is "on"), that is, $v_i = u_i$. The ideas discussed in this paper do not depend on this simplification and can be easily extended to deal with the more general case.

The state transition diagram of the gene network in figure 1, for a complete asynchronous updating rule (that is, for the case where only one gene is updated at a time), is portrayed in figure 2. Each state can be unambiguously attached to a string that indicates which transcription factors are present in the cell. For example, state 010 denotes the cellular state where the only transcription factor that is present in the cellular environment is u_2.

The state transition diagram shown in figure 2 corresponds to the open-loop behavior of the network in figure 1. Given an initial state q_0, it may be associated with a finite state automaton $G = (Q, \Sigma, \delta, q_0, Q_m)$, with:

- $Q = \{000, 001, 010, 011, 100, 101, 110, 111\}$, the set of states, where each string has the semantics previously given;

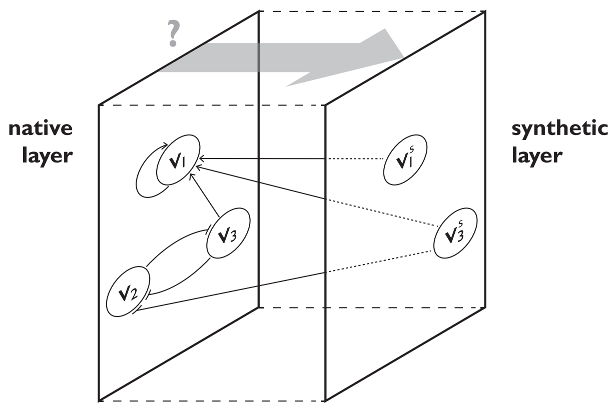


Fig. 3. Control concept employed throughout this work. A native gene network is controlled by synthetic genes. It is known which proteins are synthesized by v_1^s and v_3^s, but the impact of u_1, u_2 and u_3 upon v_1^s and v_3^s (represented by the shaded thick arrow) will be defined only after the supervisor synthesis.

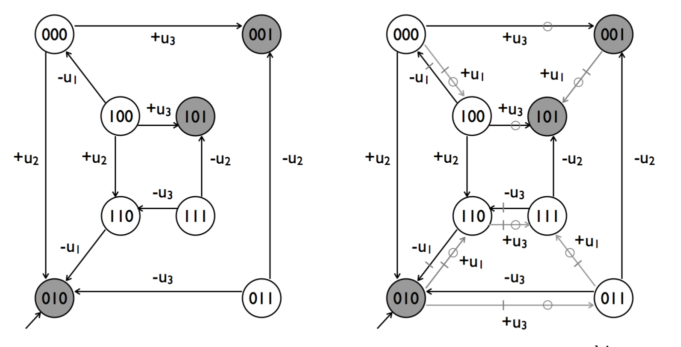
- $\Sigma = \{+u_i, -u_i\}$, for $i = \{1,2,3\}$, the set of events. The event $+u_i$ stands for the rise in the concentration level of protein u_i above a certain threshold, whereas $-u_i$, for the decrease of u_i below this threshold;
- $\delta : Q \times \Sigma \to Q$, the partial transition function, trivially derived from figure 2 and the definition of the events in Σ. For instance, $\delta(000, +u_3) = 001$;
- $q_0 \in Q$, any initial state;
- $Q_m = Q$, the set of marked states, that is, those associated with accomplished tasks. A choice of Q_m depends on the control goals.

3.2 Enlarging the Plant model to encompass the control inputs

Suppose one wants to intervene in the behavior of this gene network. In order to accomplish the task, assume that it is possible to insert in the cell two synthetic genes, v_1^s and v_3^s, that satisfy (a) v_i^s codes for the same protein as gene v_i, that is, protein u_i, and (b) the regulatory region of v_i^s may be programmed, *prior* to its insertion in the cell, to be activated at any specific state or combination of states. In this way, v_1^s could be designed, for instance, to be transcribed at a state where the only transcription factor present is u_3, that is, state 001. Figure 3 illustrates schematically the control concept that is being employed in this work.

Our choice here for synthetic genes that code for the transcription factors u_1 and u_3 was arbitrary, but it could have been a consequence of concrete biological restrictions. For example, if the synthesis of protein u_2 by the cell is energetically expensive, imposing an additional production of it could disturb cell behavior in unwanted ways. Besides that, this choice could reflect the result of a sensitivity analysis, where the proteins that have the largest impact on the native network behavior are chosen.

Assume also, for the remaining of the paper, that the transcription and translation efficiency of the synthetic genes can be made such that their dynamics are faster than those associated with the native genes. That is, the



(a) Automaton G (b) Automaton G^{hi}

Fig. 4. G describes the behavior of the native gene network, whereas G^{hi} captures the dynamics of the native network *plus* the possible control inputs provided by the synthetic genes v_1^s and v_3^s. In 4(b), features added by the introduction of synthetic genes are colored in gray. Cut arrows stand for controllable events and those with a circle, for forcible ones.

activation of a synthetic gene can *preempt* the occurrence of events associated with native genes. This assumption is not unrealistic and there are indeed reports in literature that provide support to it (Blazeck et al. [2012], Gingold and Pilpel [2011]). Additionally, one may also generalize our framework to encompass the case where not all the events in Σ can be preempted.

The automaton G, which models the open-loop behavior of the plant, can be now modified to take into account the control possibilities offered by the synthetic genes v_i^s, leading to a new automaton that we call $G^{\text{hi}} = (X, \Sigma^{\text{hi}}, \delta^{\text{hi}}, x_0, X_m)$. In G^{hi} three main changes with respect to G can be recognized:

- controllable *and* forcible events $+u_1$ and $+u_3$ are added at some states, as is the case in the transition from 000 to 100. Note that these events can now be generated by the activation of the synthetic genes at these states;
- the controllability nature of $-u_1$ and $-u_3$ is changed, as is the case in the $-u_3$ event from 111 to 110. Given that the control action can preempt other plant events, the decrease in the concentration level of u_1 and u_3 can be counteracted by the activation of the corresponding synthetic genes at these states;
- events $+u_1$ and $+u_3$ can now be forced, though not controlled, at those states where they could already occur in G, as is the case with the event $+u_3$ from 100 to 101. Here the transition between these states can occur either by an activation of the native gene v_3 (which cannot be controlled) or by means of the activation of v_3^s (which can be controlled).

4. SUPERVISOR SYNTHESIS, REALIZATION AND ITS BIOLOGICAL RESTRICTIONS

4.1 Supervisor synthesis and its realization

Suppose one wants to drive the native gene network in figure 4(a) from the attractor $x_0 = 010$ to $X_m = \{101\}$. That is, it is required to achieve a physiological state where, among the transcription factors u_1, u_2 and u_3, only u_1 and u_3 are present. These stable states (in the sense that no event departs from them in the open-loop system) could correspond, for instance, to different cell types in multicellular organisms or simply to different physiological states in a given species of bacteria.

Table 1. Definition of the supervisor $S : X \to 2^{\Sigma}$ and its implications.

States	$S(x)$	Preempted or disabled events	Required activation
010	$\{+u_1\}$	$\{+u_3\}$	v_1^s
110	$\{+u_3\}$	$\{-u_1\}$	v_3^s
111	$\{-u_2\}$	$\{-u_3\}$	v_3^s
101	$\{\}$	$\{\}$	-

To solve this problem, let a supervisor S be defined as a function that associates to each state $x \in X$ a set of events $\sigma \in \Sigma^{\text{hi}}$ which are enabled or forced by the supervisor. The admissible control inputs for this enriched case (where events can be not only disabled but also forced by S), as well as the necessary and sufficient conditions for the existence of a supervisor S that brings the plant G^{hi} from x_0 to X_m, will be laid out by us in a future publication. In this paper, we shall keep this discussion in a more intuitive level, to focus on the main ideas we want to convey.

For the example illustrated in figure 4(b), it is possible to verify that the path 010-110-111-101 constitutes one solution to the posed problem. It suffices to force event $+u_1$ at 010, force $+u_3$ at 110 and, finally, disable $-u_3$ at 111. The path 010-011-111-101 is also a solution to the problem, and finding the corresponding control input at each state is, for this simple example, a straightforward task.

Among all possible closed-loop behaviors, let us take the path 010-110-111-101 to illustrate the point of this paper. Table 1 summarizes (i) the corresponding control inputs at the states where the supervisor is defined, (ii) which events are prevented (disabled and/or preempted) by the applied control input and (iii) which synthetic genes must be expressed (that is, activated) at each state to realize the control input.

Forcing event $+u_1$ at 010 requires the activation of synthetic gene v_1^s at the same state. At 111, v_3^s must be activated to disable transition $-u_3$. At 010, though, no synthetic gene needs to be activated to disable $+u_3$ (it suffices to *not* activate it, given that v_3^s is the only gene that can generate $+u_3$ at 010). As a last remark, note that forcing $+u_3$ at 110 has the consequence of preventing the occurrence of $-u_1$ and, hence, $-u_1$ does not need to be disabled (that is, v_1^s does not need to be activated at 110).

As a result, synthetic gene v_1^s must be expressed at 010 only, whereas v_3^s, at 110 and 111. Therefore, the updating functions of v_1^s and v_3^s can be written as $f_1^s(\mathbf{v}, \mathbf{v^s}) = \overline{(v_1 \vee v_1^s)} \wedge v_2 \wedge \overline{(v_3 \vee v_3^s)}$ and $f_3^s(\mathbf{v}, \mathbf{v^s}) = (v_1 \vee v_1^s) \wedge v_2^s$, respectively, for $\mathbf{v} = [v_1 \ v_2 \ v_3]$ and $\mathbf{v^s} = [v_1^s \ v_3^s]$. Let now be U_i, $i \in \{1, 2, 3\}$, a boolean variable that has value 1 if transcription factor u_i is present in the cell and 0 otherwise. Then, f_1^s and f_3^s may be written as a function of U_i as $f_1^s = \overline{U_1} \wedge U_2 \wedge \overline{U_3}$ and $f_3^s = U_1 \wedge U_2$.

These updating functions are strictly related to the regulatory regions of synthetic genes v_1^s and v_3^s, that is, they

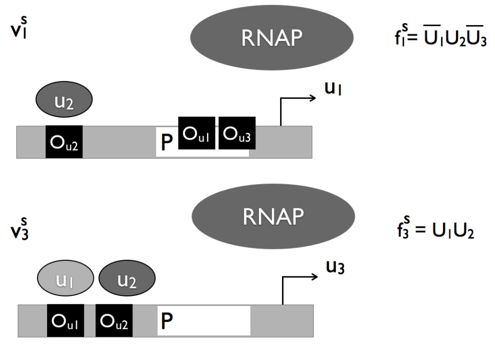


Fig. 5. Possible realizations of the required regulatory regions of synthetic genes v_1^s and v_3^s (for more details, the reader is referred to the work of Buchler et al. [2003]). This figure represents the situation in which each gene is transcribed. O_{u_i} denotes the binding site of transcription factor u_i to the DNA. 'P' stands for the promoter region, the sequence in the DNA where the enzyme RNA polymerase (RNAP) attaches to start transcription.

describe how transcription factors u_1, u_2 and u_3 affect the expression of genes v_1^s and v_3^s. One possible realization of f_1^s and f_3^s is shown in figure 5, based on the work of Buchler et al. [2003]. The cited work shows how different boolean functions can be implemented with regulatory regions of genes, by adjusting the position in the DNA of each transcription factor binding site O_{u_i}, as well as its binding affinity to the associated transcription factor u_i.

This realization procedure can be applied to updating functions f_i^s written in the disjunctive normal form, that is, to $f_i^s(\mathbf{U}) = f_i^s(U_1, \ldots, U_N) = s_{i,1} \vee s_{i,2} \vee \ldots \vee s_{i,m}$, where each minterm $s_{i,j}$ is a conjunction of variables $U_1, \ldots, U_N$ or their logical complements $\overline{U_1}, \ldots, \overline{U_N}$, and N is the total number of transcription factors in the gene network. Note that in this framework, each $s_{i,j}$ is associated with a state $x \in X$ (or a set of states) where the synthetic gene v_i^s must be activated.

The "programming" of regulatory regions to compute boolean functions has biological limitations. Take, for example, the regulatory region of v_1^s in figure 5. It can be broken down into two conditions (i) U_2 is valid *and* (ii) $\overline{U_1} \wedge \overline{U_3}$ holds. The implementation of $\overline{U_1} \wedge \overline{U_3}$, according to the cited procedure, requires that both transcription factors u_1 and u_3 have binding sites inside the promoter P, the region in the DNA to which the enzyme RNA polymerase (RNAP) attaches, in order to initiate transcription. So when either u_1 or u_3 is present, they prevent the RNAP from binding to P. But due to the limited size of the promoter, there is a natural restriction in the number of binding sites O_{u_i} that can be nested inside P (the so-called promoter overcrowding problem).

Take now the regulatory region of v_3^s in figure 5. The biological realization of boolean function $U_1 \wedge U_2$ implies (i) choosing O_{u_1} and O_{u_2} such that u_1 and u_2 alone cannot stably bind to the DNA, and (ii) putting O_{u_1} and O_{u_2} adjacent to each other, so that, with the help of cooperative interaction between u_1 and u_2, the binding of both transcription factors to the DNA occurs. For the implementation of an ideal AND behavior with several variables, an adequate adjustment of the correct place of the DNA binding sites O_{u_i} may be difficult to obtain.

Both issues treated in the last two paragraphs brings us to the task of reducing the number of variables in each conjunctive clause of the boolean functions f_i^s. The number of minterms $s_{i,j}$ in f_i^s does not pose additional complexity, given that it is possible to use m copies of v_i^s, each one implementing a specific $s_{i,j}$, $1 \leq j \leq m$.

It is worth noting that this discussion was based on one specific realization procedure, which is tailored to bacterial transcriptional control (Buchler et al. [2003]). Eukaryotic transcriptional regulation enhances the realization possibilities and has different restrictions, which are not assessed in this work (Alberts et al. [2008]).

4.2 Simplifying regulatory regions of synthetic genes: problem definition and example

The problem of simplifying the regulatory region of synthetic genes can be formally stated as follows:

Problem statement: *Given a supervisor S that achieves the closed-loop specification required for a gene network modeled by the automaton G^{hi}, find a supervisor S^* such that: (i) $S^*(x) = S(x)$, if $S(x)$ is defined, and (ii) the realization of S^* as a set of synthetic genes leads to updating functions $f_i^s(\mathbf{U})$ that, in the canonical disjunctive normal form, involve the minimum number of variables in each minterm.*

Restriction (i) is intuitive and does not require further explanation. By minimizing the number of variables in each minterm, restriction (ii) requires a reduction in the number of binding sites that must be placed in the promoter region, limiting the aforementioned promoter overcrowding problem. Besides that, restriction (ii) decreases the number of interactions between transcription factors that are needed to induce the expression of a gene, thus facilitating the programming of their regulatory regions.

Consider, then, the example introduced in the last subsection. The supervisor S defined in table 1 already satisfies restriction (ii) for v_3^s (recall that $f_3^s = U_1 \wedge U_2$). But the same is not true for v_1^s, whose updating function f_1^s has three variables in the minterm $\overline{U_1} \wedge U_2 \wedge \overline{U_3}$.

Let us define a S^* that satisfies $S^*(x) = S(x)$ for $x \in \{010, 101, 111, 110\}$ and that, also, verifies $S^*(011) = \{+u_1\}$. Such supervisor does not violate restriction (i) and, additionally, allows to simplify the regulatory region of v_1^s as follows: $f_1^s = (\overline{U_1} \wedge U_2 \wedge \overline{U_3}) \vee (\overline{U_1} \wedge U_2 \wedge U_3) = \overline{U_1} \wedge U_2$, because $S^*(011)$ requires that v_1^s be also activated at 011. Hence, the new S^* satisfies both restrictions (i) and (ii) of the problem statement.

Interestingly, the new supervisor S^* has a positive side-effect on the closed-loop behavior of the system when compared to S. If, for any modeling error, the system is unexpectedly driven to 011 from the initial state 010, for instance, the lack of a control input $S(010)$ cannot prevent the system from reaching the attractor 001 and, consequently, remaining there. With S^*, however, the system is brought back to the 'route' 010-110-111-101,

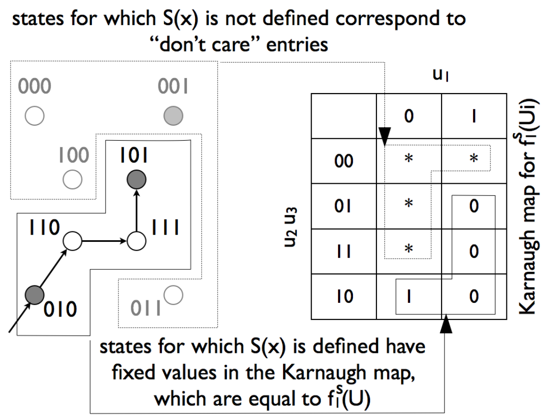

states for which S(x) is not defined correspond to "don't care" entries
000
001
100
101
110
111
010
011
u1
0
1
u2 u3
00
01
11
10
*
*
*
*
0
0
1
0
Karnaugh map for f1s(Ui)
states for which S(x) is defined have fixed values in the Karnaugh map, which are equal to f1s(U)

Fig. 6. Schematic view of the procedure adopted to devise the supervisor S^*, simplifying the regulatory region of the synthetic genes that induce the desired closed-loop behavior for the native gene network. In the figure, a Karnaugh map is shown for the simplification of f_1^s only. One map for each f_i^s to be simplified must be built.

because $S^*(011)$ forces a transition from 011 to 111. So S^* enhances, in this sense, the robustness of the closed-loop system, besides leading to a less complex realization of the regulatory region in v_1^s.

As a last remark, observe that if we had chosen $S^*(000) = \{+u_1\}$, instead of $S^*(011) = \{+u_1\}$, restriction (ii) would be equally satisfied. But the robustness side-effect of this new S^* would be negligible, given that the supervisor would not avoid the system remaining indefinitely in the cycle 000-100.

4.3 Simplifying regulatory regions of synthetic genes: general solution

The procedure explained in the previous subsection to devise S^* can be made more schematic, as shown in figure 6, laying the foundations for a general way of finding S^*.

A Karnaugh map may be used to search for S^*, if it exists. Suppose that a given f_i^s does not satisfy restriction (ii). A Karnaugh map is then built, with the variables U_j that appear in f_i^s. The entries that correspond to states for which $S(x)$ is defined have a fixed value in the map (they are equal to 0 in case $f_i^s = 0$, and 1 if $f_i^s = 1$). On the other side, entries that are associated with states for which $S(x)$ is not defined (that is, states "outside" the closed-loop trajectory) assume the "don't care" symbol in the map. The rules pertaining to the operation of a Karnaugh map (Karnaugh [1953]) are then used to find the simplest expression for f_i^s, substituting, for this purpose, the "don't care" symbols by 1s when needed.

The explanation in the previous paragraph is organized in Algorithm 1, which receives G^{hi} and S as inputs and returns an S^* that satisfies both restrictions (i) and (ii).

Observe that, to the best of our knowledge, the use of Karnaugh maps is convenient to boolean functions f_i^s with up to six variables (Karnaugh [1953]), limiting, in this way, the applicability of the procedure to gene networks of moderate size. Nevertheless, other algorithms available in the literature for simplifying boolean functions can be used instead, as the Quine-McCluskey algorithm (Quine [1952]) and other computational methods based on it.

Algorithm 1. Pseudo-algorithm to find the supervisor S^* that leads to regulatory regions of synthetic genes that satisfy restrictions (i) and (ii) of the problem statement.

```
For each $x \in X$ s.t. $S(x)$ is defined, determine which $v_i^s$ must be activated;
Based on this result, write all $f_i^s$ in the disjunctive normal form;
Simplify the boolean functions $f_i^s$;
for all $f_i^s(\mathbf{U})$ that do not satisfy restriction (ii) do
    Build a Karnaugh map with the variables in $f_i^s(\mathbf{U})$;
    for all $x$ s.t. $S(x)$ is defined do
        Set the entries that make $f_i^s(U) = 1$ to 1;
        Set the other entries to 0;
    end for
    Set all other entries to the "don't care" symbol '*';
    Simplify the expression for $f_i^s$ using the map rules;
end for
Return the simplified $f_i^s$;
```

5. CONCLUSION AND FINAL REMARKS

In this paper, we presented a method that allows to, given a supervisor S that controls the behavior of a native gene network, devise a new supervisor S^* that (a) induces the same closed-loop behavior as S, and (b) is realized by means of synthetic genes that compute simpler boolean functions than those required by S. Our method is based on the simplification of boolean functions that possess the so-called "don't care" conditions (i.e. states at which the function is not defined), a concept that is largely employed in the design of digital circuits. Therefore, our approach can profit from the algorithms already derived in this area. Specifically, we illustrated the ideas with the help of Karnaugh maps, a widely used visual tool.

We also showed, with an example, that the simplification process may result in supervisors S^* that are more robust with regard to modeling errors, in the sense that S^* can lead the system to the target state, even if the plant makes a transition to states that were not expected (and for which S was not defined). This idea, though, was not formalized and a more detailed investigation about the coupling between control robustness and supervisor simplification remains a topic for further research.

Simplifying regulatory regions of synthetic genes is an important step towards the application of our ideas to real problems. Nevertheless, this is not the only simplification that is needed. The supervisor design must take into account a tradeoff between (a) simplicity of the regulatory region and (b) the additional metabolic load that is imposed over the cell by the inserted synthetic genes, given that these extra genes employ valuable cellular resources (cell machinery and energy, for instance) to change the native gene network behavior. So if the controller is too "greedy", in terms of cellular resources, the cell being controlled may lose its viability to grow and replicate. Solving such multi-criteria optimization problem to find a practical supervisor is thus a subject for future research.

At last, we explored only one possible strategy to realize a supervisor as a set of synthetic genes, assuming specific features of bacterial transcriptional control. We plan to extend our work to encompass other possibilities and to include restrictions that arise in eukaryotes as well.

ACKNOWLEDGEMENTS

José E. R. Cury would like to thank the Brazilian National Council for Scientific and Technological Development, CNPq, for the financial support (senior grant 302628/2010-0).

REFERENCES

Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular biology of the cell*. Garland Science, New York, 2008.

Ernestro Andrianantoandro, Subhayu Basy, David K. Karig, and Ron Weiss. Synthetic biology: new engineering rules for an emerging discipline. *Molecular Systems Biology*, 2:1–14, 2006.

Sangram Bagh, Mahuya Mandal, and David McMillen. An active intracellular device to prevent lethal disease outcomes in virus-infected bacterial cells. *Biotechnology and Bioengineering*, 108:645–654, 2011.

Fabio L. Baldissera and José E. R. Cury. Application of supervisory control theory to guide cellular dynamics. In *Proceedings of WODES 2012*, pages 384–389, Guadalajara, Mexico, 2012.

Yaakov Benenson. Biomolecular computing systems: principles, progress and potential. *Nature Reviews Genetics*, 13:455–468, 2012.

John Blazeck, Rishi Carg, Ben Reed, and Hal S. Alper. Controlling promoter strength and regulation in Saccharomyces cerevisiae using synthetic hybrid promoters. *Biotechnology and Bioengineering*, 109:1–12, 2012.

Stefan Bornholdt. Less is more in modeling large genetic networks. *Science*, 310:449–451, 2005.

B. A. Brandin and W.M. Wonham. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 39:329–342, 1994.

Y. Brave and M. Heymann. On stabilization of discrete-event processes. In *Proceeding of the 28th IEEE Conference on Decision and Control*, pages 2737–2742, Tampa, FL, 1989.

Yitzhak Brave and Michael Heymann. On optimal attraction in discrete-event processes. *Information Sciences*, 67:245 – 276, 1993.

Nicolas Buchler, Ulrich Gerland, and Terence Hwa. On schemes of combinatorial transcription logic. *PNAS*, 100:5136–5141, 2003.

Christos G. Cassandras and Stéphane Lafortune. *Introduction to discrete event systems*. Springer, New York, 2008.

Madalena Chaves and Jean-Luc Gouzé. Exact control of genetic networks in a qualitative framework: the bistable switch example. *Automatica*, 47:1105–1112, 2011.

José E.R. Cury and Fabio L. Baldissera. Systems biology, synthetic biology and control theory: A promising golden braid. *Annual Reviews in Control*, 37(1):57 – 67, 2013.

Aniruddha Datta, Ranadip Pal, Ashish Choudhary, and Edward Dougherty. Control approaches for probabilistic gene regulatory networks. *IEEE Signal Processing Magazine*, 24:54–63, 2007.

Hila Gingold and Yitzhak Pilpel. Determinants of translation efficiency and accuracy. *Molecular Systems Biology*, 7:1–13, 2011.

C. H. Golaszewski and P.J. Ramadge. Control of discrete event processes with forced events. In *Proceedings of the 26th Conference on Decision and Control*, pages 247–251, 1987.

Guy Karlebach and Ron Shamir. Modeling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9:770–780, 2008.

M. Karnaugh. The map method for synthesis of combinatorial logic circuits. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 72:593–599, 1953.

Hiroaki Kitano. Systems biology: a brief overview. *Science*, 295:1662–1664, 2002.

Ratnesh Kumar, Vijay Garg, and Steven Marcus. Language stability and stabilizability of discrete event dynamical systems. *SIAM J. Control Optm.*, 35:1294–1320, 1993.

Daniel Machado, Rafael S. Costa, Miguel Rocha, Eugénio C. Ferreira, Bruce Tidor, and Isabel Rocha. Modeling formalisms in systems biology. *AMB Express*, 1:1–14, 2011.

Filippo Menolascina, Mario di Bernardo, and Diego di Bernardo. Analysis, design and implementation of a novel scheme for in-vivo control of synthetic gene regulatory network. *Automatica*, 47:1265–1270, 2011.

Andreas Milias-Argeitis, Sean Summers, and John Lygeros. In silico feedback for in vivo regulation of a gene expression system. *Nature Biotechnology*, 29:1114–1116, 2011.

W. V. Quine. The problem of simplifying truth functions. *The American Mathematical Monthly*, 59:521–521, 1952.

P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77:81–98, 1989.

J. H. van Schuppen. Realization and control problems for biochemical reaction networks. In *Proceedings 11th International IEEE Conference on Methods and Models in Automation and Robotics*, Grenoble, France, 2005.

Olaf Wolkenhauer. Systems biology: the reincarnation of systems theory applied in biology? *Briefings in Bioinformatics*, 2:258–270, 2001.