Introducing continuous time in discrete models of gene regulatory networks.

Jean-Paul Comet[1] and Gilles Bernot[1]

[1] I3S Laboratory, UMR 6070 CNRS and University of Nice Sophia Antipolis, 2000 route des Lucioles, B.P. 121, F-06903 Sophia Antipolis CEDEX

Abstract

It is becoming a routine task to build models of increasing complexity about a given gene network. While available data on the connectivity between elements of the network are more and more numerous, the kinetic data of the associated interactions remain difficult to interprete in order to identify the strength of the gene activations or inhibitions. This parameter identification problem constitutes the cornerstone of the modelling processes. In this article, we show that some information about the elapsed time that takes a trajectory between two points (and that can be experimentally measured) can be of great interest for constraining the parameters of the model. It brings us to set out various frameworks of hybrid modelling (where a model is defined as a combination of a qualitative model with additional continuous variables) in which it is possible to compute elapsed time of trajectories while maintaining powerful automated reasoning capacities. This chapter is an overview of the main formal frameworks able to treat activation or inhibition delays between genes.

1 Introduction

Computational modelling of gene regulatory networks aims at deep understanding of how their components are controlled, thus allowing the prediction of a set of non-obvious behaviours that can be experimentally tested. Unfortunately, while available data on the interaction graph between genes are more and more numerous, the kinetic data allowing us to identify the sensible parameter values are difficult to obtain experimentally and they require many indirect reasonings. This parameter identification problem constitutes the cornerstone of the modelling activities. More precise the available information about the dynamics of the system, more precise can be the model. But precision is not the main criterion. If the precision of the model is higher than the one of the knowledge of the biological system, the precision given by computer simulations is only a consequence of an arbitrary choice of parameter values. Qualitative models where parameters are easier to identify constitute the good compromise.

This comment motivated some researchers to develop methods where this identification problem is tractable. In particular René Thomas' discrete modelling [26] of gene regulatory networks (GRN) is a well-known approach to

study the dynamics resulting from a set of interacting genes. It deals with some *discrete* parameters that reflect the possible targets of trajectories. Those parameters are *a priori* unknown, but they can generally be deduced from a well-chosen set of biologically observed trajectories.

Besides, it neglects the time delay necessary for a gene to pass from one level of expression to another one, whereas information on the time necessary for the system to go from one state to another one is often experimentally available. For example, time used by the system to cover a whole turn of a periodic trajectory (*e.g.* circadian cycle) is often available. Time can also be an abstract time such as the current state of accomplishment within a phase (*e.g.* cellular cycle where "time" is connected to the measure of the mass of the cell). Such an information is not used to face up to the parameter identification problem in the "standard" Thomas' framework without delays. Such kind of information motivates several researchers to propose formal frameworks where time is explicit.

Hybrid extensions of the discrete approach of R. Thomas make time explicit: New parameters, *i.e.* delays mandatory for a gene to go from a discrete abstract level to another one, allow the determination of time along a trajectory. Hybrid modelling frameworks preserve powerful computer-aided reasoning capabilities. Adding delays, the identification problem is more difficult because of the increased number of parameters. Nonetheless computer is able to reject a large class of parameter values.

So, hybrid models (where the levels of expression of each gene remain abstracted into a finite number of possible values but where the delays elapsed inside each discrete level are continuous real numbers) seem to be the best trade-off between *precision* and *automated reasoning* capabilities :

- Differential equations give a full continuous precision both on the concentration level of the gene products and on the time along a trajectory, but parameter values are almost impossible to identify precisely with respect to the experimental and measurement capabilities in biology, and computers are unable to perform proofs on these models, they only perform simulations.

- The discrete approach (so called "logical approach") of René Thomas provides an easy way to identify, exhaustively and using computer proofs, the sensible parameter values, but the discrete models give rise to some trajectories which cannot be observed biologically because the *in vivo* delays make them impossible.

- *In vivo* cell models are somehow "in between" the differential equation models and the discrete models:

  - The number of molecules produced by a gene is finite and it can sometimes be very low, thus, the continuous differential models induce an abuse of precision (which can exhibit some limit behaviours that do not exist *in vivo*)

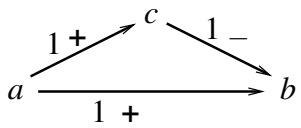


Figure 1: The incoherent type 1 feedforward loop (I1-FFL)

– The number of molecules produced by a gene is most of the time much higher than the number of discrete levels in the Thomas' models, thus, discrete modelling is a rough approximation.

It appears to be possible to define adequate hybrid frameworks for the modelling of gene networks, but the task is not so easy. Many obstacles have been encountered by us and our colleagues. This chapter is an overview of the main techniques that have been proposed; it shows the main obstacles and gives a picture of the current state of the art in hybrid modelling of gene networks.

In this chapter we first present in Section 2 the basic discrete modelling framework of gene regulatory networks without delays due to R. Thomas and an extension based on formal methods from computer science which allow the automation of the search of parameter values from experimentally known behaviours. Section 3 focuses on the now classical framework of piecewise linear differential equations and their relationships with discrete models. In Section 4 we present the first hybrid approach due to R. Thomas consisting in completing a discrete model by a set of clocks which measure the time necessary to pass through a transition. Another dual approach has been also proposed by Bockmayr and Siebert [23] and is sketched in Section 5. An alternative hybrid framework is then proposed in Section 6 in which the delays introduced in the hybrid models are coherent with the underlying piecewise linear differential equation systems. Finaly we discuss in Section 7 some parameter identification issues when considering hybrid models with delays.

In order to evaluate the consequences of introducing delays into the modelling framework, we consider in the sequel some examples all based on a particular graph pattern [22]: the feedforward loop - incoherent type 1 (I1-FFL), see Figure 1, which is one of the most common network motifs. The dynamics of such a pattern of interaction graph have been largely studied [20]. The feed-forward loop is composed of a transcription factor a that regulates a second transcription factor c and both a and c regulate a gene b. So, a regulates b *via* two paths. When the signs of both paths (that is the product of signs of each interaction along the path) are not equal, the feed-forward is said incoherent [3].

Intuitivelly, it is straightforward to comprehend that when a is switched on, both b and c are subject to change. If the delay mandatory for b to come on is less than the one associated with c, then one can observe a transitory presence of b before the presence of c inhibits b. We will also see that the situation is complex when a oscillates.

2 Discrete modelling of gene regulatory networks

René Thomas has introduced in the 70's a qualitative approach [26] in order to model gene networks and to predict their dynamics. Three main ideas constitute the foundation of this qualitative approach.

2.1 First idea

The criterion to abstract the qualitative concentration levels of a gene product is the number of other genes on which it acts in the network.

Such a criterion is based on the fact that, when a gene acts on another one, the curve that represents the production rate of the target gene with respect to the concentration level of the source gene is a sigmoid. For example, assume that x is a gene that activates a gene y and inhibits a gene z as in Figure 2, then the corresponding sigmoids allow us to consider two thresholds inside the interval of all possible real concentrations levels of the x product: τ_1 and τ_2. These two thresholds delimit three intervals within which the gene x behaves uniformly. Each interval is conventionaly identified by an integer, which is the number of genes on which x has an action. If we know the order between the thresholds, then we can additionally label the action of x on a gene by the number of the first interval that activates this action (lower left drawing of Figure 2).

So, the interaction graph contains *variables* that mostly represent genes (sometimes they represent abstract phenotypes or environmental conditions) and it contains *edges* between variables that can be labelled by a sign ($+$ for activation, $-$ for inhibition) and by an integer threshold.

2.2 Second idea

At a given global state of the network, the concentration toward which a gene product tries to go depends only on the inventory of the activations and inhibitions that act on this gene.

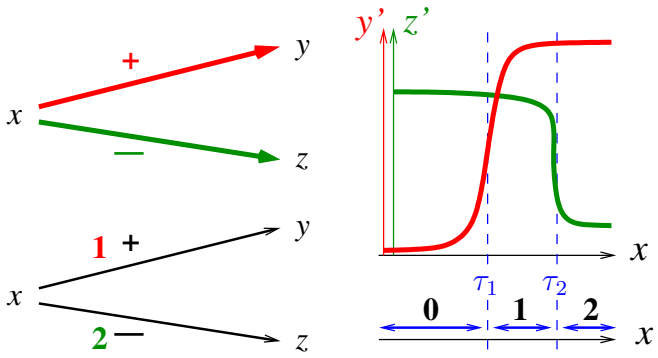
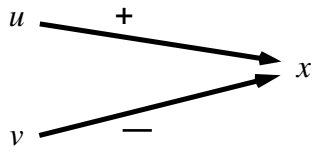


Figure 2: Multivalued regulatory graph



No help for x: K_x
The presence of u helps x: $K_{x,u}$
The absence of v helps x: $K_{x,v}$
Both help x: $K_{x,uv}$

Figure 3: Parameters

For example, if u is an activator of x and v is an inhibitor of x and if we assume that x has no other activator or inhibitor in the network, as in Figure 3, then four cases have to be considered:

K_x is the number of the interval toward which x tends to go when it has no help at all from the considered network. It means that u does not activate x, thus the current state of u is strictly less than the threshold of $(u \to x)$, and v inhibits x, thus the current state of v is greater or equal to the threshold of $(v \to x)$.

$K_{x,u}$ is the number of the interval toward which x tends to go when it benefits only from the help of u. It means that the current state of u is greater or equal to the threshold of $(u \to x)$, and the current state of v is greater or equal to the threshold of $(v \to x)$.

$K_{x,v}$ is the number of the interval toward which x tends to go when it benefits only from the help of v. It means that the current state of u is strictly less than the threshold of $(u \to x)$, and the current state of v is strictly less than the threshold of $(v \to x)$.

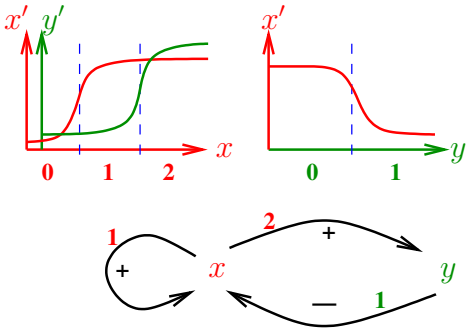
$K_{x,uv}$ is the number of the interval toward which x tends to go when it benefits both from the help of u and from the help of v. It means that the current state of u is greater or equal to the threshold of $(u \to x)$, and the current state of v is strictly less than the threshold of $(v \to x)$.

It is also possible that a gene influences itself in a given network, nevertheless auto-regulations do not change the approach at all. For each state, the parameters $K_{...}$ define the vector state toward which the system tends to go. Figure 4 gives a small example of gene network where we have arbitrarily chosen the parameters as follows: $K_x = 0$, $K_{x,x} = K_{x,y} = K_{x,xy} = 2$ and $K_y = K_{y,x} = 1$.

2.3 Third idea

The variables of a network are asynchronously updated toward their parameters, crossing at most one threshold.

The asynchronous updating is motivated by the fact that a threshold represents a very thin region of the real concentration space for each variable, thus, the probability that several variables cross their thresholds exactly at the same time is negligible. Consequently, when the network is in a state such that several variables can change (*i.e.*, such that several variables have



(x,y)	*Focal Point*
(0,0)	$(K_{x,y},\ K_y) = (2,1)$
(0,1)	$(K_x,\ K_y) = (0,1)$
(1,0)	$(K_{x,xy},\ K_y) = (2,1)$
(1,1)	$(K_{x,x},\ K_y) = (2,1)$
(2,0)	$(K_{x,xy}, K_{y,x}) = (2,1)$
(2,1)	$(K_{x,x},\ K_{y,x}) = (2,1)$

Figure 4: Table of focal points

a concentration level belonging to an interval which is different from the interval pointed by the current parameter $K_{...}$), there are as many possible next states as such variables. From such a state, the system can choose to modify any one of these variables.

For example, Figure 5 shows the state graph extracted from the network given in Figure 4. The left hand side of the figure shows what would happen if we followed a naive synchronous updating that would reflect the table of focal points: the situation would be biologically incredible for two reasons. The first reason, as already mentioned, is that x and y would be updated at the same time for example from the state (1,0). The second reason is that from the state (0,0), the variable x would cross two thresholds, which is contradictory with the fact that we are modelling a continuous change of concentration levels. The right hand side of the figure provides the correct abstract behaviours of the system. It shows in particular that it is possible to reach the stable state (0,1) from the initial state (0,0).

2.4 The boolean framework

The boolean framework was the first framework introduced by René Thomas. It modifies the first idea as follows: *the product of a gene can be "present" or "not present" in the cell*. It means that there is only one threshold for each gene; the two other ideas (the parameters K and the asynchronous state graph) remain unmodified.

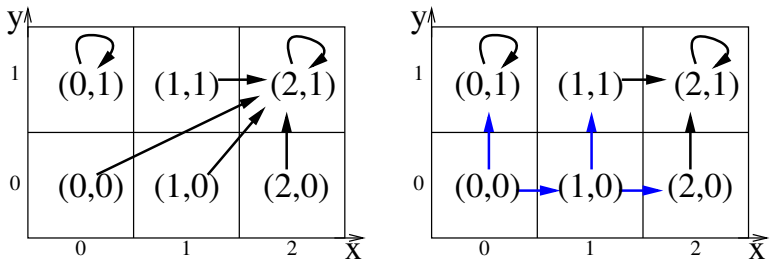
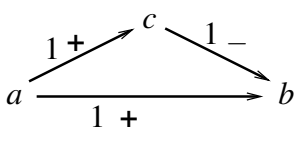


Figure 5: Synchronous and asynchronous state graphs



$$\begin{array}{llll} K_b & = 0 & K_c & = 0 \\ K_{b,a} & = 0 & K_{c,a} & = 1 \\ K_{b,c} & = 0 & & \\ K_{b,ac} & = 1 & & \end{array}$$

Figure 6: Boolean feedforward

Let us consider the "type 1 incoherent feedforward loop" introduced in Figure 1. As we are in the boolean framework, every threshold is equal to 1. Moreover, if we want that b needs the presence of its activator a *and* the absence of its inhibitor c to be synthesized, then a unique choice is possible to make all the interactions of the graph functional, see the parameter valuation in Figure 6. Remember that, in the $K_{b\ldots}$ parameters, the subscript c means that c does not pass the threshold, as it is an inhibitor of b. Lastly, the variable a being the entry point of the feedforward pattern, we do not consider K_a yet.

The question that we will address on this example all along the article is the following: *what shall be the behaviour of* b *in response to the input signal offered by* a *?*

Obviously, if a is equal to 0 for a sufficiently long time, both b and c will also be equal to 0, because b and c need a as a resource in order to reach the state 1; see Figure 7. Let us assume that the signal a goes from 0 to 1. Then,

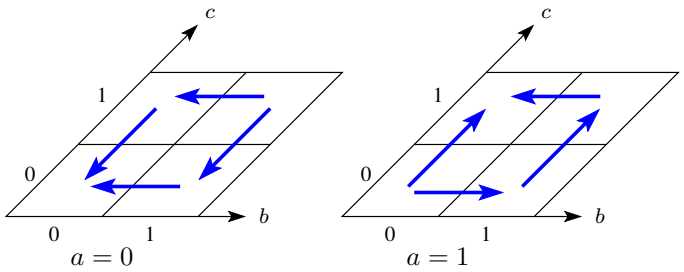


Figure 7: Asynchronous state graph for type 1 incoherent feedforward loop

the current state will move to $(a = 1, b = 0, c = 0)$: the square situated at the lower left corner of the plan $a = 1$ of Figure 7. The new stable state is $b = 0$ and $c = 1$ but, due to the asynchronous semantics, there are two different paths from the current state: either we go directly to the stable state and b remains constantly equal to 0, or we follow the other path where b is transitorily equal to 1, before being inhibited by c.

Under which conditions will b always signal the presence of a *via* a transitory production ? May-be the conjunction of resources for the variable b is not optimal, e.g., would a disjunction be better ? or any other values for the $K_{\ldots}$ parameters ?

This is more generally the usual question of identification of the parameter values, according to some biologically known behaviours or some hypothetical behaviours. Here, the example would be small enough to enumerate all

the possible parameter values, to generate the state graphs and study for each of them the answer of b. It is of course not the case when addressing real size gene networks and so, formal methods from computer science are required to perform computer-aided identification of parameters.

2.5 Temporal logic and automatic model checking

Temporal logics are languages that allow us to formalize biologically known behaviours or hypothetical behaviours in such a way that computers can automatically check if a model exhibits those behaviours or not. The building blocks of a temporal logic are atoms, connectives and temporal modalities. Let us here consider the Computation Tree Logic [12, 17], CTL for short, which is the most common temporal logic:

- Atoms in our case are simple statements about the current state of a variable of the network. For example equalities (*e.g.*, $x = 2$) or inequalities (e.g., $x \leq 1$ or $y > 1$).
- Connectives are the standard connectives: negation (*e.g.*, $\neg(x = 0)$ is the negation of the atom $x = 0$), conjunction (*e.g.*, $(x = 0) \wedge (y > 1)$), disjunction (*e.g.*, $(x = 0) \vee (y > 1)$), implication (*e.g.*, $(x = 0) \Rightarrow (y > 1)$), and so on.
- Temporal modalities are combinations of two types of information:
  - Quantifiers: a formula can be checked with respect to all possible choices of paths in the asynchronous state graph (universal quantifier, denoted by A), or one can check if it exists at least one path choice such that the formula is satisfied (existential quantifier, denoted by E).
  - Discrete time elapsing: a formula can be checked at the next state (letter X), in some future state which is not necessarily the next one (letter F), in all future states (letter G) and a formula can be checked until another formula becomes satisfied in the future (letter U).

In short:

first character	second character
A = for **A**ll path choices	X = ne**X**t state
	F = for some **F**uture state
E = there **E**xists a choice	G = for all future states (**G**lobally)
	U = **U**ntil

For example, the formula $((x = 0) \wedge (y > 0)) \Longrightarrow A[(x = 0)U(y = 0)]$ means that, starting from an initial state where $x = 0$ and y is strictly positive, there will be a state in the future such that $y = 0$ and meanwhile, x will remain equal to 0, whatever the choice of path. More generally, Figure 8 summarizes the CTL semantics with the following conventions: we start from an arbitrary initial state that constitutes the root of the tree; a blue arrow means that φ

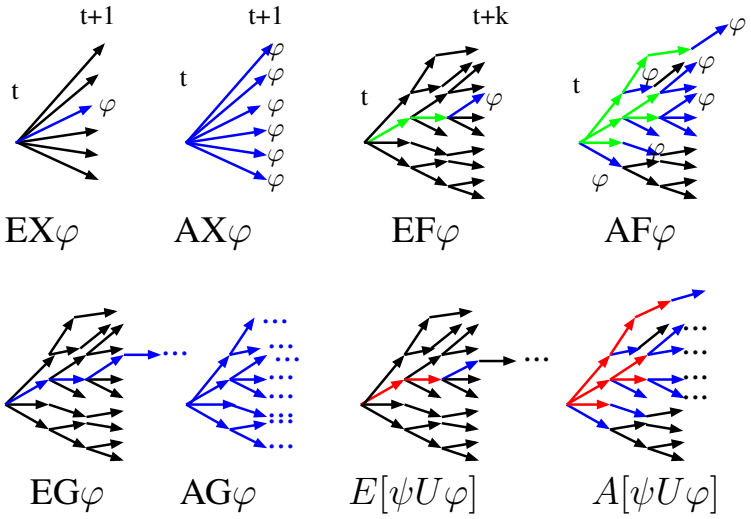


Figure 8: CTL modalities

becomes true in the target of the arrow; a green arrow means that φ is not satisfied in the target of the arrow; a red arrow means that ψ is satisfied both in the source and in the target of the arrow.

One of the main advantages of CTL is that there are very efficient model checkers, see for example [9]. A model checker is an algorithm that takes as inputs a CTL formula and a state graph, and furnishes as output the subset of states that satisfy the formula.

Model checking can be used to identify the parameters that are compatible with the known or hypothetical behaviours [7]. SMBioNet is a software platform where we can enter the influence graph between genes and where we can enter CTL formulas that describe the known behaviours: it automatically computes all the sets of parameter values that are compatible with both the graph and the behavioural properties. Technically, SMBioNet generates all the possible state graphs and performs model checking. Then, a model is proposed if and only if all its states satisfy all the behavioural properties.

For our feedforward example, the transitory activation of the gene b can be formalized in CTL as follows:

$$(b = 0 \wedge c = 0 \wedge AG(a = 1)) \Longrightarrow AF(b = 1 \wedge AXAG(b = 0))$$

It means that if the signal a becomes active when b and c are inactive, then b will become active in the future, and then it will become inactive.

We have also assumed that a is able to control c:

$$(a = 1 \wedge c = 0) \Longrightarrow EX(c = 1)$$
$$(a = 0 \wedge c = 1) \Longrightarrow EX(c = 0)$$

It means that when $a = 1$ (resp. $a = 0$) c can increase (resp. decrease). We use EX and not AX because the asynchrony can allow another variable to cross a threshold before c.

When submitting the formula to SMBioNet, we discover that there is no parameter values such that b always signals the switch of a by a transitory change of value: the direct path from $(a = 1, b = 0, c = 0)$ to $(a = 1, b = 0, c = 1)$ seems unavoidable whatever the values of the parameters.

Nonetheless, if we assume for instance that the *delay* for a gene to act on another gene is identical for all interactions in Figure 6, then a would start both the expression of b and c almost at the same time, and only after another delay, c will switch b to 0. So, b will always signal the switch of a.

The paradox comes from the fact that the standard Thomas' approach does not take delays into account. The parameters $K_{...}$ control only the functionality of combined interactions, not the delays. So, the asynchrony of variable updates always gives the possibility for a to activate c and then for c to inhibit b before the direct activation of b by a will take place. More generally, it may also depend on the real initial state inside the square $(a = 1, b = 0, c = 0)$ and it may also depend on the relative production speeds of b and c. Indeed the standard Thomas' framework has a rough notion of *time*: it is reduced to the random scheduling of variable changes. This motivates the introduction of delays into the modelling framework.

2.6 Logic programming with constraints

Before discussing the different ways to introduce delays into the Thomas' framework, let us mention the importance of constraint solving for the parameter identification problem. The current platform SMBioNet exhaustively generates the possible state graphs and checks on them the temporal properties. When time delays will be introduced, they will of course constitute additional parameter values which will need to be identified as well. Delays shall be real numbers because time passes continuously, and consequently, an exhaustive enumeration of all the possible behaviours will become impossible. Temporal properties will induce constraints on both the Thomas' parameters and the time delay parameters, in such a way that the set of solutions will involve intervals of real time delays containing an infinity of points.

In the standard discrete framework of R. Thomas, L. Trilling has already proposed to use logic programming with constraints in order to identify the $K_{...}$ values [13]. More precisely, the method extracts all the parameter values that make possible a given set of observed paths in the state graph. The method has also been extended and implemented by F. Corblin [10] in the same research team, and the results are impressive. Provided that the temporal properties under consideration can be expressed *via* a finite number of paths of fixed length, a few seconds of computing time are needed for problems where SMBioNet needs several hours.

The idea is to specify, in the PROLOG language, the Thomas' asynchronous construction of the state graph, according to symbolic representations of the $K_{...}$ parameters. Then, by specifying that a given path exists in the state graph, PROLOG will generate the constraints on the parameters that permit

each transition of the path. Lastly, constraint solving algorithms try to exhibit parameter values or to prove inconsistencies.

As an example, let us consider the path $(b = 0, c = 0) \rightarrow (b = 1, c = 0) \rightarrow (b = 1, c = 1) \rightarrow (b = 0, c = 1)$ in the plan $a = 1$ as in Figure 7. The first transition of the path generates the constraint $(K_{b,ac} > 0)$ because the variable b goes from 0 to 1 when a and (the absence of) c are resources of b. The two other transitions generate similarly $(K_{c,a} > 0)$ and $(K_{b,a} < 1)$ respectively. In order to ensure that b will always signal the presence of a *via* a transitory switching on, one has to negate the existence of the path $(b = 0, c = 0) \rightarrow (b = 0, c = 1)$, which generates the negation of $(K_{c,a} > 0)$. Lastly, constraint solving will trivially prove that the resulting global set of constraints is inconsistent. The same computations must be done for all the paths that exhibit a transitory production of b.

Of course, both constraint programming methods and model checking methods give the same result and they both raise the same "delay paradox" mentioned previously. The simple chronologic notion of random scheduling of variable changes is not sufficient; we need an explicit notion of chronometric delays in the modelling framework.

3 Piecewise Linear Differential Equations

Since chronometric information is of great importance in the dynamics of the modelled system, it seems natural to come back to the framework of differential equations because differential systems make the time explicit. Moreover in this modelling framework, the trajectories are deterministic: from an initial state, the whole trajectory can be computed.

Nevertheless, parameters of the differential equations are generally not known and have to be determined. If we want to use knowledge on the time that takes a particular trajectory between two points, in order to determine unknown parameters, one has to explicite the relationship between elapsed time along a trajectory and parameters. Thus the differential equation system has to be solved. Generally, if the differential system has no particular shape, the symbolic solving of the differential system is not possible and the large number of variables makes appear additional difficulties. The computer tools which are useful for simulations of such systems are nevertheless not well adapted for this difficult task.

3.1 Piecewise Linear Differential Equations

To simplify this task, we can restrict the form of the differential system. Snoussi [24] proposed to construct a piecewise linear differential equation system: with each qualitative situation (that is when interactions does not change) is associated a differential system which is easy to solve symbolically. The way to construct such a system of differential equations can be sketched as follows:

- With each node of the interaction graph is associated a variable of the differential equation. This variable represents the concentration of the associated protein.
- Each variable has a particular degradation rate. The degradation is supposed to be proportional to the concentration of the protein (greater the concentration, greatest the degradation).
- Each variable has a synthesis rate which depends on the activity of its regulators (greater the number of activators, greatest the synthesis rate).
- Each predecessor of a variable (in the interaction graph) has an influence on the synthesis rate of the considered variable: if it is an activator, the synthesis rate is increased when the regulator has a concentration greater than the threshold associated with the interaction; if it is an inhibition, the synthesis rate is increased when the regulator has a concentration smaller than the threshold, see Figure 9.

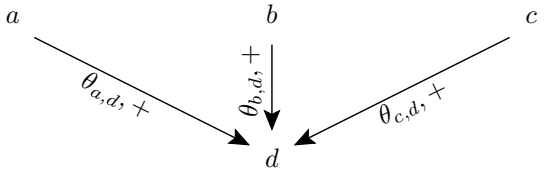


Figure 9: Example of a gene regulated by two activators and by one inhibitor.

The previous outline leads to the following differential equation system:

$$\frac{d}{dt}x_i = \left(k_0^i + \sum_{j \in A(i)} k_j^i \times \mathbb{1}_{[x_j > \theta_{j,i}]} + \sum_{j \in I(i)} k_j^i \times \mathbb{1}_{[x_j < \theta_{j,i}]} \right) - \gamma_i x_i$$

where $A(i)$ (resp. $I(i)$) is the set of activators (resp. inhibitors) of i, and $\mathbb{1}_{[\text{condition}]}$ is equal to 1 if the condition is satisfied and equal to 0 otherwise[1]. The first term is the synthesis rate which can be decomposed into three part:

- k_0^i which is the basal synthesis rate,
- the contribution of activators (each activator contributes to the synthesis rate when its concentration is greater than some threshold) and
- the contribution of inhibitors (each inhibitor contributes to the synthesis rate when its concentration is less than some threshold).

[1]Let us remark that when a concentration is on a threshold, the contribution of the associated action is not taken into consideration. It would be better to consider that the differential equation is not defined on thresholds: one does not know whether the regulation takes place or not. If one is interested in the precise behaviour of the system on thresholds, one has to embed such a differential equation into the framework of differential inclusions [14]. This work has already been done in the context of gene regulatory networks [11].

Finaly $\gamma_i x_i$ represents the degradation. Such a differential system is called a Piecewise Linear Differential Equation system, PLDE for short.

The previous differential equation system is based on the qualitative contribution of each regulator. Unfortunately, even if the additivity of contributions is not put into question, the contribution of a regulator is not a discontinuous step function. To improve the model, the set fonction $\mathbb{1}_{[x>\theta]}$ (resp. $\mathbb{1}_{[x<\theta]}$) can be replaced by a Hill function:

$$H_+(x) = \frac{x^n}{\theta^n + x^n} \qquad \left(\text{resp. } H_-(x) = \frac{\theta^n}{\theta^n + x^n}\right)$$

where n is the parameter of the Hill function which controls its roughness.

3.2 Coherence between PLDE and discrete models

Such a differential equation system has a deep relationship with the discrete models of Section 2. Let us first remark that the thresholds allow a discretization of the phase space: ranking the thresholds $\{\theta_{j,i} | i \text{ is a possible target of } j\}$ for each variable j allows one to split the concentration space of j into different subdomains numbered from 0 to b_j. The discretization of the continuous phase space is then defined by associating with each concentration state s, the discrete vector characterizing the subdomain of s. Thus the parameter of the discrete model $K_{i,\omega}$ is the discretization of the coordinate i of the equilibrium point of the differential system associated with the situation where ω is the set of regulators contributing to the synthesis rate:

$$\begin{array}{c}\text{i}^{th} \text{ coordinate of the} \\ \text{equilibrium point}\end{array} = \left(\frac{k_0^i + \sum_{j \in A(j) \cap \omega} k_j^i + \sum_{j \in I(j) \cap \omega} k_j^i}{\gamma_i}\right)$$

If each contribution to the synthesis rate $k_{i,\omega}$ is positive and if, for each i and each ω, $K_{i,\omega}$ is equal to the discretization of the i^{th} coordinate of the equilibrium point, then there exists a transition from discrete state s_1 to s_2 if and only if there exists a trajectory of the differential system starting from the domain associated with s_1 and going to the threshold separating this domain from the domain associated with s_2 [24].

3.3 A feedforward loop controled by a positive auto-regulation

To study the behavior of an incoherent type 1 feedforward loop, we first consider that the action of the transcription factor a does not change, that is, that its level of concentration does not cross the threshold of one of its interactions. The simplest way to study such a system is to consider that the transcription factor a is also a regulator of itself, see Figure 10. This positive auto-regulation leads to multi-stationarity [28] of a: if a is present (resp. absent), it remains present (resp. absent). As in Section 2.4 we suppose that b needs the presence of its activator a and the absence of its inhibitor c to

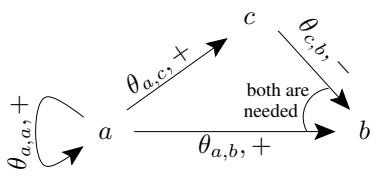


Figure 10: Incoherent type 1 feedforward loop combined with a positive autoregulation of a.

be synthesized. Mathematical modelling suggested that the I1-FFL can show two dynamical features [20] amongst whose a transient pulse of expression of b. To verify this possible behaviour, let us build the corresponding differential equation system:

$$\left\{\begin{array}{lcl} \frac{da}{dt}(t) & = & k_0^a + k_a^a \mathbb{1}_{[a>\theta_{a,a}]} - \gamma_a.a(t) \\ \frac{db}{dt}(t) & = & k_0^b + k_a^b \mathbb{1}_{[a>\theta_{a,b}]} + k_c^b \mathbb{1}_{[c<\theta_{c,b}]} - \gamma_b.b(t) \\ \frac{dc}{dt}(t) & = & k_0^c + k_a^c \mathbb{1}_{[a>\theta_{a,c}]} - \gamma_c.c(t) \end{array}\right.$$

Nevertheless this differentential system does not express that b needs the presence of its activator a *and* the absence of its inhibitor c to be synthesized. It has to be modified to take into account the condition under which the activation of b is effective: $[(a > \theta_{a,b}) \wedge (c < \theta_{c,b})]$:

$$\left\{\begin{array}{lcl} \frac{da}{dt}(t) & = & k_0^a + k_a^a \mathbb{1}_{[a>\theta_{a,a}]} - \gamma_a.a(t) \\ \frac{db}{dt}(t) & = & k_0^b + k_{ac}^b \mathbb{1}_{[(a>\theta_{a,b})\wedge(c<\theta_{c,b})]} - \gamma_b.b(t) \\ \frac{dc}{dt}(t) & = & k_0^c + k_a^c \mathbb{1}_{[a>\theta_{a,c}]} - \gamma_c.c(t) \end{array}\right.$$

where k_{ac}^b is the contribution to the synthesis rate of b due to the presence of a and the absence of c.

The global behaviour of such a system is driven by the values of parameters. For example, for some values of parameters, the protein b is synthesized before the action of the inhibitor takes place, see Figure 11-left, whereas for some other values of parameters, the synthesis of b is not so visible, see Figure 11-right. Let us remark that in both cases, the equilibrium point of variable b when a is present and c absent is then same: $\frac{k_0^b + k_{ac}^b}{\gamma_b} = \frac{40}{2} = \frac{5}{0.25} = 20$. Both models lead to the same discrete model.

Thus the identification of parameters becomes a crucial step also in the PLDE modelling framework, because a variation of parameters can lead to different qualitative behaviours, see Figure 11. Moreover the values of kinetic parameters $k_{\ldots}^{\cdot}$ are mandatory to deduce the sequence of domains the trajectory passes through, which are also necessary to compute the time that takes a trajectory passing through such a sequence of domains.

The first idea to overpass the *parameter identification problem* is to grope for parameters until a set of parameters leads to a behaviour compatible with available information about the trajectories. After having found a valuation of parameters, simulations of the mathematical model are performed (several of

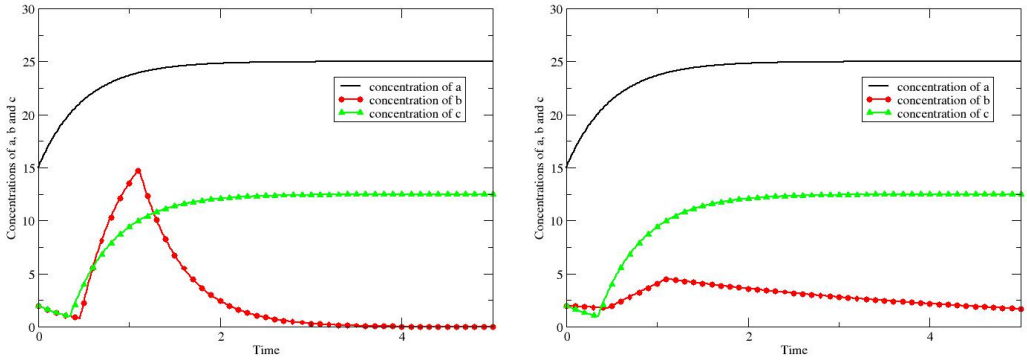


Figure 11: Feedforward loop controled by a positive auto-regulation: according to kinetic parameters, b can be activated before the effect of the inhibition of c or not. Left: $\gamma_b = 2$ and $k^b_{ac} = 40$ Right: $\gamma_b = 0.25$ and $k^b_{ac} = 5$. Other parameters are identical for both simulations: $\theta_{a,a} = 10.0$, $\theta_{a,b} = 21.0$, $\theta_{a,c} = 20.0$, $\theta_{c,b} = 10.0$, $\gamma_a = \gamma_c = 2$, $k^a_0 = k^b_0 = k^c_0 = 0.0$, $k^a_a = 50.0$, $k^c_a = 25$. Initial state is $(15, 2, 2)$.

them under perturbations) in order to evaluate its robustness, that is its ability to maintain its functions against internal and external perturbations [18]. Indeed, since robustness is one of the fundamental characteristics of biological systems and has been demonstrated many times experimentally [19], the evaluation of the robustness of the PLDE model is a indicator of its validity. Nevertheless the evaluation of the robustness of the PLDE model does not validate completely the model.

3.4 A feedforward loop controled by a negative loop

We now study the behavior of the incoherent type 1 feedforward loop when the transcription factor a oscillates. The simplest way to study such a system is to consider the interaction graph made of the incoherent type 1 feedforward loop and of the negative loop $(a \rightleftarrows a')$ containing the transcription factor a, see Figure 12. The negative feedback loop leads to oscillations of a and a'

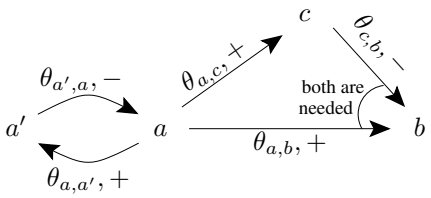


Figure 12: Incoherent type 1 feedforward loop combined with a negative loop.

under some conditions, in such a case, we say that the circuit is *functional*. When the period of oscillation of a and a' is sufficiently small, neither b nor

c is able to switch-on during a unique period. But if degradation rate is also sufficiently weak, several period can lead to the activation of b, c or both b and c. The PLDE model can be easily written:

$$\left\{\begin{array}{lcl} \frac{da}{dt}(t) & = & k_0^a + k_{a'}^a \mathbb{1}_{[a<\theta_{a',a}]} - \gamma_a.a(t) \\ \frac{da'}{dt}(t) & = & k_0^{a'} + k_a^{a'} \mathbb{1}_{[a>\theta_{a,a'}]} - \gamma_{a'}.a'(t) \\ \frac{db}{dt}(t) & = & k_0^b + k_{ac}^b \mathbb{1}_{[(a>\theta_{a,b})\wedge(c<\theta_{c,b})]} - \gamma_b.b(t) \\ \frac{dc}{dt}(t) & = & k_0^c + k_a^c \mathbb{1}_{[a>\theta_{a,c}]} - \gamma_c.c(t) \end{array}\right. \quad (1)$$

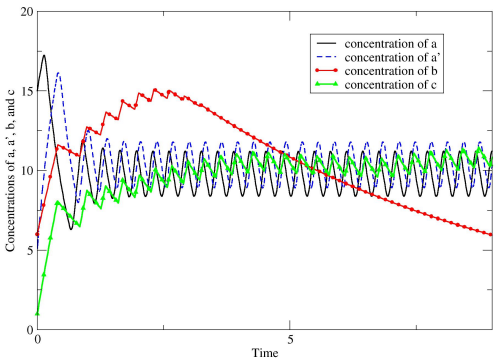
This differential system can lead to subtile behabiors. Let us first suppose that oscillations of a and a' are much faster than the increasing of b and c. Thus the order of activation of genes b and c, is intermittent and several behaviours can be obtained according to relative values of the synthesis and degradation rates:

1	neither accumulation of b nor accumulation of c
2	accumulation of b but no accumulation of c
3	no accumulation of b but accumulation of c
4	accumulation of b and c, but c is activated before b
5	accumulation of b and c, but b is activated before c

Table 1: The different possible behaviours of the feedforward loop controled by a negative loop.

Figure 13 shows the evolution of concentrations of 4 variables for some parameter values. It is clear that this choice of parameter values corresponds to the situation where both variables b and c increase because of the intermittent order due to a: synthesis rate of b (resp. c) when a does activate b (resp. c) is sufficiently high to allow, after a total oscillation period of a, a little accumulation even after the second phase of the cycle when the activation order is off. In other word, during each oscillation cycle of a, the system creates more b than it degrades b. Such an accumulation of b and c are due to low degradation rate but such values are nevertheless credible: the permease in the lactose operon system is known to be degradated very slowly. Moreover, in Figure 13, b increases faster than c. Thus b becomes present but when c becomes greater than the threshold of its action on b, variable b begins to decrease and will no more be activated. Such a behaviour corresponds to the situation 5 of the previous table.

Unfortunately, as said in the introduction, there does not exist an automated method to extract properties of kinetic parameters which have to be fullfiled to allow the system to present any known dynamical property. We then set out in the next section a hybrid framework based on the discrete one which tries to mimics the different behaviours of PLDE systems.



Thresholds:	Degradations:
$\theta_{a,a'} = \theta_{a',a} = 10$	$\gamma_a = 2$
$\theta_{ac} = 10.6$	$\gamma_{a'} = 2$
$\theta_{a,b} = 10.5$	$\gamma_b = 0.15$
$\theta_{c,b} = 9.5$	$\gamma_c = 0.5$
Synthesis rates	Initial state
$k_0^a = k_0^{a'} = k_0^b = k_0^c = 0,$	$a_0 = 15$
$k_{a'}^a = k_a^{a'} = 50$	$a'_0 = 6$
$k_{ac}^b = 15.0$	$b_0 = 2$
$k_a^c = 20$	$c_0 = 2$

Figure 13: Incoherent type 1 feedforward loop combined with a negative loop: the negative loop generates oscillations which allow b and c to accumulate.

4 First hybrid modelling approach due to R. Thomas

To try to automate the parameter identification step for a timed model of a gene regulatory network, it seems natural to propose to build a timed version of the discrete approach since this discrete framework can be viewed as a discretization of the PLDE framework. The refined modelling is based on the use of delays of activation / inhibition to specify which variable is faster affected by a change of its regulators. To be more precise, when an order of activation / inhibition rises, the biological machinery starts to increase or to decrease the corresponding protein concentration, but this action takes time. Thus the differences between the values of delays of activation / inhibition lead to decrease the non-determinism.

4.1 Qualitative states and clocks

This idea dates back to the book of Thomas and d'Ari [27]: with each variable is associated a clock which measures the elapsed time, and each transition needs some delay to be passed over. The simulation of such a model can then be sketched as follows:

1. The initial state is made of a discrete state and a initialisation of clocks (generally each clock is set to 0).

2. According to the current discrete state, the clocks associated with variables whose focal point allows them to evolve (that is whose focal point is placed outside the domain), run simultaneously at the same speed.

3. The next fired discrete transition is given by the clock which first reaches its associated delay. If two delays are equal, that is if two clocks reach their delays at the same time, non-determinism remains and several

discrete transitions can be fired. In such a case, choose at random a possible transition.

4. In the new state, some clocks are reset: the clock which allowed the transition is reset to zero, but also each clock for which the order has changed. For example, if in the previous state, the variable a was subject to an decreasing order, but in the new state, it is subject to an increasing order, its associated clock is reset also to 0.

5. Repeat steps 2, 3 and 4.

4.2 A feedforward loop controled by a positive auto-regulation

Depending of the delays associated with transitions, two behaviours can be simulated: the first one allows the switch-on of variable b, while the second does not allow it. Let us consider the boolean network described in Section 2.4 completed by the auto-regulation of a. The functionality of the auto-regulation of a, which does not allow a to evolve from its initial state, leads to following values of parameters concerning variable a:

$$K_a = 0 \qquad K_{a,a} = 1$$

Other parameters are the same, see Figure 6. Because a is not able to evolve from its initial state, the state graph is the one of Figure 7. The initial state is the boolean state $(a = 1, b = 0, c = 0)$ combined with an initialization of clocks where each clock is set to zero. On one hand, if the delay mandatory to activate c is less than the delay mandatory to activate b, then b will never be switched on because the inhivitor c becomes rapidly effective. On the other hand, if the delay to activate c is greater, c gives b the time to be switched on before becoming an effective inhibitor.

We proposed in [2] a formalisation of such a modelling approach which is based on two types of parameters, $d_v^+(x)$ and $d_v^-(x)$, which represent the time delays required to change the expression level of a variable v from level x to $x + 1$ and from level x to $x - 1$, respectively, as shown in Figure 14. Then, we add to each variable v a continuous clock h_v whose speed at state μ is 1 (when variable v can evolve) or 0 (if it cannot). At a given qualitative state μ, if the concentration of v is increasing (resp. decreasing), then, when h_v reaches $d_v^+(\mu(v))$ (resp. $d_v^-(\mu(v))$), the level of v becomes $\mu(v) + 1$ (resp. $\mu(v) - 1$) and the clock h_v is reset.

The temporal model described above belongs to the class of the so-called *stopwatch automata* [8] which is a specific type of linear hybrid automata [4, 5], LHA for short. LHA are finite state automata augmented with real variables whose values evolve continuously in a discrete state. Whereas the values of the continuous variables can be affected by discrete transitions between discrete states, evolutions of continuous variables are lines inside a discrete state. Linear hybrid automata can be subject to a reachability analysis. However,

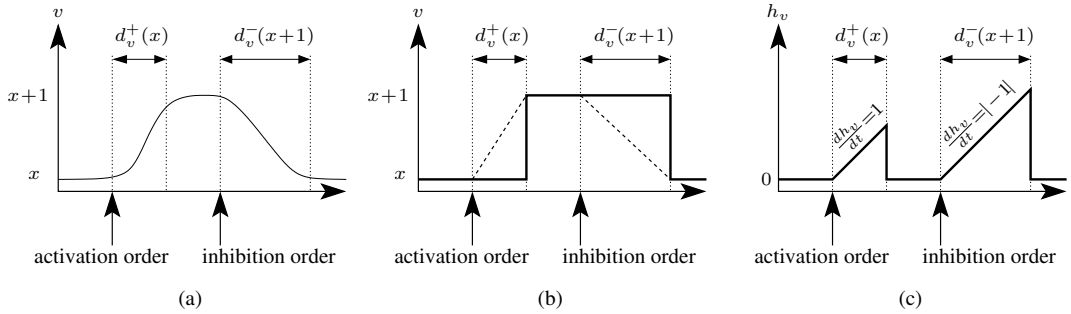


Figure 14: Evolution of a gene's expression (a), its schema in the discrete model (b) and in its extension with time delays (c).

in general, the reachability problem for linear hybrid automata is undecidable [25].

In such a modelling framework, the parameter identification problem still remains the cornerstone of the approach. The determination of discrete parameters (the $K_{v,\omega}$) can be driven by model checking as shown in section 2.5. It then remains to identify the delays. Since time delays are real numbers, it cannot exist any enumeration method (SMBioNet-like) which tries all possible combinations of delay values and retains only those which are coherent with knowledge about the behaviour. One then have to turn to constraints in order to express the conditions under which the known properties are satisfied by the model.

Let us focus on the example of the feedforward loop controled by a positive auto-regulation.

1. Let us suppose that available knowledge about the system allows one to state that when a is on, b is switched on before c. Afterwards b is switched off (because c becomes present) n minutes after the switch-on of a. In other words, the discrete path $(1,0,0) \rightarrow (1,1,0) \rightarrow (1,1,1) \rightarrow (1,0,1)$ has to be possible in the model with delays.

   - $(1,0,0) \rightarrow (1,1,0)$ leads to the constraint

     $$\delta_b^+(0) < \delta_c^+(0)$$

   - Moreover the time that takes a trajectory from discrete state $(1,0,0)$ to $(1,0,1)$, is $\delta_b^+(0) + (\delta_c^+(0) - \delta_b^+(0)) + (\delta_b^-(1))$. Then we also have the following constraint:

     $$\delta_c^+(0) + \delta_b^-(0) = n \text{ minutes.}$$

2. Let us now consider that the switch-on of b does not occur. The deduced constraint becomes:

   $$\delta_c^+(0) < \delta_b^+(0)$$

Such kind of constraints can be automated by the use of some computer science tools dedicated to analysis of linear hybrid automata. For example we used HyTech [16] for two purposes: (1) to find automatically all paths from a specified initial state to another one and (2) to synthesize constraints on the delay parameters in order to follow any specific path.

This modelling framework then seems to allow the modeller to take into account information about observed time. Indeed the parameter identification can be decomposed into two parts: the valuation of discrete parameters can be found using an exhaustive approach like SMBioNet, and, the delay parameters can be found using HyTech which allows the building of constraints.

Nevertheless, such a modelling framework present a little drawback: the succession of intermittent orders of synthesis of a variable cannot lead to its global increase. This drawback is explicit in the following example.

4.3 A feedforward loop controled by a negative loop

Let us recall that a and a' oscillate with a period which is much less than the delays mandatory for the swich-on of variables b and c. Concentrations of b and c are then increasing but at each cycle of a, the counter-order of decreasing of b (resp. c) resets the clock h_b (resp. h_c) before the clock has reached the threshold $\delta_b^+(0)$ (resp. $\delta_c^+(0)$). Then neither h_b nor h_c will reach the threshold leading to the switch-on of the corresponding variable. This modelling framework can only represent the situation where neither b nor c can switch-on, case 1 of Table 1.

Thus this modelling framework makes possible the automation of parameter identification, and allows the distinction of two different behaviours (b can be switched-on or not) amongst the 5 possible ones. Nevertheless, it does not allow the representation of accumulation.

5 Product of automata: an alternative approach

The HyTech model checker performs *symbolic model checking* on automata and we have shown that this extension of model checking allows for the extraction of parameter constraints from some given paths. Another way to use symbolic model checking in order to identify the parameters and the delays of a gene regulatory network is to perform *products of automata*. The advantage of this kind of approach is that the computation of a product of automata does not only furnish a resulting automaton; it also systematically labels the states and the transitions of the automaton by some formulas that define the conditions under which the transitions can be fired. Then, provided that we adopt an adequate "hybrid" temporal logic, there are model checking algorithms able to manage symbolic values for some parameters. They compute the constraints under which a given temporal formula is satisfied.

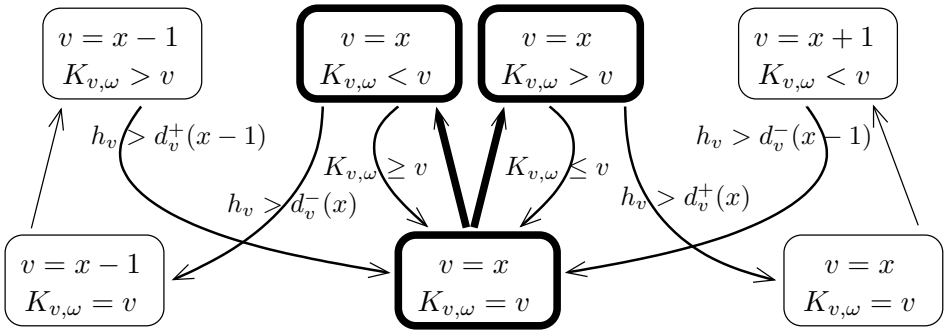


Figure 15: Timed automaton for one level

Using the UPPAAL model checker

In [23], Heike Siebert and Alexander Bockmayr made use of products of automata in order to formalize a hybrid modelling framewok for delays inspired by the approach of René Thomas. The automata that play the role of state graphs in this framewok are rather heavy, but they should be considered as purely technical mathematical objects, which will be submitted to UPPAAL [6].

The main idea is the following. For each variable v of the network, there is a clock called h_v, and for each possible discrete state of this variable, there are three possible behaviours with respect to delays:

- either the parameter $K_{v,\omega}$ (where ω is the set of resources of v according to the current state of the system) is greater than the current value of v, in which case the clock h_v measures the time of increasing of v;
- either the parameter $K_{v,\omega}$ is lower than the current value of v, in which case the clock h_v measures the time of decreasing of v;
- or the parameter $K_{v,\omega}$ is equal to the current value of v, in which case the clock h_v is off.

This principle is reflected by an "atomic" automaton structure containing three discrete states (called *locations* in the timed automata framework), as shown in the bold part of Figure 15. The central location is intuitively the default location for the state $v = x$ and every transition that changes the value of the variable v goes to this central location. Then, if $K_{v,\omega} = v$ is false, the suitable bold transition goes *immediately* to the consistent location (either $K_{v,\omega} < v$ or $K_{v,\omega} > v$) and the clock h_v starts from 0.

The product of these atomic automata is managed in such a way that the set of resources ω is properly computed in the product automaton. The formulas are rather complex, but they simply reflect the formal definition of the Thomas' framework. Lastly, as shown in the figure, if the clock h_v reaches its limit delay $d_v^+(x)$ (resp. $d_v^-(x)$) then the transition to $v = x + 1$ (resp. $v = x - 1$) is fired, and similarly, if some other variable change induces a

different comparison of $K_{v,\omega}$ with respect to x, then the location is pulled back to the central one.

This technical stuff being done, UPPAAL can be used in order to extract the constraints generated by some temporal formulas, which can for example reflect knowledge on the biological system about time delays to go from one state to another state. Let us remark that this framework still does not treat accumulation, because when the location is pulled back to the central one, it resets automatically the clock to 0.

6 Hybrid models inspired by PLDE

We saw that situations where accumulation plays a crucial role for the global behaviour are difficult to take into consideration. Nevertheless we want to overpass these difficulties and propose a new hybrid modelling framework which takes into account accumulation.

The first attempt to propose such a hybrid modelling framework [1] was based on the discrete model. With each discrete state, is associated a temporal zone, which makes hybrid the models. The temporal zone is defined as a hypercube whose dimension is the number of variables. The length of the hypercube associated with state μ in the direction v is $d_v^-(\mu) + d_v^+(\mu)$: it corresponds to the sum of the time mandatory to pass to the level $\mu_v + 1$ under increasing order and the time mandatory to pass to the level $\mu_v - 1$ under decreasing order. Because of the presence of two delays associated with a same domain and a same variable, accumulation can be represented but the decreasing and the decreasing of a variable subject to accumulation take place at the same speed, this drawback has been discussed in [1], see Figure 13 inside.

6.1 From PLDE to hybrid models

Since PLDE modelling framework is able to represent such accumulations, we present in this section yet another hybrid modelling framework based on PLDE which partially allows the building of constraints leading to a particular behaviour. The only new fundamental idea is to express a relationship between delays of the hybrid model and the PLDE model: the delay $d_v^+(\mu)$ (resp. $d_v^-(\mu)$) is an approximation of the time necessary to variable v to cross the domain from the lower bound to the upper bound (resp. from the upper bound to the lower one).

In other words, if the PLDE is known, it becomes easy to build the hybrid model since

- the thresholds defining the discretization of the PLDE are given,
- the parameters $K_{...}$ are the discretization of equilibrium points,
- the delays parameters are deducible from the PLDE.

More interesting is the inverse translation: If a hybrid model is supposed to represent a system, is it possible to construct a PLDE system whose behaviours are coherent with the possible paths in the hybrid model ? To answer this question, the work of Snoussi [24] has to be done again in the hybrid context: Snoussi has shown that it is possible to build from a discrete model, a PLDE system whose discretization is the discrete model.

Intuition. Let us consider the differential system modelling the feedforward loop controled by a positive auto-regulation:

$$\left\{\begin{array}{lcll} \frac{da}{dt}(t) & = & 50 \times \mathbb{1}_{[a>10]} & -2 \times a(t) \\ \frac{db}{dt}(t) & = & 40 \times \mathbb{1}_{[(a>21)\wedge(c<10)]} & -2 \times b(t) \\ \frac{dc}{dt}(t) & = & 25 \times \mathbb{1}_{[a>20]} & -2 \times c(t) \end{array}\right.$$

A simulation of this differential system is shown in Figure 11-left. In order to consider two different qualitative level of b, we introduce a threshold $\theta_b = 10$: if concentration of b is less than θ_b, b is said absent, otherwise it is said present. Let us suppose that a is greater than the threshold $\theta_{a,a}$. 16 domains have to be considered since a can take 4 qualitative values (less than $\theta_{a,a} = 10.0$, between $\theta_{a,a}$ and $\theta_{a,c} = 20.0$, between $\theta_{a,c}$ and $\theta_{a,b} = 21.0$, or greater than $\theta_{a,b}$), b and c can take 2 qualitative values (greater or less than θ_b and $\theta_{c,b}$).

1. In the domain $(1, 0, 0)$, a is increasing towards $50/2 = 25$

$$\left\{\begin{array}{lcll} \frac{da}{dt}(t) & = & 50 & -2 \times a(t) \\ \frac{db}{dt}(t) & = & & -2 \times b(t) \\ \frac{dc}{dt}(t) & = & & -2 \times c(t) \end{array}\right.$$

   The delay $d_a^+((1, 0, 0))$ is deduced from the solution of the differential equation: $a(t) = \frac{50}{2} - (\frac{50}{2} - a(0))e^{-\gamma_a t}$. The delay is the time necessary for the solution to go from the left boundary of the domain to the right one. Then $a(0) = \theta_{a,a} = 10$, and $a(t) = \theta_{a,c} = 20$.

$$d_a^+((1, 0, 0)) = -\frac{1}{\gamma_a} \ln\left(\frac{\frac{50}{2} - \theta_{a,c}}{\frac{50}{2} - \theta_{a,a}}\right) = -\frac{1}{2}\ln(\frac{1}{3}) = 0.55 \quad (2)$$

   Both other variables stays in the domain, then delays are not significant.

2. In the domain $(2, 0, 0)$, a and c are subject to increasing order towards $50/2 = 25$ and $25/2 = 12.5$ respectively.

$$\left\{\begin{array}{lcll} \frac{da}{dt}(t) & = & 50 & -2 \times a(t) \\ \frac{db}{dt}(t) & = & & -2 \times b(t) \\ \frac{dc}{dt}(t) & = & 25 & -2 \times c(t) \end{array}\right.$$

The delays $d_a^+((2,0,0))$ and $d_c^+((2,0,0))$ are deduced from the solution of the differential equation: $a(t) = \frac{50}{2} - (\frac{50}{2} - a(0))e^{-\gamma_a t}$ and $c(t) = \frac{25}{2} - (\frac{25}{2} - c(0))e^{-\gamma_c t}$. The delay is the time necessary for the solution to go from the left boundary of the domain to the right one.

- for variable a: $a(0) = \theta_{a,c} = 10$, and $a(t) = \theta_{a,b} = 21$.

$$d_a^+((2,0,0)) = -\frac{1}{\gamma_a} \ln \left(\frac{\frac{50}{2} - \theta_{a,b}}{\frac{50}{2} - \theta_{a,c}} \right) = 0.11 \quad (3)$$

- for variable c: $c(0) = 0$, and $c(t) = \theta_{c,b} = 10$.

$$d_c^+((2,0,0)) = -\frac{1}{\gamma_c} \ln \left(\frac{\frac{25}{2} - \theta_{c,b}}{\frac{25}{2} - 0} \right) = 0.84 \quad (4)$$

Then in the hybrid model, unless if c has accumulated before, a will pass the threshold $\theta_{a,b}$ before c will cross $\theta_{c,b}$.

3. In the domain $(3,0,0)$, b and c are subject to increasing order towards $40/2 = 20$ and $25/2 = 12.5$ respectively.

$$\begin{cases} \frac{da}{dt}(t) = 50 - 2 \times a(t) \\ \frac{db}{dt}(t) = 40 - 2 \times b(t) \\ \frac{dc}{dt}(t) = 25 - 2 \times c(t) \end{cases}$$

The solutions of the differential equation system is: $b(t) = \frac{40}{2} - (\frac{40}{2} - b(0))e^{-\gamma_b t}$ and $c(t) = \frac{25}{2} - (\frac{25}{2} - c(0))e^{-\gamma_c t}$. The delays are the times necessary for the solution to go from the left boundary of the domain to the right one.

- for variable b: $b(0) = 0$, and $b(t) = \theta_b = 10$.

$$d_b^+((3,0,0)) = -\frac{1}{\gamma_b} \ln \left(\frac{\frac{40}{2} - \theta_b}{\frac{40}{2} - 0} \right) = 0.34 \quad (5)$$

- for variable c: $c(0) = 0$, and $c(t) = \theta_{c,b} = 10$.

$$d_c^+((3,0,0)) = -\frac{1}{\gamma_c} \ln \left(\frac{\frac{25}{2} - \theta_{c,b}}{\frac{25}{2} - 0} \right) = 0.84 \quad (6)$$

Other delays are deduced similarly.

6.2 Sketch of the hybrid model

To go further, one needs to describe the evolutions of the hybrid model. In fact, with each domain is associated a temporal zone which is also defined as a hypercube whose dimension is the number of variables. According to the position of the focal point, we split the temporal zone into several subzones. Let us consider a discrete state $\mu = (\mu_i)_{i \in V}$:

- if $K_{i,\omega_i(\eta)} = \mu_i$, the coordinate i of the temporal zone is divided into 3 parts: the part where the concentration of i has to increase in order to reach the coordinate i of the focal point (the clock associated with i continues to increase until a delay denoted $d_i^+(\eta)$), the part where the concentration of i has to decrease in order to reach the coordinate i of the focal point (the clock continues to decrease until a delay denoted $d_i^-(\eta)$), and the part where the concentration of i has reached the coordinate i of the focal point (the clock is stopped).

- if $K_{i,\omega_i(\eta)} > \mu_i$ (resp. $< \mu_i$), the coordinate i of the temporal zone is not divided, since in all cases, the concentration of i has to increase (resp. decrease). The delay $d_i^-(\eta)$ (resp. $d_i^+(\eta)$) is set to 0.

The states of the hybrid model are couples $(\eta, (c_i)_{i \in V})$ where η is a qualitative state and $c_i \leq d_i^-(\eta) + d_i^+(\eta)$. Evolutions inside a qualitative state are easy to describe: the system evolves linearly until a boundary is reached: if the reached boundary corresponds to a subzone where variable i does not evolve anymore, then the clock associated with variable i stopped.

The description of the transition between two temporal zones is a little more tricky. If the reached boundary is a external face of the temporal zone, there is a qualitative jump from the current discrete state to the next state. The clock of the variable which has changed is reset in order to be coherent with the new qualitative state, and other clocks are modified to preserve the proportion of the concentration space which has already been crossed. Some particular situations lead to tricky rules explaining, for example, what is the trajectory when one successor of state μ_1 is μ_2 and one successor of state μ_2 is μ_1 (see the notion of black wall in [11]). The precise definition of this hybrid model can be found in [15].

Let us notice that delays associated with i seem to depend on the current qualitative state. Nevertheless for all qualitative states where the regulators of i are identical, the differential equation for variable i is the same. Thus, delays associated with i depends in fact on the set of active regulators, denoted in the sequel by ω_i.

6.3 Constraints on delays

More generally, it is possible to construct constraints on delays in order the system to follow a given sequence of domains. The principle of the construction of these constraints relies on the enumeration of constraints due to

paths of length 2: $\mu_0 \rightarrow \mu_1 \rightarrow \mu_2$. For a longer path, the constraint is the conjunction of constraints due to each sub-path of length 2.

We describe here only one situation among twelve. Let us consider the path $\mu_0 \rightarrow \mu_1 \rightarrow \mu_2$ where the first (resp. second) transition is due to a qualitative increasing of variable i_0 (resp. i_1). Let us suppose moreover that the vector $(c_i)_{i \in V}$ represents the clocks when entering into μ_1 and that there exists in μ_1 a variable i'_1 which can also increase. In order to allow the global path $\mu_0 \rightarrow \mu_1 \rightarrow \mu_2$, the following relation has to be satisfied:

$$(d^+_{i_1}(\mu_1) - c_{i_1}) < (d^+_{i'_1}(\mu_1) - c_{i'_1})$$

The twelve cases are exhaustively treated in [15].

6.4 Construction of constraints on FFL with auto-regulation

Let us consider the path allowing b to be switched-on before c. The sequence of domains is $(1,0,0) \rightarrow (2,0,0) \rightarrow (3,0,0) \rightarrow (3,1,0) \rightarrow (3,1,1) \rightarrow (3,0,1)$.

1. From $(1,0,0)$, there exists a unique successor domain: $(2,0,0)$. No constraint.

2. From $(2,0,0)$, there exists two possible successors: $(3,0,0)$ or $(2,0,1)$. Then, considering that clocks are reset to 0 when entering into $(2,0,0)$, we have:

$$d^+_a((2,0,0)) < d^+_c((2,0,0)) \qquad \text{see, Figure 16-left}$$

3. From $(3,0,0)$, it is possible to reach either $(3,1,0)$ or $(3,0,1)$. Then we have

$$d^+_b((3,0,0)) < d^+_c((3,0,0)) - d^+_a((2,0,0)) \qquad \text{see, Figure 16-right}$$

since during the crossing of the domain $(2,0,0)$, c has begun to increase.

4. From $(3,1,0)$ (resp. $(3,1,1)$), there exists a unique successor: $(3,1,1)$ (resp. $(3,0,1)$). No constraint.

Let us just remark that the delays deduced from the PLDE system of Figure 11-left (see equations 3, 5 and 6) does satisfy the previous contraints, whereas the delays deduced from the PLDE system of Figure 11-right does not satisfy them. Indeed taking into account parameters values of Figure 11-right, the analytic expression 5 gives $d^+_b((3,0,0)) = 2.77$ whereas expression 6 gives $d^+_c((3,0,0)) = 0.84$.
In a similar way, it is possible to build a set of constraints on delays which leads to trajectories along which the variable b is not switched on because of the fast increasing of c. The delays deduced from the PLDE system of Figure 11-left does not satisfy the contraints, whereas the delays deduced from the PLDE system of Figure 11-right does satisfy them.

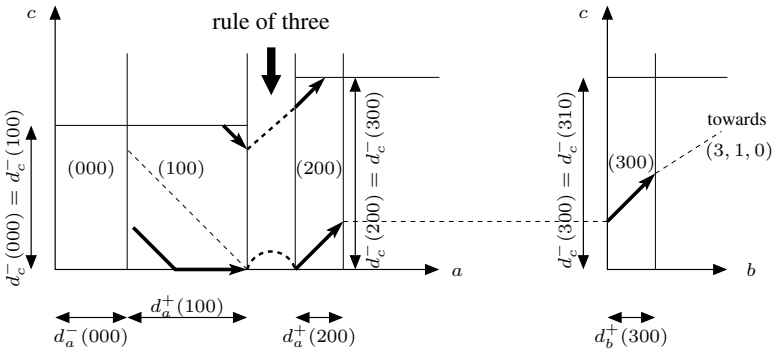


Figure 16: Illustration of the construction of constraints

6.5 A feedforward loop controled by a negative loop

We consider the boolean model, see Figure 17, of the feedforward loop controled by a negative loop which has been presented in Section 3.4: in order to build a boolean model thresholds of a on a', b and c are considered as equal. We would like to construct an hybrid system based on this boolean model

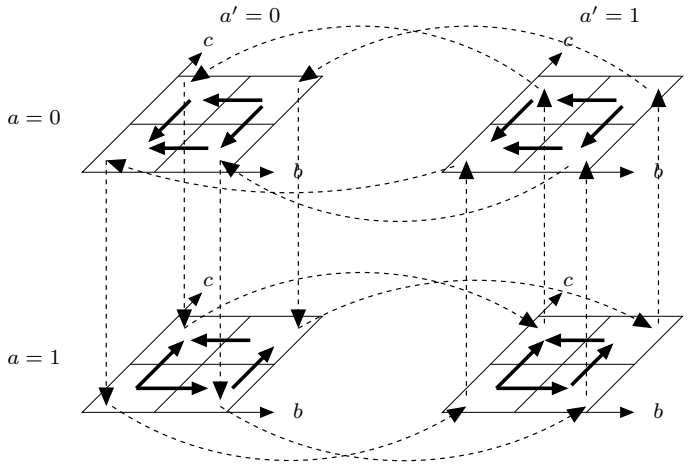


Figure 17: Boolean model of the FFL controled by a negative loop.

whose trajectories pass through the following sequence of domains:

$$(1000 \to 1100 \to 0100 \to 0000 \to)1000 \to 1100 \to$$
$$(1110 \to 0110 \to 0010 \to 1010 \to)^2 1110 \to 1111$$

This path expresses that more than one period of oscillation of a and a' are mandatory to imply the qualitative increasing of b. Two other periods are necessary to allow the qualitative activation of c which will be responsible of the degradation of b.

The expression of the corresponding constraints is very unreadable but is satisfiable. Figure 18 shows a set of values of parameters leading to the considered path.

$d^+_{a,0,\{a'\}}$	= 4	$d^+_{a',0,\{a\}}$	= 4
$d^+_{a,0,\emptyset}$	= 1	$d^+_{a',0,\emptyset}$	= 1
$d^-_{a,0,\emptyset}$	= 1	$d^-_{a',0,\emptyset}$	= 1
$d^+_{a,1,\{a'\}}$	= 1	$d^+_{a',1,\{a\}}$	= 1
$d^-_{a,1,\{a'\}}$	= 1	$d^-_{a',1,\{a\}}$	= 1
$d^-_{a,1,\emptyset}$	= 4	$d^-_{a',1,\emptyset}$	= 4
$d^+_{b,0,\{a,c\}}$	= 7	$d^+_{c,0,\{a\}}$	= 10
$d^+_{b,0,\emptyset}$	= 2	$d^+_{c,0,\emptyset}$	= 4
$d^-_{b,0,\emptyset}$	= 2	$d^-_{c,0,\emptyset}$	= 16
$d^+_{b,1,\{a,c\}}$	= 2	$d^+_{c,1,\{a\}}$	= 4
$d^-_{b,1,\{a,c\}}$	= 2	$d^-_{c,1,\{a\}}$	= 4
$d^-_{b,1,\emptyset}$	= 80	$d^-_{c,1,\emptyset}$	= 10

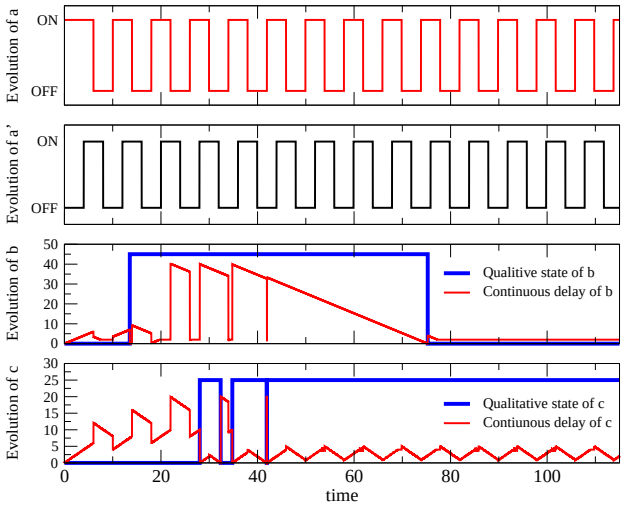


Figure 18: Simulation of a hybrid model for the FFL controled by a negative loop. The qualitative part of the initial state is $(1, 0, 0, 0)$, and its delay's part is $(2., 0., 0., 0.)$. Delays are denoted by the involved variable, its qualitative level and by the set of its effective regulators, see the description of the hybrid model.

7 Identification issues with delays

Let us remind that the cornerstone of the modelling activity is the parameter identification step. For the construction of a hybrid model based on the discrete modelling framework of R. Thomas, one has to identify both discrete parameters $K_{...}$ and delays $d^+_{...}$ and $d^-_{...}$ which correspond to approximations of times mandatory to pass through the involved domain.

In order to determine the values of delays, the modeller is going to rely on the measured elapsed time during experiments between two particular states. This global measured time has to be equal to the sum of delays of visited qualitative states. Thus, building the constraints associated with the measured elapsed time requires to know the sequence of visited qualitative states.

The parameter identification step can then be split into two subparts:

- To identify discrete parameters $K_{...}$ of the underlying discrete model. This step can be automated using model-checking or other formal methods (see sections 2.5, 2.6 and also [21, 13]).

- To identify the delays of the hybrid model. Here the built constraints on delays express relationships between a real number (the measured time) and a combination of delays. The resolutness of constraints on delays is unavoidable.

Let us mention some other approaches which do not consider a continuous time. The first way consists in discretizing the time in order to remain in a completely discrete modelling process. The underlying idea is to construct an approximation so fine as necessary as in the integral calculus. The second way consists in focusing on the duality between probabilistics approaches and models based on delays: greater the probability to fire a transition towards a particular qualitative state, smaller the associated delay. Thus all the scientific corpus of Markov chains can be useful to evaluate the probabilities of the model.

Conclusion

We have shown that different modelling frameworks for gene regulatory networks have been introduced. For all of them, the parameters have to be valuated. Fortunately this parameter identification step can be computer-aided when the modelling framework is formal and when it makes use of formal tools from computer science: for the purely discrete approach of R. Thomas, model checking, constraint programming or symbolic execution have been used to automate this stage. In order to take into consideration elapsed time and delays, it would be interesting to develop a tool that would take as inputs a PLDE system and a set of observed trajectories (and associated elapsed times) and that would give all possible valuations for parameters. Unfortunatelly such a tool is not conceivable for PLDE models. Thus, formal hybrid modellings seem to be the best candidates in order to fill up the gap between purely discrete models for which the parameter identification step can be automated and the differential models.

With such hybrid frameworks, systems biology should take advantage of the whole corpus of formal methods from computer science which opens a large horizon of research perspectives. Let us mention for instance,

- Algorithms that compute the set of parameter valuations that are compatible with reachability properties or with a known qualitative path.
- Continuous-time temporal logics adapted to the specificities of the biological application domain, and then model checking algorithms to confront a real-time temporal property to a hybrid model.
- Formal or symbolic languages to describe transition paths, taking into account populations of networks whose states are not synchronized.
- Since our aim is also to link modelling to experiments, tools to extract from the considered hybrid model several experiments which are able to refute the candidate models.

Indeed, the hybrid modellings are not the ultimate aim, they are only a guide for predictions that, in turn, suggest biological experiments whose success will be *in fine* the discriminent criterion. Thus a hybrid modelling framework

will be largely adopted only if it is able to help biology toward comprehending the biological processes through the ability of the hybrid framework to propose experiments or through its capability of refuting hypotheses. Hybrid approaches could constitute a trade-off between expressiveness and computational tractability.

Acknowledgements

In a way, the biologist Janine Guespin is our ”muse”, who motivates by her enthusiasm a lot of our works. Even if she is not a co-author of this chapter, we thought of her when writing this chapter, as our typical target reader. This work could not have succeeded without the help and support of Olivier Roux and Jonathan Fromentin who developped with us the hybrid modelling framework. We would like to thank Adrien Richard and Laurent Trilling for their comments at different stages of this work. Finally, we are very grateful to Alexander Bockmayr, Heike Siebert for very deep discussions on how to introduce delays into discrete models of gene regulatory networks.

References

[1] J. Ahmad, G. Bernot, J.-P. Comet, D. Lime, and O. Roux. Hybrid modelling and dynamical analysis of gene regulatory networks with delays. *ComPlexUs*, 3(4):231–251, 2006 (Cover Date: Nov. 2007).

[2] J. Ahmad, O. Roux, G. Bernot, J.-P. Comet, and A. Richard. Analysing formal models of genetic regulatory networks with delays: Applications to lambda phage and T-cell activation systems. *Int. J. Bioinformatics Research and Applications*, 4(3):240–262, 2008.

[3] U. Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC, 2006.

[4] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *T.C.S.*, 138(1):3–34, 1995.

[5] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, pages 209–229, 1992.

[6] J. Bengtsson and W. Yi. Timed automata: semantics, algorithms and tools. In *Lecture Notes on Concurrency and Petri Nets*, number 3098 in LNCS, pages 87–124. Springer, 2008, See also http://www.uppaal.com.

[7] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: Extending Thomas’

asynchronous logical approach with temporal logic. *J.T.B.*, 229(3):339–347, 2004.

[8] F. Cassez and K. G. Larsen. The impressive power of stopwatches. In *CONCUR'00*, pages 138–152, London, UK, 2000. Springer-Verlag.

[9] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. NuSMV Version 2: An OpenSource Tool for Symbolic Model Checking. In *CAV 2002*, volume 2404 of *LNCS*, Copenhagen, Denmark, July 2002. Springer.

[10] F. Corblin, S. Tripodi, E. Fanchon, D. Ropers, and L. Trilling. A declarative constraint-based method for analysing discrete genetic regulatory networks. *Biosystems*, 98:91–104, 2009.

[11] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselman. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biology*, 2004.

[12] E.A. Emerson and J.Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, San Francisco, California*, pages 169–180, 5-7 May 1982.

[13] E. Fanchon, F. Corblin, L. Trilling, B. Hermant, and D. Gulino. Modeling the molecular network controlling adhesion between human endothelial cells: Inference and simulation using constraint logic programming. In *CMSB 2004*, pages 104–118, 2005.

[14] A.F. Filippov. *Differential equations with discontinuous righthand sides*. Kluwer Academic Publishers, 1988.

[15] J. Fromentin, Bernot G., J.-P. Comet, and O. Roux. Hybrid modeling of gene regulatory networks: a combination of qualitative modeling with delays. *not yet published*, 2010.

[16] T.-A. Henzinger, P.-H. Ho, and H. Wong-Toi. HYTECH: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, 1997.

[17] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge University Press, 2000.

[18] H. Kitano. Biological robustness. *Nat. Rev. Genet.*, 5:826–837, 2004.

[19] H. Kitano. Towards a theory of biological robustness. *Mol. Syst. Biol.*, 3:137, 2007.

[20] S. Mangan and U. Alon. The structure and function of the feedforward loop network motif. *P.N.A.S. USA*, 100:11980–11985, 2003.

[21] D. Mateus, J.-P. Comet, J.-P. Gallois, and P. Le Gall. *Proc. of the Evry Spring school on Modelling and simulation of biological processes in the context of genomics*, chapter Inferring parameters of genetic regulatory networks with symbolic formal methods, pages 51–70. EDP Sci., 2007.

[22] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[23] H. Siebert and A. Bockmayr. Temporal constraints in the logical analysis of regulatory networks. *T.C.S.*, 391(3):258–275, 2008.

[24] E.H Snoussi. Qualitative dynamics of a piecewise-linear differential equations : a discrete mapping approach. *Dynamics and stability of Systems*, 4:189–207, 1989.

[25] A.P. Thomas, A. Henzinger, P.W. Kopke, and P. Varaiya. What's decidable about hybrid automata. Technical report, EECS Department, University of California, Berkeley, 1998.

[26] R. Thomas. Regulatory networks seen as asynchronous automata : A logical description. *J.T.B.*, 153:1–23, 1991.

[27] R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, 1990.

[28] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. I. & II. *Chaos*, 11:170–195, 2001.