Systematic Approach to Computational Design of Gene Regulatory Networks with Information Processing Capabilities

M. Moškon and M. Mraz

Citation to the original IEEE publication:

M. Moškon, M. Mraz, **Systematic Approach to Computational Design of Gene Regulatory Networks with Information Processing Capabilities**, *IEEE/ACM Transactions of Computational Biology and Bioinformatics*, Vol. 11, Issue 2, 2014 (doi: 10.1109/TCBB.2013.2295792)

Available at IEEE Xplore

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6690102

Systematic Approach to Computational Design of Gene Regulatory Networks with Information Processing Capabilities

Miha Moškon and Miha Mraz, *Member, IEEE*

***Abstract*—We present several measures that can be used in *de novo* computational design of biological systems with information processing capabilities. Their main purpose is to objectively evaluate the behaviour and identify the biological information processing structures with the best dynamical properties. They can be used to define constraints that allow one to simplify the design of more complex biological systems. These measures can be applied to existent computational design approaches in synthetic biology, i.e. rational and automatic design approaches. We demonstrate their use on a) the computational models of several basic information processing structures implemented with gene regulatory networks and b) on a modular design of a synchronous toggle switch.**

***Index Terms*—Computational Biology, Computational Design, Gene Regulatory Networks, Information Processing, Modelling and Simulation, Modular Design, Synthetic Biology.**

I. Introduction

DESIGN of novel biological systems strongly relies on computational approaches. These are based on the establishment of various computational models which are used to simulate *in-silico* dynamics of the biological system in question. Computational design approaches can be classified to two groups, a) rational design in a combination with *computer aided design* (CAD) and b) *automatic design* [1]. Rational design techniques use engineering approaches, such as modularization, rationalization and modelling, to build novel biological systems [2]. In CAD approaches circuits are designed rationally with the aid of computers. The main problem of these approaches is that human intervention is vital. They sometimes require intuitive solutions. These depend on the designer's experience and are as such far from being straightforward. However, many of the first successful implementations in the field of synthetic biology, such as the toggle switch [3] and repressilator [4], were executed this way. Several CAD tools for synthetic biology have been developed in recent years. These tools allow the designer to rapidly establish and analyse the computational models of a certain biological system. Typical representatives are e.g. TinkerCell [5], COPASI [6] and SimBiology. While these tools may be very useful in the design process, they only help the designer with the establishment of computational models and offer different types of analysis of their behaviour. However, the design process still derives from rational design techniques and therefore faces exactly the same problems.

Automatic approaches on the other hand try to establish the design of a biological system with a predefined behaviour, specified by e.g. a thruth table [7], *in-silico*, without any human intervention. Several more or less successful attempts were reported recently. All of them try to automatize the construction of biogical systems from basic modules. However, these approaches are still in many ways limited. Huynh et al. [8] present an approach that aims to find the best parts from a library of modules that will perform a given function for a fixed topology. They compare the obtained behaviour of a given configuration with the desired behaviour based on its steady states or the predefined temporal profile of the system. Their approach does not consider noise effects. The requirement for an accurate definition of the desired behaviour, which may not be achieved at all, can also present a major problem. Moreover, the topology needs to be fixed. Marchisio and Stelling [7] try to find compatible modules with the steady state separation. They do not consider effects that may be caused by noise. Their approach is applicable only to combinatorial circuits. Rodrigo and Jaramillo [9] evaluate the network configurations obtained by metaheuristic search algorithms with behaviour based fitness functions [10]. This is very similar to an approach of Huynh et al. [8] and does not solve the problems of the evaluators used by them. Moreover, in Rodrigo and Jaramillo's case each intermediate design has to be verified with the computationally expensive chemical simulation. This makes their approach useful only for biological networks of a very small complexity. The most comprehensive approach was presented by Beal et al. [11]. It considers the logic compatibility among the modules used, but does not consider switching, i.e. temporal compatibility. The approach can only be applied to the circuits with a limited functional diversity, i.e. only to combinatorial circuits. In addition like the other approaches presented here, it does not consider the effects that the receiving modules might have on the sending modules, i.e. retroactivity [12], due to the large consumption of chemical species used as a medium between interacting modules. The approach does not consider the stability of the modules.

Here, we introduce a systematic approach that employs a set of computational measures in *de novo* design of biological systems with information processing capabili-

The authors are with the Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia (e-mail: miha.moskon@fri.uni-lj.si; miha.mraz@fri.uni-lj.si).

Manuscript received ; revised .

ties (BSIPs). The introduced measures aim to solve the problems of the existing approaches. They are used to specify the criteria that have to be considered in modular construction of BSIPs with larger functional complexity. The criteria can be used to verify the compatibility between the basic modules and thus provide support for modular design of more complex circuits. No additional verification of the target design is needed after the constraints are satisfied. The introduced measures are thus able to lead the design process from the evaluation of basic modules to the modular construction of a more complex biological system. They can be used in combination with CAD tools to make rational design of novel biological systems more straightforward. The measures can be applied to a) objectively characterize the basic modules, b) identify the modules which are the most suitable for a certain task and c) determine and eliminate the potential shortcomings from the established modular design networks. After the desired *in-silico* behaviour is achieved the experimental realization of the system can be performed. The measures can also be used as a basis for the definition of objective functions or as evaluators of the quality of both, individual information processing modules and established design networks of BSIPs that were obtained with automatic design approaches.

In Sec. II we establish the measures and propose a set of constraints that can be used in order to make the modular design of novel BSIPs systems more straightforward. The measures were established regarding two aspects. First, we apply the characteristics that are used to describe the behaviour of digital electronic circuits to the observed biological systems. Second, we describe the biological systems as nonlinear dynamical systems and investigate their asymptotic behaviour, stability and bifurcations. In Sec. III we present the application of the introduced measures on the models of simple gene regulatory networks representing BSIPs and demonstrate their usage on the modular design of a more complex system, i.e. synchronous toggle switch. Even though our methodology is demonstrated only on the models of gene regulatory networks, it could be easily applied to any other BSIPs model or even to the results obtained from their experimental realisations.

II. Methods

In our case modular *de novo* design of more complex BSIPs is executed as follows:

1) establishment of basic information processing modules,
2) characterization of basic information processing modules,
3) modular design of a target system,
4) verification of the obtained design.

The strategy mimics the rational design approach in which some steps may have to be repeated many times before the final solution is obtained [13], namely step 4 may often lead to the redesign of the BSIP (step 3) or even to the redesign of basic information processing modules (step 1). In order to establish such a modular methodology one needs to define ways of assessment that can be used as objective criteria and may eventually also lead to automatic design including verification.

We propose a BSIP should be studied from two perspectives, a) from an information processing perspective and b) from a qualitative perspective. In order to use a biological system for information processing, information has to be bound to a certain chemical species present in the system. The presence of a chosen chemical species can be interpreted as different logic values. In digital electronic circuits logic values are encoded as voltage levels, therefore the concentrations of the chosen chemical species could be viewed as to play a similar role. In order to asses the dynamical properties of digital electronic circuits from an information processing perspective the process of digital circuit design studies logic levels, regions of uncertainty, etc. [14], [15], [16]. Therefore, we propose to use the following quantities in the analysis of BSIPs:

- input and output logic levels,
- input and output regions of uncertainty,
- high level, low level and maximal noise,
- noise margins,
- switching times,
- validity duration and
- input consumption.

Insights into the biological system's asymptotic behaviour can be obtained with the qualitative analysis of the Ordinary Differential Equations (ODEs) that present its deterministic model. The system's qualitative analysis can lead us to the evaluation of the system's robustness regarding:

- the size of a limit cycle in the case of oscillatory systems,
- the distance between the stable steady states in the case of multistable systems and
- the distance from the bifurcation points.

A. Analysis from an information processing perspective

1) Input output logic levels: Logic levels divide the *signal*, the concentrations of chemical species, used to encode information, into a number of logic values. In binary logic, which is for reasons of its simplicity used in the majority of modern computer systems, we are dealing with two logic values, 0 and 1, i.e. a low and a high level region. These two regions are typically separated by a region of uncertainty in which the signal does not have a valid interpretation. If the signal is in the region of the low logic level it can be interpreted as logic value 0. Conversely if the signal is in the region of the high logic level, it can be interpreted as logic value 1. In BSIPs as a contrast to typical electronic systems logic levels can be different for an input and output signal.

When dealing with an input signal one needs to define the input concentration levels that bring the system's output species to a concentration with a valid logic interpretation. In this context we introduce two measures, i.e. *minimal high*

level input concentration, $C_{IH(\min)}$, and *maximal low level input concentration*, $C_{IL(\max)}$. Depending on the BSIP's behaviour the minimal high level input concentration represents the lowest high concentration that either still keeps the concentration of the output species low enough or barely achieves the output species' concentration that is high enough to be interpreted as a valid logic value. The same applies to the maximal low level input concentration.

When dealing with an output signal one needs to define a concentration of chemical species that must be achieved after a logic switch is performed. We can thus introduce *high level output concentration*, C_{OH}, and *low level output concentration*, C_{OL}, which present the required concentrations of the species, for the signal to be interpreted as logic value 1 or 0 respectively.

When a signal travels through the system it can be distorted by noise effects. These influence the concentration of the chosen species. The concentration levels that still present a signal with a valid interpretation, are introduced for reasons of noise influence elimination. We shall denote these as *minimal high level output concentration*, $C_{OH(\min)}$, and *maximal low level output concentration*, $C_{OL(\max)}$. They define the concentrations, that are still interpreted as valid inputs to other, receiving, modules in the system.

With respect to the above the input and output *region of uncertainty* can be defined as $U_I = \left(C_{IL(\max)}, C_{IH(\min)}\right)$ and $U_O = \left(C_{OL(\max)}, C_{OH(\min)}\right)$, where U_I presents the input region of uncertainty and U_O the output region of uncertainty.

2) Noise: Deviations of the signal from its desired values are called noise and can be caused by the fluctuations within the system, i.e. intrinsic noise, or by external influences, i.e. extrinsic noise [16]. While the noise is always present in reality its influence on the interpretation of the signal can be mostly eliminated with the introduction of *noise margins*. Regarding the maximal noise present in the output signal we can define the *high level noise margin* as $NM_H = [C_{OH(\min)}, C_{OH}]$ and the *low level noise margin* as $NM_L = [C_{OL}, C_{OL(\max)}]$. The minimal high level output concentration $C_{OH(\min)}$ and maximal low level output concentration $C_{OL(\max)}$ can be evaluated regarding the maximal value of expected noise as

$$C_{OH(\min)} = C_{OH} - N_H, \tag{1}$$

$$C_{OL(\max)} = C_{OL} + N_L, \tag{2}$$

where N_H presents the maximal value of expected noise present in a high level signal and N_L the maximal value of expected noise present in a low level signal. Based on this we shall define the *maximal noise* as $N = \max(N_L, N_H)$.

3) Switching times: In binary logic there are two types of switches, the signal can transit from either low to high or from high to low. The time to perform a switch from a low to a high logic level, i.e. the time needed to increase the concentration of the output chemical species, can vary considerably from the time to perform a switch from a high to a low logic level, i.e. the time needed to decrease the concentration of the output chemical species. We therefore define two different measures, *rise time*, t_r, and *fall time*, t_f. Each of these can be measured as the time the concentration of the output chemical species remains within the uncertainty region after the switch has been initiated. We can further define the maximal time to perform any kind of switch, i.e. *switching time*, as $t_s = \max(t_r, t_f)$.

4) Validity duration: This measure is relevant only when the signal is valid for a limited amount of time, e.g. in oscillatory circuits the signal changes due to the nature of the circuit itself. Similar to switching times we can measure the time the signal has a valid interpretation after the switch has been performed. In this way we can determine the validity duration of a high level signal, t_{v_H}, and the validity duration of a low level signal, t_{v_L}. *Validity duration* can then be determined as $t_v = \min(t_{v_H}, t_{v_L})$.

5) Input consumption: When designing more complex BSIPs the input consumption, C_I, of an observed biological system has to be taken into account. The consideration of input consumption is highly important especially when using the same chemical species, i.e. the same output to drive several receiving modules, i.e. when the fan-out of the system is larger than 1. As a certain amount of chemical species is sequestrated by e.g. the promoter of the receiving module the concentration of the chemical species that is used as an input to that module cannot be used as an input to some other module at the same time. Moreover, in certain cases the receiving module may also affect the behaviour of the sending module, e.g. when using the output of the sending module also as its feedback input. This phenonenom, which is tightly correlated with the input consumption, is known as *retroactivity* and needs to be considered in the modular design of biological systems [12].

B. Analysis from a qualitative perspective

Qualitative analysis of biological systems can be performed regarding their asymptotic behaviour, i.e. behaviour that is achieved gradually without the modification of external inputs [17], [18]. In order to perform a qualitative analysis, it is preferable that the biological system is described with a deterministic model that is established with a set of ODEs. The qualitative analysis can be performed through the stability and bifurcation analysis on the phase space of ODEs that describe the concentration trajectories of the observed chemical species. Stability analysis provides the system's steady states, limit cycles and their stabilities. Bifurcation analysis gives the locations of bifurcation points.

1) Stability analysis: Regarding the stability analysis we can divide the BSIPs in three groups, i.e. a) systems with an oscillatory behaviour, b) systems that implement combinatorial circuits, and c) systems that implement sequential or memory circuits [19]. Oscillatory behaviour of the analysed biological system is conditioned with the existence of a stable limit cycle in the phase space of its deterministic model. Systems with only one steady state, which present the behaviour where concentrations of

the observed chemical species gradually converge into a fixed point in the phase space, can be used to implement combinatorial circuits. Systems that have more than one steady stable state reflect a behaviour where the current state is preserved even when an external input is changed until some predefined threshold. The current state of the system is therefore retained, memorized. Such biological systems can be used to implement sequential circuits. For binary logic we need to have a system with two steady stable states, i.e. a bistable system, in order to effectively memorize the information. It is important that the system does not change its state unless the external influence is strong enough to trigger the transition, i.e to perturbate the system from its current stable steady state to the other one.

Results of the stability analysis can be used to determine the system's qualitative behaviour. Combined with the analysis from an information processing perspective they can be used to evaluate also the robustness of the system's behaviour. Several robustness analysis approaches already exist in the field of systems biology [20]. We here propose two novel methods for robustness evaluation of bistable and oscillatory systems, i.e. *robustness regarding the size of a limit cycle* and *robustness regarding the distance between the stable steady states*, respectively.

Size of a limit cycle, d_c, is defined as the amplitude of the oscillations in an oscillatory system. All biological systems are subject to noise (see Sec. II-A2). Measuring the maximal noise N in the biological system in question we can asses robustness regarding the size of a limit cycle as

$$R_c = \frac{d_c}{N}, \tag{3}$$

where d_c presents the size of a limit cycle and N presents the maximal value of expected noise within the biological system. If R_c is close to or even smaller than 1, the system will not exhibit regular oscillations or the oscillations will gradually diminish in reality.

Distance between the stable steady states can be evaluated in systems that express two or more stable steady states. A larger distance between the states results in larger energy consumption when performing a switch. A smaller distance between the states results in smaller robustness of the system and therefore greater noise sensitivity. Similar to the size of a limit cycle distance between the stable steady states can be observed in relation to maximal noise

$$R_d = \min_{i \neq j} \frac{d_{i,j}}{N}, \tag{4}$$

where $d_{i,j}$ presents the distance between the stable steady states i and j and N presents the maximal value of expected noise within the biological system. If R_d is close to or even smaller than 1, the probability of unexpected switches between different steady stable states is very large.

2) Bifurcation analysis: The qualitative behaviour of a system is strictly dependant on the parameters that describe the properties of the chosen chemical species in a predefined environment. Playing with these parameters transitions among different types of qualitative behaviour, *bifurcations*, can be observed. In relation to our systems of interest, systems that either reflect convergence to a steady state or exhibit periodic oscillations, two types of bifurcations can occur. In a bistable system a supercritical pitchfork bifurcation occurs when a stable steady state bifurcates into two stable steady states and an unstable one. In a system with oscillatory behaviour a supercritical Hopf bifurcation occurs when a stable steady state bifurcates into an unstable steady state and a stable limit cycle. The parameter values where the bifurcation occurs are called bifurcation points. We can determine the location of a bifurcation point with the bifurcation analysis of a given system.

While there are many factors that define the parameter values within a biological system, the parameter values can vary drastically even within the same biological system. With an actual set of parameter values the robustness of the system can be evaluated with respect to the distance from a bifurcation point. This value is tightly connected to the probability that the system will reflect the desired behaviour and can be used to compare different systems with similar properties. *Robustness regarding the distance from the bifurcation points* is evaluated as

$$R_b = \min_i \frac{d_i}{b_i}, \tag{5}$$

where b_i defines the location of a bifurcation point for parameter i and d_i the distance between the bifurcation point and the current value of parameter i. A value close to 1 means we are very close to a bifurcation point. Parameters that are more or less invariant to environmental characteristics of the system (their values do not change) can be omitted from the calculation of R_b.

C. Modular design of complex BSIPs

When approaching the modular design of complex BSIPs several constraints have to be considered in order to obtain a system with the desired behaviour. The elementary modules that are used to construct the complex system should interact between each other only on the desired segments. This can be achieved with the use of orthogonal compounds of different modules [21]. The elementary modules should also be orthogonal to the biological mechanisms that are already present within the host. In order for the complex BSIP to reflect the correct behaviour the interacting modules need to be compatible. To achieve the desired behaviour two types of compatibility have to be satisfied, a) *logic compatibility* and b) *switching compatibility*.

Logic compatibility exists if the output levels of the sending modules are in accordance with the input levels of the chosen receiving modules [22], i.e. if the following conditions are satisfied

$$C_{OL(\max)}(out) \leq C_{IL(\max)}(in), \tag{6}$$

$$C_{OH(\min)}(out) \geq C_{IH(\min)}(in), \tag{7}$$

where $C_{OL(\max)}(out)$ and $C_{OH(\min)}(out)$ present the output concentration levels of the sending biological module

and $C_{IL(\max)}(in)$ and $C_{IH(\min)}(in)$ present the input concentration levels of the receiving biological module. When these conditions are satisfied the outputs of the sending modules are always able to induce a valid switch in the chosen receiving modules.

We also have to consider the time characteristics in the modules that simultaneously change the concentrations of their output species significantly, e.g. oscillatory circuits. In oscillatory circuits, the time during which an output signal can be interpreted as a valid logic value needs to be at least as long as the time needed to perform a switch in all receiving modules, i.e. all modules directly or indirectly connected to the output signal of the oscillator. Switching compatibility can be defined with the following condition

$$t_{v_O} \geq \max t_{s_I}, \tag{8}$$

where t_{v_O} presents the output signal's validity duration of the oscillatory circuit and t_{s_I} presents switching times of all modules that follow the oscillatory circuit in the design, i.e. all of the downstream modules.

In a modular design one need not forget about *input consumption* and *fan-out* of the elementary modules. If an output species is used on several different places as an input signal, large input consumption caused by one of the receiving modules could reflect in an undesired behaviour of the modular system. When the output of a sending module is used as a feedback input large input consumption of a receiving module may cause undesired behaviour of the sending module as well. In such cases insulator modules that decrease the retroactivity effect between the receiving and the sending module have to be used [23].

III. Results

In order to demonstrate the introduced analysis procedure, different computational models of sample BSIP modules were established, i.e. models of gene regulatory networks that present biological inverter (NOT gate), driver (YES gate), AND gate, NOR gate, toggle switch and repressilator. The kinetic parameters used were derived from the properties of protein CI, and O_R2 and O_R3 operators in a Bacteriophage λ system [24], [25]. The models were simulated and the efficiency of the proposed analysis procedure demonstrated by modular design of a more complex BSIP, i.e. a synchronous toggle switch.

A. Elementary modules

A biological inverter, i.e. NOT gate, can be realized with a gene regulatory network of two proteins, where the input protein represses the expression of the output protein (see Fig. 1 (a)). A complementary scenario, where the input protein activates the expression of the output protein presents a biological driver, i.e. YES gate (see Fig. 1 (b)). If we presume that the activation of the output protein is additionally dependant on the presence of an external inducer, which allows the activator to bind to the promoter and thus activate the transcription, a behaviour that is analogous to an AND gate is achieved (see Fig. 1 (c)). A biological inverter can be extended to NOR gate with the introduction of an additional repressor protein (see Fig. 1 (d)). If either of the two repressor proteins is present, the expression of the output protein ceases. The output protein is therefore expressed only in the case when both repressors are absent. Directed graph representations of the described systems are given in Fig. 1.

Elementary modules can be used to construct more complex systems. In our examples we will use two systems, i.e. the toggle switch and the repressilator. A toggle switch presents a potential memory element and can be realized with two biological NOR gates. On one hand these gates are mutually repressing each other. On the other hand two different external repressors repress each of the gates separately (see Fig. 2 (a)). A repressilator presents an oscillatory element, which can be used for synchronization. It can be realized with a circular structure of an odd number of NOT gates. Fig. 2 (b) presents the structure of a repressilator with 3 elements and Fig. 2 (c) a repressilator with 5 elements.

The described biological systems that present our elementary modules were modelled with two different approaches. Deterministic models were established with a set of ODEs, where Hill equations were used for the modelling of transcription activation [26]. Stochastic models with the Chemical Master Equation (CME) were established on the basis of a system of chemical reactions. Here, we will only present the description of the inverter and the toggle switch models. The descriptions of the other models used in Sec. III-C are available in the supplementary material accompanying this manuscript.

1) Inverter: The parameters of operator O_R3 and protein *CI* as an input protein will be used in order to model the biological inverter. The observed reactions are presented in Table I, where the reaction rates are as follows [24], [27], [28]:

- association of protein x with O_R3: $k_{a_3} = 0.000012 s^{-1} nM^{-2}$,
- dissociation of protein x from O_R3: $k_{d_3} = 0.4791 s^{-1}$,
- transcription of mRNA of protein y: $k_{trs} = 0.0715 s^{-1}$,
- translation of protein y: $k_{trl} = 0.043 s^{-1}$,
- degradation of mRNA of protein y: $k_{deg_m} = 0.0039 s^{-1}$,
- degradation of protein y: $k_{deg} = 0.0007 s^{-1}$.

TABLE I
Reactions in the model of biological inverter (NOT gate).

reaction	description
$2x$+DNA$\xrightarrow{k_{a_3}}$ x_2DNA	association of protein x with O_R3
x_2DNA$\xrightarrow{k_{d_3}}$ $2x$+DNA	dissociation of protein x from O_R3
DNA$\xrightarrow{k_{trs}}$mRNA+DNA	transcription of mRNA of protein y
mRNA$\xrightarrow{k_{trl}}$ mRNA+y	translation of protein y
mRNA $\xrightarrow{k_{deg_m}}$ $\emptyset$	degradation of mRNA of protein y
$y \xrightarrow{k_{deg}} \emptyset$	degradation of protein y

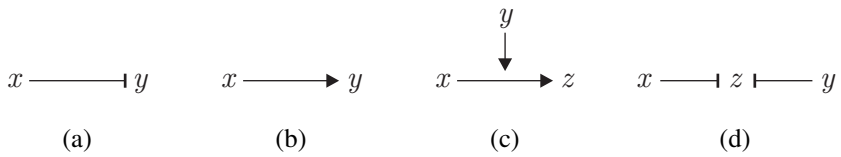


Fig. 1. Directed graph representation of basic logic gates, i.e. NOT gate (a), YES gate (b), AND gate (c) and NOR gate (d).

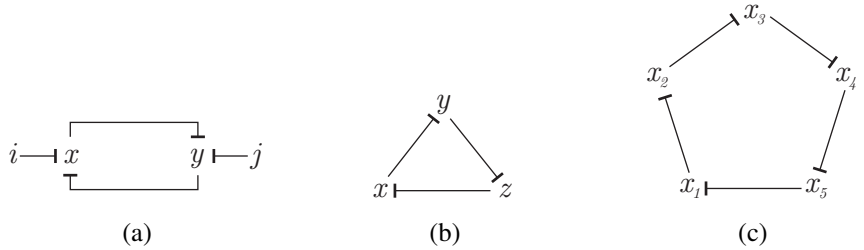


Fig. 2. Directed graph representation of toggle switch (a), repressilator circuit with three elements (b) and repressilator circuit with five elements (c).

A stochastic model can be established with the CME [29]. The trajectories of the observed chemical species were calculated with the Stochastic Simulation Algorithm (SSA) [30]. A deterministic model can be established with a set of ODEs. In order to model the transcription the following Hill equation [26] was used:

$$\frac{dm}{dt} = \frac{\alpha}{1 + \left(\frac{x}{K_{d_3}}\right)^{\gamma}} - \delta_m \cdot m, \tag{9}$$

$$K_{d_3} = \left(\frac{k_{d_3}}{k_{a_3}}\right)^{\frac{1}{\gamma}}, \tag{10}$$

where dm presents a change of mRNA concentration, α maximal transcription rate ($\alpha = k_{trs}$), γ cooperativity, i.e. Hill coefficient ($\gamma = 2$), x concentration of input protein, K_{d_3} dissociation constant for operator O_R3, δ_m mRNA degradation rate ($\delta_m = k_{deg_m}$) and m the mRNA concentration. Translation was modelled by the following differential equation

$$\frac{dy}{dt} = \beta \cdot m - \delta_y \cdot y, \tag{11}$$

where dy presents a change of output protein concentration, β translation rate ($\beta = k_{trl}$), δ_y output protein degradation rate ($\delta_y = k_{deg}$) and y the output protein concentration. Time evolution of the output protein concentration as a result of both models is presented in Fig. 3.

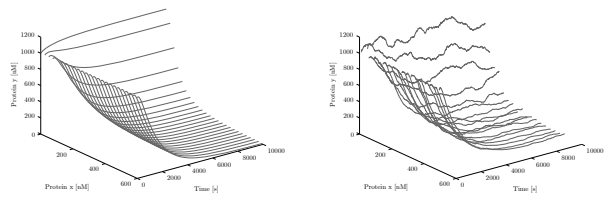


Fig. 3. Time evolution of a deterministic (a) and a stochastic (b) model of biological inverter.

2) Toggle switch: The earlier introduced inverter can be extended with an additional operator O_R2. The output protein is expressed only when both operators, i.e. O_R2 and O_R3 are not bound with their repressors. The described structure represents a biological NOR, i.e. negative OR, gate. A biological toggle switch represents a potential memory element and can be realized with two biological NOR gates mutually repressing each other over operator O_R2. Switches from one logic state to the other can be performed with an introduction of external repressors which bind to operator O_R3.

The observed reactions present an extension of the reactions used in a biological inverter (see supplementary material). The main difference is the additional operator O_R2 to which an external repressor is bound. The association reaction rate of the operator O_R2 equals $k_{a_2} = 0.012 s^{-1} nM^{-2}$ and dissociation reaction rate equals $k_{d_2} = 0.4791 s^{-1}$. A stochastic model can be built with the CME and is solved numerically with the SSA algorithm. A deterministic model can be established with the following set of ODEs:

$$\frac{dm_x}{dt} = \frac{\alpha}{1 + \left(\frac{y}{K_{d_2}}\right)^{\gamma} + \left(\frac{i}{K_{d_3}}\right)^{\gamma}} - \delta_m \cdot m_x, \tag{12}$$

$$\frac{dm_y}{dt} = \frac{\alpha}{1 + \left(\frac{x}{K_{d_2}}\right)^{\gamma} + \left(\frac{j}{K_{d_3}}\right)^{\gamma}} - \delta_m \cdot m_y, \tag{13}$$

$$\frac{dx}{dt} = \beta \cdot m_x - \delta \cdot x, \tag{14}$$

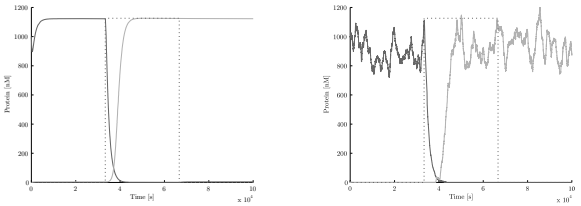
$$\frac{dy}{dt} = \beta \cdot m_y - \delta \cdot y, \tag{15}$$

$$K_{d_2} = \left(\frac{k_{d_2}}{k_{a_2}}\right)^{\frac{1}{\gamma}}, \tag{16}$$

$$K_{d_3} = \left(\frac{k_{d_3}}{k_{a_3}}\right)^{\frac{1}{\gamma}}, \tag{17}$$

where dm_x and dm_y present the changes in mRNA concentration of protein x and y respectively, dx and dy the

changes of protein concentrations, x and y the concentrations of output proteins, i and j the concentrations of input proteins that can cause the transitions from one toggle switch state, and m_x and m_y the mRNA concentrations. Constants K_{d_2} and K_{d_3} are dissociation constants for operators O_R2 and O_R3 respectively. The time evolution of the output protein in dependence of the input proteins concentrations as a result of both models is presented in Fig. 4.



(a) (b)

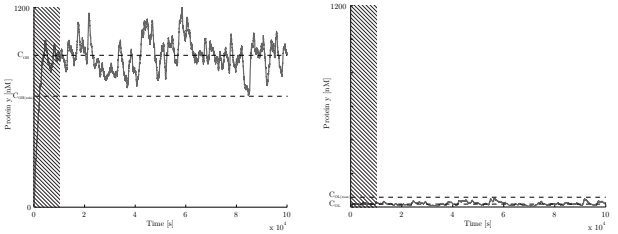
Fig. 4. Time evolution of toggle switch output proteins x and y concentrations presented with solid lines in dependence on input protein i concentrations presented with dotted line. Results are based on the deterministic model (a) and stochastic model (b) of biological toggle switch.

B. Analysis of the elementary modules

Here we will present the detailed analysis of the inverter and toggle switch circuits only. Further details are provided in the supplementary material.

1) Inverter: The dynamics of the concentration of the output protein, i.e. protein y, reflects the inverter behaviour. This same protein can serve as an input, i.e. transcription factor, to some other gene regulatory network. The output logic levels are thus evaluated regarding the concentrations and noise values of protein y. We are measuring the average signal values that are achieved when the output is in the low logic state and when the output is in the high logic state after some initial period of signal stabilisation (see Fig. 5). Based on multiple simulation runs we can set the high level output concentration as $C_{OH} = 915nM$ and low level output concentration as $C_{OL} = 5nM$. In order to compute the minimal high and the maximal low level output concentration, the maximal noise has to be estimated. In our study we estimate the noise only on the basis of chemical stochastics, but other sources of variance could also be included. Different noise estimation approaches could therefore be used straightforwardly in a combination with our analysis procedure. Based on the results of stochastic models, noise present in high level signals equals $N_H = 290nM$ and noise present in low level signals equals $N_L = 45nM$. The minimal high level output concentration is therefore $C_{OH(\min)} = 915nM - 290nM = 625nM$ and the maximal low level output concentration $C_{OL(\max)} = 5nM + 45nM = 50nM$.

The input concentration levels are evaluated regarding the concentrations of protein x that succeed to bring the output protein y to a valid logic state. As we are dealing with an inverter we can evaluate the minimal high level input



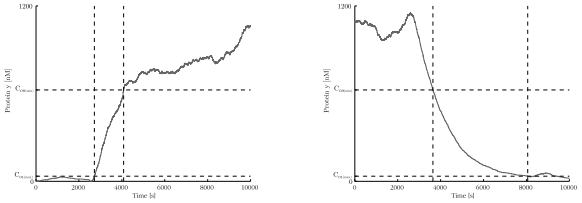
(a) (b)

Fig. 5. Evaluating the output logic levels for biological inverter on a single simulation run. Shaded areas present the signal initialisation period only after which the measurements start.

concentration as the minimal high concentration of protein x that still brings protein y to a low logic state, $C_{IH(\min)} = 330nM$. Similarly we can evaluate the maximal low level input concentration as $C_{IL(\max)} = 50nM$.

The input consumption is evaluated regarding the maximal concentration of the consumed input protein. The inverter only consumes 2 molecules of protein x, which equals approximately $2nM$ for the presumed reaction volume (for details see supplementary material).

The switching times are interpreted as the maximal time the signal is located within the uncertainty region after the switch is initiated. Based on results of multiple simulation runs of the stochastic model the rise time of the biological inverter is $t_r = 1500s$ and the fall time $t_f = 4500s$ (see Fig. 6 (a) and Fig. 6 (b)).



(a) (b)

Fig. 6. Evaluation of the rise time (a) and the fall time (b) in a stochastic model of the biological inverter.

Stability analysis indicates that the system has one steady stable state per input condition only, the analysis from a qualitative perspective is therefore omitted (Sec. II-B).

2) Toggle switch: The analysis from the information processing perspective can be performed in the same manner as in the case of the biological inverter. The main difference is that in the case of the toggle switch we are observing two output signals, i.e. protein x and protein y. The logic levels and noise values are $C_{OH} = 925nM$, $C_{OL} = 0$, $N_H = 335nM$, $N_L = 5nM$, $C_{OH(\min)} = 590nM$, $C_{OL(\max)} = 5nM$, $C_{IH(\min)} = 250nM$ and $C_{IL(\max)} = 40nM$, for both proteins, x and y. Since the same components are used as in the case of the biological inverter input consumption is the same for both inputs, i and j, $C_I = 2nM \approx 2$.

The toggle switch performs a logic switch when one of the outputs transits from the high logic state to the low logic

state and the other one from the low logic state to the high logic state. Since the two outputs are mutually exclusive there is only one switching time, $t_s = \max(t_r, t_f)$, which equals $10000s$ in our example circuit (see Fig. 7).

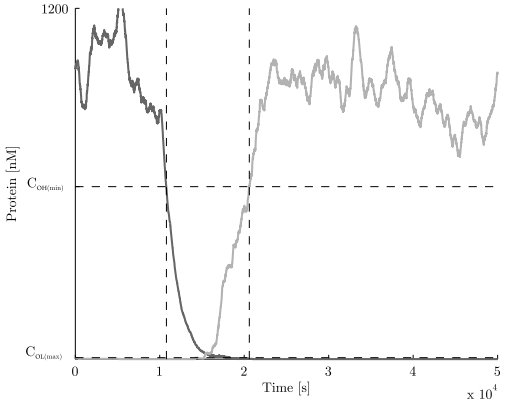


Fig. 7. Evaluation of the time needed to perform a logic switch in biological toggle switch.

Stability analysis indicates that the system has two stable and one unstable steady state. Stable steady states present the states that have a valid logic interpretation. In each state one of the output proteins represents logic value 1 and the other one logic value 0. We can thus estimate the robustness of the system by assessing its robustness regarding the distance between the steady states and robustness regarding the distance from the bifurcation points.

We can determine the location of the system's steady states with the stability analysis:

- $s_1 : x = 3.556nM, y = 1122.634nM$,
- $s_2 : x = 1122.634nM, y = 3.556nM$,
- $s_3 : x = 157nM, y = 157nM$.

A further analysis shows that the states s_1 and s_2 are stable while the state s_3 is unstable. Robustness regarding the distance between the steady states is then

$$R_d = \frac{d_{1,2}}{N} = 3.3405. \tag{18}$$

Since $R_d > 1$ the probability of unexpected switches is low.

The bifurcation points of the system, i.e. parameter values that bring the system from its bistable to monostable behaviour, can be determined with the bifurcation analysis. The intervals for each parameter in which the system reflects bistable behaviour are:

- transcription rate: $\alpha \in [0.029s^{-1}, 0.416s^{-1}]$,
- translation rate: $\beta \in [0.0176s^{-1}, 0.25s^{-1}]$,
- dissociation constant for operator O_R2: $K_{d_2} \in [10.855, 154.572]$,
- dissociation constant for operator O_R3: K_{d_3} does not affect the bistability,
- mRNA degradation rate: $\delta_m \in [0.00067s^{-1}, 0.00954s^{-1}]$,
- protein degradation rate: $\delta \in [0.000119s^{-1}, 0.001713s^{-1}]$.

Only the parameters that have a tendency to vary between the systems of the same type are included in the analysis. The cooperativity factor γ is therefore left out. Robustness regarding the distance from the bifurcation point in then $R_b = 0.59124$. Since $R_b < 1$ we are distant enough from the bifurcation point to obtain the desired system's behaviour even in the presence of noise. The measures R_d and R_b would gain a true value only when comparing our system with some other realization of the circuit with the same logic functionality. They will not be used directly in our further demonstration of the modular design procedure.

C. Modular design of the synchronous toggle switch

We will demonstrate the modular design approach by constructing a synchronous toggle switch circuit. A synchronous toggle switch can be constructed with the synchronization of the toggle switch inputs. This can be achieved with repressilator circuits. In order to perform a switch, a high repressilator signal and a high input signal have to be present at the same time. The basic design of a synchronous toggle switch circuit constructed with the elementary structures described in Sec. III-A is presented in Fig. 8.

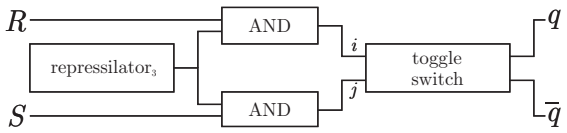


Fig. 8. Basic design of a synchronous toggle switch, where R presents the reset and S the set input and q presents the current state of the toggle switch. The circuit labelled as repressilator$_3$ presents a 3-element repressilator.

The suitability of the presented design can be verified with the verification of constraints presented in Sec. II-C. We will presume that the design is orthogonal to a degree that there is no significant unwanted interference among the basic modules. Reviewing the presented model from the point of view of the aforementioned criteria we can find several flaws in the design (see Table II and supplementary material for details).

The first problem is the logic compatibility between the AND gates and the 3-element repressilator, i.e. the high level output signal of the repressilator is not able to induce a switch in the AND gates (see Equation 7). Another problem arises because of the large input consumption of the AND gates. As the repressilator uses its output also as a feedback input, this can cause incorrect behaviour. We can solve both problems by using intermediate YES gates. These act as a buffer between the repressilator and the AND gates. As the output of the YES gates is only used by the AND gates it can therefore be fully consumed without causing additional problems. The YES gates thus serve as an insulator module which decreases the retroactivity between the AND gates and the repressilator. With this approach we achieve logic compatibility among the basic modules, but the modified system is still lacking

TABLE II
VALUES OF THE MEASURES THAT AFFECT THE COMPATIBILITY BETWEEN THE BASIC MODULES IN THE DESIGN PRESENTED IN FIG. 8.

module	$C_{OL(\max)}$	$C_{OH(\min)}$	$C_{IL(\max)}$	$C_{IH(\min)}$	t_s	C_I	t_v
repressilator$_3$	$15nM$	$330nM$					$2500s$
AND	$5nM$	$505nM$	$100nM$	$500nM$	$5000s$	∞	∞
toggle switch	$5nM$	$590nM$	$40nM$	$250nM$	10^4s	$2nM$	∞

in switching compatibility. The validity duration of the 3-element repressilator's output is too short to perform a full switch in the other modules. This problem can be solved by using a repressilator with 5 elements. The 5-element repressilator produces an output signal with more stable logic levels, longer oscillation periods, i.e. $t_v = 10000s$ and exhibits a higher robustness according to its qualitative analysis (see supplementary material). The modified design of the synchronous toggle switch circuit is presented in Fig. 9.

A further analysis shows that the extended design fulfils all of the constraints given in Sec. II-C (see Table III) and that the simulated behaviour is in accordance with our expectations. The simulation results of a stochastic model of the proposed design are presented in Fig. 10.

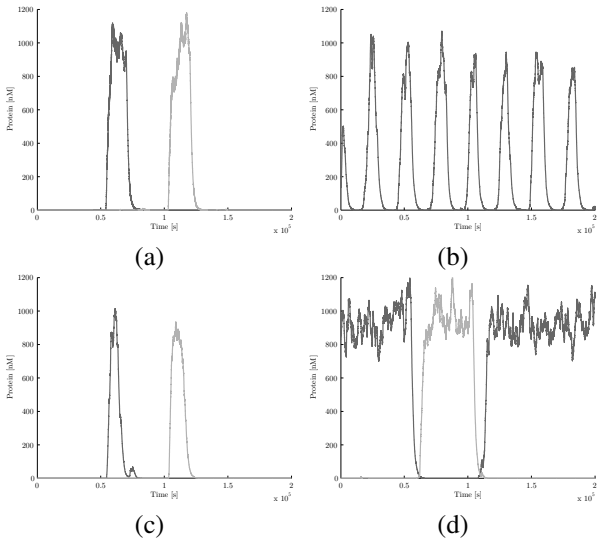


Fig. 10. Simulation results of the designed synchronous toggle switch obtained with the stochastic modelling approach; (a) shows the time evolution of the input signals R and S, (b) the time evolution of the 5-element repressilator's output, (c) the time evolution of the toggle switch input signals i and j, and (d) the time evolution of the output signals q and $\overline{q}$.

IV. CONCLUSIONS

This paper presents a new approach for the analysis and design of BSIPs. The analysis procedure allows one to objectively evaluate the information processing capabilities of a given set of biological systems from the speed, robustness and compatibility points of view. Modular design of more complex biological systems constructed from basic components can thus be established faster and more straightforwardly. The analysis can be performed using any experimental data obtained from laboratory experiments, however, if the analysis is performed on simulation results obtained from the established models they have to be accurate enough to reflect the actual dynamics in the observed biological systems.

The analysis procedure is demonstrated through an initial study of elementary modules and through modular design of a synchronous toggle switch. Verification with the established methods is able to identify the shortcomings of the initially proposed design and guide its improvements through which the problems are eliminated and an appropriate simulated behaviour achieved. As indicated by our results, no additional verification of the target design is needed after the necessary constraints are satisfied. Only the elementary modules therefore need to be analysed and the computational time needed to verify the final design can thus be drastically reduced. In our specific case the analysis and design procedures were performed manually, however, both processes will be automatized in the near future. A computational tool that is being developed will be able to automatically evaluate a given set of biological modules and automatize the modular design of functionally more complex BSIPs.[1]

ACKNOWLEDGMENT

Authors would like to thank Iztok Lebar Bajec and Mattia Petroni for many useful comments. We also thank the reviewers for their suggestions that improved the content and readability of the paper. The research was partially supported by the scientific-research programme Pervasive Computing (P2-0359) financed by the Slovenian Research Agency in the years from 2009 to 2013.

REFERENCES

[1] G. Rodrigo, J. Carrera, T. E. Landrain, and A. Jaramillo, "Perspectives on the automatic design of regulatory systems for synthetic biology," *FEBS Letters*, vol. 586, pp. 2037–2042, 2012.
[2] A. S. Khalil and J. J. Collins, "Synthetic biology: Applications come of age," *Nature Reviews Genetics*, vol. 11(5), pp. 367–379, 2010.

[1]The code used in this paper is available at http://lrss.fri.uni-lj.si/bio/material/tcbb.zip under the Creative Commons Attribution license (see also supplementary material).

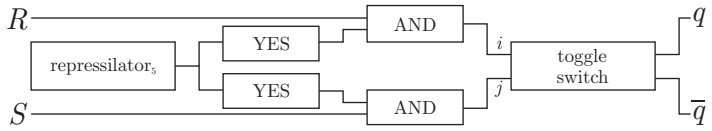


Fig. 9. Modified design of synchronous toggle switch, where R presents reset and S set input and q presents the current state of the toggle switch.

TABLE III
VALUES OF MEASURES THAT AFFECT THE COMPATIBILITY BETWEEN THE BASIC MODULES IN THE MODIFIED DESIGN PRESENTED IN FIG. 9.

module	$C_{OL(\max)}$	$C_{OH(\min)}$	$C_{IL(\max)}$	$C_{IH(\min)}$	t_s	C_I	t_v
repressilator$_5$	$15nM$	$795nM$					$10000s$
AND	$5nM$	$505nM$	$100nM$	$500nM$	$5000s$	∞	∞
toggle switch	$5nM$	$590nM$	$40nM$	$250nM$	10^4s	$2nM$	∞
YES	$20nM$	$580nM$	$15nM$	$125nM$	$5000s$	$2nM$	∞

[3] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in Escherichia coli," *Nature*, vol. 403, pp. 339–342, 2000.

[4] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, pp. 335–338, 2000.

[5] D. Chandran, F. Bergmann, and H. Sauro, "Tinkercell: modular CAD tool for synthetic biology," *Journal of Biological Engineering*, vol. 3, no. 1, p. 19, 2009.

[6] P. Mendes, S. Hoops, S. Sahle, R. Gauges, J. Dada, and U. Kummer, "Computational modeling of biochemical networks using COPASI," in *Systems Biology*, ser. Methods in Molecular Biology, I. V. Maly, Ed. Humana Press, 2009, vol. 500, pp. 17–59.

[7] M. A. Marchisio and J. Stelling, "Automatic design of digital synthetic gene circuits," *PLoS Computational Biology*, vol. 7(2), 2011.

[8] L. Huynh, J. Kececioglu, M. Koppe, and I. Tagkopoulos, "Automatic design of synthetic gene circuits through mixed integer non-linear programming," *PLoS ONE*, vol. 7(4), 2012.

[9] G. Rodrigo and A. Jaramillo, "AutoBioCAD: Full biodesign automation of genetic circuits," *ACS Synthetic Biology*, in press 2013.

[10] G. Rodrigo, J. Carrera, and A. Jaramillo, "Computational design of synthetic regulatory networks from a genetic library to characterize the designability of dynamical behaviors," *Nucleic Acids Research*, vol. 39, 2011.

[11] J. Beal, R. Weiss, D. Densmore, A. Adler, E. Appleton, J. Babb, S. Bhatia, N. Davidsohn, T. Haddock, J. Loyall, R. Schantz, V. Vasilev, and F. Yaman, "An end-to-end workflow for engineering of biological networks from high-level specifications," *ACS Synthetic Biology*, vol. 1, no. 8, pp. 317–331, 2012.

[12] D. Del Vecchio, A. J. Ninfa, and E. D. Sontag, "Modular cell biology: retroactivity and insulation," *Molecular Systems Biology*, vol. 4:161, 2008.

[13] P. Marguet, F. Balagadde, C. Tan, and L. You, "Biology by design: reduction and synthesis of cellular components and behaviour," *J. R. Soc. Interface*, vol. 4, pp. 607–623, 2007.

[14] A. K. Maini, *Digital Electronics: Principles, Devices and Applications*. John Wiley & Sons, 2007.

[15] R. F. Tinder, *Engineering Digital Design, Revised Second Edition*. Academic Press, 2000.

[16] J. F. Wakerly, *Digital Design: Principles and Practices Package, 4th Edition*. Prentice Hall International, Inc., 2005.

[17] L. Edelstein-Keshet, *Mathematical Models in Biology*. Siam, Philadelphia, 2005.

[18] P. Fall, E. S. Marland, J. M. Wagner, and J. J. Tyson, *Computational Cell Biology (Interdisciplinary Applied Mathematics)*. Springer, 2002.

[19] S. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, ser. Studies in nonlinearity. Perseus Books Group, 2008.

[20] J. Kim, D. G. Bates, I. Postlethwaite, L. Ma, and P. A. Iglesias, "Robustness analysis of biochemical network models," *IEE Proceedings Systems Biology*, vol. 153(3), pp. 96–104, 2006.

[21] B. Wang, R. I. Kitney, N. Joly, and M. Buck, "Engineering modular and orthogonal genetic logic gates for robust digital-like synthetic biology," *Nature Communications*, vol. 2, pp. 508–516, 2011.

[22] F. Yaman, S. Bhatia, A. Adler, D. Densmore, and J. Beal, "Automated selection of synthetic biology parts for genetic regulatory networks," *ACS Synthetic Biology*, vol. 1, no. 8, pp. 332–344, 2012.

[23] E. Franco, E. Friedrichs, J. Kim, R. Jungmann, R. Murray, E. Winfree, and F. C. Simmel, "Timing molecular motion and production with a synthetic transcriptional clock," *Proceedings of the National Academy of Sciences*, 2011.

[24] H. S. Booth, C. J. Burden, M. Hegland, and L. Santoso, *Mathematical Modeling of Biological Systems, Volume I*. Birkhauser Boston, 2007, ch. A Stochastic Model of Gene Regulation Using the Chemical Master Equation, pp. 71–81.

[25] H. Kitano, "Perspectives on systems biology," *New Generation Computing*, vol. 18, pp. 199–216, 2000.

[26] U. Alon, *An Introduction to Systems Biology*. Chapman & Hall/CRC, 2007.

[27] J. Stricker, S. Cookson, M. R. Bennett, W. H. Mather, L. S. Tsimring, and J. Hasty, "A fast, robust and tunable synthetic gene oscillator," *Nature*, vol. 456, pp. 516–520, 2008.

[28] M. Tigges, T. T. Marquez-Lago, J. Stelling, and M. Fussenegger, "A tunable synthetic mammalian oscillator," *Nature*, vol. 457, pp. 309–312, 2009.

[29] H. El Samad, M. Khammash, L. Petzold, and D. Gillespie, "Stochastic modeling of gene regulatory networks," *International Journal of Robust and Nonlinear Control*, vol. 15, pp. 691–711, 2005.

[30] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The Journal of Physical Chemistry*, vol. 81, pp. 2340–2361, 1977.

Miha Moškon was born on October 28, 1983 in Ljubljana, Slovenia. He received his BSc degree in Computer Science from the Faculty of Computer and Information Science, University of Ljubljana, Slovenia in 2007 and his PhD in 2012.

He is currently employed as a teaching assistant in the Computer Structures and Systems Laboratory at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia, where he is in charge of the several laboratory courses. He is also a member of Computational biology group at the same faculty. His research interests are mainly directed towards unconventional processing methods and computational biology. He has published his work in several international journals and conference proceedings

Miha Mraz was born on August 20, 1966 in Ljubljana, Slovenia. He received his BSc, MSc and PhD degree in computer science from the Faculty of Computer and Information Science, University of Ljubljana, Slovenia in 1992, 1995 and 2000.

He is currently employed as a full professor at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia and is also a head of Computational biology group at the same faculty. His research interests are recently directed towards unconventional processing methods, such as fuzzy logic, synthetic biological systems and QCA structures. He has published his work in many distinguished journals such as Nanotechnology, Journal of Theoretical Biology, International Journal on Unconventional computing, Japanese journal of applied physics, etc. He is also in the editorial board of newly founded Journal of Synthetic Biology. He is a member of IEEE professional society.

SYSTEMATIC APPROACH TO COMPUTATIONAL DESIGN OF GENE REGULATORY NETWORKS WITH INFORMATION PROCESSING CAPABILITIES
Supplementary material

Miha Moškon and Miha Mraz
University of Ljubljana
Faculty of Computer and Information Science
Tržaška cesta 25, SI-1001 Ljubljana, Slovenia
e-mail: miha.moskon@fri.uni-lj.si

1 Parameters and units

We presume a fixed reaction volume in all the models, i.e. $V = 2 \cdot 10^{-15}L$. We used the following reaction rates [1, 2]:

- association of protein with operator O_R2: $k_{a_2} = 0.012s^{-1}nM^{-2}$,
- dissociation of protein from operator O_R2: $k_{d_2} = 0.4791s^{-1}$,
- association of protein with operator O_R3: $k_{a_3} = 0.00012s^{-1}nM^{-2}$,
- dissociation of protein from operator O_R3: $k_{d_3} = 0.4791s^{-1}$,
- binding of inducer to activator: $k_{b_i} = 0.00012s^{-1}nM^{-1}$,
- unbinding of inducer from activator: $k_{u_i} = 0.4791s^{-1}$,
- mRNA transcription: $k_{trs} = 0.0715s^{-1}$,
- protein translation: $k_{trl} = 0.043s^{-1}$,
- protein degradation: $k_{deg} = 0.0007s^{-1}$,
- mRNA degradation: $k_{deg_m} = 0.0039s^{-1}$.

1.1 Deterministic models

Reaction parameters used in the determinstic models are as follows:

- maximal transcription rate: $\alpha = k_{trs} = 0.0715s^{-1}$,
- translation rate: $\beta = k_{trl} = 0.043s^{-1}$,
- Hill coefficient for activator and repressor: $\gamma = 2$,
- Hill coefficient for inducer: $\gamma_i = 1$,
- O_R2 dissociation constant : $K_d = K_{d_2} = \left(\frac{k_{d_2}}{k_{a_2}}\right)^{\frac{1}{\gamma}} = 6.3168$,
- O_R3 dissociation constant: $K_d = K_{d_3} = \left(\frac{k_{d_3}}{k_{a_3}}\right)^{\frac{1}{\gamma}} = 63.1862$,
- inducer-activator dissociation constant: $K_d = K_{d_i} = \left(\frac{k_{u_i}}{k_{b_i}}\right)^{\frac{1}{\gamma_i}} = 3992.5$,
- protein degradation rate: $\delta = k_{deg} = 0.0007s^{-1}$,
- mRNA degradation rate: $\delta_m = k_{deg_m} = 0.0039s^{-1}$.

1.2 Stochastic models

The concentrations of chemical species are observed in different units in deterministic models, where molar concentration is used (M), than in stochastic models, where numbers of molecules are used. In order to compare the results of both modelling approaches, concentrations obtained with stochastic modelling approaches were converted to units used in deterministic models using the following equation:

$$x_i(t) = \langle y_i(t) \rangle \cdot \Omega, \tag{1}$$

where $x_i(t)$ defines the molar concentration of the chemical species i, $\langle y_i(t) \rangle$ average number of molecules of the chemical species i at time t and $\Omega = V \cdot N_A$ (N_A is Avogadro constant).

Conversion of reaction rates to reaction parameters used in stochastic models is dependant on the number of reactants in an observing reaction [3]. Reaction parameters used in the stochastic models can be obtained as follows:

- association of protein with operator O_R2: $c_{a_2} = 2 \cdot \frac{k_{a_2}}{\Omega^2} = 0.0165s^{-1}$,
- dissociation of protein from operator O_R2: $c_{d_2} = k_{d_2} = 0.4791s^{-1}$,
- association of protein with operator O_R3: $c_{a_3} = 2 \cdot \frac{k_{a_3}}{\Omega^2} = 1.6544 \cdot 10^{-4}s^{-1}$,
- dissociation of protein from operator O_R3: $c_{d_3} = k_2 = 0.4791s^{-1}$,

- binding of inducer to activator: $c_{b_i} = \frac{k_{b_i}}{\Omega} = 1.4453 \cdot 10^{-4} s^{-1}$,
- unbinding of inducer from activator: $c_{u_i} = k_{u_i} = 0.4791 s^{-1}$,
- mRNA transcription: $c_{trs} = k_{trs} = 0.0715 s^{-1}$,
- protein translation: $c_{trl} = k_{trl} = 0.043 s^{-1}$,
- protein degradation: $c_{deg} = k_{deg} = 0.0007 s^{-1}$,
- mRNA degradation: $c_{deg_m} = k_{deg_m} = 0.0039 s^{-1}$.

2 Models

2.1 YES gate

Observed reactions for the YES gate are presented in Table 1.

Table 1: Reactions that present the model of biological driver (YES gate).

reaction	**description**
$2x$+DNA$\xrightarrow{k_{a_3}} x_2$DNA	association of protein x with O_R3
x_2DNA$\xrightarrow{k_{d_3}} 2x$+DNA	dissociation of protein x from O_R3
x_2DNA$\xrightarrow{k_{trs}}$mRNA+DNA	transcription of mRNA of protein y
mRNA$\xrightarrow{k_{trl}}$ mRNA+y	translation of protein y
mRNA$\xrightarrow{k_{deg_m}} \emptyset$	degradation of mRNA of protein y
$y \xrightarrow{k_{deg}} \emptyset$	degradation of protein y

The deterministic model can be established with the following equations:

$$\frac{dm}{dt} = \alpha \frac{x^\gamma}{K_{d_3}^\gamma + x^\gamma} - \delta_m \cdot m, \tag{2}$$

$$\frac{dy}{dt} = \beta \cdot m - \delta_y \cdot y. \tag{3}$$

2.2 NOR gate

Observed reactions for the NOR gate are presented in Table 2.

The deterministic model can be established with the following equations:

$$\frac{dm}{dt} = \frac{\alpha}{1 + \left(\frac{x}{K_{d_3}}\right)^\gamma + \left(\frac{y}{K_{d_3}}\right)^\gamma} - \delta_m \cdot m, \tag{4}$$

Table 2: Reactions that present the model of biological NOR gate.

reaction	**description**
$2x$+DNA$\xrightarrow{k_{a_3}} x_2$DNA	association of protein x with operator O_R3
x_2DNA$\xrightarrow{k_{d_3}} 2x$+DNA	dissociation of protein x from O_R3
$2y$+DNA$\xrightarrow{k_{a_3}} y_2$DNA	association of protein y with operator O_R3
y_2DNA$\xrightarrow{k_{d_3}} 2y$+DNA	dissociation of protein y from O_R3
DNA$\xrightarrow{k_{trs}}$mRNA+DNA	transcription of mRNA of protein z
mRNA$\xrightarrow{k_{trl}}$ mRNA+z	translation of protein z
mRNA$\xrightarrow{k_{deg_m}} \emptyset$	degradation of mRNA of protein z
$z \xrightarrow{k_{deg}} \emptyset$	degradation of protein z

$$\frac{dz}{dt} = \beta \cdot m - \delta_z \cdot z. \tag{5}$$

2.3 AND gate

Observed reactions for the AND gate are presented in Table 3.

Table 3: Reactions that present the model of biological AND gate.

reaction	**description**
$x + y \xrightarrow{k_{b_i}} x^*$	binding of inducer y to activator x
$x^* \xrightarrow{k_{u_i}} x + y$	unbinding of inducer y from activator x
$2x^*$+DNA$\xrightarrow{k_{a_3}} x_2^*$DNA	associaction of complex with O_R3
x_2^*DNA$\xrightarrow{k_{d_3}} 2x^*$+DNA	disociation of complex from O_R3
x_2^*DNA$\xrightarrow{k_{trs}}$mRNA+x_2^*DNA	transcription of mRNA of protein z
mRNA$\xrightarrow{k_{trl}}$ mRNA+z	translation of protein z
mRNA$\xrightarrow{k_{deg_m}} \emptyset$	degradation of mRNA of protein z
$z \xrightarrow{k_{deg}} \emptyset$	degradation of protein z

The deterministic model can be established with the following equations:

$$x^* = x \cdot \frac{y^{\gamma_i}}{y^{\gamma_i} + {K_{d_i}}^{\gamma_i}}, \tag{6}$$

$$\frac{dm}{dt} = \frac{\alpha \cdot {x^*}^{\gamma}}{K_{d_3}^{\gamma} + {x^*}^{\gamma}} - \delta_m \cdot m, \tag{7}$$

$$\frac{dz}{dt} = \beta \cdot m - \delta_z \cdot z. \tag{8}$$

2.4 Toggle switch

Observed reactions for the toggle switch are presented in Table 4.

Table 4: Reactions that present the model of toggle switch.

reaction	**description**
$2x$+DNA$_y \xrightarrow{k_{a_2}} x_2$ DNA$_y$	association of protein x with O_R2 of gene y
$2y$+DNA$_x \xrightarrow{k_{a_2}} y_2$ DNA$_x$	association of protein y with O_R2 of gene x
x_2 DNA$_y \xrightarrow{k_{d_2}} 2x$+DNA$_y$	dissociation of protein x from O_R2 of gene y
y_2 DNA$_x \xrightarrow{k_{d_2}} 2y$+DNA$_x$	dissociation of protein y from O_R2 of gene x
$2i$+DNA$_x \xrightarrow{k_{a_3}} i_2$ DNA$_x$	association of protein i with O_R3 of gene x
$2j$+DNA$_y \xrightarrow{k_{a_3}} j_2$ DNA$_y$	association of protein j with O_R3 of gene y
i_2 DNA$_x \xrightarrow{k_{d_3}} 2i$+DNA$_x$	dissociation of protein i from O_R3 of gene x
j_2 DNA$_y \xrightarrow{k_{d_3}} 2j$+DNA$_y$	dissociation of protein j from O_R3 of gene y
DNA$_x \xrightarrow{k_{trs}}$ DNA$_x$+mRNA$_x$	transcription of mRNA of protein x
DNA$_y \xrightarrow{k_{trs}}$ DNA$_y$+mRNA$_y$	transcription of mRNA of protein y
mRNA$_x \xrightarrow{k_{trl}}$ mRNA$_x + x$	translation of protein x
mRNA$_y \xrightarrow{k_{trl}}$ mRNA$_y + y$	translation of protein y
$x \xrightarrow{k_{deg}} \emptyset$	degradation of protein x
$y \xrightarrow{k_{deg}} \emptyset$	degradation of protein y
mRNA$_x \xrightarrow{k_{deg_m}} \emptyset$	degradation of mRNA of protein x
mRNA$_y \xrightarrow{k_{deg_m}} \emptyset$	degradation of mRNA of protein y

2.5 Repressilator

Observed reactions for the repressilator with 3 elements are presented in Table 5.

The deterministic model of the repressilator with 3 elements can be established with the following equation:

$$\frac{dm_x}{dt} = \frac{\alpha_x}{1 + \left(\frac{z}{K_{d_2}}\right)^{\gamma}} - m_x \cdot \delta_{m_x}, \tag{9}$$

$$\frac{dm_y}{dt} = \frac{\alpha_y}{1 + \left(\frac{x}{K_{d_2}}\right)^{\gamma}} - m_y \cdot \delta_{m_y}, \tag{10}$$

Table 5: Reactions that present the model of a repressilator with 3 elements.

$2x$+DNA$_y$ $\xrightarrow{k_{a_2}}$ x_2 DNA$_y$
$2y$+DNA$_z$ $\xrightarrow{k_{a_2}}$ y_2 DNA$_z$
$2z$+DNA$_x$ $\xrightarrow{k_{a_2}}$ z_2 DNA$_x$
x_2 DNA$_y$ $\xrightarrow{k_{d_2}}$ $2x$+DNA$_y$
y_2 DNA$_z$ $\xrightarrow{k_{d_2}}$ $2y$+DNA$_z$
z_2 DNA$_x$ $\xrightarrow{k_{d_2}}$ $2z$+DNA$_x$
DNA$_x$ $\xrightarrow{k_{trs}}$ DNA$_x$+mRNA$_x$
DNA$_y$ $\xrightarrow{k_{trs}}$ DNA$_y$+mRNA$_y$
DNA$_z$ $\xrightarrow{k_{trs}}$ DNA$_z$+mRNA$_z$
mRNA$_x$ $\xrightarrow{k_{trl}}$ mRNA$_x$ + x
mRNA$_y$ $\xrightarrow{k_{trl}}$ mRNA$_y$ + y
mRNA$_z$ $\xrightarrow{k_{trl}}$ mRNA$_z$ + z
x $\xrightarrow{k_{deg}}$ $\emptyset$
y $\xrightarrow{k_{deg}}$ $\emptyset$
z $\xrightarrow{k_{deg}}$ $\emptyset$
mRNA$_x$ $\xrightarrow{k_{deg_m}}$ $\emptyset$
mRNA$_y$ $\xrightarrow{k_{deg_m}}$ $\emptyset$
mRNA$_z$ $\xrightarrow{k_{deg_m}}$ $\emptyset$

$$\frac{dm_z}{dt} = \frac{\alpha_z}{1 + \left(\frac{y}{K_{d_2}}\right)^{\gamma}} - m_z \cdot \delta_{m_z}, \tag{11}$$

$$\frac{dx}{dt} = \beta_x \cdot m_x - \delta_x \cdot x, \tag{12}$$

$$\frac{dy}{dt} = \beta_y \cdot m_y - \delta_y \cdot y, \tag{13}$$

$$\frac{dz}{dt} = \beta_z \cdot m_z - \delta_z \cdot z. \tag{14}$$

The model can be extended straightforwardly to a model of the repressilator with 5 elements.

3 Evaluation of the introduced measures

3.1 NOT gate

Measures evaluated on the NOT gate model:

- $C_{OL} = 5nM$,
- $C_{OH} = 915nM$,
- $N_L = 45nM$,
- $N_H = 290nM$,
- $N = 290nM$,
- $C_{OL(\max)} = 5nM + 45nM = 50nM$,
- $C_{OH(\min)} = 915nM - 290nM = 625nM$,
- $NM_O =]50nM, 625nM[$,
- $C_{IL(\max)} = 50nM$,
- $C_{IH(\min)} = 330nM$,
- $NM_I =]50nM, 330nM[$,
- $C_I = 2nM$,
- $t_r = 1500s$,
- $t_f = 4500s$,
- $t_s = 4500s$,
- $t_v = \infty$.

3.2 YES gate

Measures evaluated on the YES gate model:

- $C_{OL} = 5nM$,
- $C_{OH} = 915nM$,
- $N_L = 15nM$,
- $N_H = 335nM$,
- $N = 335nM$,

- $C_{OL(\max)} = 5nM + 15nM = 20nM$,
- $C_{OH(\min)} = 915nM - 335nM = 580nM$,
- $NM_O =]20nM, 580nM[$,
- $C_{IL(\max)} = 15nM$,
- $C_{IH(\min)} = 125nM$,
- $NM_I =]15nM, 125nM[$,
- $C_I = 2nM$,
- $t_r = 2500s$,
- $t_f = 5000s$,
- $t_s = 5000s$,
- $t_v = \infty$.

3.3 NOR gate

Measures evaluated on the NOR gate model:

- $C_{OL} = 5nM$,
- $C_{OH} = 915nM$,
- $N_L = 45nM$,
- $N_H = 290nM$,
- $N = 290nM$,
- $C_{OL(\max)} = 5nM + 45nM = 50nM$,
- $C_{OH(\min)} = 915nM - 290nM = 625nM$,
- $NM_O =]50nM, 625nM[$,
- $C_{IL(\max)} = 35nM$,
- $C_{IH(\min)} = 330nM$,
- $NM_I =]35nM, 330nM[$,
- $C_{I_x} = C_{I_y} = 2nM$,
- $t_r = 1500s$,
- $t_f = 4500s$,
- $t_s = 4500s$,
- $t_v = \infty$.

3.4 AND gate

Measures evaluated on the AND gate model:

- $C_{OL} = 0nM$,
- $C_{OH} = 840nM$,
- $N_L = 5nM$,
- $N_H = 335nM$,
- $N = 335nM$,
- $C_{OL(\max)} = 0nM + 5nM = 5nM$,
- $C_{OH(\min)} = 840nM - 335nM = 505nM$,
- $NM_O =]5nM, 505nM[$,
- $C_{IL(\max)} = 100nM$,
- $C_{IH(\min)} = 500nM$,
- $NM_I =]100nM, 500nM[$,
- $C_{I_x} = C_{I_y} = \infty nM$,
- $t_r = 1500s$,
- $t_f = 5000s$,
- $t_s = 5000s$,
- $t_v = \infty$.

3.5 Toggle switch

Measures evaluated on the toggle switch model:

- $C_{OL} = 0nM$,
- $C_{OH} = 925nM$,
- $N_L = 5nM$,
- $N_H = 335nM$,
- $N = 335nM$,
- $C_{OL(\max)} = 0nM + 5nM = 5nM$,

- $C_{OH(\min)} = 925nM - 335nM = 590nM$,
- $NM_O =]5nM, 590nM[$,
- $C_{IL(\max)} = 40nM$,
- $C_{IH(\min)} = 250nM$,
- $NM_I =]40nM, 250nM[$,
- $C_{I_i} = C_{I_j} = 2nM$,
- $t_s = 10000s$,
- $t_v = \infty$,
- $R_d \approx \frac{d_{1,2}}{N} = 3.3405$,
- $R_b \approx 0.59124$.

3.6 Repressilator with 3 elements

Measures evaluated on the 3-element repressilator model:

- $C_{OL} = 5nM$,
- $C_{OH} = 660nM$,
- $N_L = 10nM$,
- $N_H = 330nM$,
- $N = 335nM$,
- $C_{OL(\max)} = 5nM + 10nM = 15nM$,
- $C_{OH(\min)} = 660nM - 330nM = 330nM$,
- $NM_O =]15nM, 330nM[$,
- $t_v = 2500s$,
- $R_c \approx \frac{d_c}{N} = 1.3$,
- $R_b \approx 0.8$.

3.7 Repressilator with 5 elements

Measures evaluated on the 5-element repressilator model:

- $C_{OL} = 5nM$,
- $C_{OH} = 995nM$,
- $N_L = 10nM$,
- $N_H = 200nM$,
- $N = 200nM$,
- $C_{OL(\max)} = 5nM + 10nM = 15nM$,
- $C_{OH(\min)} = 995nM - 200nM = 795nM$,
- $NM_O =]15nM, 795nM[$,
- $t_v = 10000s$,
- $R_c \approx \frac{d_c}{N} = 5.475$,
- $R_b \approx 0.83$.

4 Supplementary code

The code used in the paper is available at lrss.fri.uni-lj.si/bio/material/tcbb.zip under the Creative Commons Attribution license. Help is provided for each file and can be accessed with the command `help filename`. We only give a brief description of the available files here.

- `and_det.m`: deterministic model of and gate,
- `and_det_fall.m`: estimates the signal falling time of and gate on the basis of deterministic model,
- `and_det_rise.m`: estimates the signal rising time of and gate on the basis of deterministic model,
- `and_det_steady.m`: plots the steady states of the and gate depending on the input,
- `and_driver_ssa_inputs.m`: stochastic model of the and gate buffered by the biological driver (inputs are forced by external values),
- `and_inputs.m`: evalutes the input logic levels of and gate,
- `and_outputs.m`: evaluates the output logic levels and noise of and gate,

- `and_ssa.m`: stochastic model of the AND gate,
- `and_ssa_fall.m`: estimates the falling time of and gate on the basis of stochastic model,
- `and_ssa_rise.m`: estimates the rising time of and gate on the basis of stochastic model,
- `conc_mol_to_num.m`: converts number of molecules to moles,
- `conc_num_to_mol.m`: converts moles to number of molecules,
- `driver_det.m`: deterministic model of the biological driver,
- `driver_det_fall.m`: estimates the falling time of biological driver on the basis of deterministic model,
- `driver_det_multiple.m`: multiple simulations of deterministic driver with different inputs,
- `driver_det_rise.m`: estimates the rising time of biological driver on the basis of deterministic model,
- `driver_det_steady.m`: plots the steady states of the biological driver depending on the input,
- `driver_inputs.m`: evaluates the input logic levels of biological driver,
- `driver_multiple.m`: multiple stochastic runs of biological driver,
- `driver_outputs.m`: evaluates the output logic levels and noise of biological driver,
- `driver_ssa.m`: stochastic model of the biological driver,
- `driver_ssa_inputs.m`: stochastic model of the biological driver (inputs are forced by external values),
- `driver_ssa_rise.m`: estimates the rising time of biological driver on the basis of stochastic model,
- `inverter_det.m`: deterministic model of the biological inverter,
- `inverter_det_fall.m`: estimate the falling time of biological inverter on the basis of deterministic model,
- `inverter_det_multiple.m`: multiple simulations of deterministic driver with different inputs,
- `inverter_det_rise.m`: estimates the rising time of biological inverter on the basis of deterministic model,

- `inverter_det_steady.m`: plots the steady states of the biological driver depending on the given input,
- `inverter_inputs.m`: evaluates the input logic levels of biological inverter,
- `inverter_multiple.m`: multiple stochastic runs of biological inverter,
- `inverter_outputs.m`: evaluates the output logic levels and noise of biological inverter,
- `inverter_ssa.m`: stochastic model of the biological inverter,
- `inverter_ssa_fall.m`: estimates the falling time of biological inverter on the basis of stochastic model,
- `inverter_ssa_inputs.m`: stochastic model of the biological inverter (inputs are forced by external values),
- `inverter_ssa_rise.m`: estimates the rising time of biological inverter on the basis of stochastic model,
- `nor_det.m`: deterministic model of the nor gate,
- `nor_det_fall.m`: estimates the falling time of nor gate on the basis of deterministic model,
- `nor_det_rise.m`: estimates the rising time of nor gate on the basis of deterministic model,
- `nor_det_steady.m`: plots the steady states of the nor gate depending on the given input,
- `nor_inputs.m`: evaluates the input logic levels of nor gate,
- `nor_outputs.m`: evaluates the output logic levels and noise of nor gate,
- `nor_ssa.m`: stochastic model of the NOR gate,
- `nor_ssa_fall.m`: estimates the falling time of nor gate on the basis of stochastic model,
- `nor_ssa_rise.m`: estimates the rising time of nor gate on the basis of stochastic model,
- `repressilator_bifurcation.m`: searches for the location of the bifurcation point in the repressilator model,
- `repressilator_det.m`: deterministic model of the repressilator with 3 elements,
- `repressilator_extended_bifurcation.m`: searches for the location of the bifurcation point in the extended repressilator model,

- `repressilator_extended_det.m`: deterministic model of the repressilator with 5 elements,
- `repressilator_extended_limit_cycle.m`: determines the limit cycle of the extended repressilator,
- `repressilator_extended_outputs.m`: evaluates the output logic levels and noise of extended repressilator,
- `repressilator_extended_ssa.m`: stochastic model of the repressilator with 5 elements,
- `repressilator_extended_validity.m`: evaluates the validity duration of extended repressilator,
- `repressilator_limit_cycle.m`: determines the limit cycle of the repressilator,
- `repressilator_outputs.m`: evaluates the output logic levels and noise of repressilator,
- `repressilator_ssa.m`: stochastic model of the repressilator with 3 elements,
- `repressilator_validity.m`: evaluates the validity duration of repressilator,
- `ssa.m`: SSA implementation,
- `ssa_input.m`: SSA implementation with forced inputs,
- `toggle_bifurcation.m`: searches for the location of the bifurcation point in the toggle switch model,
- `toggle_det.m`: deterministic model of the toggle switch,
- `toggle_det_steady.m`: determines the steady states of the toggle switch,
- `toggle_det_switch.m`: estimates the switching time of toggle switch on the basis of deterministic model,
- `toggle_inputs.m`: evaluates the input logic levels of toggle switch,
- `toggle_outputs.m`: evaluates the output logic levels and noise of toggle switch,
- `toggle_ssa.m`: stochastic model of the toggle switch ,
- `toggle_ssa_inputs.m`: stochastic model of toggle switch (inputs are forced by external values),
- `toggle_ssa_switch.m`: estimates the switching time of toggle switch on the basis of stochastic model,
- `toggle_sync.m`: simulates the behaviour of synchronous toggle switch on the basis of stochastic models.

References

[1] H. S. Booth, C. J. Burden, M. Hegland, and L. Santoso, *Mathematical Modeling of Biological Systems, Volume I.* Birkhauser Boston, 2007, ch. A Stochastic Model of Gene Regulation Using the Chemical Master Equation, pp. 71–81.

[2] J. Goutsias, "Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems," *The Journal of Chemical Physics*, vol. 122, pp. 1–15, 2005.

[3] H. El Samad, M. Khammash, L. Petzold, and D. Gillespie, "Stochastic modeling of gene regulatory networks," *International Journal of Robust and Nonlinear Control*, vol. 15, pp. 691–711, 2005.