

# Reprogrammable Biological Logic Gate that Exploits Noise

Anna Dari<sup>1</sup>, Adi. R. Bulsara<sup>2</sup>, William L. Ditto<sup>1</sup>, and Xiao Wang<sup>1</sup>

<sup>1</sup>*School of Biological and Health Systems Engineering, Arizona State University, Tempe, AZ 85287-9709, USA*

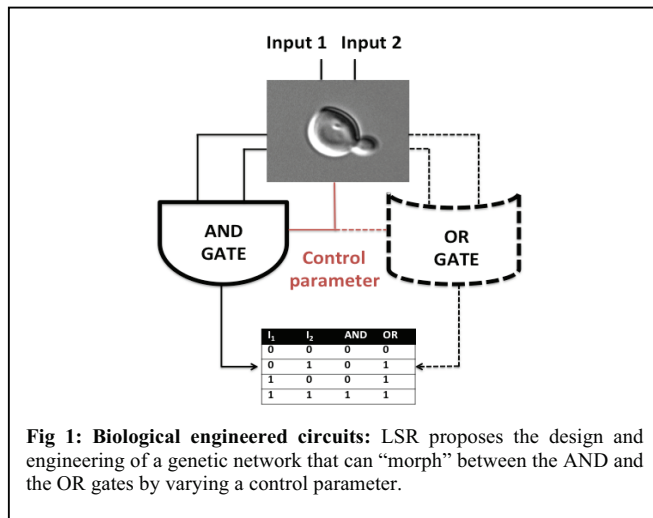
<sup>2</sup>*SPAWAR Systems Center Pacific - San Diego, CA 92152-5001, USA*

xiaowang@asu.edu

**Abstract**— Computation underlies the genetic regulatory network activities. Previous studies have designed and engineered systems that can perform single logic gate functionalities, trying to avoid external and internal random fluctuations. In this work, we demonstrate the possibility to exploit noise when it cannot be eliminated. In particular, we adapt the LSR paradigm to a single-gene network derived from the bacteriophage  $\lambda$  and to a more robust two-gene network derived from the yeast *S. cerevisiae*. Our results demonstrate that in both cases there is an *optimal* amount of noise where the biological logic gate can be externally reprogrammed (i.e. switch from the AND to the OR gate) and perform well according to the truth table.

## I. INTRODUCTION

Synthetic biology offers new strategies to address emerging problems in medical, pharmaceutical, environmental, and industrial areas. For example, a gene regulatory network (GRN) can deliver specific proteins or regulate treatments in response to a particular environment [1]. It can be implemented by just assembling simple components (modules) into more complex GRNs, similar to electronic circuits.



**Fig 1: Biological engineered circuits:** LSR proposes the design and engineering of a genetic network that can “morph” between the AND and the OR gates by varying a control parameter.

The core of this work is to demonstrate, through calculations and simulations, the possibility of building a GRN that can switch between the AND and OR logic gate

functionalities. The intriguing aspect of this *biological logic gate* is its ability to operate consistently and robustly in presence of random fluctuations and *to change the logic gate functionality* (see Fig. 1). The realization of this biological system is achieved through the application of the Logical Stochastic Resonance (LSR) paradigm [2]. LSR requires controls of external parameters to change the system’s nonlinearity and noise level in order to obtain the desired output signal (according to the logic gate truth tables). Noise is critical for the implementation and operation of the gates.

## II. LSR IN A SINGLE-GENE NETWORK:

### THE MAIN IDEA

#### A. The Single-Gene Network

Several applications have made use of the LSR paradigm [3, 4]. To test the possibility of adapting LSR [2] also to biology, we have chosen a synthetic gene network derived from the bacteriophage  $\lambda$ .

To implement LSR, the biological system needs to be bistable: a logic gate (AND or OR in our case) has two expected output values, “1” or “0”. Moreover, this bistability can be represented by a double-well potential, where the left well is set as “0”, and the right well is set as “1”. LSR uses the external inputs to overbalance the potential to the left or to the right, so that the well representing the desired output (as defined by the truth tables [2]) becomes deeper (see Fig. 2).

The bacteriophage  $\lambda$ , after infecting the bacteria *E. coli*, can follow two pathways: *lytic* (the bacteriophage  $\lambda$  replicates its DNA autonomously, releasing hundreds phages) and *lysogenic* (the bacteriophage  $\lambda$  genome is incorporated into the bacteria genome). The decision between these two pathways depends on a positive feedback loop that regulates the repressor protein  $\lambda$  [5]. We have considered a DNA plasmid made of a promoter region  $P_{RM}$  that regulates the *cI* gene.  $P_{RM}$  is characterized by three tandem operator sites:  $O_{R1}$ ,  $O_{R2}$  (activate transcription), and  $O_{R3}$  (represses transcription). The system’s bistable configuration is reached only when the protein production is comparable to the protein degradation.

## B. The Deterministic Model

For a complete comprehension of the genetic network, we have considered the slow and fast biochemical reactions that control the system (we are not reporting them in the present manuscript, see [6]). According to these reactions and considering high copy number plasmids, the  $\lambda$  repressor concentration dynamics can be described using the Langevin equations. The following equations represent the monomer and dimer concentration forms:

$$\begin{aligned} \dot{x} &= -2k_1x^2 + 2k_{-1}x_2 + nk_1p_0(d_1 + \beta d_2) - k_x x + \epsilon d_0 \\ \dot{x}_2 &= k_1x^2 - k_{-1}x_2 - k_y x_2, \end{aligned} \quad (1)$$

where we assume that the RNA polymerase concentration  $p_0$  is constant, while  $\epsilon$  is the basal expression rate. In particular,  $x$ ,  $x_2$ ,  $d_0$ ,  $d_1$ , and  $d_2$  are the concentrations of the repressor protein in the monomer and dimer forms [6], the DNA promoter region with no bound protein, one or two bound proteins, respectively. Moreover, the slow and fast reactions are characterized by reaction rates:  $k_i$  and  $k_{-i}$  are rate constants for the dimerization reaction,  $k_t$  is the transcription rate while one dimer is bound to the  $O_{R1}$  site,  $\beta k_t$  is the enhanced rate by a factor  $\beta$  when a second dimer binds the  $O_{R2}$  site,  $k_x$  is the degradation rate and  $k_y$  is the dilution rate.

We can now assume that the dimerization reactions are faster than the other reactions. This consideration allows us to reduce the system in (1) to a single equation. After considerable calculations [6] and without loss of generality, we can define the dimensionless variables  $\tilde{x} = x\sqrt{K_1K_2}$ ,  $\tilde{t} = t r K_2 / 4$ ,  $r = \epsilon d_0$ . The constant  $d_0$  is the total concentration of the DNA promoter sites. We finally obtain (overbars on  $x$  and  $t$  are omitted for simplicity):

$$\dot{x} = \frac{(\alpha - 1)x^2 + \sigma_1(\alpha\beta - 1)x^4 - \sigma_1\sigma_2x^6}{(\tau + x)(1 + x^2 + \sigma_1x^4 + \sigma_1\sigma_2x^6)} + \frac{1 - \gamma x - \gamma_y x^2}{\tau + x}, \quad (2)$$

where we have introduced the dimensionless parameters  $\alpha = nk_1p_0d_0/r$ ,  $\gamma = k_x/(\sqrt{K_1K_2}r)$ ,  $\gamma_y = 2k_y/(rK_2)$ , and  $\tau = \sqrt{K_1K_2}/4K_1$ . TABLE I shows the definitions and values of the remaining parameters.

Before implementing the LSR paradigm on the synthetic genetic network, we need to consider the general functioning of a logic gate. Its logic inputs can either be “0” or “1”. We then have the following input sets: (0,0), (0,1), (1,0), and (1,1). Because of a lack of externally tunable parameters in our biological system, we have proposed that the two inputs of each set enter in (2) as the sum. In other words, the input set reduces to three combinations: (0,0), (0,1)/(1,0), and (1,1), where the sum of (0,1) and (1,0) gives the same value. Following this simplification, we will need only one biological parameter to encode the logic gate data inputs (that enters the systems through  $\alpha$ ) and another parameter to switch from the AND to the OR gate (that enters the system through  $\gamma$ ). In particular,  $\alpha$  is connected to the repressor protein basal rate of production, and  $\gamma$  is related to the degradation rate. Finally, we set the output signal as “0” if

the system is in the left well (or low protein concentration), and “1” if the system is in the right well (high protein concentration).

TABLE I. PARAMETERS RELEVANT TO THIS SINGLE-GENE NETWORK

Parameter value	Definition
$\beta=11$	Degree of transcriptional activation
$K_1=k_1/k_{-1}=5.0 \times 10^7 M^{-1}$	Equilibrium constant dimerization
$K_2=k_2/k_{-2}=5.0 \times 10^7 M^{-1}$	Equilibrium constant dimer- $O_R$
$\sigma_1=2$	Binding affinity for the dimer to $O_{R2}$ relative to $O_{R1}$
$\sigma_2=0.08$	Binding affinity for the dimer to $O_{R3}$ relative to $O_{R1}$

For a deeper comprehension of the LSR implementation, we have analytically calculated the biological system potential function,  $U(x)$ , by integrating the right-hand side of (2). It is important to notice that the  $\alpha$  and  $\gamma$  parameters directly influence  $U(x)$ : modifying the shape of each well. We used simulations to exhaustively search for the best  $\alpha$  and  $\gamma$  values, limited to the interval of biological accessible parameters. We have then noticed that the bistable area in the accessible range of parameter values is narrow, and this influences the possibility to experimentally realize the LSR. To solve this problem, we have followed the theoretical suggestion from [7]. The idea is to choose the external parameter values so that the unwanted well almost disappears: it means that the bistable potential function in Fig. 2 overbalances on the left (configuration on the left of Fig. 2) or on the right (configuration on the right of Fig. 2), lowering as much as possible the barrier between the wells. In addition, LSR exploits Stochastic Resonance [8] for such cases where the potential function shows bistability. For example, it usually happens when the input sets are (0,1) or (1,0). In other terms, it can happen that the (0,0) or (1,1) cases can be realized when the potential function is almost monostable.

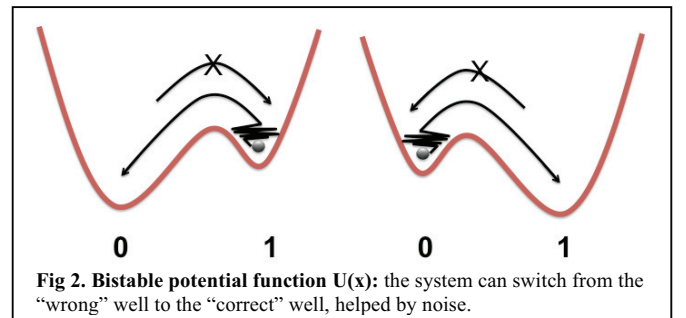
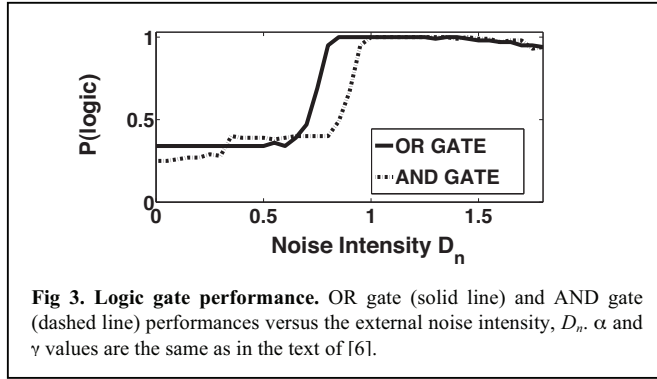


Fig 2. Bistable potential function  $U(x)$ : the system can switch from the “wrong” well to the “correct” well, helped by noise.

Finally, we have obtained  $\alpha=6.3$ , 9.8, and 13.3 to respectively encode (0,0), (0,1)/(1,0), and (1,1), while  $\gamma=50$  gives the AND gate functionality and  $\gamma=36$  gives OR gate functionality. In almost all potential configurations obtained from this chosen values,  $U(x)$  presents two wells separated by a barrier. Moreover, if we consider the system dynamics without noise, and our initial condition is in the “wrong” well, the system will be trapped in this well giving undesired

signal output. As a consequence, we decided to exploit the presence of natural fluctuations affecting the GRN.



### C. In presence of Noise

At the gene level, noise is generated from many sources. Scientists have classified it into two main classes [9]: *extrinsic* (inherent in other cellular components) and *intrinsic* (inherent in biochemical processes of gene expression) noise. To simply test our idea, here we focus on the (additive) *extrinsic* noise (the whole noise characterization is reported in [6]). Considering  $x$  in (2) as the repressor concentration, noise is the fluctuation of the “background” repressor production that influences the basal production rate,  $r$ . We can now write the stochastic model, under the assumption that fluctuations are small, by adding the random perturbation  $D_n \xi(t)$  term on the right hand side of (2). In particular,  $\xi(t)$  is a zero-mean Gaussian noise with  $\langle \xi(t) \xi(t') \rangle = \delta(t-t')$  (we take into account that random fluctuations happen faster than other reaction time scale in the system) and with  $D_n$  the noise intensity.

To analyze our results, we need to define a quantity that measures the system robustness in presence of noise and the possibility to realize the logic gate. In our case, this quantity is the logic gate performance,  $P(\text{logic})$ . It is defined as the ratio of successes in obtaining the desired output signal over the total number of trials (see Fig. 3). In particular, for each noise value, we have a success when the simulated outputs for all the three data input sets ((0,0), (0,1)/(1,0), and (1,1)) are equal to the respective truth table values. When this doesn’t happen, we consider it a failure. We have repeated this procedure 500 times. Moreover, we have solved it through simulating the stochastic model via the Euler-Maruyama method, on the dimensionless time interval [0, 7000]. In simulations, we noticed that the time interval is longer than the escape time the system takes to reach the “correct” well.

Fig. 3 is the main result of this section: it shows  $P(\text{logic})$  over the additive noise intensity  $D_n$ . We can notice that, in absence of a noise floor or when  $D_n$  is smaller than the barrier value between the two wells, the logic gate does not work properly:  $P(\text{logic})$  is lower than 0.5. Moreover, if the noise intensity is very large, higher than the barrier height, the system loses the possibility to be confined in one of the wells and it quickly changes its state. Also in this case the logic gate outputs don’t match the truth table values. Between these

two configurations, there is a range of noise values where the logical output occurs consistently and robustly. As Stochastic Resonance predicts [8], we have demonstrated that there is an *optimal* noise range where the biological logic gate works interchangeably as an AND gate or OR gate, in presence of noise, and  $P(\text{logic}) \sim 1$ .

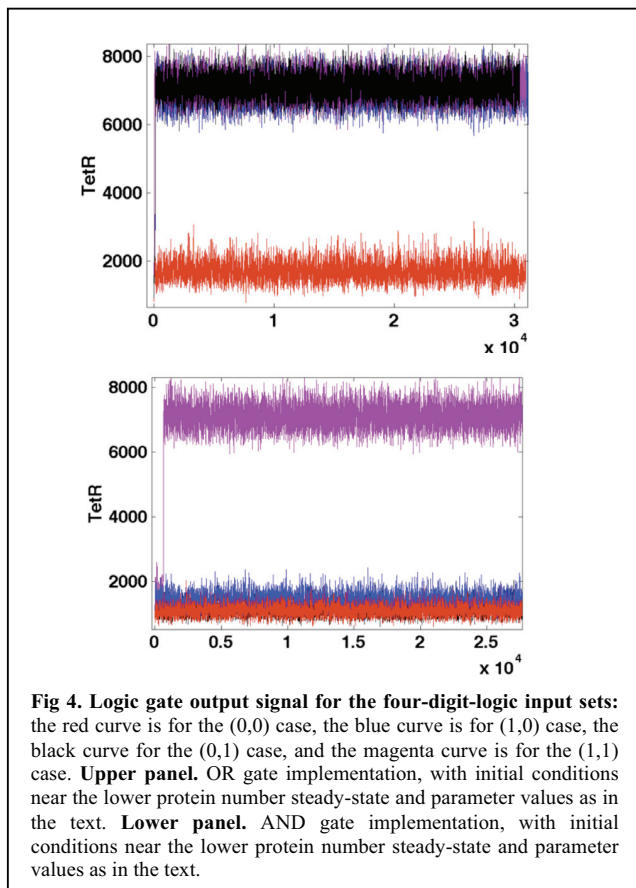
### III. LSR IN A TWO-GENE NETWORK

We have shown that LSR can function in a single-gene network. However, it is oversimplified. To better guide future experimental verifications, we have developed a more detailed model. In this Section, we describe the possibility to apply LSR in a more robust genetic regulatory network.

We have considered the mutual repressor motif of the genetic switch from the yeast *S. cerevisiae* [10]: the system consists of two repressor genes, LacI and TetR. This GRN is suitable for applying our idea: when LacI is in the higher state, TetR is in the lower one. This happens because of the mutual repression. Exploiting this effect, there is no need to measure the signal of both genes, but simply focus on one of them. Moreover, in Sec. II, we have pointed out that the single-gene network lacks accessible parameters necessary for the LSR implementation. This restriction forced us to enter each logic input set as the sum of the two data inputs. Instead, this new GRN allows the external control of all three parameters: two for encoding the logic gate data inputs and one for letting the system to switch from the AND gate to the OR gate. From [10], we know that it is possible to externally control the system dynamics by increasing or decreasing the inducer values (i.e., IPTG and ATc inactivate respectively the LacI and TetR repressor concentrations). Consequently, IPTG and ATc are the two data inputs of our biological model. Furthermore, to choose what logic gate functionality the system performs, we can tune the LacI degradation rate. Previous work has shown that it is possible to control the protein degradation rate through degradation machinery [11]. In our work, we simplified and adapted the proposed system in [11], by using copper as an external control parameter. In other words, more copper is present in the cell, below a toxic level, the higher is the degradation rate of the protein.

In our study, we have firstly computed several bifurcation diagrams for different parameter values. In order to implement LSR, we have also explored the possibility to obtain a bistable region over a larger range of external parameter values. Our final goal is not only to build the LSR on a GRN, but also to look for the range of parameter values where LSR is optimally performed. With this in mind, we have selected the best bifurcation diagrams (one for the AND gate and the other for the OR gate) among all the possible cases.

As in the single-gene network, the new system’s bistable region is very small. Consequently, we recurred to the same solution explained in Sec. II. We have then selected the suitable parameters and studied the system dynamics in presence of internal noise. The data input encoded by IPTG has values 0 mM (for the logic value “0”) and 10 mM (for the logic value “1”); instead, the encoded data input through ATc has values 8.5 ng/ml (for the logic value “0”) and 7.5 ng/ml



(for the logic value “1”). Finally the control parameter is copper that is approximately 0  $\mu\text{M}$  (for the AND gate) and 40  $\mu\text{M}$  (for the OR gate). In this case, we have used the Gillespie algorithm [12] to reproduce step by step the biochemical reactions that happen in our GRN. This method is more realistic and able to take into account only internal fluctuations.

We have run simulations fixing IPTG, ATc, and copper, for two different initial conditions: one near low protein number steady-state and the other near the high protein number steady-state. In Fig. 4, we have reported the output signal for the four input sets: (0,0), (0,1), (1,0) and (1,1). In this work, we only plot our simulations starting near the small protein number state. We show that also in presence of internal random fluctuations the system, after a short amount of time, reaches the expected well and remains there. For example, we can notice the magenta curve in the lower panel of Fig. 4: initially the system is in the “wrong” state and after a short amount of time it switches and remains in the expected configuration of high protein numbers.

In Fig. 4, the upper panel represents the OR gate where only the (0,0) case has output in the lower state (that we set at “0”), while the lower panel represents the AND gate where only the (1,1) case has output in the higher state (that we set

at “1”). In particular, the red curve is for the (0,0) case, the blue curve is for (1,0) case, the black curve for the (0,1) case, and the magenta curve is for the (1,1) case.

In summary, we have demonstrated that it is possible to apply the LSR paradigm to a biological system. We have firstly used a single-gene network derived from a bacteriophage  $\lambda$ . Through mathematical modeling and simulations, we have searched for the best  $\alpha$  and  $\gamma$  parameters, and we have consequently obtained the logic gate performance,  $P(\text{logic})$ . Results show that in an *optimal band of noise*,  $P(\text{logic}) \sim 1$  both for the AND gate and the OR gate: this demonstrates that our system can operate as two logic gates.

Moreover, our single-gene network seems oversimplified for future experimental applications. This brought us to choose a two-gene network from the yeast *S. cerevisiae*. In this case, we have introduced a degradation machinery to make the *biological logic gate* reconfigurable. We have shown also for this system that there exists parameter values where the GRN, exploiting internal noise, can still perform as an AND or OR gate (according to the truth tables), and be reconfigured.

LSR on a GRN offers the possibilities to utilize the computational power and functionality of engineered biological systems.

## REFERENCES

- [1] J. C. Anderson, E. J. Clarke, A. P. Arkin, and C. A. Voigt, “Environmentally controlled invasion of cancer cells by engineered bacteria,” *J. Mol. Biol.*, vol. 355, p. 619, 2006.
- [2] K. Murali, S. Sinha, W. L. Ditto, and A. R. Bulsara, “Reliable logic circuit elements that exploit nonlinearity in the presence of a noise floor,” *Phys. Rev. Lett.*, vol. 102, p. 104101, 2009.
- [3] D. N. Guerra, A. R. Bulsara, W. L. Ditto, S. Sinha, K. Murali, and P. Mohanty, “A noise-assisted reprogrammable nanomechanical logic gate,” *Nano Lett.*, vol. 10, p. 1168, 2010.
- [4] L. Worshech et al., “Universal and reconfigurable logic gates in a compact three-terminal resonant tunneling diode,” *Appl. Phys. Lett.*, vol. 96, p. 042112, 2010.
- [5] J. Hasty, J. Pradines, M. Dolnik, and J.J. Collins, “Noise-based switches and amplifiers for gene expression,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 97, p. 2075, 2000.
- [6] A. Dari, B. Kia, X. Wang, A. R. Bulsara, and W. L. Ditto, “Noise-aided computation within a synthetic gene network through morphable and robust logic gates,” *Phys. Rev. E*, vol. 83, p. 041909, 2011.
- [7] A. Dari, B. Kia, A. R. Bulsara and W. L. Ditto, “Creating morphable logic gates using logical stochastic resonance in an engineered gene network,” *EPL*, vol. 93, p.18001, 2011.
- [8] L. Gammaitoni, P. Hanggi, P. Jung, and F. Marchesoni, “Stochastic Resonance,” *Rev. Mod. Phys.*, vol. 70, p. 223, 1998.
- [9] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, “Stochastic gene expression in a single cell,” *Science*, vol. 297, p. 1183, 2002.
- [10] T. Ellis, X. Wang, and J. J. Collins, “Diversity-based, model-guided construction of synthetic gene networks with predicted functions,” *Nat. Biotechnol.*, vol. 27, p. 465, 2009.
- [11] C. Grilly, J. Stricker, W. L. Pang, M. R. Bennett, and J. Hasty, “A synthetic gene network for tuning protein degradation in *Saccharomyces cerevisiae*,” *Mol. Syst. Biol.*, vol. 3, p. 127, 2007.
- [12] D. T. Gillespie, “Exact Stochastic Simulation of Coupled Chemical Reactions,” *J. Phys. Chem.*, vol. 81, p. 2340, 1977.