Synthetic gene networks as potential flexible parallel logic gates

2011 EPL 93 50001

(http://iopscience.iop.org/0295-5075/93/5/50001)

A Letters Journal Exploring the Frontiers of Physics

March 2011

EPL, **93** (2011) 50001 www.epljournal.org
doi: 10.1209/0295-5075/93/50001

Synthetic gene networks as potential flexible parallel logic gates

Hiroyasu Ando[1(a)], Sudeshna Sinha[2], Remo Storni[3] and Kazuyuki Aihara[3,4]

[1] *RIKEN Brain Science Institute - 2-1 Hirosawa, Wakoshi, Saitama 351-0198, Japan*
[2] *Indian Institute of Science Education and Research (IISER) Mohali, Transit Campus: MGSIPAP Complex, Sector 26 - Chandigarh 160 019, India*
[3] *Institute of Industrial Science, University of Tokyo - 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan*
[4] *FIRST, Aihara Innovative Mathematical Modelling Project - Japan*

received 16 September 2010; accepted in final form 4 February 2011
published online 4 March 2011

PACS 05.45.-a – Nonlinear dynamics and chaos

Abstract – We show how a synthetic gene network can function, in an optimal window of noise, as a robust logic gate. Interestingly, noise enhances the reliability of the logic operation. Further, the noise level can also be used to switch logic functionality, for instance toggle between AND, OR and XOR gates. We also consider a two-dimensional model of a gene network, where we show how two complementary gate operations can be achieved simultaneously. This indicates the flexible parallel processing potential of this biological system.

Logic gates, such as AND, OR, XOR, NOR and NAND, form the basis of universal general-purpose computation [1]. So different physical principles that can yield logic outputs, consistent with the truth tables of different logic functions (see table 1), are of paramount interest. A new direction in this endeavour uses the interplay between noise and nonlinearity to enhance the reliability of logic gates, a phenomenon named *logical stochastic resonance* [2].

In this work we investigate the possibility of obtaining robust logic outputs from a noisy biological system of considerable interest, namely a synthetic single gene network. We demonstrate the crucial role of noise in the enhancement of the logic performance in this system. We also show that the power of varying noise levels can flexibly yield different types of logic: namely the genetic network can effectively behave as a *reconfigurable biological logic gate* with noise acting as a logic pattern selector. Further, we show the capacity of a higher-dimensional system for *parallel processing logic functions*. The latter observation indicates that more complex systems may have inherently greater computational capability arising from greater parallel processing capacity.

First we describe the synthetic single gene network model below, and then go on to define what constitutes inputs and outputs in this system.

Table 1: Relationship between the two inputs and the output of the fundamental OR, AND, NOR, NAND and XOR logic operations. Note that the four distinct possible input sets (0, 0), (0, 1), (1, 0) and (1, 1) reduce to three conditions as (0, 1) and (1, 0) are symmetric. Note that *any* logical circuit can be constructed by combining the NOR (or the NAND) gates [1].

Input set (I_1, I_2)	OR	AND	NOR	NAND	XOR
(0, 0)	0	0	1	1	0
(0, 1)/(1, 0)	1	0	0	1	1
(1, 1)	1	1	0	0	0

Single-gene network model. – We use the quantitative model, developed in [3], describing the regulation of the operator region of the λ phage, whose promoter region consists of three operator sites. The chemical reactions describing this network, is given by suitable rescaling (and considering the total concentration of DNA promoter sites to be a constant) as [3]:

$$\dot{x} = \frac{m(1 + x^2 + \alpha\sigma_1 x^4)}{1 + x^2 + \sigma_1 x^4 + \sigma_1\sigma_2 x^6} - \gamma x = F(x), \qquad (1)$$

where x is the concentration of the repressor.

Such equations often arise in modelling genetic circuits. In its functional form, the right-hand side represents production of repressor due to transcription. The even polynomials in x arise due to dimerization and subsequent binding to the promoter region.

(a) E-mail: hiroyasu_ando@brain.riken.jp

For the operator region of the λ phage, $\sigma_1 \sim 2$, $\sigma_2 \sim 0.08$ and $\alpha \sim 11$. The integer m represents the number of plasmids per cell. It is possible to design a plasmid with a given copy number. We take it to be 1 in the numerics here.

The parameter γ is directly proportional to the protein degradation rate, and in the construction of artificial networks it can be considered a *tunable parameter* [3]. The nonlinearity of eq. (1) leads to a double well potential, and different γ introduces varying degrees of asymmetry in the potential.

Generally speaking, cells are intrinsically noisy biochemical reactors, and low reactant numbers can lead to significant statistical fluctuations in molecule numbers and reaction rates [4]. So it is of considerable interest to study properties that may emerge from noisy genetic systems. With this in mind, consider the above nonlinear system given by eq. (1), driven by a low-amplitude input signal I, under noise as follows:

$$\dot{x} = F(x) + I + D\eta(t), \qquad (2)$$

where $\eta(t)$ is an additive zero-mean Gaussian noise with unit variance and intensity D; the noise is taken to have correlation time smaller than any other time scale in the system, so that it may be represented, theoretically, as delta correlated. Broadly speaking, the inherent stochasticity of biochemical processes such as transcription and translation, as well as fluctuations in the amounts or states of other cellular components, leads to variation in the expression of a particular gene. Specifically, for a single gene it was found that noise is essentially determined at the translational level, and that the fluctuations in the concentrations of a regulatory protein can propagate through a genetic cascade [4].

Now for a two-input logic gate, the logic inputs can be either 0 or 1, giving rise to 4 logic input sets (I_1, I_2): (0, 0), (0, 1), (1, 0), (1, 1), with (0, 1) and (1, 0) being symmetric for basic logic functions (see table 1). In our model, when the inputs are I_1, I_2, we drive the system with the signal $I = I_1 + I_2$. That is, the system evolves as $\dot{x} = F(x) + D\eta(t) + I_1 + I_2$.

With no loss of generality, consider two inputs $I_{1,2}$ to take the value 0 when the logic input is 0, and the value 0.75 when the logic input is 1. That is, the input signal I can have 3 distinct levels, corresponding to (0, 0), (0, 1)/(1, 0) and (1, 1). Also, since the inputs I_1, I_2 can stream in any random order, the external driving I is, in general, completely *aperiodic*.

The output of the system is determined by its state, *e.g.*, the output can be considered a logical 1 if it is in one well, and logical 0 in the other well. Specifically, the output corresponding to this 2-input set, for a system with potential wells at x_u and x_l, is taken to be 1 (or 0) when the system is in the well at x_u, and 0 (or 1) when the system is in the other well. Hence, when the system switches wells, the output is "flipped".

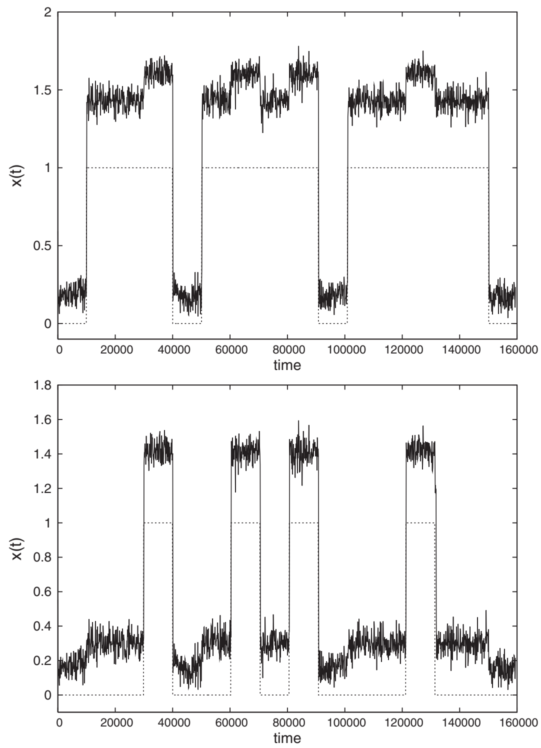

Fig. 1: Time series of $x(t)$, with γ equal to (upper panel) 5.7 and (lower panel) 6.3, and noise intensity $D = 0.15$. The dashed line shows the desired logic response (OR for the upper panel and AND for the lower panel). Evidently, the upper panel shows consistent OR and the lower panel consistent AND, for instance by taking $x > 0.75$ as logic output 1 and $x < 0.75$ as logic output 0 (or alternately, NOR and NAND, by taking $x > 0.75$ as logic output 0 and $x < 0.75$ as 1).

In fig. 1 (upper panel) we show the response of the system for $\gamma = 5.7$ (see footnote [1]). We observe that, under optimal noise, interpreting $x(t) < x^*$ as logic output 0 and the voltage $x(t) > x^*$ as logic output 1 yields a clearly defined robust logical OR whereas interpreting $x(t) > x^*$ as logic output 0 and $x(t) < x^*$ as logic output 1 yields a clean logical NOR. In a completely analogous way, when $\gamma = 6.3$ in fig. 1 (lower panel), we can realize clean AND and NAND gates. The output is determined by taking x^* to be a point midway between the wells. Specifically $x^* = 0.75$ in the numerics presented here. However, similar results are robustly obtained with $0.5 < x^* < 1$.

In order to rigorously quantify the performance of the system as a logic gate we calculate the probability of the system yielding the right gate output, by scanning a range of input sequences and initial states. Namely, the consistency of obtaining a given logic output is given by the probability, P(logic), defined as the ratio of the number of correct logic outputs to the total number of runs, where each run samples over the four input sets (0, 0), (0, 1), (1, 0), (1, 1), in different permutations.

[1]Here we use the Euler-Maruyama scheme for integrating stochastic differential equations.

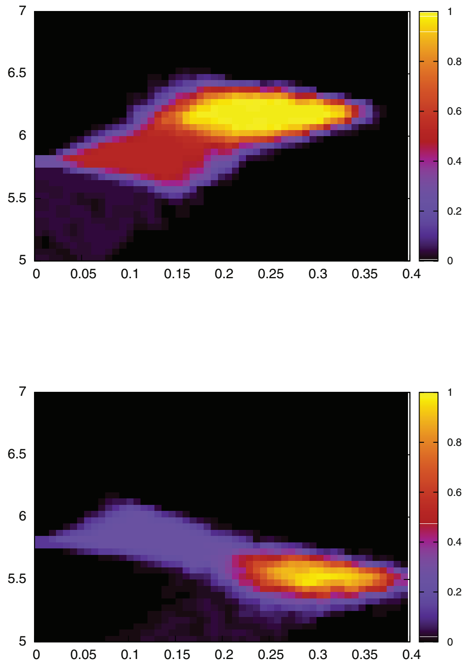

Fig. 2: (Colour on-line) Probability of obtaining logic operations (see text): (top) $P(\text{NAND})/P(\text{AND})$; (bottom) $P(\text{NOR})/P(\text{OR})$. The x-axis displays noise intensity D and the y-axis displays γ.

A run is considered a success if the output is correct *over the entire period of the given input signal* (allowing for small latency after input switching), for *all* four input sets in the run.

Figure 2 shows this stringent reliability criterion obtained from numerical simulations over 100 different runs, with the input signals being held over $t \sim 10^5$ before switching. We can thus ascertain, both the steadiness of the logic output over long times, and its robustness with respect to changing input streams. It is evident that the fundamental logic operation NAND (and, analogously, NOR) is realized, consistently, in an optimal band of moderate noise. Namely, $P(\text{logic}) \sim 1$ in relatively wide windows of moderate noise. The parameter γ determines the asymmetry of the wells, and so varying γ results in the system being driven to different wells. For smaller γ one obtains the NOR logic operation, and for larger γ one obtains the NAND logic operation[2].

One can also analyse the probability $P(x)$ of obtaining the system in state x by solving for the steady-state distribution arising from the relevant Fokker-Planck equation, namely $P(x) = A\exp(-2\phi(x)/D)$, where A is a normalization constant, D is the noise intensity and $-\partial\phi(x)/\partial x = F(x)$ [4,5]. This analysis yields results completely consistent with the observations (for example see fig. 3).

Alternate input-output association. – Consider again the above genetic system under the influence of

[2]Note that the above results are robust with respect to different types of noise, for instance $1/f$ noise.

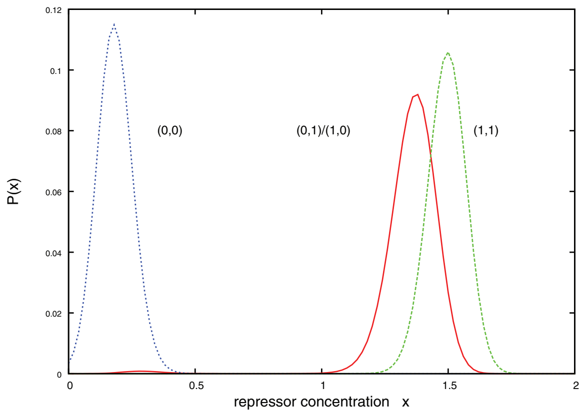

Fig. 3: (Colour on-line) $P(x)$ (see text) *vs.* x, for different input sets, with $\gamma = 5.7$ and $D = 0.1$, reflecting a clean OR/NOR logic association.

noise: $\dot{x} = F(x) + D\eta(t)$, where $F(x)$ is given by eq. (1) and the noise intensity

$$D = D_{gate} \times I,$$

where $I = I_1 + I_2$ is a low-amplitude external input signal encoding the two logic inputs. So the inputs here determine the noise intensity, or alternately one can view the *input signal as being modulated by noise.*

Specifically, I_1 and I_2 can have value 0 when logic input is 0 and value 0.5 when logic input is 1. So the effective noise amplitude D varies with the inputs, being 0 for input set $(0, 0)$, $\frac{1}{2}D_{gate}$ for input sets $(0,1)/(1,0)$, and D_{gate} for input set $(1, 1)$.

In order to obtain a logic operation one needs a clean one-to-one relation between some physical quantity and the output. Here we associate the *average* value of the repressor concentration x, $\langle x \rangle$, with different outputs. Note that this average can be over very short times for AND/NAND and OR/NOR determination.

For instance, with no loss of generality, the logic output can be determined as follows: i) if the average x, $\langle x \rangle < x^*$, then the logic output is 0; ii) if the average x, $\langle x \rangle > x^*$, then the logic output is 1, where x^* is the critical output determination level. Specifically, the threshold level x^* used to determine the 0/1 output can be chosen appropriately depending on the well positions, well depths, well symmetry, etc.

Now, varying the level D_{gate} yields different kinds of logic behavior. Representative values of D_{gate} are $D_{AND} = 0.5$ for AND gate and $D_{OR} = 1.0$ for OR gate, using $x^* \sim 0.75$.

It is also possible, using longer time averages, to adopt more sensitive cut-offs to obtain a larger variety of gates. For instance, using $x^* = 1.35$, one can obtain the XOR gate with $D_{XOR} = 1$, the OR gate with $D_{OR} = 0.6$ and the AND gate with $D_{AND} = 0.3$ (see fig. 4).

So the noise level has a one-to-one correspondence with the distinct logic input sets that can occur, as well as the type of logic function. Or another way of viewing this

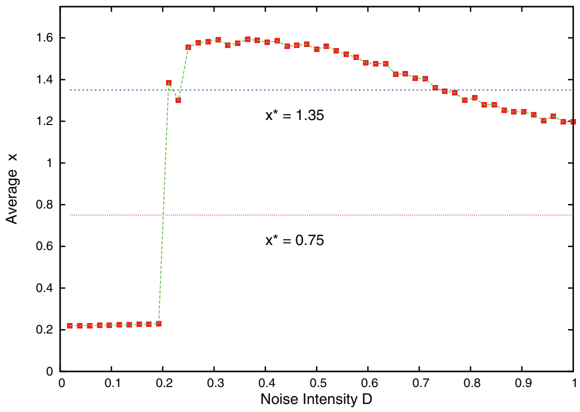

Fig. 4: (Colour on-line) The average value of repressor concentration $\langle x \rangle$, averaged over $t = 50000$, *vs.* noise intensity D. Here $m = 1$, $\gamma = 4.75$, and the initial state of the system is $x = 0$ (*i.e.* the system is in the lower well). The cut-offs $x^* = 0.75$ and $x^* = 1.35$ are also displayed for reference.

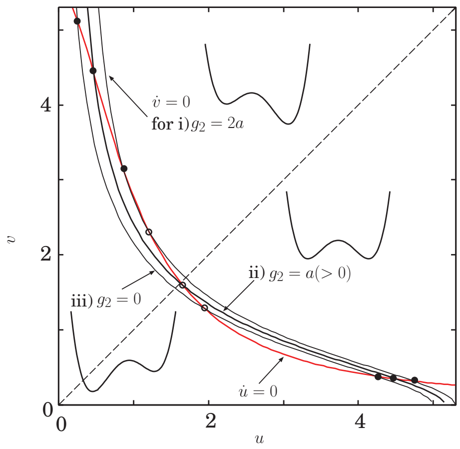

Fig. 5: (Colour on-line) Null-clines for the system (3) and (4) in the phase space. We consider three cases for the null-cline of (4). The open circles are the unstable fixed point and the filled circles are stable ones. The schematic 1D double-well potentials are shown corresponding to the condition of each null-cline.

is that the input signal is modulated with superimposed noise, whose level determines the nature of logic response obtained. The distinct feature here is the *multiplicative* nature of the noise and signal, *vis-à-vis* the earlier additive noise [2].

In particular, we have now explicitly shown the example of parallel AND and XOR on a set of inputs —which constitutes a half-adder (this is a set of operations that typically need to be done an enormous number of times in arithmetic computation). We show how this example of combinational logic, involving two logic functions, can be done by a single system here.

Now, in order to gauge the generality of these results under different types of noise, and the effects of increasing dimensionality, we will consider a two-dimensional model genetic system below.

Two-dimensional model of a toggle switch as a logic gate. – Here we use the model of toggle switch composed of two repressors and two constructive promoters [6]. Each promoter is inhibited by the repressor that is transcribed by the opposing promoter. This model can be described by the following two-dimensional system, which is slightly modified from original one [7]:

$$\dot{u} = \frac{\alpha_1}{1 + v^{n_1}} - d_1 u + g_1, \tag{3}$$

$$\dot{v} = \frac{\alpha_2}{1 + u^{n_2}} - d_2 v + g_2, \tag{4}$$

where u is the concentration of one repressor (R1), v is that of the other (R2). α_1 and α_2 are the effective rates of synthesis of R1 and R2, $n_1(n_2)$ is the cooperativity of repression of promoter 2 (promoter 1). d_i and g_i $(i = 1, 2)$ are the degradation rates and the basal synthesis rates, respectively. We fix the parameters as $\alpha_1 = \alpha_2 = 5$, $n_1 = n_2 = 1.6$, $d_2 = 1$, $g_1 = 0$, and we vary the value of d_1 and g_2 in the following simulations.

In fig. 5, we show the crossing null-clines in the phase space of the system (3) and (4), where $d_1 = 1$ and g_2 (> 0) takes three different values ($n_1 = n_2 = 1.8$ only for the figure). There are two stable fixed points and one unstable fixed point in each case. We must have a two-dimensional potential with double wells for the system. In the figure, we also show the corresponding double-well potential in the one-dimensional case. It is conceivable that one can tune the shape of the wells in the two-dimensional potential by varying the parameters of the system. Therefore, logical stochastic resonance can be observed in the two-dimensional model too, as we show below.

Here, let us consider the noise is added to the degradation rates, which means the noise is multiplicative to the system, *i.e.* $d_1 u \to (d_1 + D\eta_1(t))u$, $d_2 v \to (d_2 + D\eta_2(t))v$. As in the 1D case, $\eta_{1,2}$ are zero-mean white Gaussian noise with the amplitude D. Further, we consider an input signal $I = I_1 + I_2$ driving eq. (4). With no loss of generality, for logic input 0, the $I_{1,2}$ take value 0 here, and for logic input 1, they take value 0.1.

In fig. 6 we show the response of the system with respect to one of two variables, *i.e.* $v(t)$ for $d_1 = 0.9$ and different g_2. Interpreting $v(t) < v^*$ as logic output 0, and the value of $v(t) > v^*$ as logic output 1, gives correct OR (upper panel) and AND (lower panel) logical operations under optimal noise. We take $v^* \sim 1.5$ here, but note that the results are quite robust with respect to small variations of this level. We take $g_2 = 0.1$ for the OR response and $g_2 = 0.0$ for the AND. Here the different gates are obtained by changing the well shape determined by varying g_2.

Now, most interestingly, the dynamics of the other variable $u(t)$ (which evolves simultaneously of course) yields the *complementary* logical output. That is, $u(t)$ operates as NAND or NOR, while $v(t)$ yields AND or OR response. This is significant, as it allows the system to yield

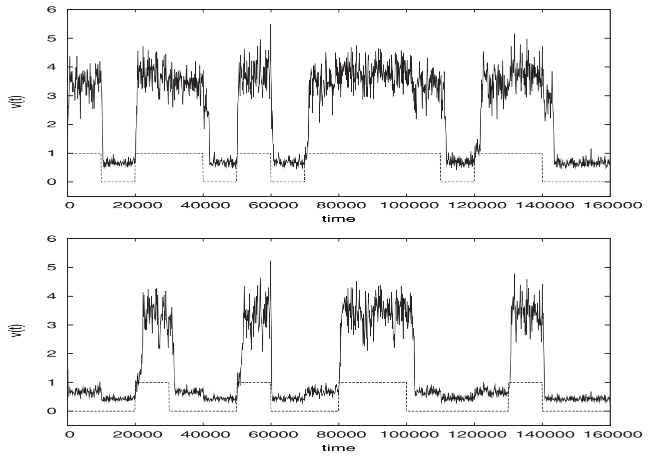

Fig. 6: Time series of $v(t)$, with $d_1 = 0.9$ and g_2 equal to (upper panel) 0.1 and (lower panel) 0.0, and the noise intensities are both $D = 0.1$. The dashed line shows the desired logic response (OR for the upper panel and AND for the lower panel). Clearly, the upper panel shows consistent OR and the lower panel consistent AND, for instance by taking $v > 1.5$ as logic output 1 and $v < 1.5$ as logic output 0. Moreover, NAND (top) and NOR (bottom) responses are obtained by $u(t)$ simultaneously (not shown).

a logic operation and its complement in parallel, without necessitating the additional NOT operation.

Generalized parallel logic. – Above, we were successful in obtaining complementary logical operations simultaneously, say OR/AND for $v(t)$ and NOR/NAND for $u(t)$ in fig. 6. Now, we extend the combination of parallel logic operations, to realize OR-XOR, AND-OR and AND-XOR simultaneously for a shared input set. The AND-XOR combination for a common set of inputs is particularly important, as it forms the basis of bit-by-bit addition, which is the most fundamental arithmetic operation. Further, we will then go on to show parallel logic operations on two *independent* input sets. The key to parallel processing is the higher dimensionality of the system. For instance, in a two-dimensional dynamical system, the two variables can give two different logical outputs in parallel.

The parallel logic output AND-NAND (OR-NOR) above, could be achieved due to the location of the potential wells in 2D phase space. Namely, one of the two potential wells was located at low u and high v, and the other at high u and low v. Clearly, to obtain a variety of combinations of parallel operations (not just complementary logic), we must exploit the location and depth of the wells in two-dimensional phase space.

Specifically, consider four regions of phase space in a 2D system: A) (x_{low}, y_{low}), B) (x_{low}, y_{high}), C) (x_{high}, y_{low}), D) (x_{high}, y_{high}). The "low" and "high" are determined by appropriate thresholds, for instance, with the "low" corresponding to less than 3 and "high" corresponding to greater than 3. So the parallel logic outputs (0, 0) can be associated with region A), outputs (0, 1) with

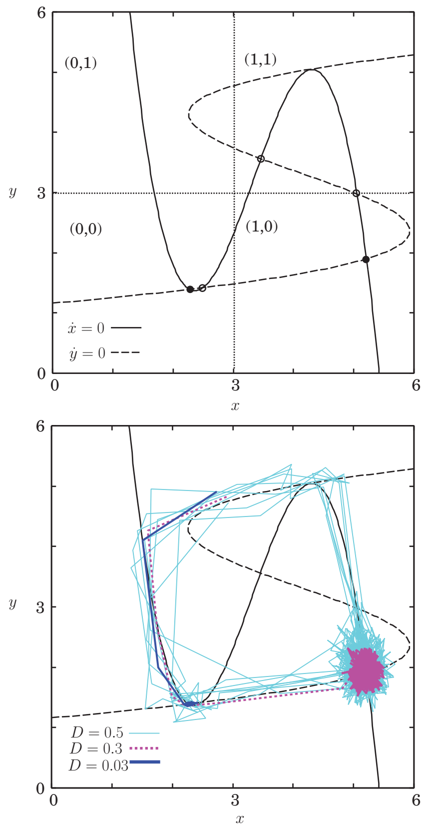

Fig. 7: (Colour on-line) Top: null-clines for the system (5) and (6) in the phase space, for (I_1, I_2) equal to $(1, 1)$. Corresponding pairs of logical outputs are indicated in the region segmented by the dotted lines. The open circles are the unstable fixed point and the filled circles are stable ones. Bottom: three trajectories for $D = 0.03$, $D = 0.3$ and $D = 0.5$ are shown.

region B), outputs (1, 0) with region C) and outputs (1, 1) with region D). Thus, the different potential well locations realize different pairs of logic truth tables in parallel.

For a particular realization of such potential wells, consider the following 2D system with independent sets of inputs $I = I_1 + I_2$ and $I' = I'_1 + I'_2$:

$$\dot{x} = f(x) - y + h_1(I,\ I') + D\eta_1(t), \tag{5}$$

$$\dot{y} = x - g(y) + h_2(I,\ I') + D\eta_2(t), \tag{6}$$

where f, g are cubic functions: $f(x) = a_1x^3 + b_1x^2 + c_1x + d_1$, $g(y) = a_2y^3 + b_2y^2 + c_2y + d_2$. $\eta_{1,2}$ are zero-mean white Gaussian noise and with the amplitude D. h_i is an appropriate function to design the location of the well depending on the inputs. Here, we fix the parameter $a_1 = -a_2 = -1, b_1 = -b_2 = 10, c_1 = -c_2 = -30.5$.

In fig. 7, we show the null-clines of the system (5) and (6) with respect to $(I_1, I_2) = (I'_1, I'_2) = (1, 1)$, and $h_i(x, y) = A_i(x+y)/2 + B_i s((x+y)/2, 0.5)$ with the sigmoidal function $s(x, \epsilon) = 1/(1+\exp(-1000(x-\epsilon)))$. $A_1 = -0.3, B_1 = 1.5, A_2 = 2, B_2 = -3.1$, and $d_1 = 29.9$ and $d_2 = -24.4$. Here we consider the case of shared inputs. In the figure, the deepest potential well corresponding to the bottom-right

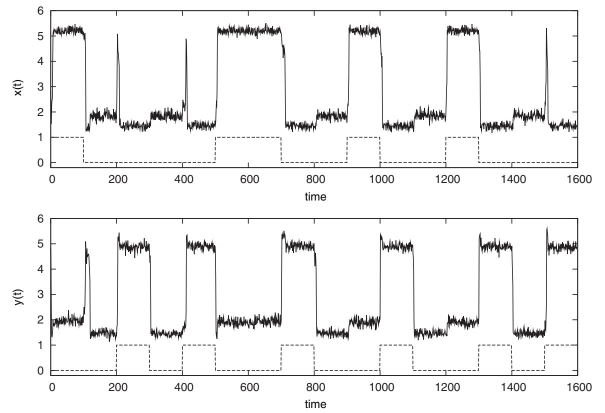

Fig. 8: Time series of $x(t)$ (upper panel) and $y(t)$ (lower panel), with noise intensity D equal to 0.3. The dashed line shows the desired logical outputs (AND for the upper panel and XOR for the lower panel). Almost certainly, the upper panel shows consistent AND (taking $x > 3$ as output 1; $x < 3$ as output 0) response, and the lower shows XOR response in the same threshold.

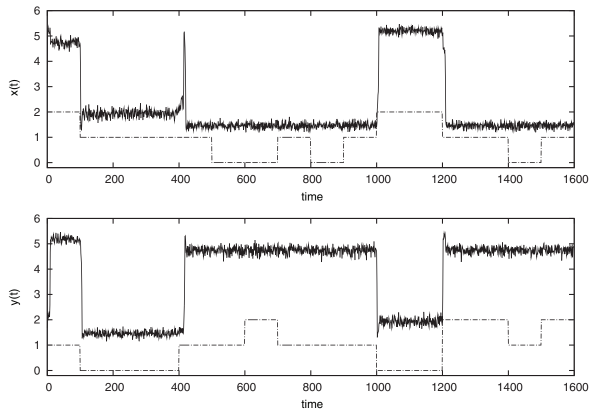

Fig. 9: Time series of $x(t)$ (upper panel) for AND gate and $y(t)$ (lower panel) for OR gate, with $D = 0.3$ in the case of independent input sets. The dash-dotted lines show the inputs $I = I_1 + I_2$ (upper panel) $I' = I'_1 + I'_2$ (lower panel). The upper panel shows consistent AND (taking $x > 3$ as output 1; $x < 3$ as output 0) response most likely, and the lower panel shows OR response in the same threshold.

stable fixed point would be located in the region C) (top panel). Additionally, we show the trajectories with noise intensity $D = 0.03$, $D = 0.3$, and $D = 0.5$ (bottom panel). Apparently, under optimal noise, the system goes to the deepest potential well.

In fig. 8, we show the response of the system for parallelized AND-XOR operations in a window of optimal noise. This parallelization allows bit-by-bit addition, namely the "half-adder", to be implemented in one step.

Now we move on to the case of independent input sets. In principle, as for shared inputs, the idea again is that the location of the deepest well is shifted depending on the inputs, with the only difference being that the well location is changed by function h_i which depends on the combination of the inputs. For example, in order to operate parallelized AND-OR gates with independent inputs, we can choose h_i for (I, I') as follows: $h_1(I, I') = A_1 s(I, 1.5) + B_1 s(I', 0.5)$ and $h_2(I, I') = A_2 s(I, 1.5) + B_2 s(I', 0.5)$. Figure 9 shows the successful response for AND and OR parallel operations in response to a sequence of independent input sets. The input $I(I')$ takes three different levels for (0, 0), (0, 1)/(1, 0), and (1, 1), and the parameter values are $A_1 = 0.67$, $B_1 = 0.83$, $A_2 = 0.83$, $B_2 = -0.67$, and $d_1 = 30.176$, $d_2 = -24.34$. Note that the sigmoidal functions provide the Boolean operations with regard to locating the deepest well. However, the outputs are obtained robustly, not directly from the Boolean operations, but due to the noisy dynamics that prevents the trajectories from being trapped in shallow wells.

The above-mentioned capacity of a complex system to operate in parallel as different combinations of gates indicates exciting possibilities for significantly increasing computational power, not only in synthetic genetic networks [8], but also in many other natural and engineered systems.

In summary, our results here extend the scope, and indicate the generality of the recently observed phenomena of logical stochastic resonance (LSR). Further, these observations may provide an understanding of the information processing capacity of synthetic genetic networks, with noise acting as the logic pattern selector. It also may have potential applications in the design of biological gates with added capacity of reconfigurability of logic operations and parallel processing.

* * *

This work is partly supported by KAKENHI 21800089.

REFERENCES

[1] Mano M. M., *Computer System Architecture* (Prentice Hall) 1993; Bartee T. C., *Computer Architecture and Logic Design* (McGraw-Hill) 1991.

[2] Murali K. *et al.*, *Phys. Rev. Lett.*, **102** (2009) 104101; Murali K. *et al.*, *Appl. Phys. Lett.*, **95** (2009) 194102; Guerra D. N. *et al.*, *Nano Lett.*, **10** (2010) 1168; Sinha S. *et al.*, *EPL*, **86** (2009) 60003; Worschech L. *et al.*, *Appl. Phys. Lett.*, **96** (2010) 042112; Fierens P. I. *et al.*, *Phys. Lett. A*, **374** (2010) 2207; Dari Anna *et al.*, *EPL*, **93** (2011) 18001.

[3] Hasty J. *et al.*, *Chaos*, **11** (2001) 207.

[4] Hasty J., Pradines J., Dolnik M. and Collins J. J., *Proc. Natl. Acad. Sci. U.S.A.*, **97** (2000) 2075.

[5] Risken H., *The Fokker-Planck Equation: Methods of Solutions and Applications* (Springer) 1989.

[6] Gardner T. S. *et al.*, *Nature*, **403** (2000) 339.

[7] Wang J. *et al.*, *BMC Syst. Biol.*, **1** (2007) 50.

[8] Chen L., Wang R., Li C. and Aihara K., *Modeling Biomolecular Networks in Cells: Structures and Dynamics* (Springer) 2010.