# [Re] A Multi-Functional Synthetic Gene Network

Gabriel Baiocchi de Sant'Anna[1, ID] and Mateus Favarin Costa[1]
[1]Federal University of Santa Catarina (UFSC), Florianópolis, Brazil

R E S C I E N C E C

Abstract Studies in the field of synthetic biology are constantly making additions to biological circuit repositories, as well as to the theoretical understanding of their capabilities. The ability to compute with biomolecules has been demonstrated by many synthetic gene networks, but until recently the majority of such models were engineered to implement the functionality of a single circuit part. Purcell et al. have proposed a network capable of multiple functions, switching between three different behaviours in a programmable fashion. This work provides an open-source implementation in which their in silico experiments were replicated.

1 Introduction

The field of biomolecular computing – and DNA-based computing in particular – has advanced remarkably over the last years [1]. There are numerous designs of biological parts which implement the behavior of digital logic gates [2], continuous-time systems [3], oscillators [4], memory components [5], asynchronous circuits [6] and so on. Such recent developments allow one to consider the possibility of exploting biologically derived materials and their aspects of massive parallelism and self-replication to build practical computing systems on biological *substrata* [7].

Amidst forward-engineered biochemical systems, genetic oscillators have been a focus of research [8] as they are required for the correct operation of synchronous sequential circuits and can also provide persistent periodic *stimuli* to other regulatory networks which may rely on them [9]. Genetic switches present another functionality specially useful [9] to digital logic: the ability to toggle between on or off states by either activating or repressing the expression of a certain gene makes them equivalent to a cellular memory unit [7]. One study has shown that combining an oscillator with a toggle switch under certain circumstances will result in the generation of a clock-like near square wave signal [10]. Until recently, though, there was no genetic network capable of exhibiting both of these behaviours and alternating between them.

Purcell et al.[11] presented the *in silico* design of a novel genetic regulatory network which performs frequency division on an oscillating input. During experiments, that model was also discovered able to behave as a self-induced oscillator or toggle switch – thus ressembling the most successful electronic Integrated Circuit (IC) ever developed, the 555 timer IC [12]. We believe such multi-functionality may lead to more programmable and reusable components in biological computing.

In this work, we develop an open-source implementation of their multi-functional synthetic gene network in order to reproduce the original simulations and verify the simplifying assumptions used in the equations which model their genetic circuit.

2 Methods

The multi-functional synthetic gene network and its dynamics are wholly described in the original study. Supplementary material ("File S1", in [11]) contains the complete Ordinary Differential Equation (ODE) system that models the genetic circuit under mass-action kinetics. The Quasi-Steady-State Assumption (QSSA) exploited to derive the reduced model is also provided, together with all reaction parameterization and initial state of each experiment. These factors allowed for an easy replication of the model, even without direct reference to source code or any usage of the proprietary tools (MAPLE and MATLAB) originally employed.

The QSSA equations defining the reduced model are given below, where $[\cdot]$ denotes protein concentration. Here, h^+ and h^- represent activating and repressing Hill functions with Hill coefficient N and half-saturation concentration k_A. Additional network constants include protein translation rate k_{tl}; maximum and unrepressed transcription rates β and P_{tc}; and degradation rates for mRNA (δ_m) and repressor proteins (δ_x). Thus, we borrow the network's mathematical description from the original paper to replicate each of the hereby presented experiments. Unless otherwise stated, experimental results on the next section have protein concentrations initially set to $R1 = R2 = 50nM$, $R3 = R4 = 0nM$ and reaction parameters are the same as given in the reference work.

$$[\dot{R1}] = \alpha h^+([I])h^-([R2]) + \gamma h^-([R3]) - \delta_x[R1]$$
$$[\dot{R2}] = \alpha h^+([I])h^-([R4]) + \gamma h^-([R3])h^-([R4]) - \delta_x[R2]$$
$$[\dot{R3}] = \alpha h^+([I])h^-([R4]) + \gamma h^-([R1]) - \delta_x[R3]$$
$$[\dot{R4}] = \alpha h^+([I])h^-([R2]) + \gamma h^-([R1])h^-([R2]) - \delta_x[R4]$$

$$h^+([X]) = \frac{[X]^N}{k_A^N + [X]^N} \qquad \alpha = \frac{k_{tl}\beta}{\delta_m}$$
$$h^-([X]) = \frac{1}{1 + \frac{[X]^N}{k_A^N}} \qquad \gamma = \frac{k_{tl}P_{tc}}{\delta_m}$$

All numerical simulations were performed in Octave 5.2.0, with additional packages `signal-1.4.1` and `control-3.2.0`. We used two integration methods throughout: a hand-written Euler's method implementation with a step size of 60 seconds and Octave's Dormand-Prince method `ode45`, both yielding similar results on all of our experiments. This differs from the MATLAB solvers – `ode45` for the deterministic simulations and `ode4` for the stochastic ones – used by Purcell et al.[11], but Euler's method is justified by the duration of experiments, the shortest of which take at least four days (virtual simulated time) in order to observe a couple of periods on the oscillating output of the frequency divider. Due to this difference, some numerical mismatch is to be expected, while qualitative behaviour should remain the same.

Every deterministic experiment was carried in two systems: one considering the whole set of ODEs and another with the QSSA approximation that is used throughout the original study. Quantitative results shown refer to the full model and deviations between that and the reduced one are highlighted in the text. While stochastic simulations are only briefly mentioned in the reference work, more details can be found inside supporting information documents ("File S6", in [11]). The Chemical Langevin Equations (CLEs) described therein were implemented with Gaussian noise being generated through the use of built-in Octave functions.

3 Results

3.1 Frequency Divider

The network was originally designed as a frequency divider such that the concentrations of repressor proteins oscillate in approximately one half of the input frequency. This behaviour can be observed by feeding the model with a continuously oscillating input – this would be often the case considering existing genetic oscillators [8] – but also works with square waves (Figure 1).
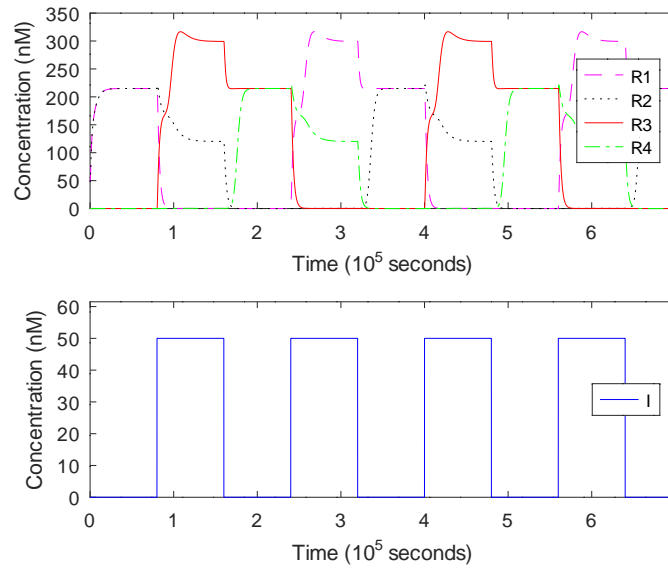

Figure 1. Frequency division of clock-like input. Here we replicate the original paper's Figure 3 with a square wave input having 50 nM amplitude, period of 1.6×10^5 seconds and 50% duty cycle.

(a) Frequency division in the full model.

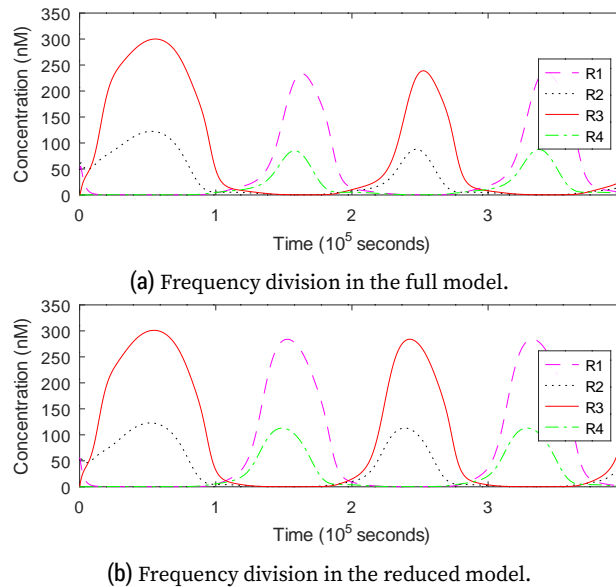(b) Frequency division in the reduced model.

Figure 2. Comparing frequency division between full and reduced models. This experiment replicates Figure 4 in the reference work. Input varies as a sinusoidal signal with amplitude of 50 nM, minimum of 6 nM and a period of 0.9×10^5 seconds.

With a sinusoidal input, output signals from the QSSA model have constant amplitude, whereas in the full model the first concentration peak of proteins R2 and R3 are higher than the following ones. This suggests mRNA reactions stabilize quicker in the first 10^5 seconds and thus the approximation is more accurate at those instants [13]. After that moment, however, the reduced model exhibits a persistent offset in relation to the full one.

As illustrated in Figures 2a and 2b, while R1 and R4 concentrations in the full model reach *maxima* valued at 238.29 nM and 87.62 nM respectively, the reduced model goes up to 283.96 nM and 113.05 nM for each of these proteins (peak values of R3 follow R1 closely and the same happens with R2 and R4). This offset distinguishing the two models happens because the QSSA used to derive the reduced ODE system considers a separation of time-scales between reactions which regulate mRNA production and those which describe protein translation. In the approximation, the former reactions are assumed to reach equilibrium instantaneously relative to the latter. Thus, the difference comes from the fact that the original model maintains itself in a dynamic state that never actually reaches equilibrium [13], as it perpetually oscillates.

As mentioned in the original study, frequency division functionality can only be observed after a specific period threshold. We ran experiments over a range of input frequencies and found the period-doubling bifurcation to be located near values of 0.8×10^5 seconds (~ 22 hours) in both full and reduced models. Although this confirms the network's capability to interface with long-period oscillators, these results – shown in Figure 3 – greatly diverge from those in the reference work, which state this threshold could be observed at input periods of approximately 0.275×10^5 seconds (~ 8 hours). We note that varying the integration time step between 1 and 60 seconds had little to no effect on the location of the period-doubling bifurcation and neitheir did changing the integration method: the `ode45` solver yields the same results.
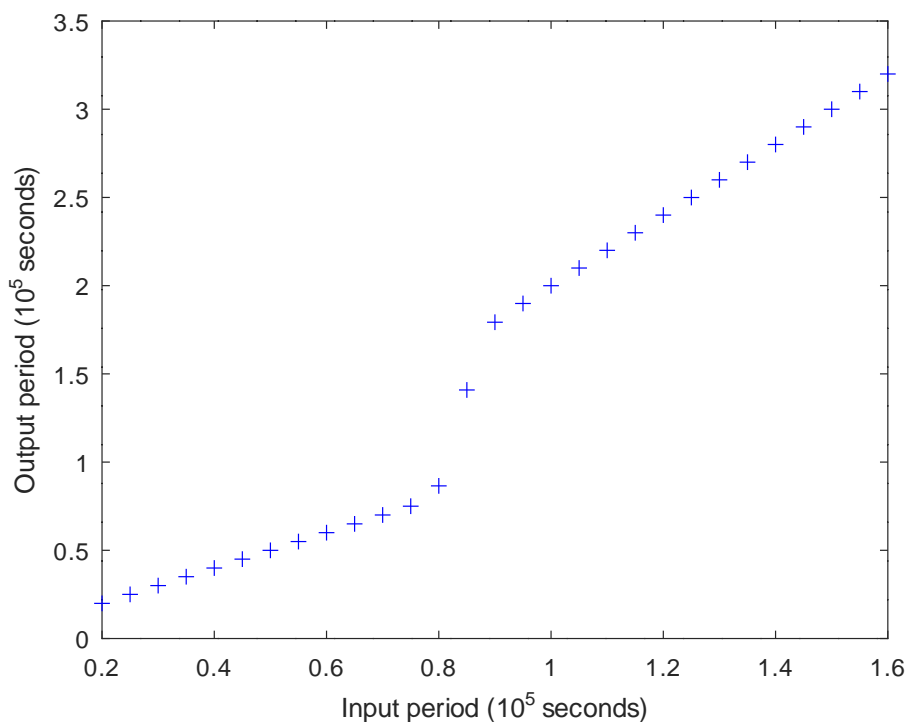

Figure 3. **Locating the period-doubling bifurcation.** Through this set of successive simulations, we attempted to replicate Figure 7A from Purcell et al.[11]. Output period is detected by measuring the distance between R1 concentration peaks. Other than input frequency and simulation length (each run was configured to take as long as five times the input period), parameters are the same as in Figure 2a.

3.2 Bifurcation Analysis

Purcell et al.[11] discovered the model's multiple extra functionalities by verifying different behaviours could be attained when input concentration was held constant at specific ranges. Experiments regarding the so-called bifurcation analysis were reproduced and Figures 4-7 show results under the full model. These correspond to the simulation panels displayed within Figure 5 in the reference paper.

The original study states experiments labeled $4c_1$ and $1b$ use the same initial conditions as $4c_2$ and $1a$ respectively. We believe these were typographical errors, as such settings would lead those pairs of experiments to the exact same results under deterministic semantics and this is not the exhibited behaviour. Instead, whereas reaction parameters and input levels are the same as in the original work, initial conditions used are $R1 = R2 = 0nM$ and $R3 = R4 = 50nM$ for experiments $1b$ and $4c_2$ and $R1 = R2 = 50nM$ and $R3 = R4 = 0nM$ for all others.
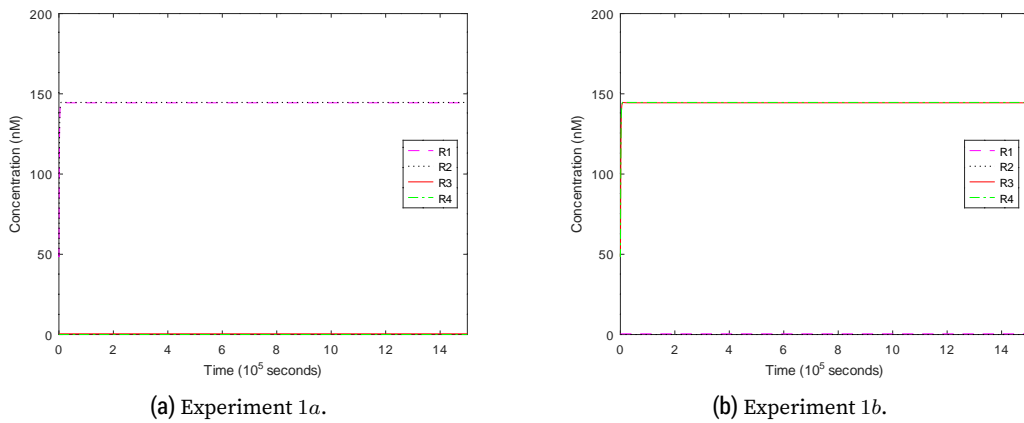

(a) Experiment $1a$.
(b) Experiment $1b$.

Figure 4. Low concentration bistable behaviour. $I = 0.1nM$.
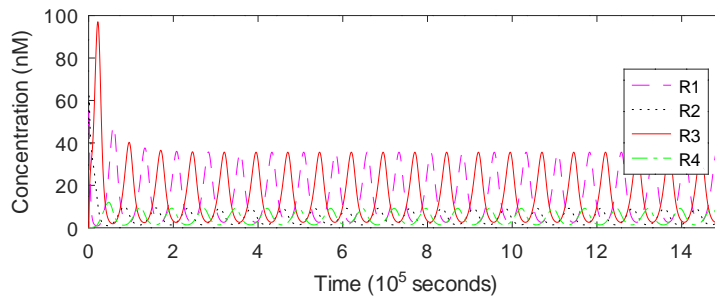
Figure 5. Experiment 2. $I = 5nM$.

Figure 6. Experiment 3. $I = 7.5nM$.

(a) Experiment $4c_1$.

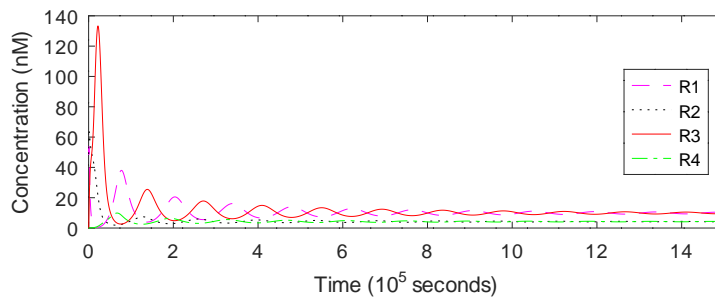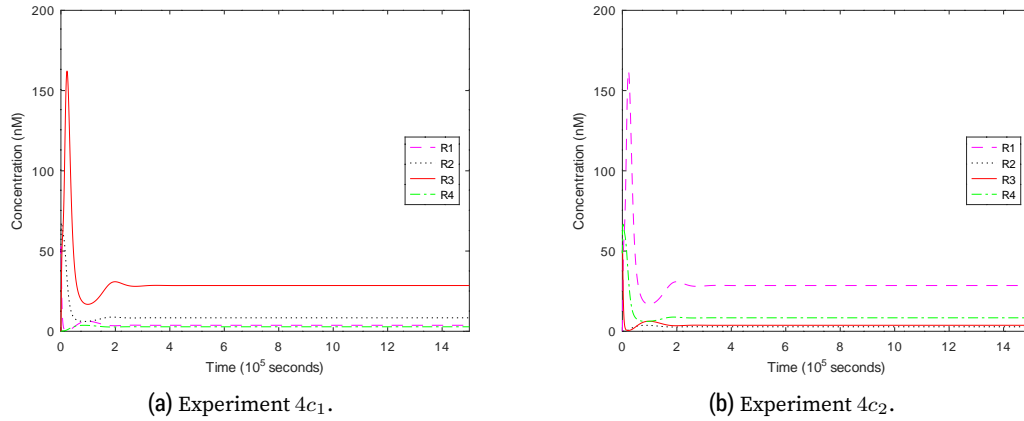(b) Experiment $4c_2$.

Figure 7. High concentration bistable behaviour. $I = 10nM$.

3.3 Self-induced Oscillator

We verified the network's oscillatory dynamics in the region between saddle-node and Hopf bifurcations by measuring its output period for each given input concentration level. Results illustrated in our Figure 8 describe a relation with similar behaviour and the same near-vertical increase in period when input approaches the lower bistability region as Figure 6A in the original paper.
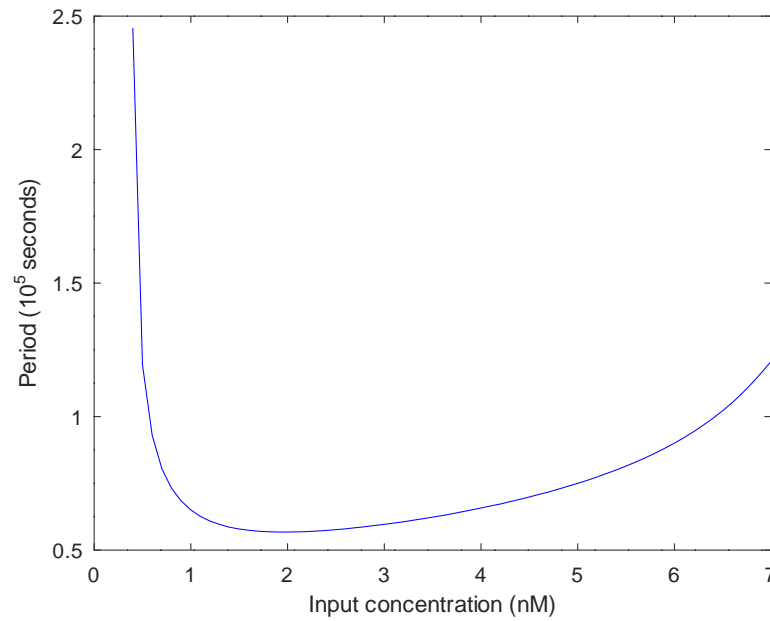

Figure 8. **Analysing oscillatory dynamics.** Simulation configuration is the same as in Figure 5 except for input levels, which are held between 0.4 nM and 7 nM. Output period is detected by measuring the distance between R1 concentration peaks.

3.4 Toggle Switch

As revealed during bifurcation analysis, the network exhibits bistability when input concentration is held outside the oscillatory range, that is, at levels lower than 0.4 nM or greater than 7 nM. Figure 9 illustrates the system being used as a toggle switch which is "triggered" by varying binding affinity of particular repressors, altering k_A momentarily

from $6 \times 10^{-10} M$ to $4 \times 10^{-6} M$, as described in the reference work.

It is important to note that this behaviour can be observed even when the simulated system does not start from "clean" (i.e. unused) state. This means the switch can be toggled on and off repeatedly (not shown), which is to be expected in any non-trivial synthetic biology application [2].
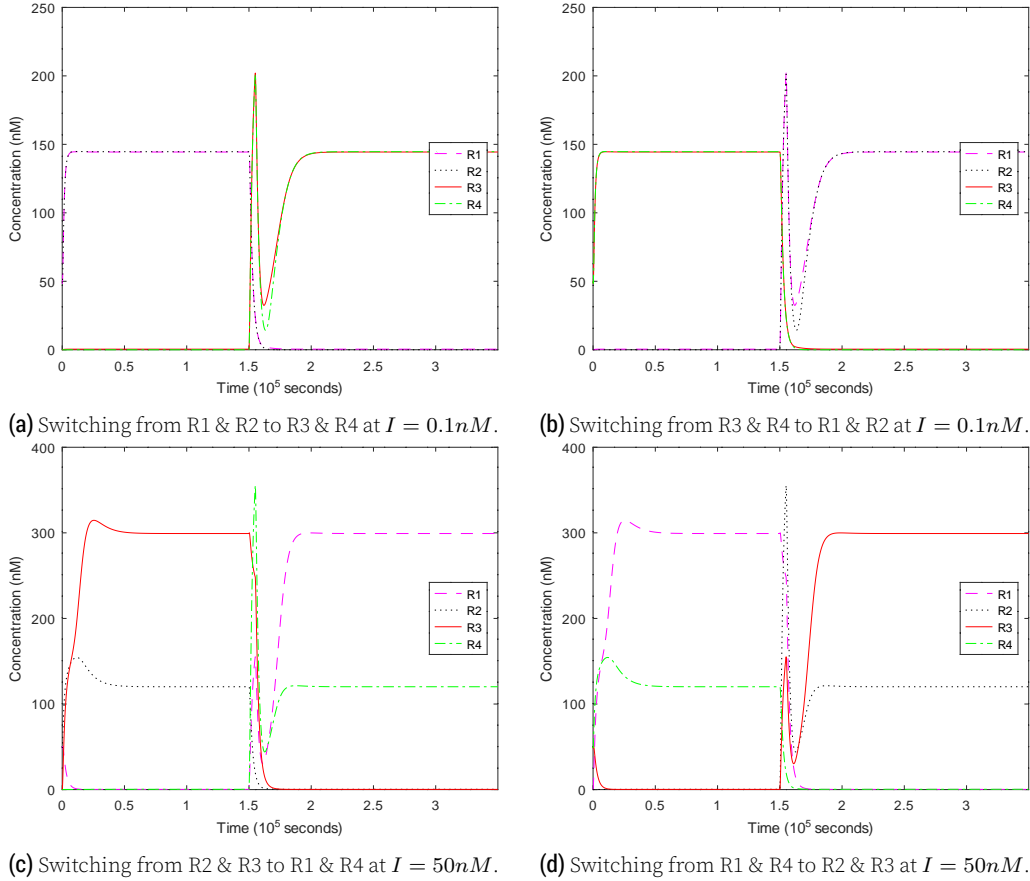

(a) Switching from R1 & R2 to R3 & R4 at $I = 0.1 nM$.

(b) Switching from R3 & R4 to R1 & R2 at $I = 0.1 nM$.

(c) Switching from R2 & R3 to R1 & R4 at $I = 50 nM$.

(d) Switching from R1 & R4 to R2 & R3 at $I = 50 nM$.

Figure 9. Demonstrating toggle-switch function. k_A is altered simultaneously for R1 and R2 in 9a and 9c, while in 9b and 9d the switch is performed by increasing the k_A for R3 and R4. This variation is held between times 1.50×10^5 and 1.55×10^5 seconds. These experiments replicate what is shown in Figure 8 of the reference work.

3.5 Stochastic Simulations

We implemented the CLEs shown in the related supporting information document ("File S6") provided by Purcell et al.[11], but it was not possible to verify similar results without modifications to the noise-inducing function. The reference work states the usage of Gaussian noise with zero mean and variance of 1, but employing Octave's `randn` function (which provides such a distribution [14]) proved being too noisy, as stochastic fluctuations began dominating the model's behaviour.

This phenomenon was reproduced using both the fixed-step Euler method and the adaptive `ode45` integrator. We are led to believe this is due to the usage of standard ODE solvers (instead of a more appropriate stochastic method), which have the effect of increasing noise variance based on the total number of simulated steps. In order to approximate the results in the reference work, random numbers are scaled by a factor – found empirically – $s \in [\frac{1}{100}, \frac{1}{10}]$, consequently downscaling variance by s^2. Figures 10,

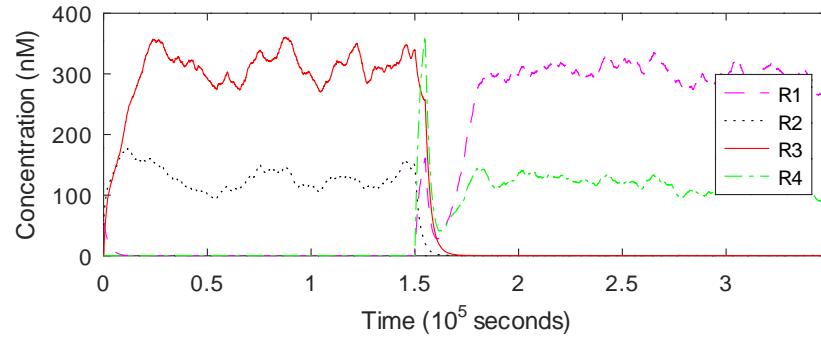11 and 12 show some stochastic simulations with this compensation factor applied.

Figure 10. Effect of noise on switching function. Random seed is set to 73544911520192 and fluctuations are scaled by $\frac{1}{55}$ for a fixed input of $50\,\mathrm{nM}$.
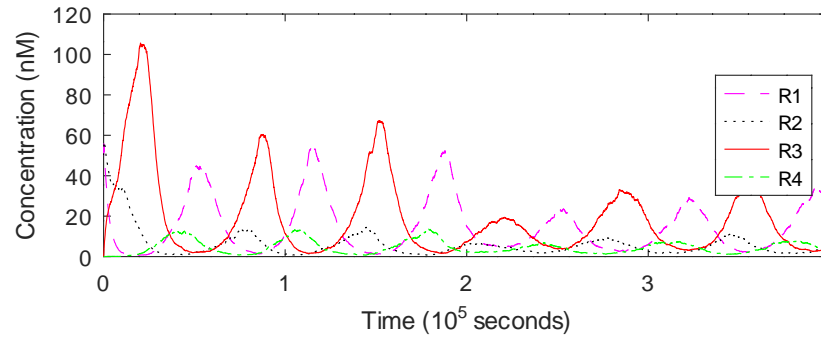

Figure 11. Effect of noise on oscillator functionality. Input is set to $5\,\mathrm{nM}$ and noise is scaled by $\frac{1}{55}$. Random seed: 73544911520192.
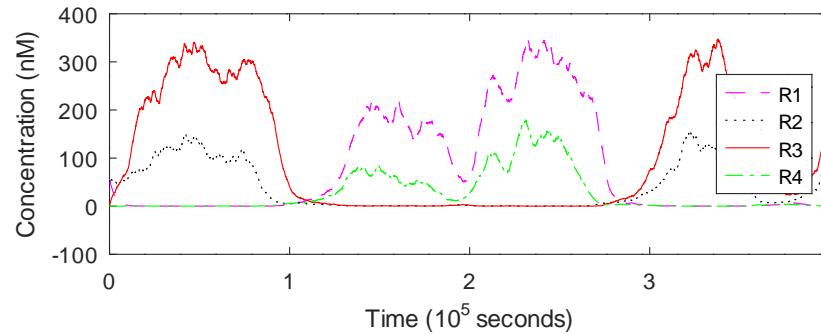

Figure 12. Irregular oscillations caused by stochastic fluctuations. Input is the same as in Figure 2a. Random seed is set to 5.83346446892352 and noise is scaled by $\frac{1}{26}$.

We verified that among the three functionalities, frequency division appears less robust to intrinsic noise: when input is applied and proteins are all at low concentrations, random fluctuations may cause unintended oscillations as a pair of repressors rise in concentration and prevent transcription of the other two. An example of such irregular oscillations is given in Figure 12, in which we use a specific pairing of random seed and scaling factor to illustrate this behaviour.

4 Conclusions

All operation configurations in the multi-functional synthetic gene network were verified and to a large degree the results presented in [11] were replicated. It should be emphasised that the original work stands as an example of reproducible science, with detailed descriptions of model derivation, parameter decisions and additional experiments in openly available supplementary material. The QSSA simplification proved being a good approximation of the full ODE system, since even though some small quantitative differences were observed – and those could be due to the different integration methods as well – there was no impact on the network's overall dynamics.

However, stochastic simulations would not provide the expected results without an expressive reduction to noise variance. In fact, Gaussian noise seems capable of undermining frequency division functionality to some degree. Additionally, the period-doubling bifurcation was found at a much lower frequency than what is stated in the reference study. Upon correspondence with the authors, it was speculated that the displacement could have been caused by some mismatch between the parameters in the paper and those actually used in their simulation, since qualitative behaviour was mostly maintained and all other deterministic simulations match.

We highlight that, as synthetic networks grow in complexity and size, multiple functions may arise more frequently and even become difficult to avoid. While this could allow for reusable programmable components, it might also become a nuisance if models start behaving unexpectedly under the influence of certain inputs. At last, we believe further extensions of this work could focus on a deeper exploration of parameter space. The motivation for this is double: it will allow for the design of *in vivo* implementations with reusable biological parts and may uncover the complete characterisation of this programmable genetic network, much like the datasheet of an electronic component.

References

1. D. E. Cameron, C. J. Bashor, and J. J. Collins. "A brief history of synthetic biology." In: **Nature Reviews Microbiology** 12 (5 Apr. 2014), pp. 381–390.
2. A. Goni-Moreno and M. Amos. "A reconfigurable NAND/NOR genetic logic gate." In: **BMC Systems Biology** 6 (Sept. 2012), p. 126.
3. S. A. Salehi, H. Jiang, M. D. Riedel, and K. K. Parhi. "Molecular Sensing and Computing Systems." In: **IEEE Transactions on Molecular, Biological and Multi-Scale Communications** 1.3 (Sept. 2015), pp. 249–264.
4. M. B. Elowitz and S. Leibler. "A synthetic oscillatory network of transcriptional regulators." In: **Nature** 403 (Jan. 2000), pp. 335–338.
5. C. Lin, T. Kuo, and Y. Chen. "Implementation of a genetic logic circuit: bio-register." In: **Systems and Synthetic Biology** 9.1 (Dec. 2015), pp. 43–48.
6. L. Cardelli, M. Kwiatkowska, and M. Whitby. "Chemical reaction network designs for asynchronous logic circuits." In: **Natural Computing** 17.1 (Mar. 2018), pp. 109–130.
7. G. H. Moe-Behrens. "The Biological Microprocessor, or How to Build a Computer with Biological Parts." In: **Computational and Structural Biotechnology Journal** 7.8 (2013), e201304003.
8. T. Mahajan and K. Rai. "A novel optogenetically tunable frequency modulating oscillator." In: **PLOS ONE** 13.2 (Feb. 2018), pp. 1–29.
9. A. Khalil and J. Collins. "Synthetic Biology: Applications Come of Age." In: **Nature reviews. Genetics** 11 (May 2010), pp. 367–379.
10. C. Lin, P. Chen, and Y. Cheng. "Synthesising gene clock with toggle switch and oscillator." In: **IET Systems Biology** 9.3 (2015), pp. 88–94.
11. O. Purcell, M. di Bernardo, C. S. Grierson, and N. J. Savery. "A Multi-Functional Synthetic Gene Network: A Frequency Multiplier, Oscillator and Switch." In: **PLOS ONE** 6.2 (Feb. 2011), pp. 1–12.
12. T. R. Kuphaldt. **Lessons In Electric Circuits: Volume VI - Experiments**. URL: https://www.ibiblio.org/kuphaldt/electricCircuits/Exper/EXP_8.html (visited on 12/30/2019).
13. B. P. Ingalls. **Mathematical Modeling in Systems Biology: An Introduction**. The MIT Press, Jan. 2013.
14. Octave Forge Community. **Octave randn Documentation**. URL: https://octave.sourceforge.io/octave/function/randn.html (visited on 11/11/2019).