A synthetic approach towards building a custom biological circuit

Martin Stražar*, Nikolaj Zimic, Miha Mraz, Miha Moškon

Faculty of Computer and Information Science, University of Ljubljana, Tržaska cesta 25, Ljubljana, Slovenia
Phone: +(386)14768371; Fax: +(386)14264647

Email: Martin Stražar*- martin.strazar@gmail.com;

*Corresponding author

Abstract

1 Introduction

Over the last decades, great understanding has been reached in the field of biology, in terms of one of the key life processes in the cell - protein production from the instructions encoded in an organism's DNA. Protein production is controlled by means of transcription factors, which bind to their appopriate sites on the DNA strand, acting as transcription activators or repressors. By utilizing recombinant transcription factor binding sites, transcription and consequently protein production rates can be controlled. Technological advances in terms of processes, methods and infrastructure have made it possible to look on the subject from an engineer's point of view. It opens up the potential to realize a living device with an merely arbitrary functionality (up to the constraints on complexity (1)), which could solve problems using formal logic (2). The vast range of applicable problems include curing various kinds of diseases, hereditary disorders, biochemical disasters etc. The basic logical operations analogous to digital circuits have been sucessfully realized (3; 4; 5). Our goal is to take the achievements further towards tangible control of biological circuit functionality.

The process of a live realization of a biological circuit can be substantially time and money consuming. DNA sequencing and cell culture incubation are an example of the steps that can take hours to days. Detailed planning and modelling is needed in order to eliminate as many mistakes as possible in the early design phase. Mathematical models of biological circuits tend to convey their dynamics as accurately as possible.

As opposed to electronic digital circuits, the depth of subprocesses involved in protein production is much larger. Intermediate steps in protein production, such as RNA polymerase binding, mRNA translation, protein transcription, multimerization or phosphorilation all take a significant amount of time and have to be accounted for. Consequently, the tendency is to model the dynamics in depth (6).

The very details of a cell functioning for any organism are not yet fully understood. To model a biological circuit, the knowledge gathered is conveyed in the form of ordinary differential equations or chemical equations. A common approach towards modelling a biological circuit goes as follows: after gathering the measured experimental data, the appropriate model is chosen amongst a set of possible models by using statistical methods (7; 8). Parameters that best fit the measured experimental data are then estimated, using some measure of error, such as least mean square (9). A thorough overview of parameter estimation methods using ordinary differential equations is given in (10). The parameter search space is vast, where some parameters can be experimentally measured (e.g. reaction rate constants), while others cannot (e.g. non-linearity in protein production) and are most often fitted (11).

To solve a problem *in vivo*, one has to adapt to spatial and time constraints. An example is insulin secretion in human beings. As an example, insulin is a signal for the body to store the blood glucose, which oscillates with the period of approximately 5-15 minutes (12). The proper timing in insulin intake is key in treating *Diabetes type 1*, from where it follows that the system would have to obey the given time constraints. A synthetic approach towards tackling the problem consists of the definition of a desired function of the circuit, presenting it in a proper model and determining the parameter values that will produce the desired resulting behaviour. Knowing the resulting parameter values, the decisions in chosing parts and materials while building a biological circuit are facilitated.

The systems of differential equations are tipically composed of a large number of parameters. Hence, the analitical approaches towards integration are often unfeasible. For our purposes, numerical integration methods with sufficient accuracy (13) are used.

Achieving the desired behaviour with a large number of parameters is a non-trivial task. Again, analitically solving the problem includes a complex search for bifurcation points (14). A common workaround are natural computing techniques, which exhibit natural phenomena as inspiration for solving the problem (15). One such phenomena, the survival of the fittest, is used in genetic algorithms.

We demonstrate a concept of a bottom up approach towards building a biological circuit by using a model of an oscillator, based on a time delayed, negative feedback loop (13). In spite of accuracy of the model being crucial for the accuracy of the result, our test model lies upon a generalization of intermediate

processes (grouped under a time delay variable), which does not affect the proof of concept. The desired behaviour of the oscillator is essentially described by the amplitude and the frequency of the oscillations. An application of a genetic algorithm is then used to explore the parameter space and return the parameter values that produce the desired behaviour.

In section 2, the computatinal framework for solving the problem is defined. Furthermore, the parameter space search algorithm and its improvement are presented. As a case example, the model of an oscillator circuit is examined. In section 3, the algorithm is run on three test cases and the results are analyzed. Section 4 comments on the overall achievement and discusses future work and applications.

2 Computational framework

Ordinary differential equations are a common way to represent a biological circuit. In essence, their components can be classified as observed chemical species and the equation parameters, hereafter refered to as *a parameter set*. First can be interpreted as input and output variables of the system, while the second determine its dynamics. The problem of achieving desired system behaviour is then reduced to an optimization problem, subject to finding the appropriate values of the parameter set.

In order to employ a genetic algorithm, the parameter sets are modelled as members of a population. Accross the population, parameter sets differ in their values. Each set determines the behaviour of the system, which is in turn evaluated by a *fitness function*. The population evolves through generations of new members, introducing random changes in parameter values, also known as *mutations*. At the end of each such iteration, the best members are selected and chosen for reproduction. In such manner, analogous to natural selection, the population converges towards finding the optimal result (16). The basic outline of a genetic algorithm, adapted from (15), is sketched in Figure 1.

1. Initialize the population.
2. Evaluate the population
3. While the desired result is not reached:
   (a) Select parents
   (b) Recombine selected parents
   (c) Mutate the resulting offspring
   (d) Evaluate new members on the fitness function
   (e) Select individual members for the next generation

Figure 1: The basic outline of a generic genetic algorithm.

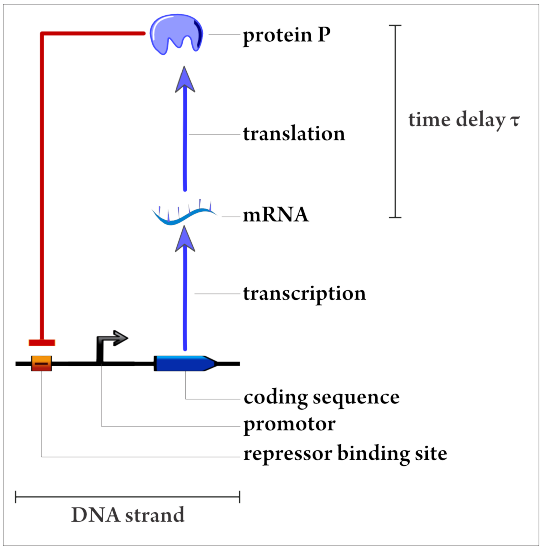

Figure 2: Schematic representation of the model biological circuit

2.1 Case study: a biological oscillator

So far, many mathematical models of biological oscillators have been proposed. A common property is the negative feedback loop in the synthesis of the observed protein. Without further assumptions, the negative loop itself leads into a stable steady state (17). Assuming an additional delay between transcription and translation phases and nonlinearity in protein synthesis cascade, the system produces oscillations in concentrations of observed chemical species. Due to their dynamic nature, many processes involved during protein synthesis are difficult to quantify. Examples include protein phosphorylation or protein folding, which is its own field of research altogether. Consequently, numerical approximations are used in order to get as close as possible to a formal description of the observed system. For the purpose of our proof of concept, we decided to keep the model as simple as possible. Scheper et al. (13) propose a model of intracellular circadian oscillator, based on the following Hill equations:

$$\frac{dM}{dt} = \frac{r_M}{1 + (P(t)/k)^n} - q_M \cdot M(t) \tag{1}$$

$$\frac{dP}{dt} = r_P \cdot M(t - \tau)^m - q_P \cdot P(t) \tag{2}$$

where $M(t)$ and $P(t)$ represent the concentrations of mRNA and the resulting protein, repectively. The model biological circuit is depicted in Figure 2. The parameters in the two equations compose a parameter set and are described in Table 1.

Parameter label	Meaning	Initial value
r_M	Rate of production (mRNA)	1.0
r_P	Rate of production (protein)	1.0
q_M	Rate of degradation (mRNA)	1.0
q_P	Rate of degradation (protein)	1.0
τ	Transcription-translation delay	1.0
m	Nonlinearity in protein synthesis cascade	3.0
n	Hill coeficient	2.0
k	Scaling constant	1.0

Table 1: The observed parameter set and the initial values.

2.2 Problem definition

In (13), a parameter subspace search was performed, where not all the parameters in the equation were considered. In this work, we look at the problem from another angle: given the oscillator behaviour, we derive the parameter values that will produce it.

Since it would not be feasible to find all the solutions of the two equations analitically, we make use of a customized genetic algorithm to derive the parameter values that will result in a system with minimal error regarding the desired behaviour. The input of the algorithm are the arguments of a sine wave, which describes the desired behaviour:

$$y_{wave}(t) = A_{wave} \cdot sin(\frac{1}{\lambda_{wave}} \cdot t) + \frac{A_{wave}}{2}, \tag{3}$$

where A_{wave} and λ_{wave} stand for amplitude and wavelength, respectively. In order to make the problem solution as general as possible, all quantities in the system are to be interpreted on an arbitrary unit scale.

2.3 Parameter set evaluation

Given the time interval and time step, the values of the equations 1 and 2 are computed using a numerical integration method for each parameter set (denoted *pset* in the following equations). Depending on the parameter values, the system can produce oscillations or end up in a stable steady state. From the obtained numerical data, the following properties are derived: a flag, whether the observed parameter set produces oscillations (B_{pset}), oscillating amplitude (A_{pset}), wavelength (λ_{pset}), and the rise and fall times ($t_{pset/fall}$ and $t_{pset/rise}$). These are involved in computing the criteria, which in turn are used to evaluate the fitness function. The former is the core of the selection process. The criteria are computed as described in Table 2. The smaller the value of a single criterium, the better the observed parameter set matches it.

Shorthand	Criterium	Evaluation
C_A	$Amplitude$	$\lvert A_{wave} - A_{set}\rvert$
C_λ	$Wavelength$	$\lvert\lambda_{wave} - \lambda_{set}\rvert$
C_{sym}	$Symmetry$	$\lvert t_{set/rise} - t_{set/fall}\rvert$

Table 2: Criteria used to compute the fitness function.

The fitness function[1] is then defined as follows:

$$F_{pset} = \begin{cases} \frac{C_A + C_\lambda + C_{sym}}{3}, & B_{pset} = 1 \\ \infty, & B_{pset} = 0 \end{cases} \tag{4}$$

2.4 Deriving the optimal solution

The initial population of parameter sets is derived by introducing mutations in the initial values, defined in Table 1. The fitness function (equation 4) is evaluated for each parameter set and the top half of the population (parameter sets with the values of the fitness funtion less than the population mean) is preserved in the next generation. Additionaly, new parameter sets are derived as offspring from the existing population. The closer a parameter set is to the desired behaviour, the more offspring will it produce, as the probability to get further close to the desired behaviour is larger. The number of offspring for an individual parameter set in population $\mathcal{P}$ is calculated as follows:

$$F_{cut} = E(\sum_{pset} F_{pset}) \tag{5}$$

$$\mathcal{P}_{new} = \{pset\ \epsilon\ \mathcal{P}; F_{pset} \leq F_{cut}\} \tag{6}$$

$$O_{total} = 1 + |\mathcal{P}_{new}| \tag{7}$$

$$O_{pset} = \lfloor \frac{1}{F_{pset}} \cdot \frac{1}{\sum_{x \epsilon \mathcal{P}_{new}} \frac{1}{F_x}} \cdot O_{total} \rfloor \tag{8}$$

where F_{cut} is the mean fitness of the population, $\mathcal{P}_{new}$ is a subset of the population to be reproduced, O_{total} is the total number of offspring produced for the current generation, of which each population member gets its share by calculating O_{pset} in equation 8. The child parameter set is derived from its parent with a pretedermined mutation probability at any of the definitive eight parameter values. In such manner, the procedure is repeated from generation to generation for a predefined number of times. In each generation new individuals are derived and the best are selected. As shown in (16), such system will monotonically converge towards global maxima and sucessfully avoid local minima, which was the case in our tests as well.

[1] In our case, the lesser the value of the fitness function, the better the parameter set matches the desired behaviour.

2.5 Improving the search time

During the development phase, the initial solution was improved in the following way. The simulation of the life of a population is repeated and the solution at the end of one simulation is the input to the next. We repeat such *step* for multiple times. In each step, the fitness function is modified, such that only a subset of the criteria (see Table 2) is evaluated. A single step is then repeated until the desired user defined accuracy is reached. Using the resulting solution, the algorithm then proceeds in evaluating the next criteria subset. The criteria subset for a given step is shown in Table 3.

Criteria subset number	Evaluated criteria subset
1	Amplitude
2	Wavelength
3	Amplitude, wavelength, symmetry

Table 3: Choosen subsets of evaluated criteria depending on the current step.

This modification allows us to find the solution more quickly, escape local maxima or to bias one of the criteria. The final solution outline is given in Figure 3. Note that the recombination step from Figure 1 has been skipped in our solution, since the experiments have shown better results without it.

1. Initialize the parameter set population.
2. Evaluate the population
3. While the desired behaviour is not reached:
   (a) Select parents and produce offspring, inversely proprotional to their fitness value
   (b) Mutate the resulting offspring
   (c) Evaluate new members
   (d) Select individual members for the next generation

Figure 3: The solution outline.

3 Results

The algorithm was tested for various input behaviours. To reproduce the behaviour of (13), with input amplitude of 16 nM and wavelength of 24 h, we got perfectly symetric stable oscillations as close as 15.72 nM amplitude (0.017 % deviation) and 24.14 h (0.0062% deviation). The sampling time was 1000 h and sampling step was 6 min. The resulting parameters are shown in Table 4, column Test 1, and differ from the ones proposed in (13). To achieve the result, 90 generations of oscillators were analyzed with maximum population

size of 3200. The wavelength and amplitude convergence graphs are shown in figures [] and [].

In accordance with the tendency to achieve faster frequencies and tunable amplitudes, we tested the system for other input behaviours. In the following example, the input amplitude was 8.0 nM and wavelength of 12 h. The end system resulted in perfectly symetric stable oscillations of 7.18 nM (10.23 % deviation) and 12.31 h wavelength (0.025 % deviation) and the resulting parameters are shown in Table 4, column Test 2.

Some parameters are difficult to quantify (i.e. nonlinearity in protein synthesis cascade, Hill's coefficient and scaling constants, ...). In order to test the flexibility of the algorithm, a subset of parameters can be left intact (i.e. be unchanged during the course of the algorithm). Again, the solution was tested against values in (13), leaving the m parameter anchored at it's initial value. The resulting amplitude and wavelength were 15.607 nM (0.024% deviation) and 25.299 (0.054% deviation) respectively. Resulting parameters are shown in Table 4, column Test 3.

Parameter label	Units	Test 1	Test 2	Test 3
r_M	h^-1	1.976	2.986	2.027
r_P	h^-1	1.666	2.731	1.610
q_M	h^-1	0.536	0.739	0.662
q_P	h^-1	0.996	1.134	0.928
τ	h	9.236	4.262	8.449
m	None	2.858	0.802	3.0
n	None	6.148	6.429	3.056
k	None	2.211	3.856	0.088

Table 4: Parameters of the system an their resulting values

The results show that there are multiple points in parameter space that produce similar behaviours for a given system, as well as that with a given desired behaviour, the required parameter values can be derived. Parameter values can be controlled to some degree (using promoters with various strengths, proteins and operator sites with different binding rates, post-transcriptional regulation to modify the delay, etc.), allowing us to engineer arbitrary structures with desired behaviours.

4 Conclusions and future work

The advances in synthetic biology, genetic engineering and bioinformatics will allow researchers to shift towards a synthetic approach in buildic biological logic components. The growing registries of DNA parts (promoters, regulator binding sites, coding sequences etc.) help us to predict the properties of the systems that are to be built (example: http://www.biobricks.org). A number of health conditions require treatment

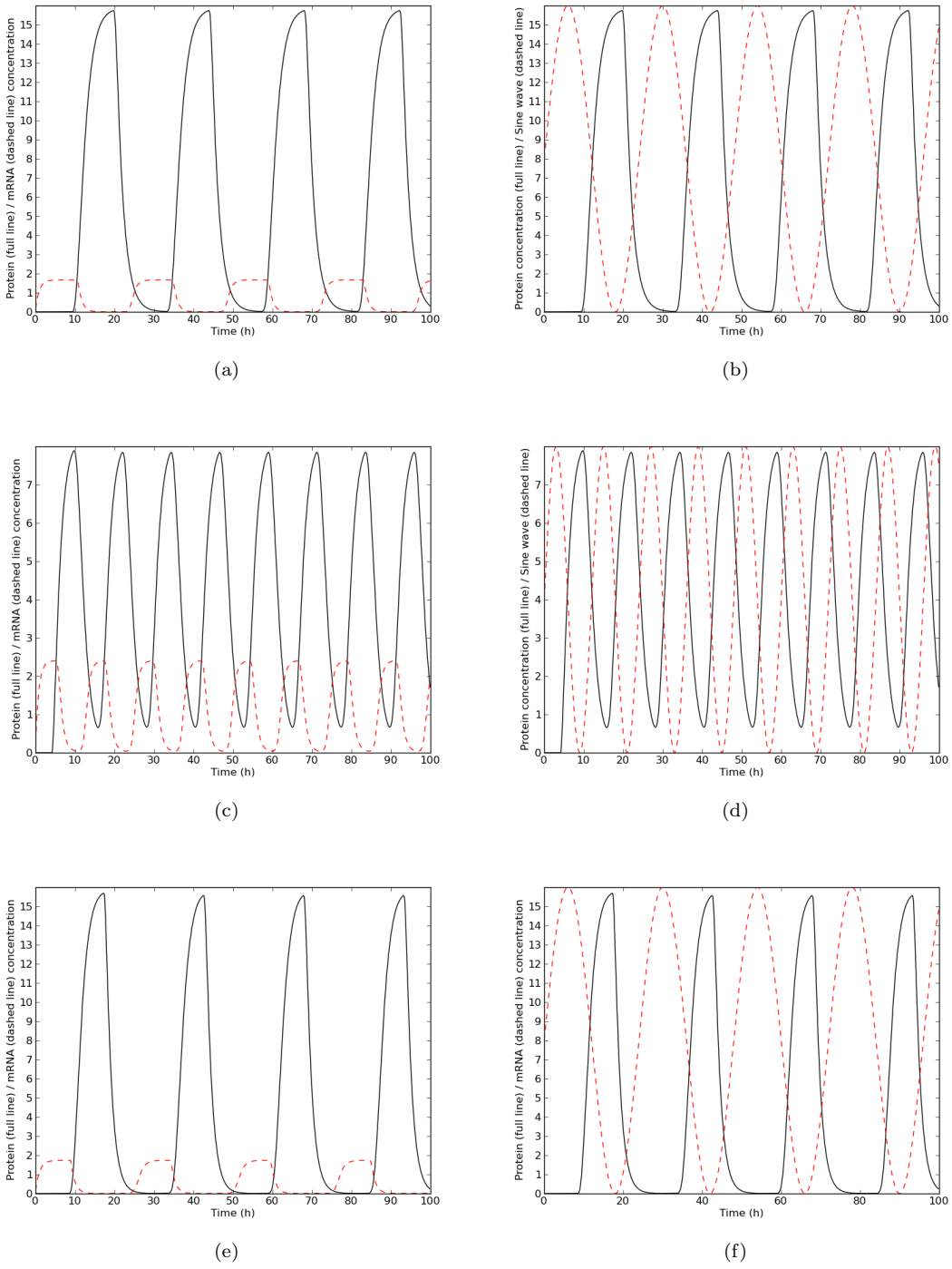

Figure 4: (a) Test 1: concentrations of protein (full line) and mRNA (dashed line). (b) Test 1: Comparison with protein concentration (full line) and target behaviour sine wave (dashed line). (c) Test 2: concentrations of protein (full line) and mRNA (dashed line). (d) Test 2: Comparison with protein concentration (full line) and target behaviour sine wave (dashed line). (e) Test 3: concentrations of protein (full line) and mRNA (dashed line). (f) Test 3: Comparison with protein concentration (full line) and target behaviour sine wave (dashed line).

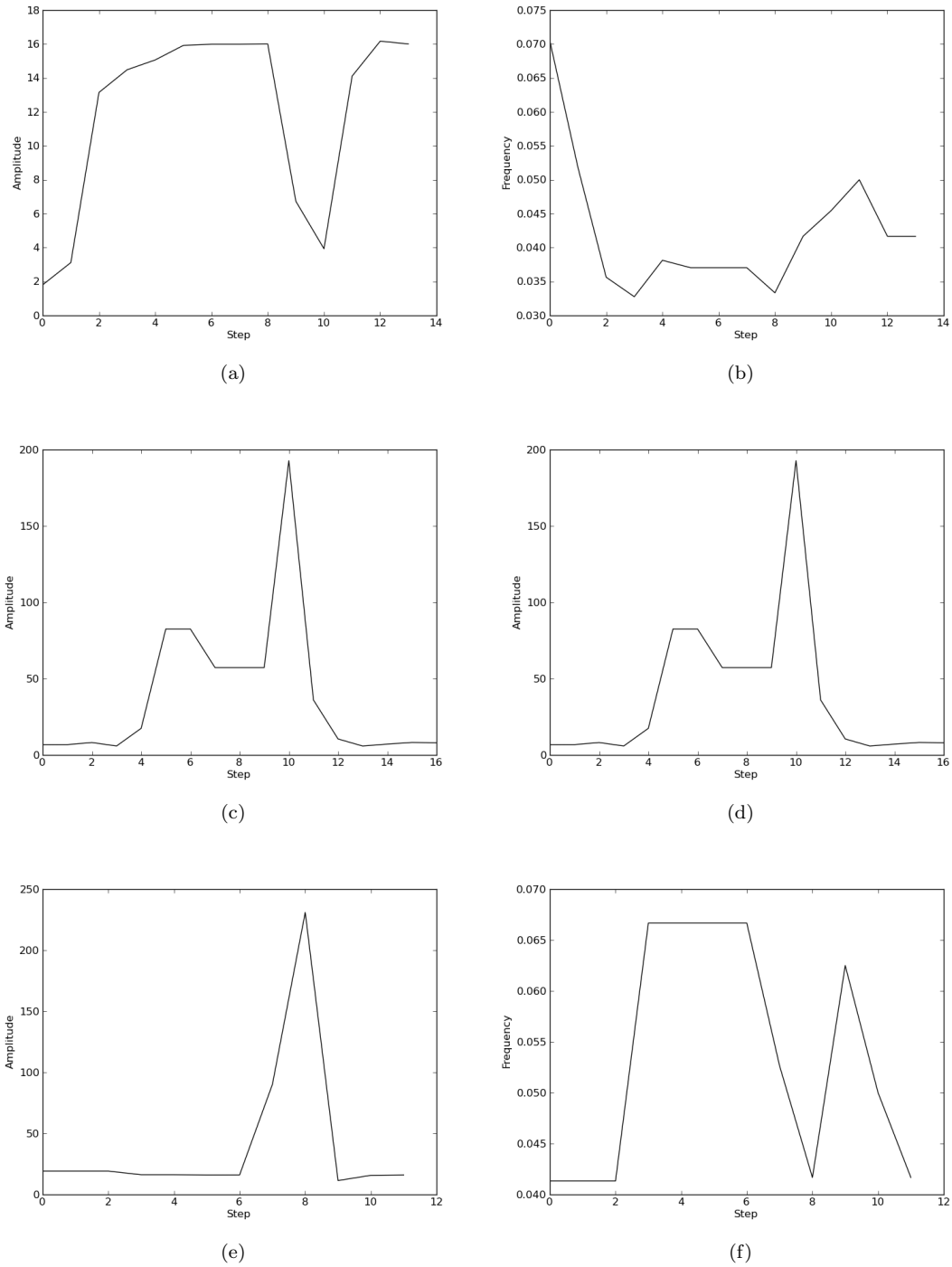

Figure 5: (a) Test 1: Amplitude value convergence. (b) Test 1: Frequency ($1/\lambda$) value convergence. (c) Test 2: Amplitude value convergence. (d) Test 2: Frequency ($1/\lambda$) value convergence. (e) Test 3: Amplitude value convergence. (f) Test 3: Frequency ($1/\lambda$) value convergence.

in a predetermined and widely distributed intervals of time. For instance, to facilitate problems caused by insomnia, melatonine is taken after in defined periods of time. To treat this and similar conditions, systems with predetermined oscillation period are ought to be built. By knowing the behaviour of parts and using proper modelling and prediction algorithms, the development is significantly faster.

5 Acknowledgements

The research was supported by the scientific-research programme Ubiquitous Computing (P2-0359) financed by Slovenian Research Agency in years from 2009 to 2012. Results presented here are in scope of PhD thesis that is being prepared by Martin Strazar.

References

1. Thomas F Knight. Cellular gate technology. *Unconventional Models of Computation*, 1997.

2. Ron Weiss and GE Homsy. Toward in vivo digital circuits. *Evolution as Computation*, 2003.

3. Beat P Kramer, Cornelius Fischer, and Martin Fussenegger. BioLogic gates enable logical transcription control in mammalian cells. *Biotechnology and bioengineering*, 87(4):478–84, August 2004.

4. Wilfried Weber and Martin Fussenegger. Engineering of synthetic mammalian gene networks. *Chemistry & biology*, 16(3):287–97, March 2009.

5. Marcel Tigges, Nicolas Dénervaud, David Greber, Joerg Stelling, and Martin Fussenegger. A synthetic low-frequency mammalian oscillator. *Nucleic acids research*, 38(8):2702–11, May 2010.

6. Jesse Stricker, Scott Cookson, Matthew R Bennett, William H Mather, Lev S Tsimring, and Jeff Hasty. A fast, robust and tunable synthetic gene oscillator. *Nature*, 456(7221):516–9, November 2008.

7. Chris P Barnes, Daniel Silk, Xia Sheng, and Michael P H Stumpf. Bayesian design of synthetic biological systems. *Proceedings of the National Academy of Sciences of the United States of America*, 108(37):15190–15195, September 2011.

8. Gabriele Lillacci and Mustafa Khammash. Parameter estimation and model selection in computational biology. *PLoS computational biology*, 6(3), March 2010.

9. Maksat Ashyraliyev, Johannes Jaeger, and Joke G Blom. Parameter estimation and determinability analysis applied to Drosophila gap gene circuits. *BMC systems biology*, 2:83, January 2008.

10. Denis Dochain. State and parameter estimation in chemical and biochemical processes: a tutorial. *Journal of Process Control*, 13(8):801–818, December 2003.

11. Gabriele Lillacci and Mustafa Khammash. A distribution matching method for parameter estimation and model selection in computational biology. *International Journal of Robust and Nonlinear Control*, 2012.

12. Niels Porksen, Malene Hollingdal, Claus Juhl, and Peter Butler. Pulsatile insulin secretion: detection, regulation, and role in diabetes. *Diabetes*, pages 30–33, 2002.

13. T Scheper, D Klinkenberg, C Pennartz, and J van Pelt. A mathematical model for the intracellular circadian rhythm generator. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 19(1):40–7, January 1999.

14. Christopher P Fall. *Computational Cell Biology*, volume 20 of *Interdisciplinary Applied Mathematics*. Springer New York, New York, NY, 2004.

15. Sami Halawani. Modelling Biological Networks Using Soft Computing Technique. *International Journal of Research*, 3(1):1436–1443, 2012.

16. KF Man and KS Tang. *Genetic algorithms: concepts and designs*. Springer New York, 1999.

17. U Alon. *An introduction to systems biology: design principles of biological circuits*. Chapman & Hall/CRC, 2007.