

BioBrick Type Modelling

Karolis Kidykas, iGEM2007, Glasgow

September 7, 2007

Contents

1	Abstract	1
2	The Problem	2
3	In the Making	4
4	BioBrick Type Modelling	4
5	Nonlinear to Linear approximation	5
6	Recommendations	7

1 Abstract

By default, there are two common approaches in modeling the Biological Systems, Deterministic and Stochastic. While I am going to re-frame myself from Stochastic method, Deterministic methods are based on ordinary differential equations (ODE's). ODE describes the rate of change and, obviously, we need some rate constants to model such systems. If we could 'find' rate constants for our model, the problem would be solved ¹. The bigger the model, the more constants we need to find and more errors it introduces. But modellers needs to provide answers for laboratory before the actual experiment is going to take place. And the answer for the following question is not always available. Which topological path will lead to better output? BioBrick concept seems to hold these answers; even if it is only in evolutionary stage for a dry lab? [1]

¹Assuming that the ODE model itself is accurate

2 The Problem

For iGEM07 competition Glasgow team decided to present a pollutant detecting - self powering biosensor. BioBricks[2] were prepared to be put into their respective E.coil vectors and several possible pathways were proposed. Figure 1. Model a. represents the initial design stage; however it was uncertain if additions of feedback loop b, will force the system to behave differently, as well as constitutive expression of phzM in c. d. and e. cases.

As a first iteration, the following ODE models were brainstormed by the team. ²

Standard Construction a.

$$T\dot{F}S = \beta_{TFS} \frac{s}{\gamma_{TFS} + s} - k_d TFS \quad (1)$$

$$\dot{s} = -\delta_s s - \beta_{TFS} \frac{s}{\gamma_{TFS} + s} + k_d TFS \quad (2)$$

$$Phz\dot{M} = \beta_{PhzM} \frac{TFS}{\gamma_{PhzM} + TFS} - \delta_{PhzM} PhzM \quad (3)$$

$$Phz\dot{S} = \beta_{PhzS} \frac{TFS}{\gamma_{PhzS} + TFS} - \delta_{PhzS} PhzS \quad (4)$$

$$\begin{aligned} MPCAB\dot{} = & \beta_{MPCAB} \frac{PhzM}{\gamma_{MPCAB} + PhzM} - \delta_{MPCAB} MPCAB - \\ & - \beta_{PYO} \frac{MPCAB}{(\gamma_{PYO_{MPCAB}} + MPACB)} \frac{PhzS}{(\gamma_{PYO_{PhzS}} + PhzS)} \end{aligned} \quad (5)$$

$$\begin{aligned} PYO\dot{} = & \beta_{PYO} \frac{MPCAB}{(\gamma_{PYO_{MPCAB}} + MPACB)} \frac{PhzS}{(\gamma_{PYO_{PhzS}} + PhzS)} - \\ & - \delta_{PYO} PYO \end{aligned} \quad (6)$$

Feedback Loop b.

²It must be noted that this model merely represents the final model. It was only first attempt to model the system and in no way it is the final release by iGEM07 Glasgow. It was chosen here to high lite the approach to the problem only

$$T\dot{F} = \alpha_{TF} + \beta_{TF} \frac{TFS}{\gamma_{TF} + TFS} - \delta_{TF}TF \quad (7)$$

$$T\dot{F}S = \beta_{TFS} \frac{s}{(\gamma_{TFS_s} + s)} \frac{TF}{(\gamma_{TFS_{TF}} + TF)} - k_d TFS \quad (8)$$

$$\dot{s} = -\delta_s s - \beta_{TFS} \frac{s}{(\gamma_{TFS_s} + s)} \frac{TF}{(\gamma_{TFS_{TF}} + TF)} + k_d TFS \quad (9)$$

$$Phz\dot{M} = \beta_{PhzM} \frac{TFS}{\gamma_{PhzM} + TFS} - \delta_{PhzM} PhzM \quad (10)$$

$$Phz\dot{S} = \beta_{PhzS} \frac{TFS}{\gamma_{PhzS} + TFS} - \delta_{PhzS} PhzS \quad (11)$$

$$\begin{aligned} MPCAB = & \beta_{MPCAB} \frac{PhzM}{\gamma_{MPCAB} + PhzM} - \delta_{MPCAB} MPCAB - \\ & - \beta_{PYO} \frac{MPCAB}{(\gamma_{PYO_{MPCAB}} + MPACB)} \frac{PhzS}{(\gamma_{PYO_{PhzS}} + PhzS)} \end{aligned} \quad (12)$$

$$\begin{aligned} PYO = & \beta_{PYO} \frac{MPCAB}{(\gamma_{PYO_{MPCAB}} + MPACB)} \frac{PhzS}{(\gamma_{PYO_{PhzS}} + PhzS)} - \\ & - \delta_{PYO} PYO \end{aligned} \quad (13)$$

First iteration was made while research to find rate constants on sources available was in early stage. And it was not clear if model represents the real system accurately. One could say, that some unknown constants could be found using optimisation after experiment is carried out and behavior of the system is known, but that contradicts our goal to advise laboratory before experiment. On top of it, I would also like to note, that optimisation is reliable only up to 5 unknown rate constants and becomes real tedious and unreliable with every additional one. For our experiment we were using Matlab's [3] built in optimisation protocol 'fmincon' (used to solve constrained minimisation problems) for least squares objective function, and it proved, that the smaller error after optimisation will not necessary guarantee more accurate answer³. It is clear that one can't expect to estimate systems quantitative behavior without the experiment being carried out first. But maybe we can judge systems qualitative behavior?

³We gained this knowledge using various simulation techniques. One team assumed some *rate constants* while others were using every mathematical approach available to guess them. It became clear that, for more than 5 constants, lowest optimization error have not necessarily given the most accurate answers

3 In the Making

If one compares models (a) and (b). It can be seen, that they share most of rate constants and equations 3,4,5 and 6 are the same as 10,11,12 and 13 respectively. Since we don't know real rate constants for model (a) we can make them up, say, equate all to 1 (effectively assuming that we use 100% value of actual rate constant). However model (b) has 5 extra rate constants which relation to model (a) are unknown and two models cannot be compared qualitatively. Upon closer inspection the reader can see, that if β_{TF} is set to zero. It would have the same effect as cancellation of feedback loop. And if system (a) should be plotted, time against output of PYO, (using assumed rate constants equal to 1) system (b) should demonstrate exactly the same plot thus opening the option for least square optimisation for remaining 4 unknowns. Matlab's built in function `fmincon` handles four parameters pretty easy and gives accurate results $\alpha_{TF} = 1.55$, $\delta_{TF} = 0.08$, $\gamma_{TFS_s} = 0.99$, $\gamma_{TFS_{TF}} = 0.08$. Having these results we can reset β_{TF} to 1 (again assuming that we use 100% value of actual rate constant) and compare two Pathways qualitatively. The comparison of the outcome showed that there is no difference to the output of PYO if the feedback loop for `dntR` is present. To be sure that there are no mistakes we used Multi Parametric Sensitivity Analysis (MPSA) [4] compiled by Glasgow team using Matlab. Conclusion was the same; the addition of feedback loop will not influence the system's output of PYO⁴. Thus we were able to compare, two dynamical systems qualitatively, but what about the remaining systems (c), (d) and (e). The above statements forms basis for BioBrick Type Modelling.

4 BioBrick Type Modelling

The more one knows about biological system, the more accurate model one can implement. The most accurate way of modelling would be to model every single process in cell. Since it is impossible to do so with our current technological achievement (See Recommendation.1) we should use the most detailed model known to date, which is to model every BioBrick at lowest level (Promoter, RBS, mRNA, Coding Region and etc.). The up to date brick mathematical model can be found in Openwetware [5].

However, the equations found on Openwetware contains at least four times more rate constants and as mentioned in Recommendation 1 those constants are not standard, that laboratory could provide us with. Model

⁴Method described above, with conjunction of MPSA software, proved that model was wrong and new iterations followed as a result

itself, becomes too tedious to analyze. So why bother using this type of modelling? If you look at the Figure.1 more closely, you will note, that all five models contain the same bricks. No new bricks are being added or deleted. It follows, that if we move a particular brick to a different location, but in the same vector, we will not be changing its properties, thus we keep the same rate constant. Effectively we can assume that we know all the rate constants (say, equal to 1 or 100% of its real value) and compare models qualitatively and keep our model in data base for future quantitative analysis (Recommendation 1). But before we can proceed, we need to eliminate all nonlinearities in our bricks. Linear models have reliable, robust relationships between corresponding equations, so we can copy and paste them anywhere in topological pathway without losing quality of model itself. Nonlinear systems have complex relationships and get more and more complicated with every extra brick one adds. Figure 3 (a) has one nonlinear element remaining in the model. In comparison, (b) all nonlinear elements are replaced by linear ones.

This type of modelling is not confined to models, made using same bricks, only. We can search for similarities (see In The Making chapter) between two models that share some of the same bricks as well as have new different ones. We have to keep in mind that the more new bricks we add, the more unreliable optimisation becomes. So even though this method is limited, it can be used as the first input to Laboratory. This method is only in evolutionary stage, but could become very powerful tool if combined with Multi Parametric Sensitivity Analysis and advances in constrained optimisation.

5 Nonlinear to Linear approximation

As mentioned above, we need to eliminate nonlinear elements behind Bio-Brick, so we could cut copy and paste them in any topological pathway without losing quality of the system.

Some of the first order differential equations could be linearised using frequency (Laplace) domain. That includes rate of change of mRNA, protein translation and mass action equations:

$$\begin{array}{ccc} \text{Time Domain(Non-linear)} & \rightarrow & \text{Frequency Domain(Linear)} \\ \frac{d[mRNA]}{dt} = PoPs - K_{deg}[mRNA] & \rightarrow & \frac{mRNA[out]}{PoPs[in]} = \frac{1}{s + K_{deg}} \end{array} \quad (14)$$

$$\begin{array}{ccc} \frac{d[Protein]}{dt} = SynthesisRate - K_{deg}[Protein] & & (15) \\ & \rightarrow & \frac{Protein[out]}{SynthesisRate[in]} = \frac{1}{s + K_{deg}} \end{array}$$

Mass action and activated promoter requires additional treatment, since their equations can not be expressed in the same manner as in previous example . Comprehensive description of frequency domain can be found in many dynamical systems texts, but the general rule of transformation is:

$$\dot{x}(t) \rightarrow sX(s) \quad (16)$$

$$x(t) \rightarrow X(s) \quad (17)$$

Activated promoter has following time domain equation:

$$PoPS = \frac{maxK_r[activator]}{K_d + [activator]} \quad (18)$$

This equation cannot be translated into frequency domain so linear approximation is needed. Activated promoter uses Michaelis-Menten dynamics, so it will reach saturation and curve, before it happens, can be approximated linearly Figure 4

For our project we were using Simulink . Simulink is a Matlab's package dedicated for modeling and simulating dynamical systems. It has a built in saturation dynamics models making it straight forward to approximate response of activated promoter [6]. Mass action could be approximated using the same principle although the output of product could be translated into frequency domain:

$$[A] + [B] \rightarrow [C] \quad (19)$$

$$\begin{array}{ccc} \text{Time Domain(Non-linear)} & \rightarrow & \text{Frequency Domain(Linear)} \\ \frac{d[C]}{d(t)} = K_a[A][B] - K_d[C] & \rightarrow & \frac{[C][out]}{[A][B][in]} = \frac{K_a}{s + K_d} \end{array} \quad (20)$$

We cannot do the same for substrates [A] and [B] so linear approximation as above is needed, or one could use Simulink to create dynamical bricks to avoid that step. For more information on how to use Simulink for BioBrick type modeling you can refer to ElectrEcoBlu Simulink Manual [7]

6 Recommendations

1. It is no doubt, that concept of BioBrick, is in early stages of development, but experimental are focusing their resources to develop an end product with very little input from computational informatics:

We can not guaranty that the final outcome is going to be successful. there are too many unknown factors involved in this system.

Our current technology limits us to the Mass action, Michaelis-Menten rate constants in the best case scenario. Even if rate constant can be extrapolated from experiment the other lab could barely use them, because too many assumptions were made during this process. To make things more complicated, building ODE model is not so straight forward. iGEM07 Glasgow team went through at least five iterations untile ODE model was frozen. The best comment about the current situation I believe was given by iGEM Edinburgh adviser:

We are trying to build stone Castles, but know little about stone's properties.

We are not paying enough attention to fundamental research, while concentrating mostly on final product. PoPs, SynthesisRate and et., seems to be a future rate constants for Modellers, replacing existing V_{max} , transcription and translation rates. These are the key focus points for future research if Modeller are to give Quantitative input to lab's people to judge the outcome of upcoming experiments.

References

- [1] Harvard MIT and UCSF joint project. <http://www.biobricks.org/>.
- [2] Glasgow iGEM 2007. <http://parts.mit.edu/igem07/index.php/Glasgow>.
- [3] 2007. www.mathworks.com.

- [4] T.Friend, M.Trybilo, and Xu Gu. Multi parametric sensitivity analysis program. *iGEM Glasgow*, 2007.
- [5] 2007. http://openwetware.org/wiki/Registry_of_Standard_Biological_ModelsGeneral_Architecture_Discussion.
- [6] 2007. <http://www.mathworks.com/products/simulink/>.
- [7] K.Kidykas. ElectrEcoBlu Simulink Manual iGEM 2007 Glasgow.

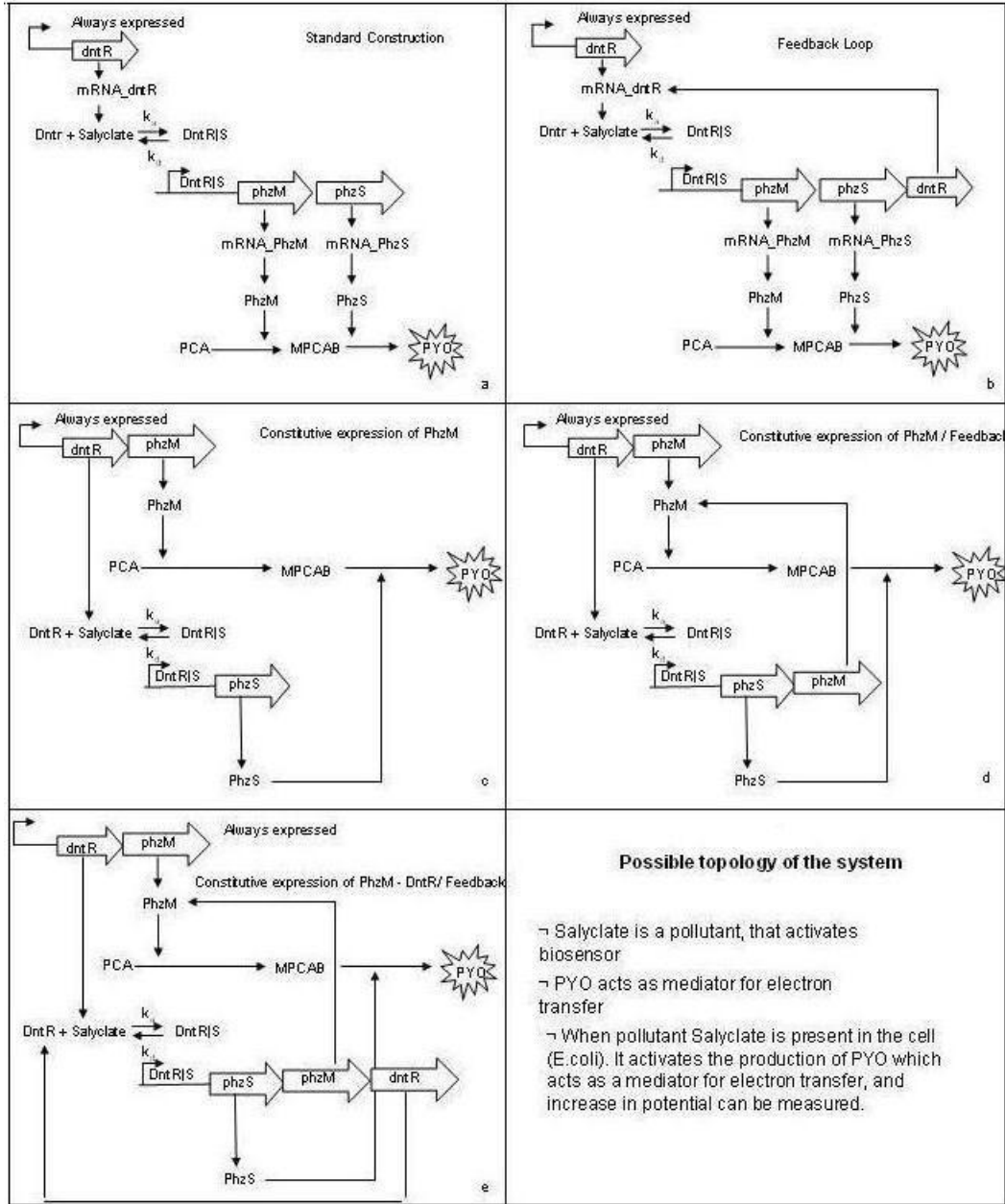


Figure 1: Project ElectrEcoBlu [2]

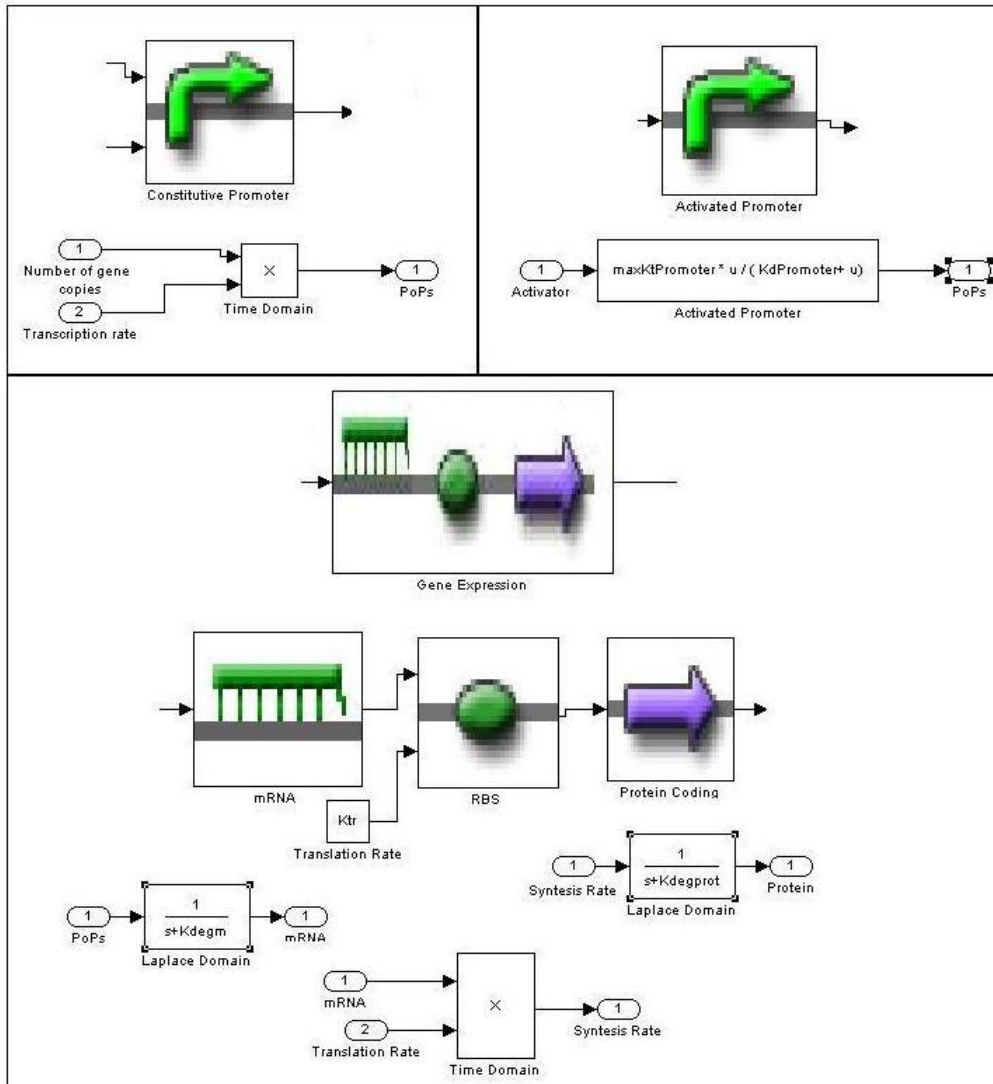


Figure 2: Gene expression modelling

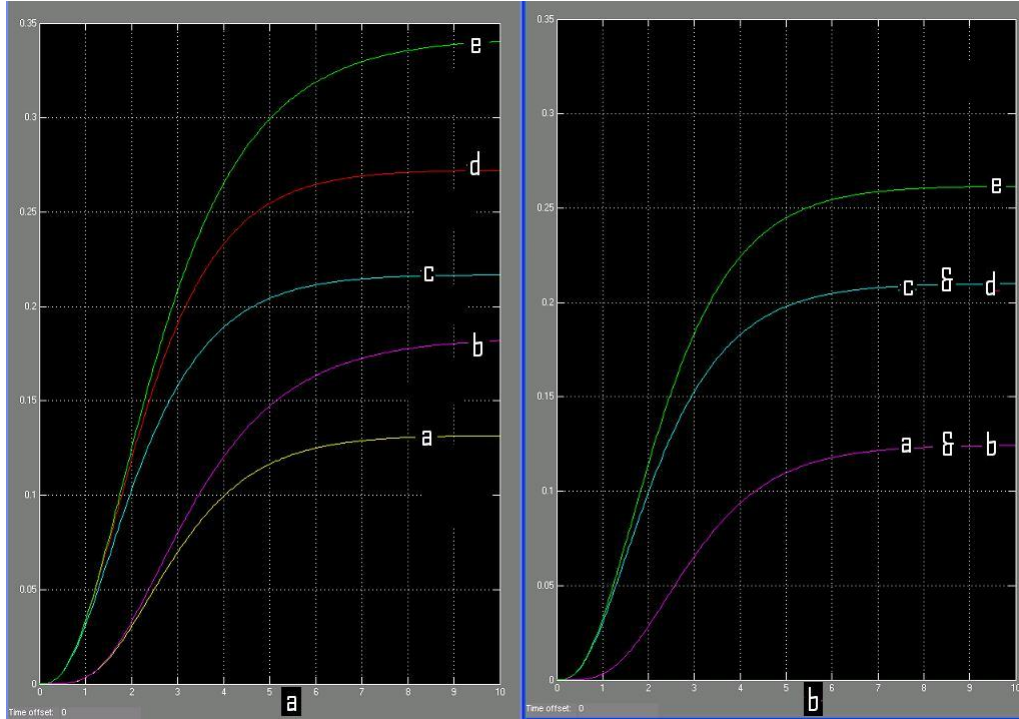


Figure 3: Error due to nonlinearity: a) With one nonlinear element remaining, b) Linear Modelling

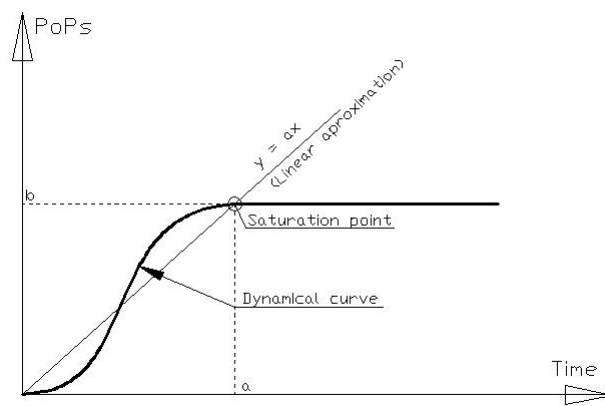


Figure 4: Linear approximation for activated promoter