ElectrEcoBlu Simulink Manual

Karolis Kidykas, iGEM2007, Glasgow

September 6, 2007

Contents

1 Abstract

It is assumed here, that user has basic skills using matlab and simulink[1]. The BioBrick type modeling's comprehensive algorithm can be found in iGEM07 Glasgow documentation [2] and up to date equations from open wet ware[3]. The general idea is not to simplify models, in order to use Michaelis-Menten or Mass action rate constants, but to have as detailed

model as current technical base allows us. Thus, we can introduce brick concept [4] to our modeling.

2 BioBrick equations

To model any biological system we need to use following time domain equations

$$ConstitutivePromoter$$
$$PoPs_{out} = K_t \quad (1)$$

$$ActivatedPromoter$$
$$PoPs_{out} = \frac{maxK_r[activator]}{K_d + [activator]} \quad (2)$$

$$RepressedPromoter$$
$$PoPs_{out} = \frac{maxK_t}{K_d + [repressor]} \quad (3)$$

$$mRNA$$
$$\frac{d[mRNA]}{dt} = PoPs_{in} - K_{deg}[mRNA] \quad (4)$$

$$RBS$$
$$SynthesisRate = K_{tr}[mRNA] \quad (5)$$

$$Protein$$
$$\frac{d[protein]}{dt} = SynthesisRate - K_{deg}[protein] \quad (6)$$

$$MassAction$$
$$\frac{d[C]}{dt} = k_a[A][B] - k_d[C] \quad (7)$$

Only equations 1,5 are linear. To use BioBrickType modeling, all of them must be linerised. The most convenient way of doing it, is to use frequency domain[1]:

[1]Input variables [in] must be updated dynamically for each time step

$$\underset{mRNA}{\frac{mRNA[out]}{PoPs[in]}} = \frac{1}{s + K_{deg}} \tag{8}$$

$$\underset{Protein}{\frac{Protein[out]}{SynthesisRate[in]}} = \frac{1}{s + K_{deg}} \tag{9}$$

$$\underset{Mass\ Action}{\frac{C][out]}{[A][B][in]}} = \frac{K_a}{s + K_d} \tag{10}$$

Activated and repressed promoters cannot be expressed in frequency domain, thus piece wise linearisation is used. Figure 1 shows possible linearisation of activated promoter:

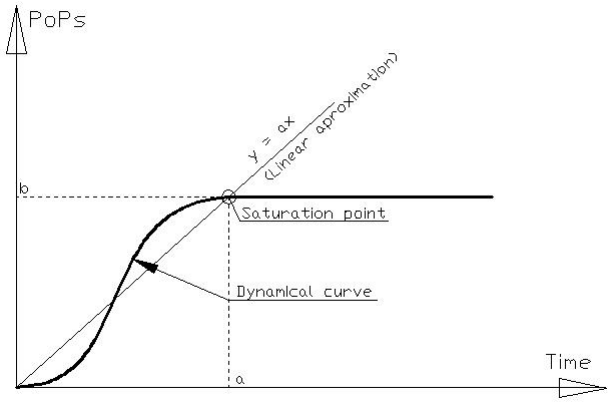

Figure 1: Activated promoter linearisation

$$PoPs(t) = \begin{cases} grad[Time], & 0 < x < a \\ b, & x > a \end{cases}$$

Where $grad = \frac{b}{a}$

Figure (incoming) shows possible linearization of repressed promoter (incoming)[2]

[2]Not attempted here because of a short time frame

3 Building the system

1. Create folder where your model will be. Copy folder **icons** and file *RateConstantDataBase.m*.
2. Run *RateConstantDataBase.m* to load your variables into workspace.
3. Start a new model.
4. Open Library *BioBricks.mdl*
5. Copy and paste required bricks into your model.
6. Change variable names in your model and update *RateConstantDataBase.m* file accordingly.
7. Run *RateConstantDataBase.m* to reload variables into workspace.
8. Simulate your model using desired parameters.
9. Add scopes, graphs, buss lines etc. to observe outcome.

4 BioBrick Library

BioBrick templates can be found in *BioBricks.mdl* file Figure 2

4.1 Promoters

Double clicking on *Promoters* will pop up promoters library [3]. Figure 3.

Double clicking on any type of promoter will bring it's mathematical equation. Figure 4.

You can use Nonlinear Activated promoter for linearisation simulation (See section Simulating the system) or use your own a and b parameters. Figure 5.

4.2 Gene Expression

You can use *Protein Expression* brick as a single entity or double click on *Protein Expression* for the library of mRNA, RBS (ribosome binding site) and protein translation bricks. Figure 6

Each element has its own mathematical equation. You can view it by double clicking it. Figure 7.

[3] In version V1 represor promoter is not included. Expected in V2.

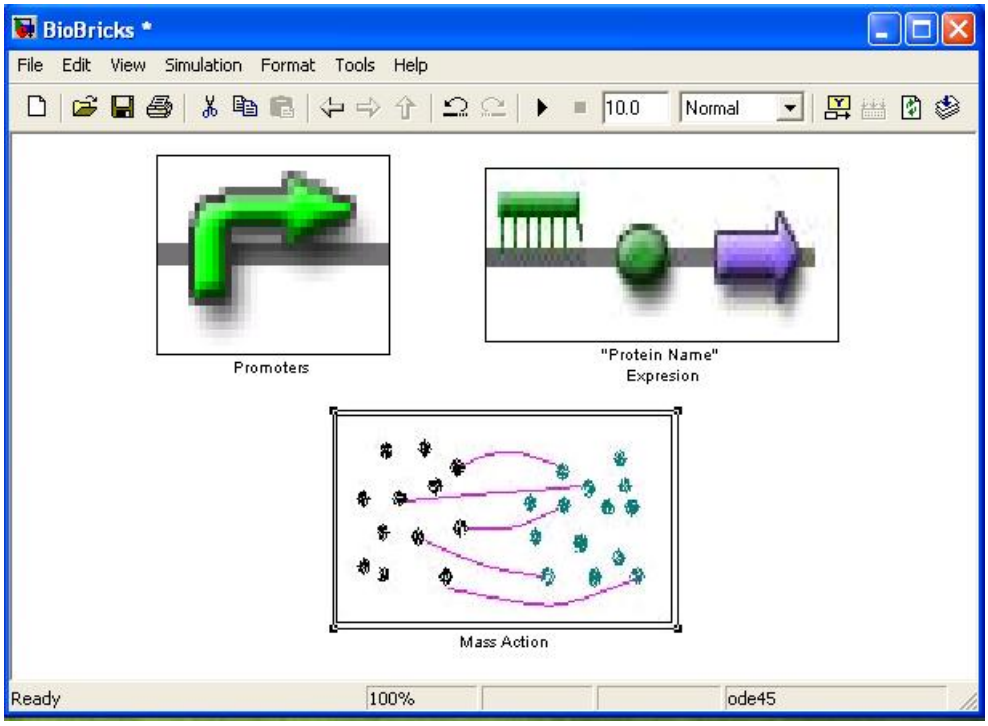

Figure 2: BioBricks.mdl

4.3 Mass Action

Double clicking on *Mass Action* will bring the mass action bricks library. Figure 8

You need to judge carefully which brick to use. Even tough all bricks are for to substrates going to one product, the input type of those substrates could differ.

4.3.1 Two Static Inputs

Use this brick if all substrates has some initial concentrations, but not being updated by a dynamical input from previous action. Figure 9.

4.3.2 Static and Dynamic Inputs

Use this brick if one substrate is being updated dynamically from previous action and the other is static. All substrates has some initial concentrations, although dynamic substrate's initial concentration is usually zero. Figure10.

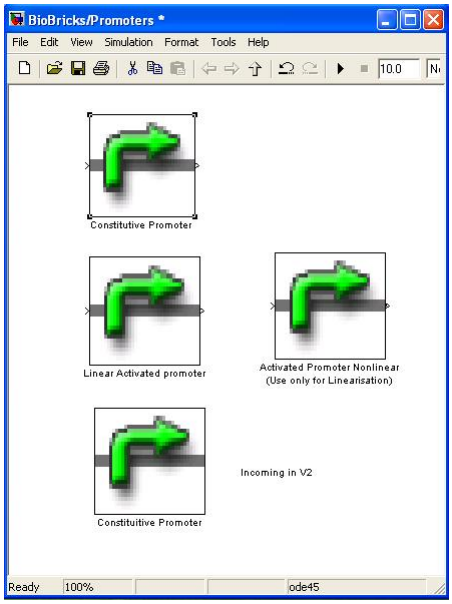

Figure 3: Library of promoters

4.3.3 Dynamic and Dynamic Inputs

Use this brick then all substrates are being updated dynamically from previous action. You have to give initial static concentrations, although in mosts cases it is zero. Figure 11.

5 Example System

5.1 Constitutively expressed protein

Follow steps described in section Building the system. Copy *constitutive promoter* and *Protein expression bricks*. Figure 12. Add scope from *Simulink Library Browser/Commonly used blocks*. Connect arrows as shown in Figure 12. Select simulation stop time and press Run. Double click on scope to observe results.

5.2 Signal activated promoter

We want to model CMS expression Figure 13. CMS production can only be activated by signal S. Protein F is always expressed and then it binds to S, CMS is being expressed . In order to proceed with CMS simulation we need

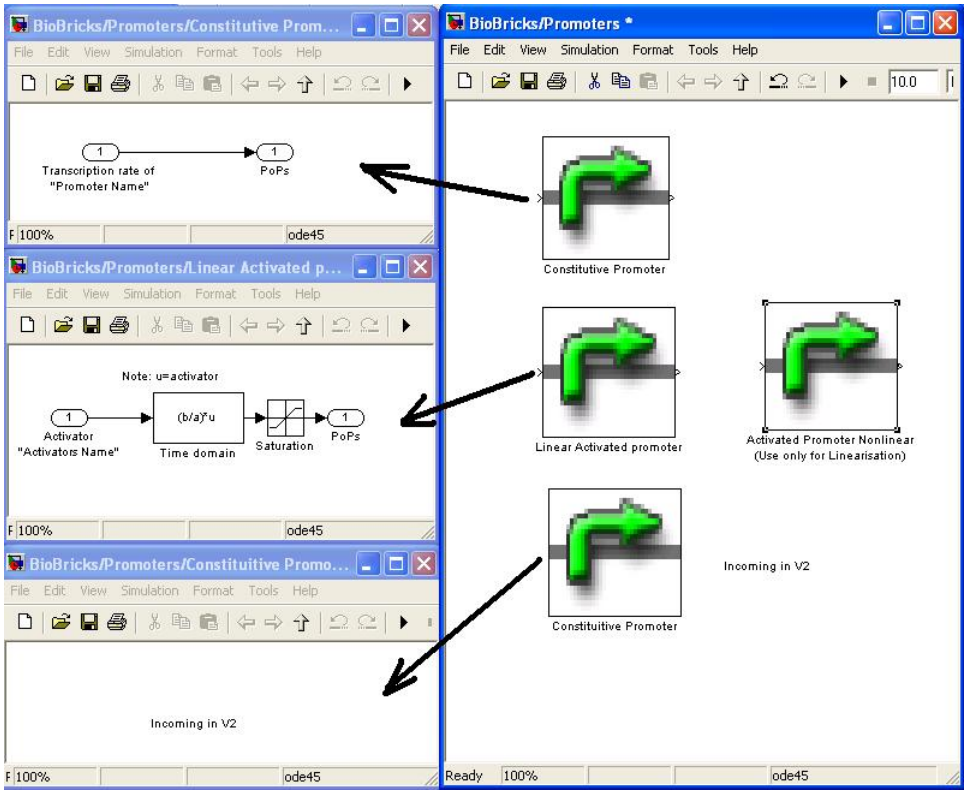

Figure 4: Promoters and their mathematical models

to find F's steady state value since it is always expressed and find constants a and b for activated promoter (see Figure 1)

5.2.1 F's steady state

Example system, described in Signal activated promoter section, effectively simulates F expression. So when you create a new model, to simulate F, copy and paste previous elements but add Display from *sinks* library to observe exact saturation value. If default run time was not enough to reach saturation point, increase it and run the simulation again, until steady state occurs. When fished with this step, add the rest of pathway's components, Figure 14, except, components highlighted for second simulation[4]. If you decide not to skip second step you can find *linearization logics* block in file *extras.mdl*

[4]Second simulation is optional if you use your own a and b parameters

5.2.2 Activated Promoter's linearisation

If you decide to use your own linearisation method skip this step. To linearise the response of activated promoter, add extra elements to your model ass described in F's steady state section Figure 14. **Important!!!** Double click *Linearisation Logics* block to connect stop sign as shown in Figure 15. Run simulation for second time. With this block active simulation will stop when saturation is reached and you can record a and b values for third simulation. **Delete line connecting stop sign afterward!**

5.2.3 CMS expression

When all values are ready run the third simulation to observe the output of CMS.

References

[1] 2007. http://www.mathworks.com/products/simulink/.

[2] Karolis Kidykas. Biobrick type modeling. *iGEM Glasgow*, 2007.

[3] 2007. http://openwetware.org/wiki/Registry_of_Standard _Biological_ModelsGeneral_Architecture_Discussion.

[4] Harvard MIT and UCSF joint project. http://www.biobricks.org/.

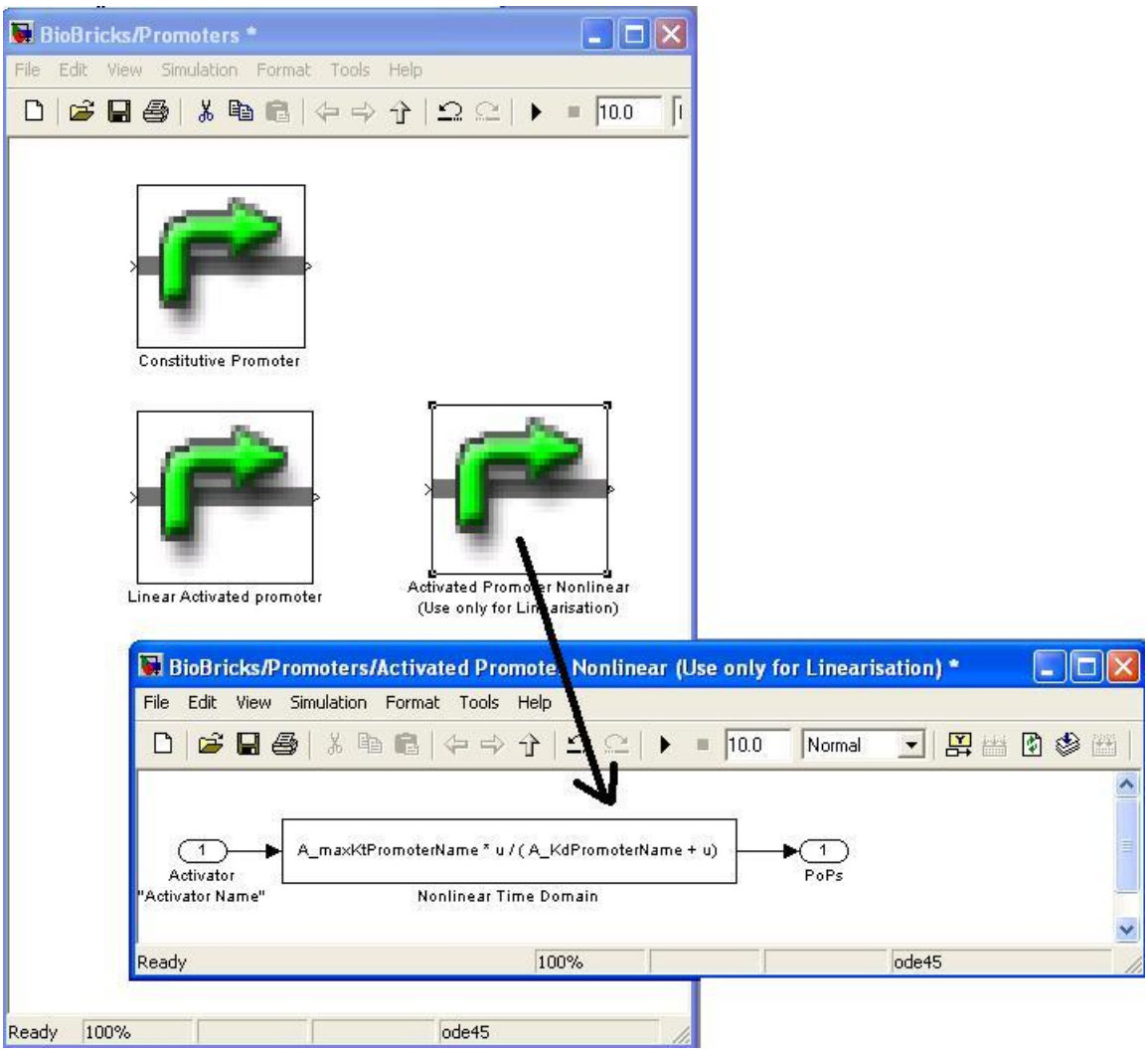

Figure 5: Nonlinerised Activated Promoter

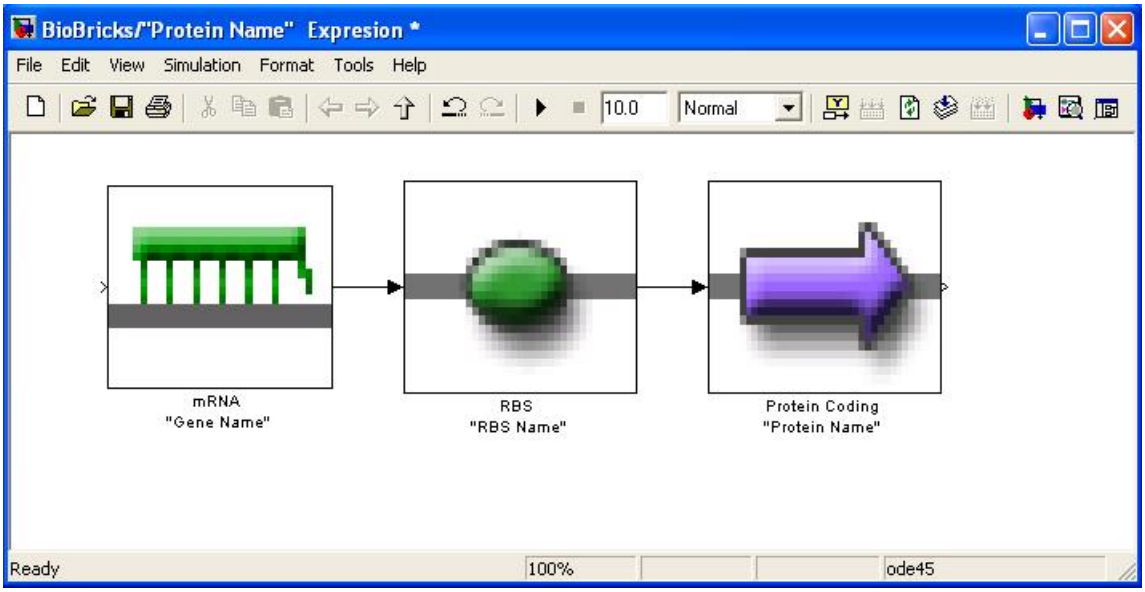

Figure 6: Protein Expression Library

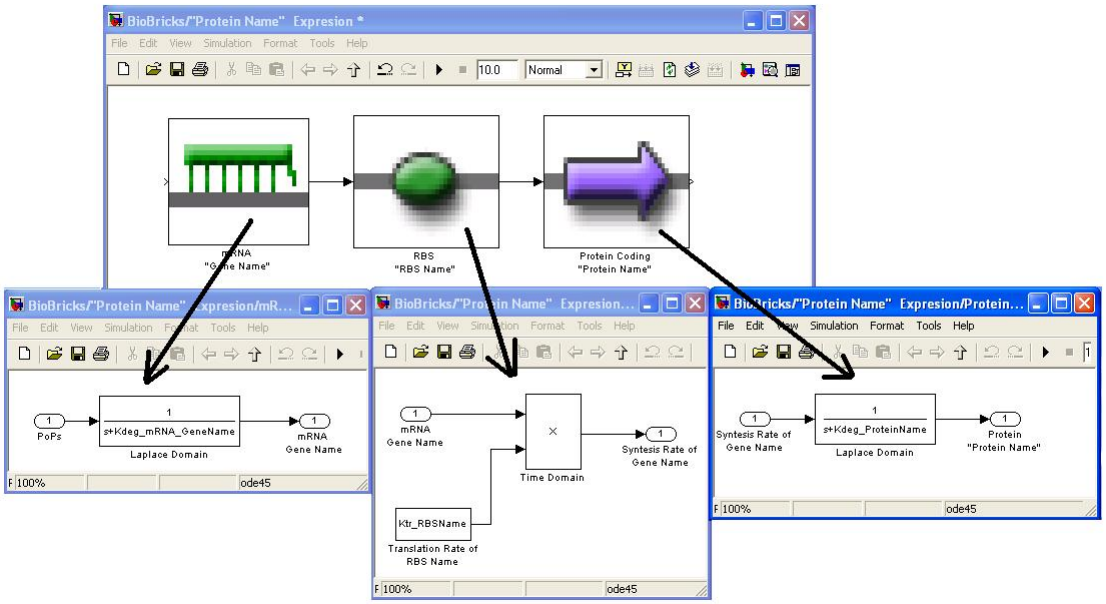

Figure 7: Mathematics behind protein expression

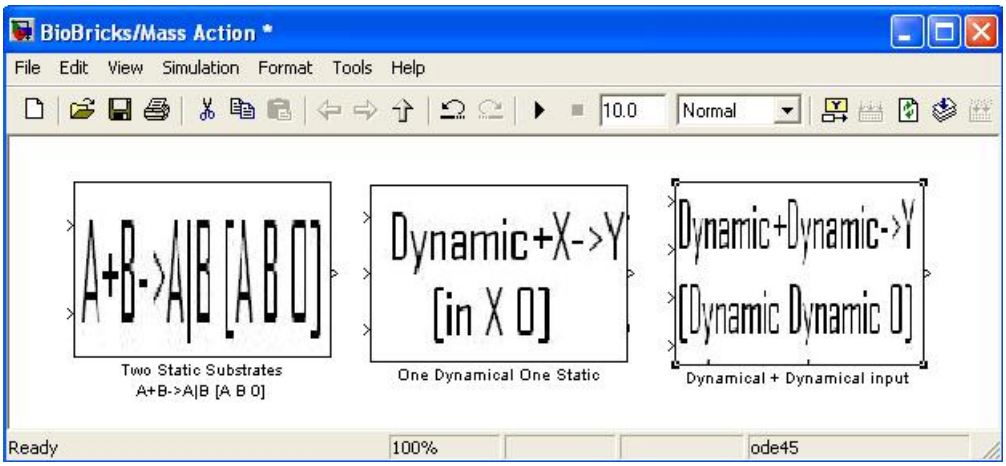

Figure 8: Mass Action

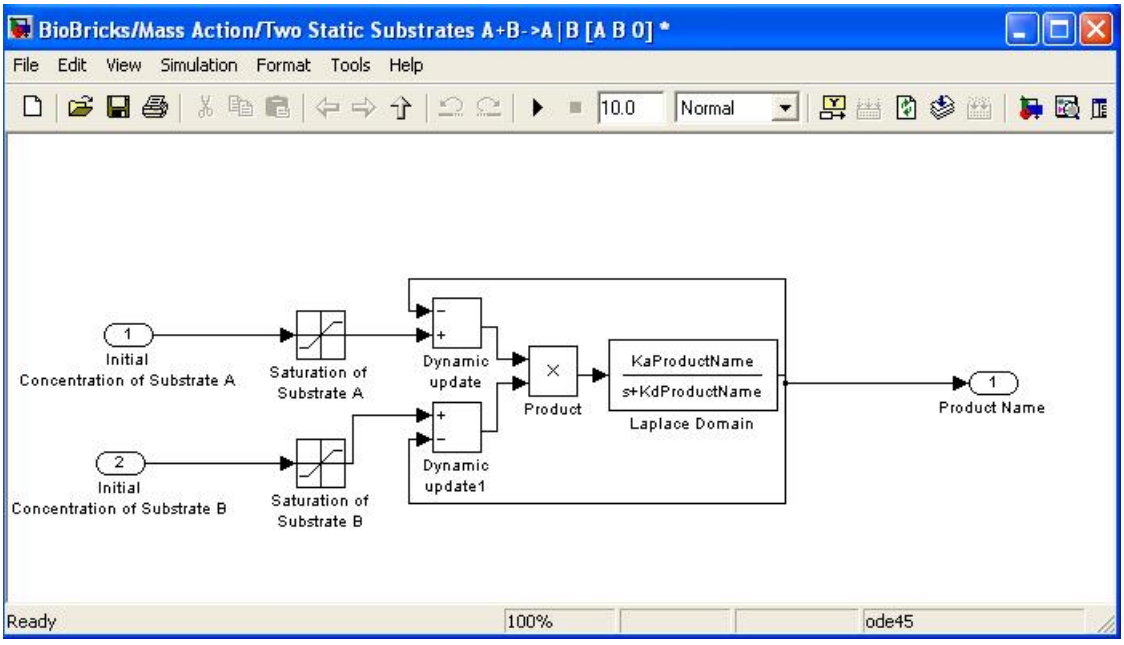

Figure 9: Static Static mass action brick

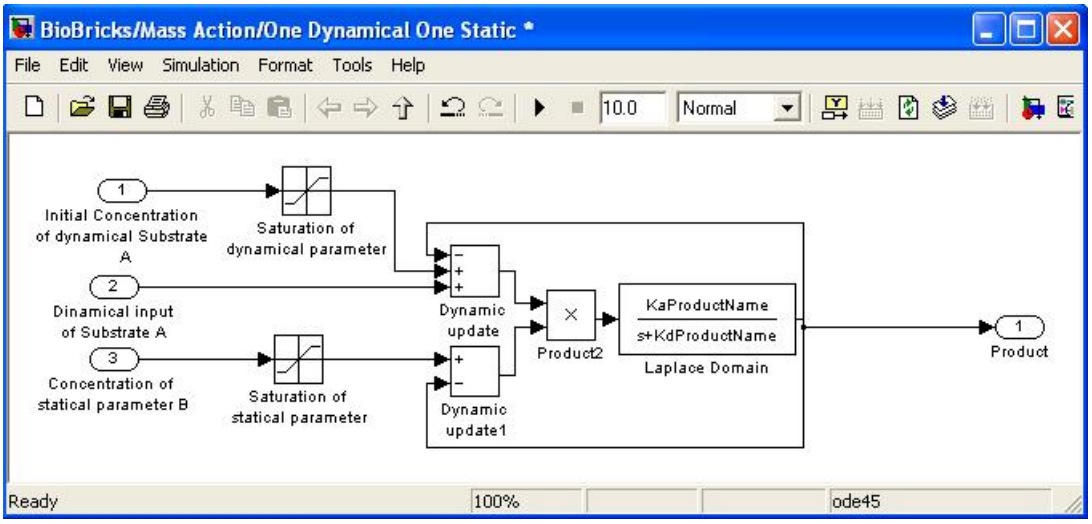

Figure 10: Dynamic Static mass action brick

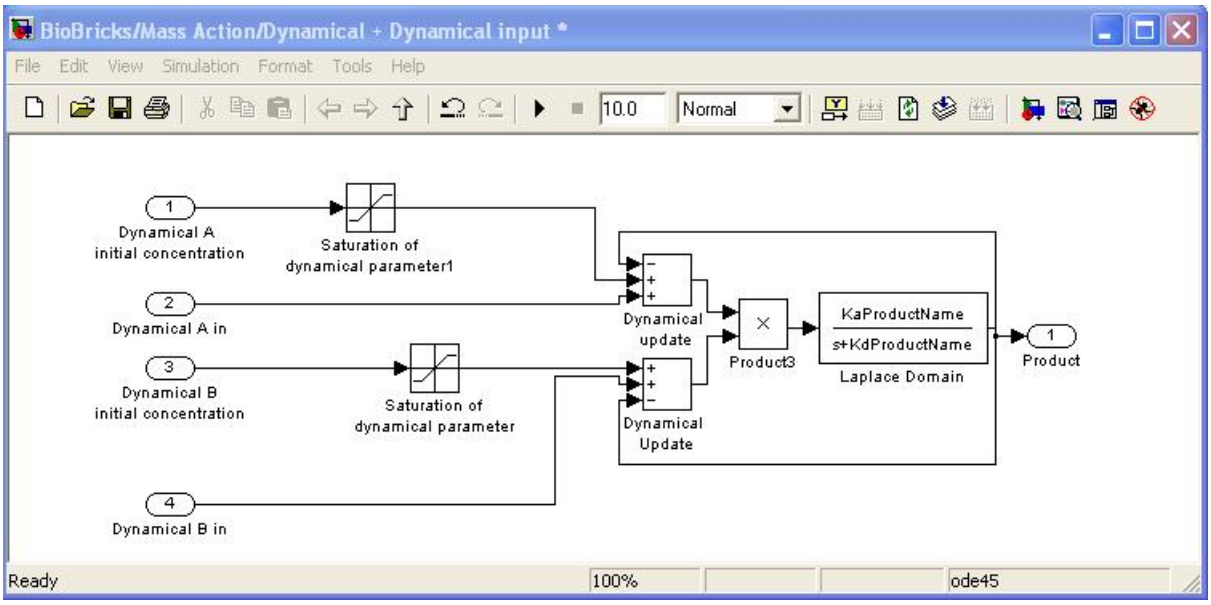

Figure 11: Dynamic Dynamic mass action brick

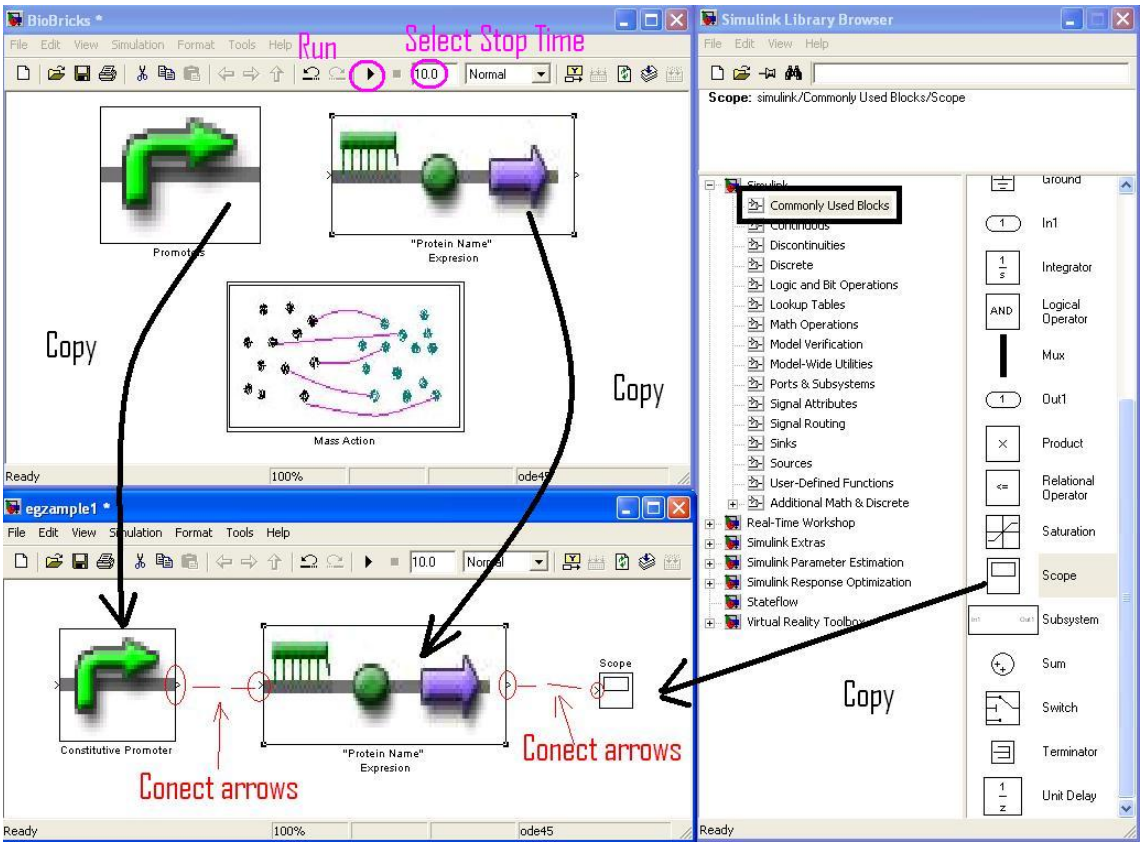

Figure 12: Creating simple system

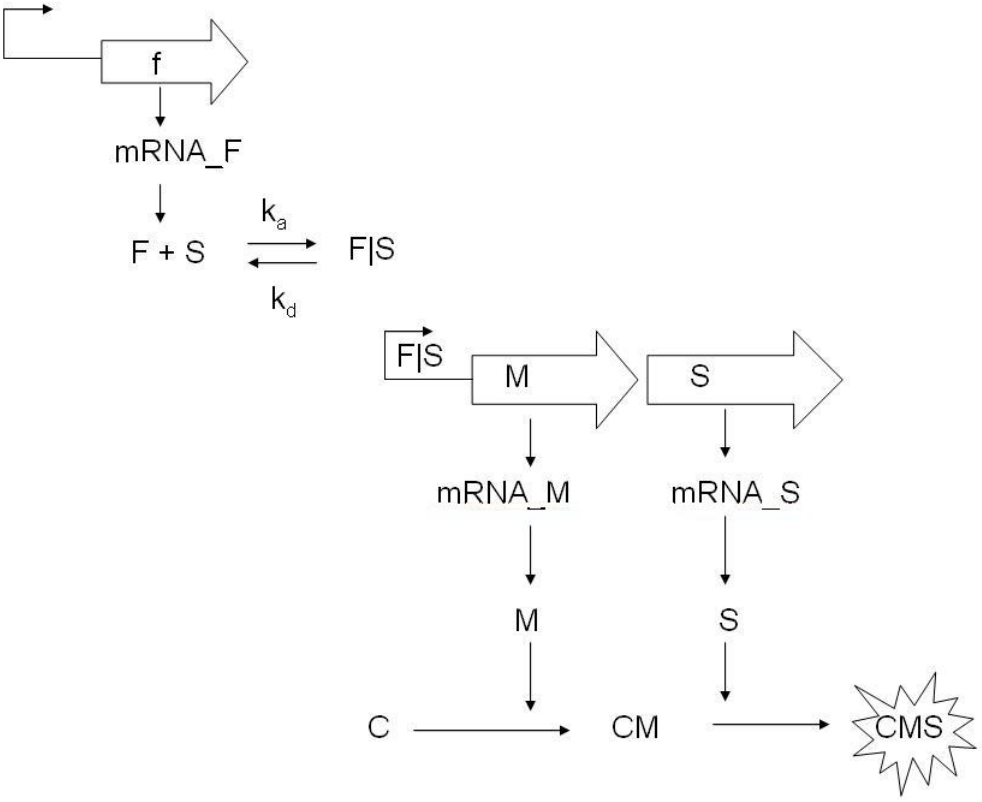

Figure 13: CMS expression topology

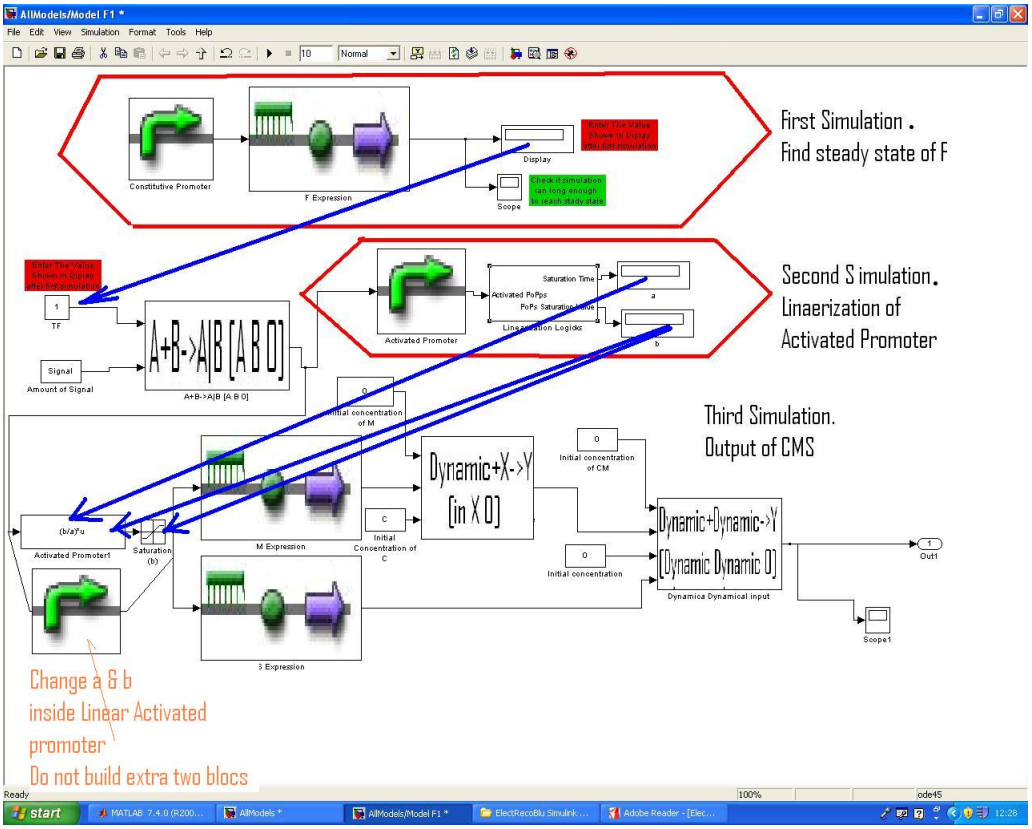

Figure 14: Signal Activated Expression of CMS

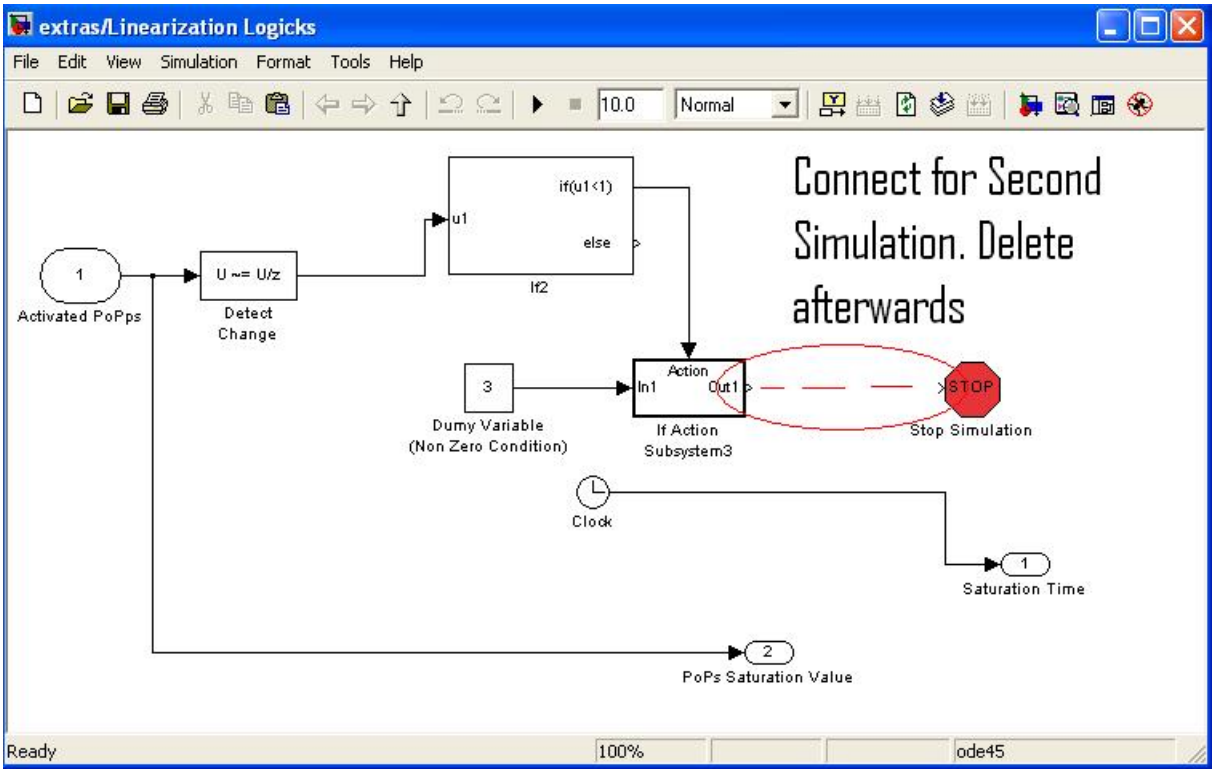

Figure 15: Linearization Logics (optional)