

Design Project 1 – Protein Networks

Individual Project Sections due when Indicated

Whole Project Due: October 18th @ 11:59 am (that's **noon**, not midnight)

General Guidelines and Directions:

- i. This is an individual project; absolutely no collaboration is allowed
- ii. The professors and TA's are here to help you with any design challenges, are available for assistance by appointment, but are not available to debug code.
- iii. There is more than one answer for these projects so be creative and don't get stuck thinking there is only one solution!
- iv. Regardless of if your system works, you will need to answer all parts of the project.
- v. Label axes and identify units where appropriate
- vi. Cite relevant literature where appropriate
- vii. Label all pages of your project with your name and title and be sure to include page numbers
- viii. Submit MATLAB scripts online in the following format:
lastname_firstname_codename.m
For grading purposes, your code should run seamlessly and shouldn't require additional manipulations to complete the simulation of your system. The TA's should be able to hit run and see plots of all the results, with no need for additional input or modifications or anything.
- ix. Submit reports and supplementary material to Course website with the following format:
LastName_FirstInitial_Design1_PartA.pdf
- x. Assignments that don't follow the guidelines won't be graded

Project Write-up Specifications:

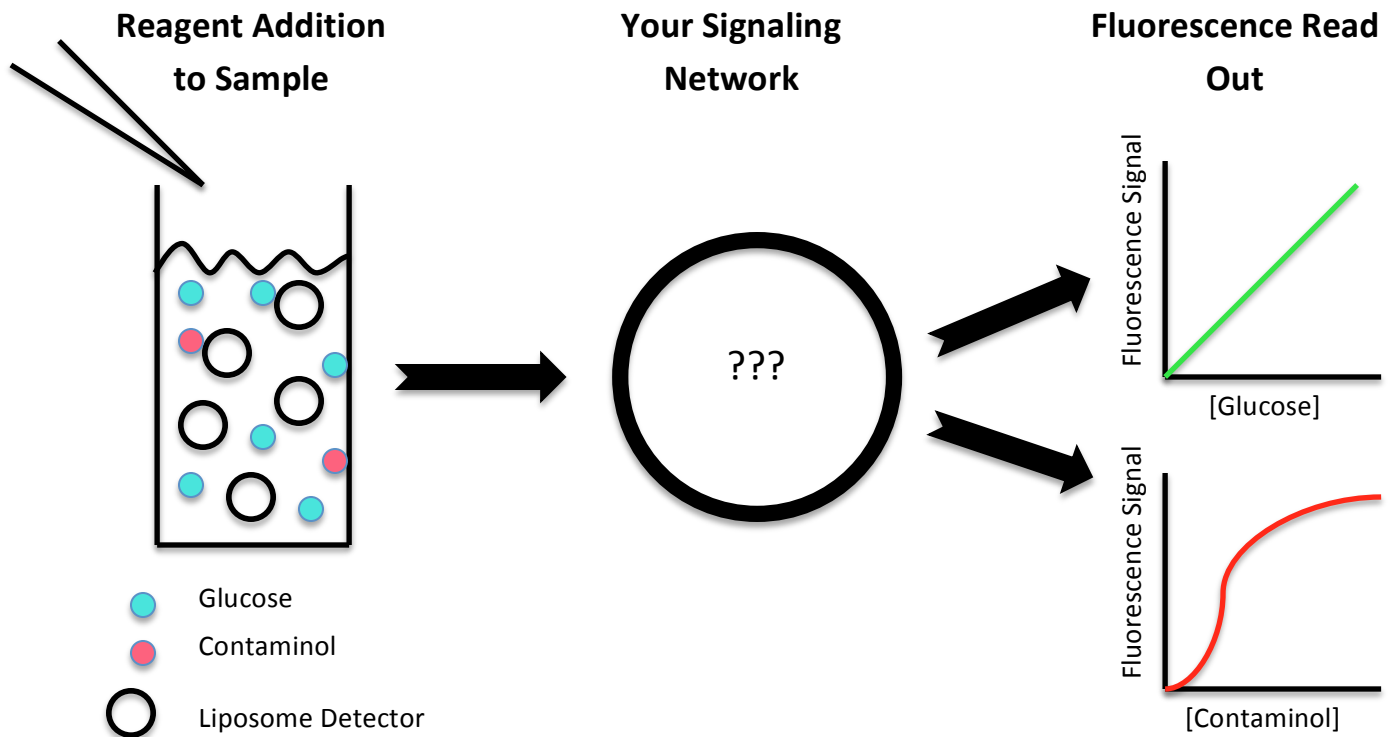
Your project must be submitted in a PDF format generated from a typed document. No components of your project can be scanned from handwritten sources or the project will not be graded (Note: this includes figures). Each portion of your document should be labeled according to the project part number it addresses and the sections should follow the same order as outlined below. When submitting later parts of the project, your submission should contain all previous parts as well. The final report is due on October 18th at 11:59 am.

Introduction and Motivation

Bioreactors are steady-state mixed tanks that keep bacteria in permanent exponential growth while continuously diluting with nutrient-rich broth and removing both cells and reaction products. They are great tools for the biochemical engineering industry, but monitoring them is not trivial. There are many factors that can affect the health of the system's bacteria, and only some of them (temperature, pressure) can be measured physically. Things like nutrient levels and the presence of contaminants must be measured through sampling, and this is the process that you'll try to simplify in this project.

One elegant solution to this sensing and reporting problem is to use the same systems that bacteria would use, but putting them all into a synthetic system that has no agenda of its own. The objective of this project is to develop self-contained liposome signaling systems capable of indicating whether reactor conditions are suitable for bacterial growth. In our simplified bio-reactor model, there are only two substances whose level concerns us: glucose and contaminol. Glucose represents every nutrient that bacteria could need; its levels should be kept high, but can fluctuate quite a bit. Contaminol is a small alcohol, a marker produced by a bacterium that competes with the E coli in our bio-reactor. Contaminol is not dangerous by itself, but even the smallest amount is a marker for contamination by other bacteria that can outgrow our E coli and ruin our product.

Build your liposome as a one-stop check of reactor conditions. An operator should be able to extract a liquid sample from the reactor, add a given amount of liposomes, incubate for a few minutes, and use a fluorescence reader to see all relevant substance concentrations in the reactor. You can assume that nutrient and marker levels stay constant in your sample once you extract it from the tank. Your reporter should produce two signals. The first should be proportional to the amount of glucose, and can be used as a rough gauge. The second should be an ultra-sensitive marker, such that even minute amounts of contaminol should set off a strong signal.



[Note on the diagram: Your response curves don't need to look exactly like the graphs depicted here, with a straight line for glucose and that particular red line for contaminol. The former just needs to be proportional and the latter needs to show switch-like behavior].

Your liposomes are ready-made, self-contained biosensors that use the same pallet of receptors and signal transduction pathways used by bacteria and eukaryotes but don't have to transcribe and translate them on the spot. You can assume the liposomes come pre-loaded with all the parts you design them to have at the beginning of the experiment. You can design your sensing pathway with existing proteins, or invent new proteins that behave in a similar way to known enzymes. Stay within the archetypes that we know exist, but be creative in putting circuits together! There are many ways to design this system. You may invent a receptor for the substances you're trying to detect, but remember that this is not the only way in which they can have an effect on your cell (and you are not required to use a receptor). Also remember that proteins are modular, and so useful parts from different known proteins can be brought together into a useful chimera of your own design.

If you decide to use transcription to control your fluorescent reporting, you can use a simplified model with constant rates of RNA transcription and degradation. You may ignore the creation and degradation of all your pre-loaded proteins, but you may also include fluorophore degradation as a way of controlling steady-state levels (if you wish). It is crucial to be realistic about the number of players you can put into your system! A simple protein circuit is much easier to model and debug, and may produce a result that is almost as good as a network of 100 enzymes.

Your glucose signal should have a dynamic range between 0.02 and 2 g/L. Your contaminol signal should be sensitive to levels as low as 50nM.

Part 1: Conceptual design

Due date: September 28th @ 11:59 am.

a) Draw a schematic of an engineered regulatory system that provides an appropriate response to glucose and contaminol concentrations in the sample. Label all system components in your schematic.

Note 1: We repeat. The simpler the design, the easier the project will be.

Note 2: You may design the conditions inside the liposome to be useful (choosing concentrations of ATP and other components). A good starting point is the condition inside normal mammalian or bacterial cells, which will be the origin of many of the parts you might be using. Your chosen concentrations must be biologically realistic/relevant.

b) Justify why/how your system is biologically plausible. Clearly describe and motivate any assumptions you have made to support the design. If you use external resources, cite them appropriately.

Note: Although your system should be biologically plausible, it does not need to be an established method. You may use the literature as a guide, but the project is written such that you should be applying principles from class.

c) Clearly identify the state variables, parameter rate constants, and system inputs.

d) Determine realistic ballpark values for these parameters (i.e. specify relevant orders of magnitude).

Note: This is a good place to use the literature as a guide!

Part 2: Mathematical design

Due date: October 12th @ 11:59 am, along with the most up to date previous section (see the Note below).

a) Write out the system of ODEs that characterizes your network and clearly state any assumptions you make.

b) What are the state variable units? Parameter units? Clearly show all work and thought processes.

c) Input the system into MATLAB and describe/demonstrate how you fit the parameters. NOTE: Most of the script has been written for you! Refer to files `run_sensorODE.m`, `sensorODE.m` and `sensorODE_solver.m` on the course website. Please do not modify the `sensorODE_solver.m` code. Input your model in the `sensorODE.m` script and follow the commented directions in `run_sensorODE.m` to appropriately simulate and plot system dynamics.

d) Simulate the sensor's response to the given (noise-free) input combinations. In your report, include the following plots:

One figure that consists of 2 subplots with the following dynamic outputs:

1. The first subplot should depict all state dynamics as a function of time. (How do your different species vary with time?)
2. The second subplot should depict fluorescent output as a function of time. (Think about what you are measuring for fluorescence. Is it a certain species, transcription, etc.?)

Generate the subplot described above for the following conditions:

1. Low Glucose (0.02 g/L)
2. High Glucose (2 g/L)
3. Low Contaminol (10 nM)
4. High Contaminol (70 nM)

Note that your two systems can be completely independent such that you only need consider certain species for each condition above. If your two systems are dependent on each other, you must consider combinations of the above (i.e. low glucose and low contaminol, low glucose and high contaminol etc.).

Finally, include a separate figure that is an equilibrium plot that depicts how the fluorescence changes with glucose or contaminol concentration for your two different systems. These should look something like the sample plots provided in the project description above.

For all plots, clearly label titles, axes (with units!), and legends. If your system isn't able to achieve your desired response, explain why it might not work.

e) Does your design require that certain parameters be less than or greater than others? Explain why.

f) Discuss interesting response dynamics: For instance, does the system handle input switching? Is there oscillatory behavior?

Note: If you are having difficulties implementing your original conceptual design or want to make changes, you may alter and improve it at this point. If altering your design, repeat Part 1 and include the final version with this section.

Part 3: Sensitivity analysis

Due date: October 18th @ 11:59 am, along with every previous part.

a) Parametric sensitivity analysis: Any modeled system is affected by the empirical parameters derived to relate biological processes and reactions. Sensitivity analysis is one way to quantitatively analyze how much your system depends on these numerical values. Identify **three** relevant properties of the system output (such as transient time and/or amplitude) that could change as a result of parametric perturbations. These properties should be quantifiable from your system output figures.

b) Choose **two** unique parameters that affect system performance and discuss how/why changing these parameters would lead to changes in your three output properties from Part a. Discuss how you might employ parametric sensitivity analysis to refine your model. Now vary these two parameters (consider a range of about five orders of magnitude) and quantify the change of each of the perturbations on your three system output responses.

c) Speed up the response of your network by redefining a single parameter value (or many, if you want to optimize further). Clearly state which parameter values have changed and copy/paste the resulting simulation figure.

d) State sensitivity analysis: Identify whether your system is susceptible to changes in your initial conditions. Why or why not?

Part 4: Robust output response with respect to environmental perturbations

Due date: October 18th @ 11:59 am, along with every previous part.

a) How would your system's response change if it was implemented in an open *in vitro* system instead of in isolation in the liposome? Is your network robust to measurement noise, or chatter? Use the supplied code in *sensorODE_solver.m* to help you answer this question.

Note: Noise due to chatter is a Bernoulli process random variable where the probability of unsuccessful transmission is set to 0.02. Therefore, the static Boolean type input will "chatter" 2-percent of the time, flipping between 'true' and 'false', or vice versa."

b) Is your network robust to stochastic variation? Again use the provided code in *sensorODE_solver.m* to help you answer this question

Note: Noise due to stochastic effects is defined as a white Gaussian variable with a mean of 0 and a variance of 1-percent of the magnitude of the input."

c) Is your network robust to fluctuations in the magnitude of your input? What happens when input concentrations are outside of the design parameter ranges? Does the system "break" easily? How does this change affect system dynamics?

Hint: Change the `maxInputAmpl` value within *run_sensorODE.m* with input concentrations that are a few orders of magnitude above and below those suggested in the project motivation section.

Part 5: Future Work

Due date: October 18th @ 11:59 am, along with every previous part.

a) What other useful environmental parameters might a system like ours be used to detect? In what other contexts could it be helpful? Industry, research, medicine, and total speculative fantasy are all fair game for you to think of an answer to this part.

b) If your system was sensitive to noise (either instrument "chatter" or stochastic fluctuations), are there ways to re-design the system to smooth out these behaviors? Can you make your output more robust to these sorts of perturbations?

Supplementary Information

Protein Network Design Project

General Comments

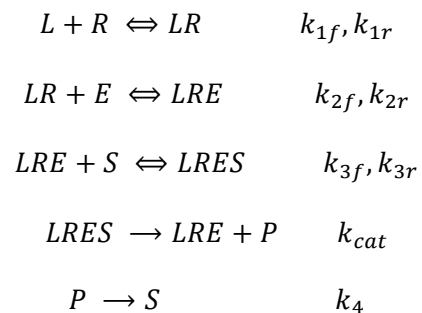
- Your signal for presence of contaminant should be qualitatively different from your signal for nutrient levels (not, say, a super-high level of the same signal).

Design Guidance

- Finding rate constants and parameters is difficult, especially if you are designing novel interactions. Often times, it is a good idea to start with biologically relevant parameter values (i.e. $K_d = 10^{-6}$ M) and vary the order of the magnitude to obtain a desired system response (i.e., test constants at 10^{-9} , 10^{-7} , 10^{-5} , 10^{-3} , etc., to get a feel for how your system responds). Where appropriate, you should cite the literature for the order of magnitude that you selected to represent a binding event.
- Again, *proteins and receptors are modular*. You can combine domains and pieces of proteins to create chimeras with different functions.

Sensitivity Analysis

When building a model, sensitivity analysis is a way of quantifying the extent to which your model output is affected by the rate constants that you have selected for your model. For instance, consider the system below:



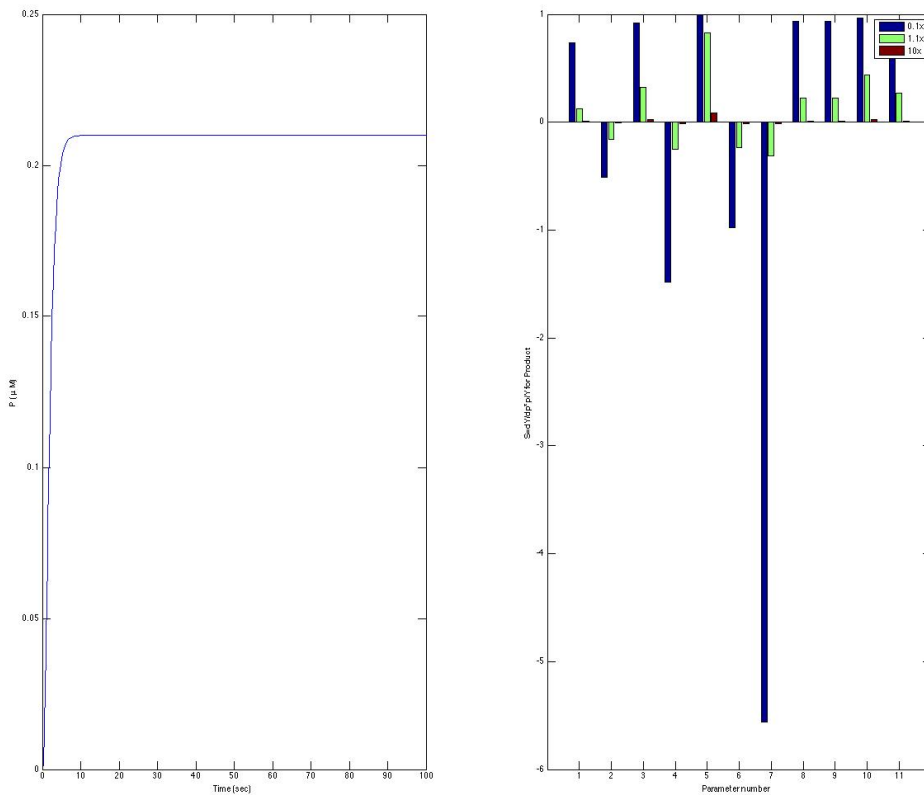
Where ligand (L) binds receptor (R) with forward and reverse rate constants (k_{1f} and k_{1r}) and LR complex binds enzyme (E) with forward and reverse rate constants (k_{2f} , k_{2r}). The LRE complex can now bind the substrate, S with forward and reverse rate constants (k_{3f} and k_{3r}). The new complex LRES can then catalyze the reaction of substrate (S) to product (P) with a rate constant k_{cat} . Lastly, product is converted back to substrate at an arbitrary rate k_4 .

You do some research and come up with a set of differential equations to represent the system and now want to know how changing each of the rate parameters affects your system output. One way to do this is to vary the system's parameters over several orders of magnitude and then measure the amount of product formed at equilibrium. In this example, we have varied the 11 system parameters over three

orders of magnitude (from 1/10 of their original value to 100 times their original value and then calculated the normalized sensitivity to these perturbations.

$$S = \frac{\left(\frac{P_{newparam} - P_{orig.param}}{P_{orig.param}} \right)}{\left(\frac{K_{modified} - K_{original}}{K_{original}} \right)}$$

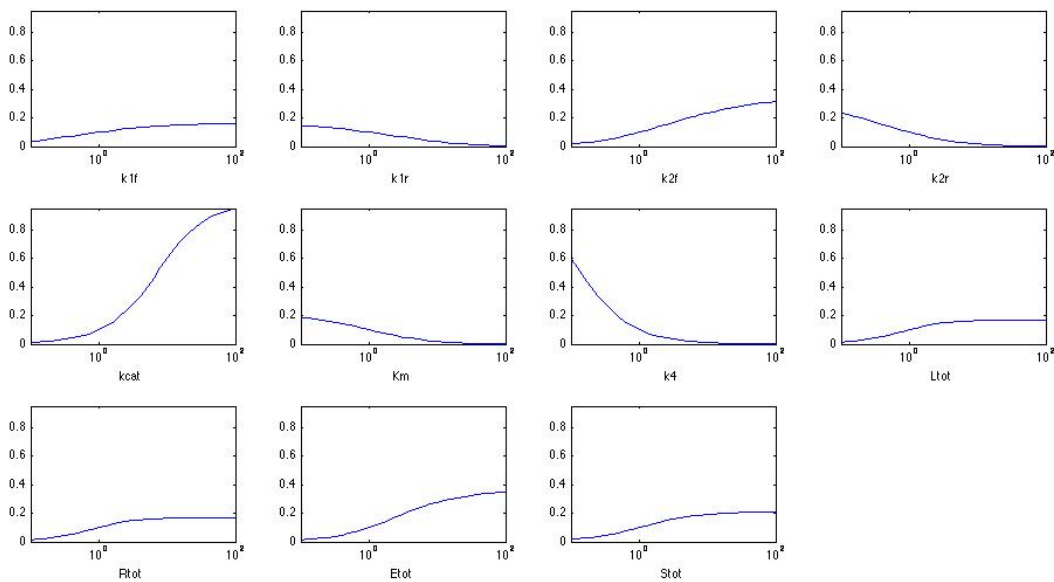
Where P is still the product formed and K can represent any of the 11 rate constants in the model.



On the left, the figure shows the amount of product formed in time with the initial parameters for the system (for this toy model, they were all set to 1). On the right, the normalized sensitivity is plotted per parameter: [k1f, k1r, k2f, k2r, kcat, Km, k4, Ltot, Rtot, Etot, Stot].

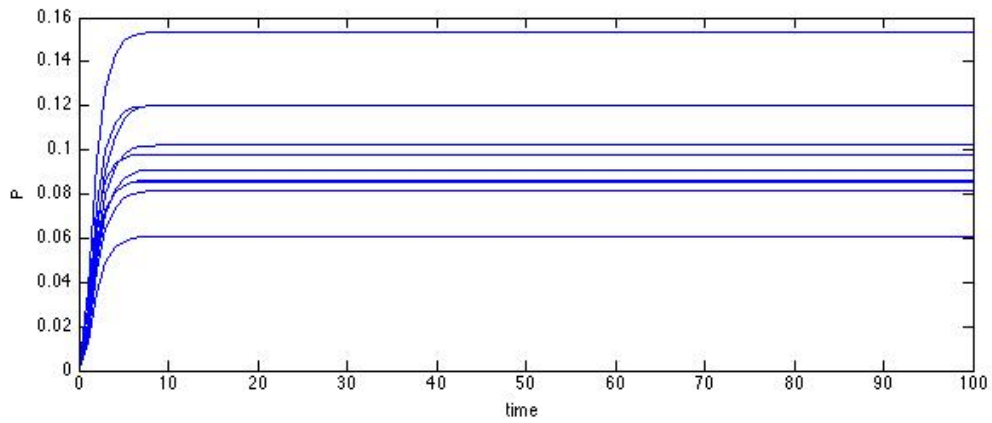
The plot on the right shows the model is most sensitive to parameters 7 and 4 (or k_4 and k_{2r}) meaning that when we vary the rates at which product is returned to substrate, and when LRE complex falls apart, we see the most drastic changes in system output.

Be careful not to confuse sensitivity with absolute product formation. This particular sensitivity parameter is measuring the change in output compared to the change in the parameter value. We can visualize this better by plotting the equilibrium product concentrations against the altered parameter values:



You'll notice that the amount of product formed drastically increases when we increase the catalytic efficiency of the enzyme. The system does respond to this parameter change but the change in output is more proportionate to the change in parameter value (see figure 1).

For your project you have been asked to identify three system properties that you can quantify (and for this example we only looked at steady state concentration). You might get more ideas for system properties to quantify by looking at product formation in time and may be able to quantify rate to equilibrium. For this system, if we look at product formation in time while varying our parameters, we can clearly see that there is a change in the time-dependent dynamics and might see different areas where we can quantify the parameter's affect on system output:



Note: another difference for your project is that we have only asked you to perform the sensitivity analysis for two of your parameters and we have asked you to vary them over 5 orders of magnitude. This example was merely to expose you to sensitivity analysis and get a feel for what it might look like when implemented for a set of ODE equations.

MIT OpenCourseWare
<http://ocw.mit.edu>

20.320 Analysis of Biomolecular and Cellular Systems
Fall 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.