20.320 — Problem Set # 5

October 22nd, 2010

Due on October 29th, 2010 at 11:59am. No extensions will be granted.

General Instructions:

1. You are expected to state all your assumptions and provide step-by-step solutions to the numerical problems. Unless indicated otherwise, the computational problems may be solved using Python/MATLAB or hand-solved showing all calculations. Both the results of any calculations and the corresponding code must be printed and attached to the solutions. For ease of grading (and in order to receive partial credit), your code must be well organized and thoroughly commented, with meaningful variable names.

2. You will need to submit the solutions to each problem to a separate mail box, so please prepare your answers appropriately. Staples the pages for each question separately and make sure your name appears on each set of pages. (The problems will be sent to different graders, which should allow us to get the graded problem set back to you more quickly.)

3. Submit your completed problem set to the marked box mounted on the wall of the fourth floor hallway between buildings 8 and 16.

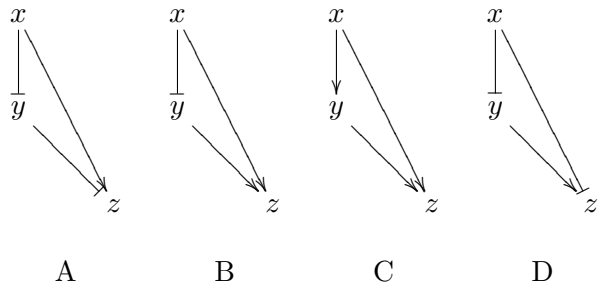
4. The problem sets are due at noon on Friday the week after they were issued. There will be no extensions of deadlines for any problem sets in 20.320. Late submissions will not be accepted.

5. Please review the information about acceptable forms of collaboration, which was provided on the first day of class and follow the guidelines carefully.

106 points + 8 EC for problem set 5.

1 Feed-forward loops, gene circuit design, and cellular logic

a) Consider the following feed-forward loop motifs:



i) For each motif, state whether it is coherent or incoherent.

Solution:

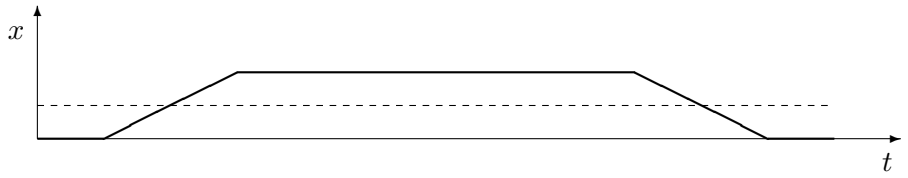
A,C,D: coherent. B: incoherent.
2 points: 0.5 points each.

ii) FFL (C) is commonly found regulatory networks governing metabolism. Find an example of it in glycolysis, state the molecular identity of x, y, and z, and *briefly* state the biological function of the FFL.

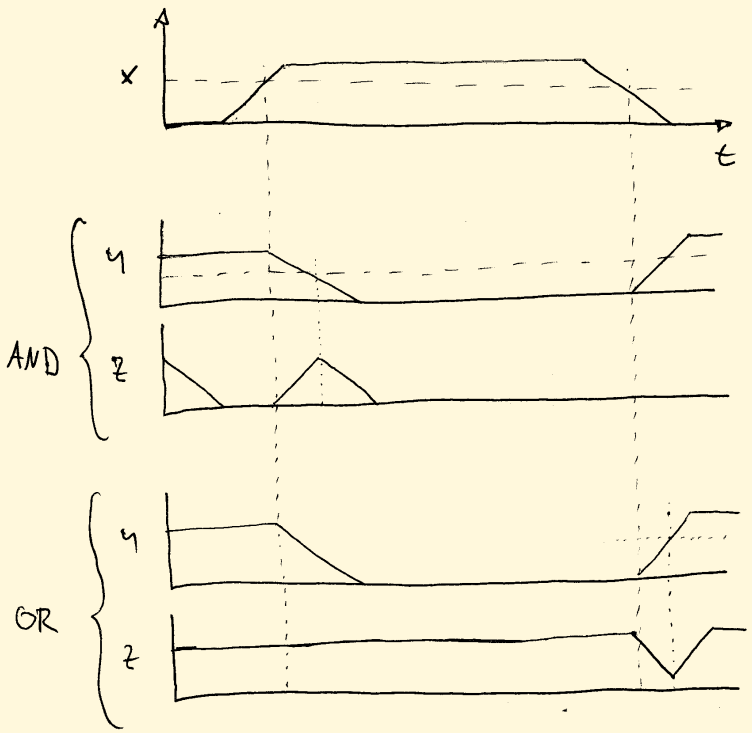
Solution:

- Example 1: (x,y,z) = (glucose-6-phosphate, phosphoenol-pyruvate, pyruvate kinase)
- Example 2: (x,y,z) = (fructose-6-phosphate, PFK2, fructose-2,6-bisphosphate)
- Note that in these examples, the "arrows" mean nothing more than a positive or negative influence on concentrations – these are not examples of strictly transcriptional regulation.
- Simple explanation of biological function: The FFLs make the levels of glycolytic enzymes more robust to brief fluctuations in substrate availability (this is enough for full credit).
- Longer explanation: The coherent OR-gated FFL of type C as depicted above *delays downregulation* of z when either x or y dip. That way, the enzymes of the glycolytic machinery remain active even during short fluctuations in substrate supply (or drops in the levels of intermediates due to shunting into other pathways). That way, instead of going through the metabolic expense of frequently up-and down-regulating enzyme levels in a fluctuating environment, the cell can save energy assuming (as it often can) that glucose will be available for extended periods of time when it is available at all.
  6 points: 3 for example, 3 for explanation of function.

iii) For FFL (B), draw the response of y and z to the input x sketched below with initial conditions $(x_0, y_0, z_0) = (0, 1, 1)$ for both the AND-gated and the OR-gated case at z:



Solution:

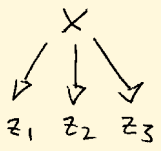


4 points: 2 for AND and 2 for OR case.

b) You are designing a synthetic gene network that will enable an engineered gut bacterium to secrete three specific toxins, z_1, z_2, and z_3, in a strictly sequential fashion upon detecting a pathogen. Using the input, x, only once in your network motif, design a simple FFL circuit which will

i) start secreting them sequentially such that z_1 secretion begins before z_2 secretion and z_2 secretion begins before z_3 secretion.

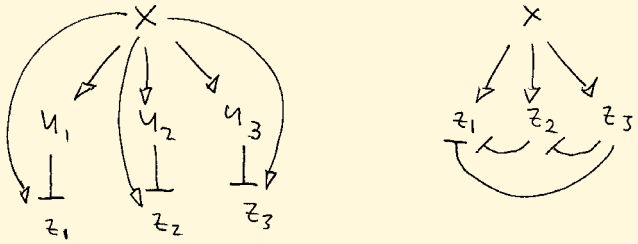
Solution:



This will work with appropriately chosen thresholds.
Any reasonable network should receive full credit. Graders, please deduct points only if you can show the network will *fail*.
3 points incl. suggestion why useful.

ii) secrete z_1 briefly, then stop; secrete z_2 briefly starting after secretion of z_1 has ceased, then stop; and secrete z_3 briefly starting after secretion of z_2 has ceased, then stop. z_1 and z_3 secretion should not overlap in time at all; they each may overlap a little with z_2 secretion.

Solution:



The left example will work with appopriately chosen thresholds. The right example should work without tuning the threshold values.
Any reasonable network should receive full credit. Graders, please deduct points only if you can show the network will *fail*.
5 points incl. suggestion why useful.

For each case, suggest why this might be useful.

Solution:

- "Sequential on" might be useful if the toxins increase in potency from z_1 to z_3, and z_2 and z_3 should only be used if z_1 alone is not sufficient to eliminate the pathogens within some timeframe.
- Alternation might be useful to prevent escape of resistant mutants.
- Any reasonable justification is ok.

20 points overall for problem 1.

2 Transcriptional regulation, feedback, and sensitivity

Let $x = [pX]$ denote the concentration of a transcription factor, pX.

a) Consider the following differential equation for x:

$$\dot{x} = \beta_0 + \beta_x \frac{x}{K_\mathrm{D} + x} - \alpha x$$

Comment on the origin and meaning of each term in the equation. Identify which quantities are state variable(s) and which quantities are parameter(s) of the system.

Solution:

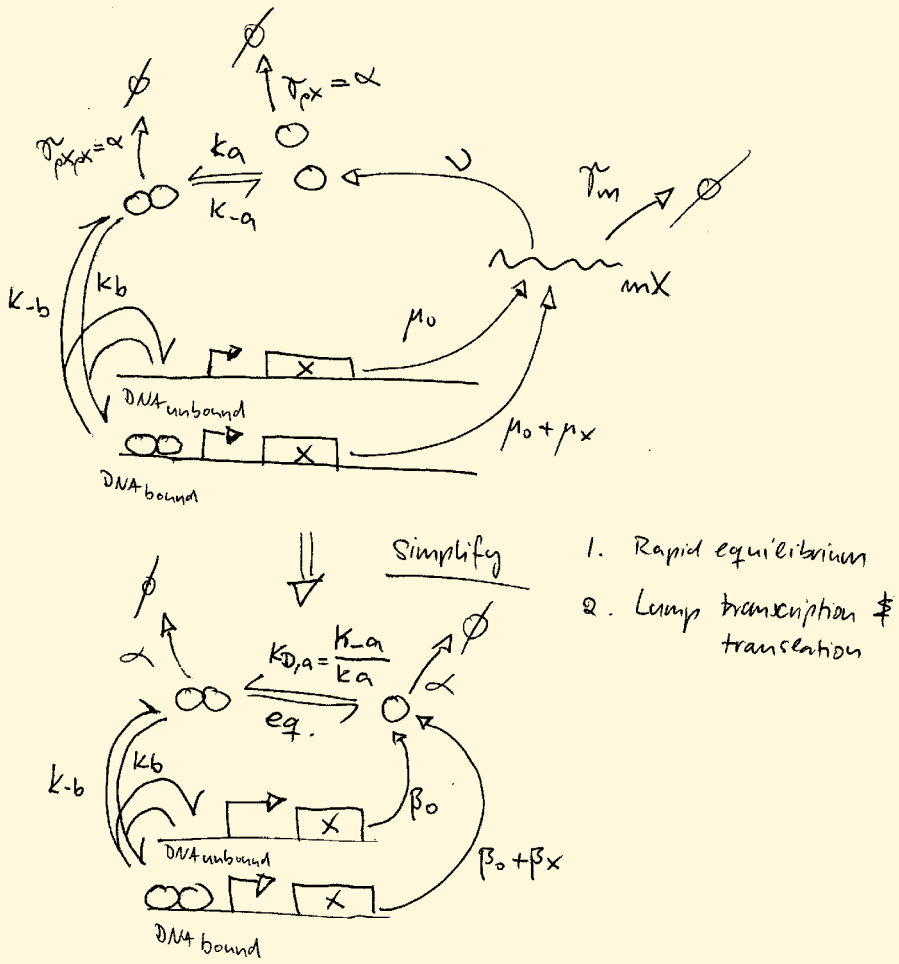
- β_0: basal rate of expression
- $\beta_x \frac{x}{K_\mathrm{D}+x}$: rate of expression due to self-activation of pX
- αx: rate of constitutive degradation or dilution
- x is a state variable
- $\beta_0, \beta_x, K_\mathrm{D}$, and α are parameters

4 points total: 0.5 for each term, and 0.5 for each paramater or state variable.

b) How would this equation change if only the dimeric species, pXpX, were capable of binding to its cognate promoter sequence? Derive the relevant expression, starting with a full (not yet simplified) reaction network. You may simplify your initial model by assuming rapid equilibrium at the dimerization stage (that is, the rates of dimer formation and dissociation, k_a and k_{-a}, are much greater than those of dimer binding to DNA, k_b and k_{-b}) and by lumping transcription and translation.

Solution:

Reaction network:



Derivation of rate due to autoactivation via dimer:

Let $D_b = \text{DNA}_{\text{bound}}$, $D_u = \text{DNA}_{\text{unbound}}$, and $D_T = \text{DNA}_{\text{total}}$. Rate is $\beta_x \left(\frac{D_b}{D_T}\right)$, where β_x is the maximal induced rate when all promoters are fully occupied by pXpX. Now find fractional saturation:

At equilibrium,

$$\dot{D_b} = 0 = k_b D_u [pXpX] - k_{-b} D_b$$

so

$$D_u = \frac{\left(\frac{k_{-b}}{k_b}\right) D_b}{[pXpX]} = \frac{K_{\text{D,b}} D_b}{[pXpX]}$$

Fractional saturation:

$$\frac{D_b}{D_T} = \frac{D_b}{D_b + D_u} = \frac{D_b}{D_b + \left(\frac{K_{\mathrm{D,b}}}{[pXpX]}\right) D_b} = \frac{1}{1 + \left(\frac{K_{\mathrm{D,b}}}{[pXpX]}\right)}$$
$$= \frac{[pXpX]}{K_{\mathrm{D,b}} + [pXpX]}$$

Assuming dimerization equilibrates rapidly,

$$\dot{[pXpX]} \approx k_a x^2 - k_{-a}[pXpX] = 0$$
$$k_a x^2 = k_{-a}[pXpX]$$
$$[pXpX] = \left(\frac{k_a}{k_{-a}}\right) x^2 = \frac{x^2}{K_{\mathrm{D,a}}}$$

Substituting this expression for [pXpX] into the fractional saturation above,

$$\frac{D_b}{D_T} = \frac{\frac{x^2}{K_{\mathrm{D,a}}}}{K_{\mathrm{D,b}} + \frac{x^2}{K_{\mathrm{D,a}}}}$$
$$= \frac{x^2}{K_{\mathrm{D,b}} K_{\mathrm{D,a}} + x^2}$$
$$= \frac{x^2}{K_{\mathrm{D,eff}}^2 + x^2}$$

Note that in general,

$$K_{\mathrm{D,eff}} \neq K_{\mathrm{D,a}} \neq K_{\mathrm{D,b}}!$$

New rate equation:

$$\dot{x} = \beta_0 + \beta_x \frac{x^2}{K_{\mathrm{D,eff}}^2 + x^2} - \alpha x$$

14 points total: 4 for full reaction network, 2 for correctly implementing the simplifications (can draw out simplified network, but don't have to), 2 for fractional saturation in terms of [pXpX], 2 for expressing [pXpX] in terms of x via rapid equilibrium, 2 for DNA fraction bound in terms of x, 2 for full final rate equation. Graders, please feel at liberty to exercise your judgment in instances where steps were skipped: if you feel convinced the student did the correct steps in his head, please award full credit. If the reasoning seems muddled or illogical, penalize as you see fit.

c) **EXTRA CREDIT:** The term $-\alpha x$ can arise by dilution even in the absence of any enzy-

matic action. Explain how, and derive it.

Solution:

As cells proliferate, they *grow* as well as divide, at approximately equal rates such that each generation has same average volume per cell. Assume that increase in volume is approximately uniform in time. Know that cells proliferate exponentially, so $\frac{dV}{dt} = \alpha V$ for the volume. Concentration is $C = \frac{n}{V}$ where n is the *amount of substance*, in moles.

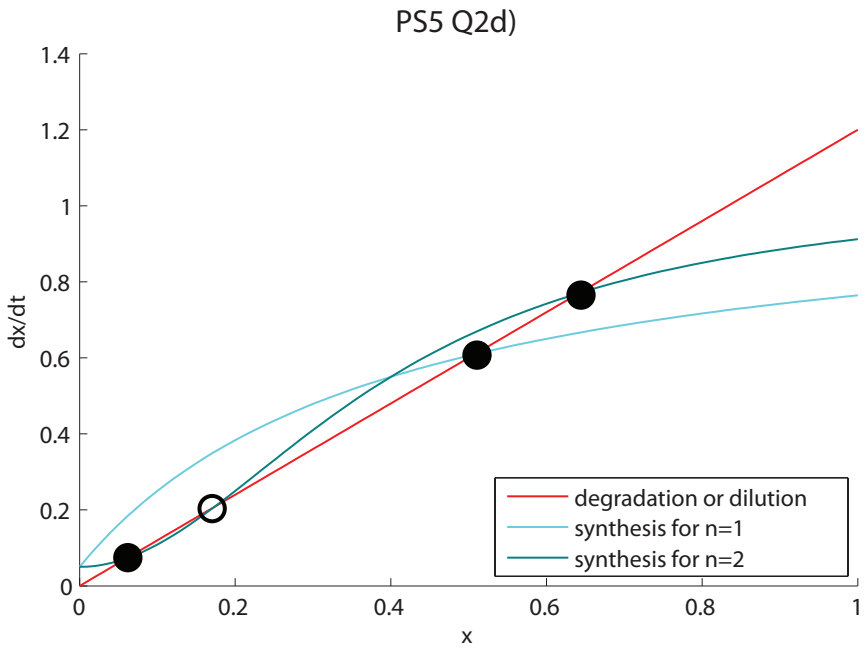
$$\begin{aligned}\frac{dC}{dt} &= \frac{dC}{dV}\frac{dV}{dt} = \frac{d}{dV}\left(\frac{n}{V}\right)\cdot \alpha V \\ &= n\cdot(-1)\cdot V^{-2}\cdot \alpha V = -\frac{\alpha n}{V} \\ &= -\frac{\alpha n}{\frac{n}{C}} \\ \frac{dC}{dt} &= -\alpha C\end{aligned}$$

Thus, protein dilution due to cell proliferation and growth can be modelled as an exponential decrease in concentration such that $\dot{c} = -\alpha c$ where c denotes protein concentration. To relate change in cell volume to change in protein concentration, it's of course possible to invoke the inverse function theorem instead of the chain rule.
8 points total: 3 for idea that thisis driven by cell proliferation, 3 for correctly relating increase in cell number to increase in volume to decrease in concentration, and 2 for correct mathematical derivation of the final answer.

d) In , analyze the transcription factor concentration as a function of time as follows, for both the monomeric and the dimeric activator for $\alpha = 1.2$, $K_{\mathrm{D}} = 0.4$, $\beta_o = 0.05$, and $\beta_x = 1$ (relative units). *Note that you do not need to run any simulations here.*

i) Plot the absolute value of the positive terms and the absolute value of the negative term in the same figure for each case (i.e. create a rate plot).

Solution:



3 points.

ii) What do the intersections of your lines signify?

Solution:

Whereever the lines intersect, synthesis and degradation of pX balance and the system is at steady state. If production exceeds degradation to the left of a steady state and degradation dominates to the right, the steady state is stable (solid circles); if the converse is true, it is unstable (empty circle).
1 point.

iii) For the monomeric and the dimeric activation case separately, state if and justify *briefly* why or why not this feedback loop can govern a cell decision process.

Solution:

For the monomeric case, the system only has one steady state, and it is stable. No matter what the initial conditions, the system will always approach that SS concentration of pX; likewise, if perturbed by an external stimulus, it will always return to it. Hence in the monomeric case, the system cannot govern a decision process.
In the dimeric case, three steady states can be seen, two of them stable and one unstable. Initial conditions or perturbations that take the system to the left of the *unstable* steady state will cause it to end up in the lower steady state. Conversely, once the system is to the right of the unstable steady state, it will tend toward its high steadty state. Thus, the system can stably assume two different states, and can be switched between the two by strong-enough perturbations. This enables it to govern cell-decision processes.
3 points.

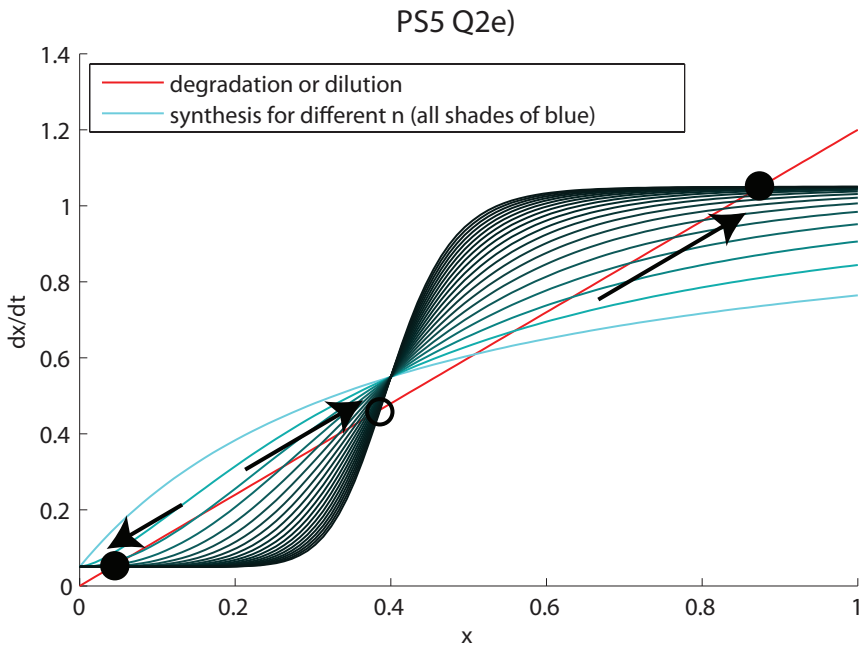
e) The expression for transcriptional activation you obtained above can be generalized to the Hill equation,

$$\theta = \frac{x^{n_{\mathrm{H}}}}{K_{\mathrm{D}}^{n_{\mathrm{H}}} + x^{n_{\mathrm{H}}}}$$

for the fractional activation of a promoter by pX. Use it to replace the corresponding term in your model to answer the following questions:

i) How does the system's behavior change as n_{H} is varied? Is the system robust or sensitive with respect to n_{H}?

Solution:



The system acquire two new steady states — one stable, one unstable — as n_{H} increases from 1 to 2 (this is called a saddle-node bifurcation). Here the system's behavior changes qualitatively even as n_{H} is varied only slightly, making it very sensitive. Past this threshold, increasing n_{H} further moves the steady states apart; but this is only a gradual change, and even its marginal magnitude decreases as n_{H} increases. Thus, the sensitivity of the system with respect to n_{H} has a peak around $n_{\mathrm{H}} = 2$, is much lower elsewhere in the range for n_{H}, and slowly but steadily decreases as n_{H} increases further.
5 points: 3 for plot, 2 for discussion (must note that sensitivity is high around the value of n_H where the monostable and bistable regimes border each other and low elsewhere, e.g. at very high values).

ii) What is this kind of sensitivity analysis called?

Solution:

Parametric sensitivity analysis (since n_{H} is a parameter).
1 point.

iii) How can you use it to *improve* a computational model? Give one example of improvements it can help you implement.

Solution:

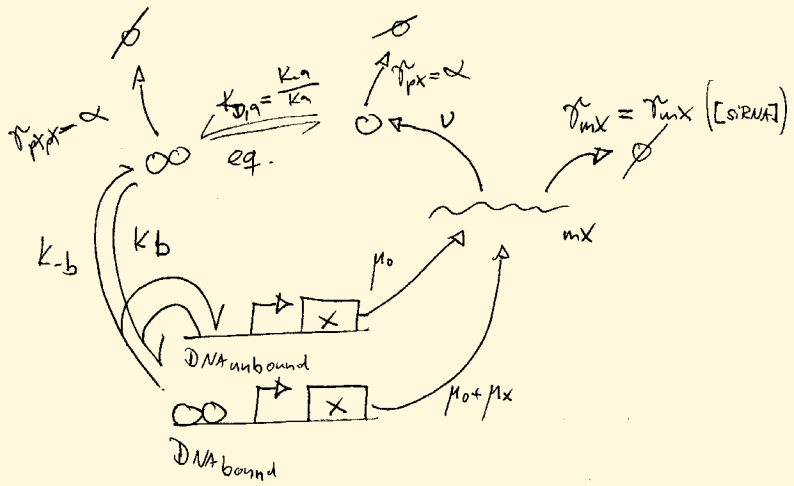
- Parameters or variables to which the system is insensitive can be lumped or altogther eliminated, simplifying the model.
- If the system is highly sensitive to a specific parameter, a more accurate measurement of its value may be prudent.

2 points.

f) You want to model the therapeutic effect of RNAi-mediated downregulation of pX. Let r denote the concentration of an siRNA (treated as an input to the system). How do you need to modify your equations? Draw a full reaction network and write down a system of ODEs to follow the minimal set of relevant species over time.

Solution:

Now, trancription and translation can no longer be lumped.
Reaction network:



To make the notation less cluttered, first define rates for individual reaction steps:

$$
\begin{aligned}
r_1 &= k_{-b}[D_u] \\
r_2 &= k_b[D_b][pXpX] \\
r_3 &= \mu_0[D_u] \\
r_4 &= (\mu_0 + \mu_x)[D_b] \\
r_5 &= \gamma_{mX}([\text{siRNA}])[mX] \\
r_6 &= \nu[mX] \\
r_7 &= \gamma_{pX}[pX] \\
r_8 &= k_a[pX]^2 \\
r_9 &= k_{-a}[pXpX] \\
r_{10} &= \gamma_{pXpX}[pXpX]
\end{aligned}
$$

The ODEs then are

$$
\begin{aligned}
[\dot{D_u}] &= r_1 - r_2 \\
[\dot{D_b}] &= r_2 - r_1 \\
[\dot{mX}] &= r_3 + r_4 - r_5 \\
[\dot{pX}] &= r_6 - r_7 - r_8 + r_9 \\
[\dot{pXpX}] &= r_8 - r_9 - r_{10} + r_1 - r_2
\end{aligned}
$$

10 points total: 5 for network and 1 for eachof the 5 ODEs. Note that modeling siRNA explicitly as another species is more complicated than the problem demands, but perfectly ok.

47 points +8 EC overall for problem 2.

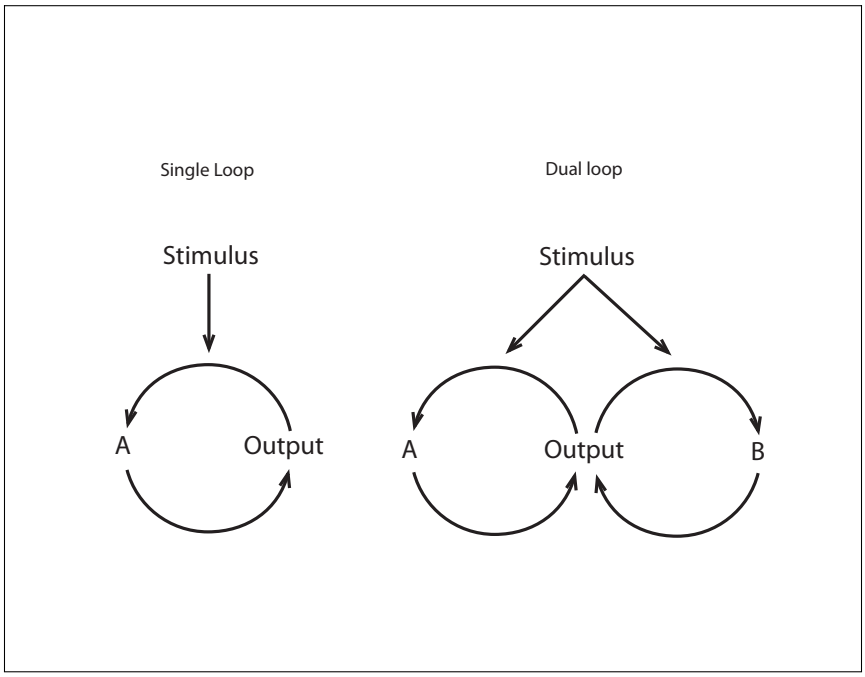
MATLAB code for Problem 2

ps5q2.m:

```
function ps5q2()
clc;
close all;

alpha = 1.2;
k = 0.4;
bo = 0.05;
bx = 1;
x=linspace(0,1,100);

% 2)d)i): Monomeric and dimeric binding
figure()
hold on;
plot(x,alpha.*x,'r-', 'LineWidth', 1);
for n=[1, 2]
  plot(x,syn(x,k,bx,bo,n),'Color', [0 1/n 1/n], 'LineWidth', 1);
end
legend('degradation or dilution', 'synthesis for n=1', ...
   'synthesis for n=2','Location','SouthEast');
title('PS5 Q2d)','FontSize', 16, 'FontWeight', 'bold');
xlabel ('x','FontSize', 12, 'FontWeight', 'bold');
ylabel ('dx/dt', 'FontSize', 12, 'FontWeight', 'bold');
set(gca,'FontSize',12, 'FontWeight', 'bold');
hold off;

% 2)e)i): General Hill equation
figure()
hold on;
plot(x,alpha.*x,'r-', 'LineWidth', 1);
for n=linspace(1,10,20)
  plot(x,syn(x,k,bx,bo,n),'Color', [0 1/n 1/n], 'LineWidth', 1);
end
legend('degradation or dilution', ...
   'synthesis for different n (all shades of blue)', 'Location','NorthWest');
title('PS5 Q2e)','FontSize', 16, 'FontWeight', 'bold');
xlabel ('x','FontSize', 12, 'FontWeight', 'bold');
ylabel ('dx/dt', 'FontSize', 12, 'FontWeight', 'bold');
set(gca,'FontSize',12, 'FontWeight', 'bold');
hold off;

function dx = syn(x,KD,bx,bo,n)
  dx=bo + bx.*(x.^n) ./(KD.^n + x.^n);
```

3 Interlinked positive feedback loops

Positive feedback allows systems to convert graded inputs into decisive, all-or-none outputs. Here you will explore the benefits of multiple interlinked positive feedback loops rather than using a single one. This exercise is highly relevant since many biological processes rely on these networks such as calcium signaling, polarization of budding yeasts or p53 regulation. Here we consider two systems described by the schematic below:



The system is governed by the following differential equations:

$$
\begin{aligned}
\dot{OUT} &= \mathrm{k_{out,\ on}} \cdot (\mathrm{A} + \mathrm{B}) \cdot (1 - \mathrm{OUT}) - \mathrm{k_{out,\ off}} \cdot \mathrm{OUT} + \mathrm{k_{out,\ basal}} \\
\dot{A} &= \left(\mathrm{Stimulus} \cdot \frac{\mathrm{OUT}^n}{\mathrm{OUT}^n + \mathrm{EC}_{50}^n} \cdot (1 - \mathrm{A}) - \mathrm{A} + \mathrm{k_{basal}} \right) \cdot \alpha_{\mathrm{A}} \\
\dot{B} &= \left(\mathrm{Stimulus} \cdot \frac{\mathrm{OUT}^n}{\mathrm{OUT}^n + \mathrm{EC}_{50}^n} \cdot (1 - \mathrm{B}) - \mathrm{B} + \mathrm{k_{basal}} \right) \cdot \alpha_{\mathrm{B}}
\end{aligned}
$$

a) This model neglects a great deal of biological processes such as translation, transport, post-translational modifications. Furthermore, the term (1-OUT) allows the system to be stable. What is the significance of all the other terms and parameters of this system?

Solution:

- Parameters:

Parameter	Significance
$k_{out,on}$	Inducible max transcription rate of OUT
$k_{out,off}$	Degradation rate of OUT
$k_{out,basal}$	Basal transcription rate
k_{basal}	Basal transcription rate of A or B
n	Hill coefficient for promoter activation
EC_{50}	Half-activation concentration
$\alpha_{A,B}$	Proportionality coefficient

- State variables: OUT, A, B
- $k_{out,\ on} \cdot (A + B)$: Inducible transcription of OUT by A or B.
- $k_{out,\ off} \cdot OUT$: First order degradation rate of OUT
- -A or -B: First order degradation rate of A or B
- Stimulus $\cdot \frac{OUT^n}{OUT^n + EC_{50}^n} \cdot (1 - A,B)$: Stimulus induced activation of A or B as a function of promoter occupancy with Hill coefficient n.

Total 10 points: 3 points for all 6 constants, 2 points for state variables, 5 points for description of functional terms.

b) Using the parameters given in the table bleow, plot the simulation over 1000 seconds for 5 different systems with three constant stimulus from t_1 = [30,50], t_1 = [100,600], t_1 = [750,770]. *Hint: for this problem, we will solve the system numerical using Euler's integration method. You have been provided with a code that will solve the system for you. All you need to do is plot the right conditions.*

i) One slow loop

ii) Two slow loops

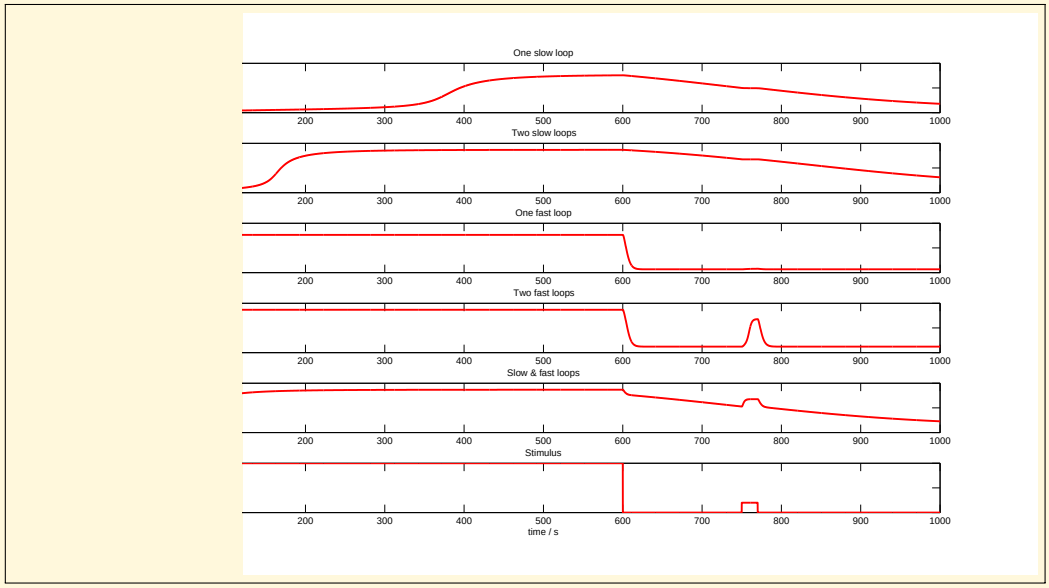
iii) One fast loop

iv) Two fast loops

v) One slow loop + one fast loop

Parameter	Value
$k_{out,on}$	2
$k_{out,off}$	3
$k_{out,basal}$	0.001
k_{basal}	0.01
n	3
EC_{50}	0.35
Fast loop α	0.5
Slow loop α	0.008

Solution:



8 points: 4 points for euler numerical integration implementation. 4 points for plots.

c) What differences do you see between these networks?

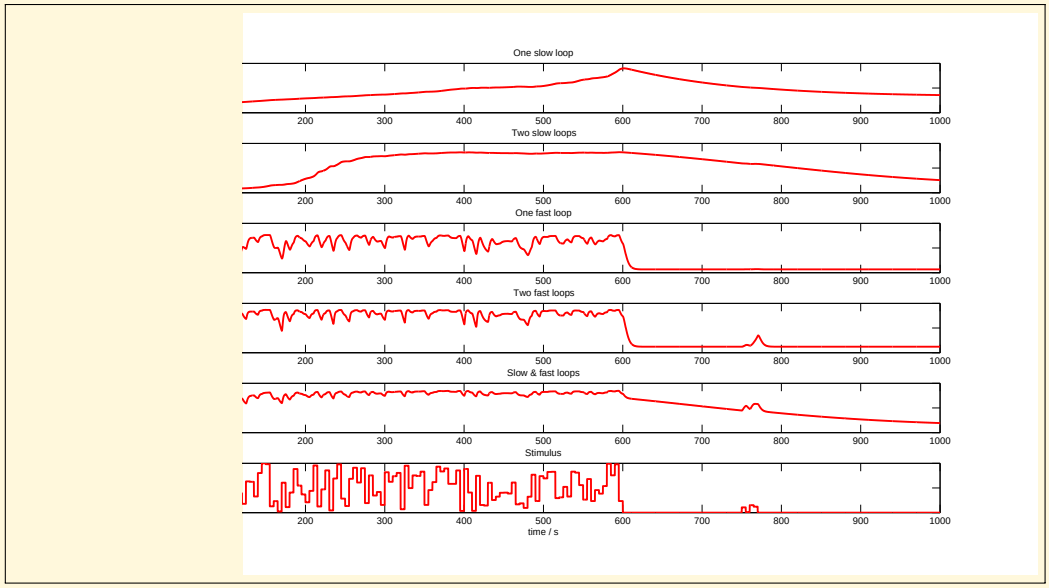
Solution:

Slow networks are have both slow activation and inactivation of OUT during the large stimulus region, but insensitive to the small stimulus regions. The one fast loop is insensitive to short stimulus but activates/inactivates OUT very rapidly in the long stimulus region. The two fast loops on the other hand is sensitive to short stimulations. The combination of fast and slow loop allows little sensitivity to short stimulus with a yet robust swtich-like response for the activation period.
4 points

d) As we have discussed in class, the robustness to noise is an important attribute for some networks. Now, instead of a constant stimulus, repeat your simulations in question b but

with a noisy signal. *Hint: Use randomly generated stimulus between 0 and 1 alternating with* Δ *t of 5 seconds. For this you may use the function* `rand!`.

Solution:



Total 8 points: 4 points for the implementation of noisy stimulus, 4 points for graphs. Different reasonable values of stimulus are ok.

e) How do these networks differ now?

Solution:

One slow loop is to weak to considerably activate OUT during the stimulus period. Two slow loops seems somewhat identical to the non noisy scenarios, indicative of high noise robustness. The fast loop and two fast loops are considerably noisy. However interlinked slow and fast loops maintain a somewhat constant signal during the stimulation region. The stimulus looks like manhattan.
4 points

f) Can you envision a condition when the system would be even more resistant to noise?

Solution:

In stead of a OR logic, we could have an AND logic.
3 points: other valid answers accepted

g) What is the advantage of having interlinked fast and slow positive feedback loops?

Solution:

Interlinked fast and slow positive have switch-like activation and are robust to noise.
2 points: other valid answers accepted

39 points overall for problem 3.

MATLAB code for Problem 3

interlinkedloops.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              Problem Set # 5
%                  Interlinked Positive feedback loops
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function interlinkedloops()

clear all;
clc;

k = initk;


stim = init_stim(); % Create stimulus
alpha_A = [0.008 0.008 0.5 0.5 0.008];
alpha_B = [0     0.008 0   0.5 0.5];
titles = {'One slow loop','Two slow loops', 'One fast loop', 'Two fast loops',...
          'Slow & fast loops'};

figure(1);
j = 0;
for i=1:5
    subplot(6,1,i);
    j = j + 1;
    k(7) = alpha_A(i);
    k(8) = alpha_B(i);
    [T, a, b, out] = feedback([0 0 0], k, stim);
    nice_plot(T,out, '', 'Out', titles(j), [1 0 0]);
end
subplot(6,1,6);
nice_plot(T,stim, 'time / s', 'Stimulus', 'Stimulus', [1 0 0]);

figure(2);
stim_noisy = init_noisy_stim(); % Create noisy stimulus
j = 0;
k = initk;
for i=1:5
    subplot(6,1,i);
    j = j + 1;
    k(7) = alpha_A(i);
    k(8) = alpha_B(i);
    [T, a, b, out_noisy] = feedback([0 0 0], k, stim_noisy);
    nice_plot(T,out_noisy, '', 'Out', titles(j), [1 0 0]);
end
subplot(6,1,6);
nice_plot(T,stim_noisy, 'time / s', 'Stimulus', 'Stimulus', [1 0 0]);



function k = initk()
k = [2        .3        0.001   0.01      3 0.35   0        0];
%    kout_on kout_off kout_basal k_basal n ec50 alpha_A alpha_B
```

```
end

function stim = init_stim()
    stim =zeros(1,10000);
    for i=301:501    % small first stimulus
        stim(i) = .2;
    end
    for i=1001:6001  % Long second stimulus
        stim(i) = 1;
    end
    for i=7501:7701  % Third short stimulus
        stim(i) = .2;
    end
end

function stim = init_noisy_stim()
    stim =zeros(1,10000);
    start = 301;
    for k= 1:4
        val = 0.2*rand();
        for i=start+50*(k-1):start+50*k
            stim(i) = val;
        end
    end
    start = 1001;
    for k= 1:100
        val = rand();
        for i=start+50*(k-1):start+50*k
            stim(i) = val;
        end
    end
    start = 7501;
    for k= 1:4
        val = 0.2*rand();
        for i=start+50*(k-1):start+50*k
            stim(i) = val;
        end
    end

end



% Euler numerical integration of system
function [t,a, b, out]=feedback(x0,k,stim)
 % k = [kout_on, kout_off, kout_basal, kbasal, n, ec50, alpha_A alpha_B]
 % k    ---1--------2---------3----------4-----5--6-------7------8
t=linspace(0,1000,10000);
dt=t(2)-t(1);
ns=length(t);

% initialization
a=zeros(ns,1);
b=zeros(ns,1);
out=zeros(ns,1);

% Initial Conditions
a(1)=x0(1);
b(1)=x0(2);
```

```
out(1)=x0(3);

%%Euler method
for i=1:ns-1
    hill = out(i)^k(5)/(out(i)^k(5)+k(6)^k(5));
    a(i+1)=a(i) + dt*[stim(i)*hill*(1-a(i))-a(i)+k(4)]*k(7);
    b(i+1)=b(i) + dt*[stim(i)*hill*(1-b(i))-b(i)+k(4)]*k(8);
    out(i+1)=out(i)+dt*(k(1)*(a(i)+b(i))*(1-out(i))-k(2)*out(i)+k(3));
end

end


function [t,a, out]=oneloop(x0,k,stim)
 % k = [kout_on, kout_off, kout_basal, kbasal, n, ec50, alpha_A alpha_B]
 % k    ---1--------2---------3----------4-----5--6-------7------8
t=linspace(0,1000,10000);
dt=t(2)-t(1);
ns=length(t);

% initialization
a=zeros(ns,1);
out=zeros(ns,1);

% Initial Conditions
a(1)=x0(1);
out(1)=x0(2);

%%Euler method
for i=1:ns-1
    hill = out(i)^k(5)/(out(i)^k(5)+k(6)^k(5));
    a(i+1)=a(i) + dt*[stim(i)*hill*(1-a(i))-a(i)+k(4)]*k(7);
    out(i+1)=out(i)+dt*(k(1)*a(i)*(1-out(i))-k(2)*out(i)+k(3));
end

end
function nice_plot(x,y, Xlab, Ylab, Title, ColorCode)
    plot(x,y, 'Color', ColorCode, 'LineWidth', 2);
    xlabel(Xlab, 'Fontsize', 10);
    ylabel(Ylab, 'Fontsize', 10);
    title(Title, 'Fontsize', 10);
end

end
```

20.320 Analysis of Biomolecular and Cellular Systems
Fall 2012