Systems Biology, Homework # 7
Models of Gene Regulation

Grayson Gerlich

1. **Response time of autoinhibitory genes** Consider the product of a constitutively expressed gene as modelled in equation 7.2 in the notes:

$$\frac{d}{dt}p(t) = \alpha_0 - \delta p(t) \tag{1}$$

For comparison, consider an autoinhibitory gene whose expression can be modelled as in equation 7.8 in the notes:

$$\frac{d}{dt}p(t) = \alpha\frac{1}{1+p(t)/K} - \delta p(t), \tag{2}$$

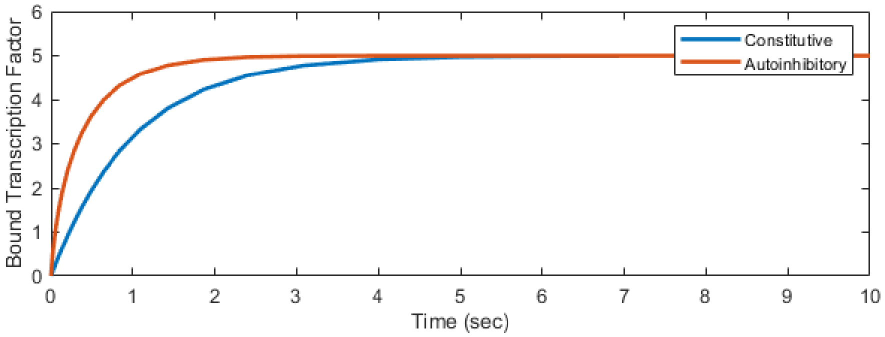
(a) Take $\delta = 1$ (time^{-1}) in models (1) and (2) and let $K = 1$ (concentration) for the autoinhibited gene. Verify that both genes generate the same steady-state protein concentration when $\alpha = \alpha_0(\alpha_0 + 1)$.

$$\frac{d}{dt}p(t) = \alpha_0 - \delta p(t) \longrightarrow 0 = \alpha_0 - p^{ss} \longrightarrow \alpha_0 = p^{ss}$$

$$\frac{d}{dt}p(t) = \alpha\frac{1}{1+p(t)/K} - \delta p(t) \longrightarrow 0 = \alpha\frac{1}{1+p^{ss}} - p^{ss}$$

$$p^{ss} = \alpha_0(\alpha_0+1)\frac{1}{1+a_0} \longrightarrow \alpha_0 = p^{ss}$$

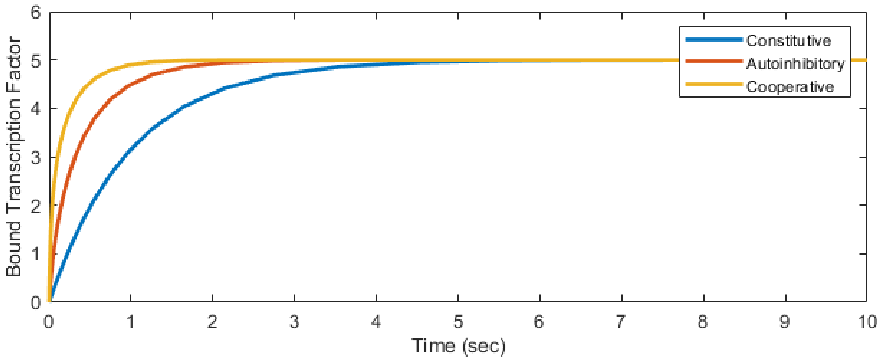
(b) Simulate the two models with $\alpha_0 = 5$ and $\alpha = 30$ (concentration · time^{-1}). Take the initial concentrations to be zero. Verify that, as a result of having a higher maximal expression rate, the autoinhibited gene reaches steady state more quickly than the unregulated gene.



(c) How would you expect the response time to be affected by cooperative binding of multiple copies of the repressor? Verify your conjecture by comparing your results from part (b) with the model

$$\frac{d}{dt}p(t) \quad = \quad \alpha_2 \frac{1}{1+(p(t)/K)^2} - \delta p(t).$$

Take $\alpha_2 = 130$ (concentration $\cdot$ time^{-1}).
I would expect the cooperative binding case to reach steady state even faster because binding accelerates further binding. This is confirmed by the simulation results.



The code I used for this part of the problem is basically identical to the last, so I will just include the model and the driver for this part.

```
function dYdt = q1_model(t,Y)
    % Extract KE, KEP, KD, KDP from input vector Y
    P = Y(1); %P
    P1 = Y(2); %P1
    P2 = Y(3); %P2

    % Define mass action laws
    a0 = 5;
    a = 30;
    a2 = 130;

    % Define dadt, dbdt, dcdt from the ODEs

    dPdt = a0-P;
    dP1dt = a*(1./(1+P1))-P1;
    dP2dt = a2*(1./(1+P2.^2))-P2;


    % Create output column vector dYdt
    dYdt = [dPdt; dP1dt; dP2dt];
end
```

```
clear all

% TODO define the timespan to simulation
tRange = [0 10];
%tRange1 = [0 10];

```

```
% TODO define the initial conditions
Y0 = [0, 0, 0];
%Y1 = [0, 0];

% call the solver of choice (ode45 is fine)
[tSol,YSol] = ode23(@q1_model,tRange,Y0);
%[tSol1,YSol1] = ode23(@q1_model,tRange1,Y1);

% plot solutions to look like figure in lab
plot(tSol,YSol(:,1),'LineWidth',2)
hold on
plot(tSol,YSol(:,2),'LineWidth',2)
plot(tSol,YSol(:,3),'LineWidth',2)
legend('Constitutive','Autoinhibitory','Cooperative','Location','northeast')
xlabel('Time (sec)')
ylabel('Bound Transcription Factor')
hold off
```

2. **The lac operon: CAP** Consider the model of the lac operon presented in Section 7.2.1. Suggest a way to modify Equation (7.11) to include the transcription factor CAP, which represses expression from the lac operon whenever glucose levels are sufficiently high, regardless of the lactose level.

   The simplest method would probably be to represent it similarly to the lac repressor, but dependent on glucose concentration. Letting $g(t)$ represent the glucose concentration, something like:

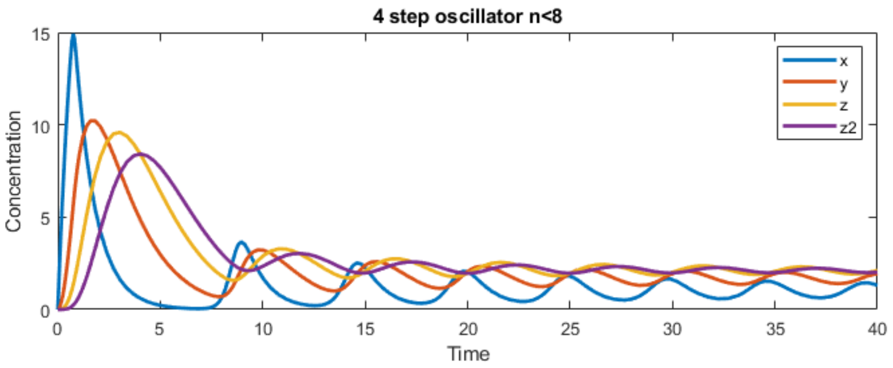
$$\frac{1}{1 + g(t)cap(t)/K_1}$$

3. **Exploration of Goodwin Oscillator** Recall the generic model of an oscillating autoregulatory gene proposed by Goodwin (equation 7.22):

$$\begin{aligned}
\frac{d}{dt}x(t) &= \frac{a}{k^n + (z(t))^n} - bx(t)\\
\frac{d}{dt}y(t) &= \alpha x(t) - \beta y(t)\\
\frac{d}{dt}z(t) &= \gamma y(t) - \delta z(t)
\end{aligned}$$

   This system exhibits limit-cycle oscillations provided the Hill coefficient n is sufficiently large. Unfortunately, for reasonable choices of the other parameter values, n has to be chosen very high (> 8) to ensure oscillatory behaviour. Modifications that generate oscillations with smaller Hill coefficients are as follows. (In exploring these models, make sure simulations run for sufficiently long that the asymptotic behaviour is clear.)

   (a) Taking parameter values as in Figure 7.17, modify the model by adding a fourth step to the activation cascade. (Use dynamics identical to the third step.) Verify that the additional lag introduced by this fourth component allows the system to exhibit sustained oscillations with $n < 8$.

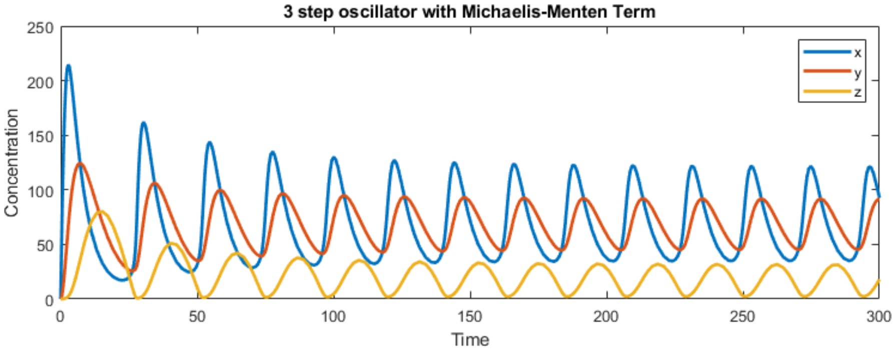


```
function dYdt = q2_model(t,Y)
    % Extract x, y, z from input vector Y
    x = Y(1); %x
    y = Y(2); %y
    z = Y(3); %z
    z2 = Y(4);

    % Define mass action laws
    a = 360;
    k = 1.368;
    b = 1;
    A = 1;
    B = 0.6;
    G = 1;
    D = 0.8;
    E = 1;
    F = 1;
    n = 8;

    % Define dadt, dbdt, dcdt from the ODEs

    dxdt = a/(k^n+z.^n)-b*x;
    dydt = A*x-B*y;
    dzdt = G*y-D*z;
    dz2dt = E*z-F*z2;


    % Create output column vector dYdt
    dYdt = [dxdt; dydt; dzdt; dz2dt];
end
```

```
clear all

% TODO define the timespan to simulation
tRange = [0 40];
%tRange1 = [0 10];

% TODO define the initial conditions
Y0 = [0, 0, 0, 0];
%Y1 = [0, 0];

% call the solver of choice (ode45 is fine)
[tSol,YSol] = ode23(@q2_model,tRange,Y0);
%[tSol1,YSol1] = ode23(@q1_model,tRange1,Y1);

% plot solutions to look like figure in lab

plot(tSol,YSol(:,1),'LineWidth',2)
```

```
hold on
plot(tSol,YSol(:,2),'LineWidth',2)
plot(tSol,YSol(:,3),'LineWidth',2)
plot(tSol,YSol(:,4),'LineWidth',2)
legend('x','y','z','z2','Location','northeast')
xlabel('Time')
ylabel('Concentration')
title('4 step oscillator n<8')
hold off
```

(b) Returning to the original model, replace the term for degradation of Z by a Michaelis-Menten term: $-\delta z/(K_M+z)$. Verify that this modified system oscillates with no cooperativity (i.e. with $n = 1$). Take $a = 150$, $k = 1$, $b = \alpha = \beta = \gamma = 0.2$, $\delta = 15$, and $K_M = 1$. (Units as in Figure 7.17.)



```
function dYdt = q22_model(t,Y)
    % Extract x, y, z from input vector Y
    x = Y(1); %x
    y = Y(2); %y
    z = Y(3); %z

    % Define mass action laws
    a = 150;
    k = 1;
    b = 0.2;
    A = 0.2;
    B = 0.2;
    G = 0.2;
    D = 15;
    n = 1;
    K = 1;

    % Define dadt, dbdt, dcdt from the ODEs

    dxdt = a/(k^n+z.^n)-b*x;
    dydt = A*x-B*y;
    dzdt = G*y-D*z/(K+z);


    % Create output column vector dYdt
    dYdt = [dxdt; dydt; dzdt];
end
```

```
clear all

```

```
% TODO define the timespan to simulation
tRange = [0 300];
%tRange1 = [0 10];

% TODO define the initial conditions
Y0 = [0, 0, 0];
%Y1 = [0, 0];

% call the solver of choice (ode45 is fine)
[tSol,YSol] = ode23(@q22_model,tRange,Y0);
%[tSol1,YSol1] = ode23(@q1_model,tRange1,Y1);

% plot solutions to look like figure in lab

plot(tSol,YSol(:,1),'LineWidth',2)
hold on
plot(tSol,YSol(:,2),'LineWidth',2)
plot(tSol,YSol(:,3),'LineWidth',2)
legend('x','y','z','Location','northeast')
xlabel('Time')
ylabel('Concentration')
title('3 step oscillator with Michaelis-Menten Term')
hold off
```