I represented the network with the stoichiometric matrix

$$V = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 0 & 1 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where the rows correspond to the species

```
gene 1:    g1 = X(1)
gene 2:    g2 = X(2)
protein 1: p1 = X(3)
protein 2: p2 = X(4)
dimer 1:   d1 = X(5)
dimer 2:   d2 = X(6)
complex 1: c1 = X(7)
complex 2: c2 = X(8)
```

and the columns correspond to the reactions

```
g1 + d2 <-- k1/k2 --> c1
g2 + d1 <-- k3/k4 --> c2
p1 + p1 <-- k5/k6 --> d1
p2 + p2 <-- k7/k8 --> d2
g1      ----  k9  ---> g1 + p1
g2      ----  k10 ---> g2 + p2
c1      ----  k11 ---> c1 + p1
c2      ----  k12 ---> c2 + p2
```

The following is the Matlab script I used to simulate the network. I had little intuition for the kinetic rates and initial conditions, so I was unable to elicit the desired oscillatory behavior.

```
% Daniel S. Standage
% BCB 570
% 17 Feb 2012
%
% Adapted from a script written by D.J. Higham, accessed at
% http://personal.strath.ac.uk/d.j.higham/chem/ssa_plot.m

clf

rand('state',100)

% Define stoichiometric matrix
V = [-1 0  0  0  0  0  0  0;
     0  -1 0  0  0  0  0  0;
     0  0  -2 0  1  0  1  0;
     0  0  0 -2  0  1  0  1;
     0  -1 1  0  0  0  0  0;
     -1 0  0  1  0  0  0  0;
     1  0  0  0  0  0  0  0;
     0  1  0  0  0  0  0  0];

% Parameters
nA = 6.023e23; % Avagadro's number
vol = 1e-15;   % volume of system
```

```
% Initial conditions for 8 biochemical species
X = zeros(8,1);
X(1) = 1e-6*nA*vol;    % gene 1
X(2) = 1e-6*nA*vol;    % gene 2
X(3) = 1e-7*nA*vol;    % protein 1
X(4) = 1e-7*nA*vol;    % protein 2
%X(5)   % dimer 1
%X(6)   % dimer 2
%X(7)   % complex 1
%X(8)   % complex 2

% Kinetic rates for 8 (12) reactions
c(1) = 1e-4; c(2) = 1e-10;  % g1 + d2 <-- k1/k2 --> c1
c(3) = 1e-4; c(4) = 1e-10;  % g2 + d1 <-- k3/k4 --> c2
c(5) = 1e-2; c(6) = 1e-6;   % p1 + p1 <-- k5/k6 --> d1
c(7) = 1e-2; c(8) = 1e-6;   % p2 + p2 <-- k7/k8 --> d2
c(9) = 1e-6;                % g1      ---- k9  ---> g1 + p1
c(10) = 1e-6;               % g2      ---- k10 ---> g2 + p2
c(11) = 1e-2;               % c1      ---- k11 ---> c1 + p1
c(12) = 1e-2;               % c2      ---- k12 ---> c2 + p2

% Algorithm
t = 0;
tfinal = 50;

count = 1;
tvals(1) = 0;
Xvals(:,1) = X;

while t < tfinal
     a(1) = -c(1)*X(1)*X(6) + c(2)*X(7);
     a(2) = -c(3)*X(2)*X(5) + c(4)*X(8);
     a(3) = 0.5*-c(5)*X(3)*X(3) + c(6)*X(5) + c(9)*X(1) + c(11)*X(7);
     a(4) = 0.5*-c(7)*X(4)*X(4) + c(8)*X(6) + c(10)*X(2) + c(12)*X(8);
     a(5) = -c(3)*X(2)*X(5) + c(4)*X(8) + 0.5*c(5)*X(3)*X(3) - c(6)*X(5);
     a(6) = -c(1)*X(1)*X(6) + c(2)*X(7) + 0.5*c(7)*X(4)*X(4) - c(8)*X(6);
     a(7) = c(1)*X(1)*X(6) - c(2)*X(7);
     a(8) = c(3)*X(2)*X(5) - c(4)*X(8);
     asum = sum(a);
     j = min(find(rand<cumsum(a/asum)));
     tau = log(1/rand)/asum;
     X = X + V(:,j);

     count = count + 1;
     t = t + tau;
     tvals(count) = t;
     Xvals(:,count) = X;
end

%%%%%%%%%%% Plots

L = length(tvals);
tnew = zeros(1,2*(L-1));
tnew(1:2:end-1) = tvals(2:end);
tnew(2:2:end) = tvals(2:end);
tnew = [tvals(1),tnew];

Svals = Xvals(1,:);
```

```
ynew = zeros(1,2*L-1);
ynew(1:2:end) = Svals;
ynew(2:2:end-1) = Svals(1:end-1);
plot(tnew,ynew,'go-')
hold on

Pvals = Xvals(4,:);
ynew = zeros(1,2*L-1);
ynew(1:2:end) = Pvals;
ynew(2:2:end-1) = Pvals(1:end-1);
plot(tnew,ynew,'r*-')

text(40,240,'Product','FontSize',16)
text(30,50,'Substrate','FontSize',16)

xlabel('Time','FontSize',14)
ylabel('Molecules','FontSize',14)

axis([0 55 0 310])

set(gca,'FontWeight','Bold','FontSize',12)
grid on
```