

Models with Delays

Geneva Porter

11 December 2018

*San Diego State University
Professor J Mahaffy, Math 636*

Goodwin Models for Genetic Expression

The Goodwin models describe genetic regulation. They are given by:

$$\begin{pmatrix} dm(t)/dt \\ dp(t)/dt \\ dr(t)/dt \end{pmatrix} = \begin{pmatrix} \frac{1}{1 + r(t - k)^n} - b_1 m(t) \\ m(t) - b_2 p(t) \\ p(t) - b_3 r(t) \end{pmatrix}$$

In this model, b_1 , b_2 , and b_3 are decay constants, k is the delay for transcription and translation, and n is the Hill coefficient, which we will give a value of 4. The functions $m(t)$, $p(t)$, and $r(t)$ represent the concentration of mRNA, the protein formed from the mRNA, and the end product repressor created by the protein, respectively.

Genetic Repression in Bacteria

Bacteria possess mechanisms that allow for genetic regulation in response to environmental changes. Regulatory molecules in bacteria can activate and attach to DNA near a specific group of genes, then either help or block the expression of those genes. This determines whether or not the specific group of genes will be transcribed into mRNA and expressed or blocked from transcription and repression.

A group of genes, or an operon, is repressible if they can be “turned off” by a corepressor molecule. One example of this is the *trp* operon found in *E. coli* bacteria, which facilitates the creation of tryptophan. When there are high levels of tryptophan present, the tryptophan can act as a corepressor and “turn off” the *trp* operon. Likewise, when tryptophan levels are low, there is little repressor activity and the *trp* operon can be “turned on” or activated to allow the biosynthetic production of the needed tryptophan.

The Goodwin model can be applied to *trp* operon repression by examining the cyclical nature of the genetic repression-expression feedback loop found in bacteria. Let’s say that low levels of tryptophan are present and the *trp* operon is active. The operon continues to signal tryptophan synthesis through transcription to mRNA, and little repression is occurring because tryptophan is needed. As more synthesis occurs, the concentration of mRNA increases, and eventually a high level of tryptophan will facilitate the repression of *trp*. As tryptophan is used up by the bacteria, more will be needed. If there is none available in the environment, repression of *trp* ceases so that tryptophan can again be produced.

The equations above describe this cycle. As the concentration of mRNA increases, the rate of mRNA production dm/dt decreases due to the term $-b_1m(t)$. The increased presence of repressor proteins also slows the rate of mRNA production, as seen in the first term of dm/dt . High amounts of mRNA means an increased rate of protein production, seen in the first term of dp/dt . As the population of proteins increase, its production rate slows, as indicated by the $-b_2p(t)$ term. The rate of change of the repressor created by the proteins (dr/dt) increases along with the $p(t)$ term when high levels of proteins are present, but decreases along with the $-b_3r(t)$ term as more repressors are produced. Decreasing amount of repressor proteins increases the the rate of mRNA production, and the cycle begins again.

Stability Analysis: Undelayed

We can examine the Goodwin models with the parameter values $k = 0$, $b_1 = 1$, $b_2 = 0.1$, and $b_3 = 0.05$. Setting the system to $(0,0,0)$ with these parameters, we get a singe real equilibrium at $(e_m, e_p, e_r) = (0.08934, 0.89340, 1.78681)$. Taking the partial derivatives of each term to form the Jabobian matrix and plugging in the equilibrium values, we get:

$$J \begin{pmatrix} e_m \\ e_p \\ e_r \end{pmatrix} = \begin{pmatrix} -1 & 0 & -0.18213 \\ 1 & -0.1 & 0 \\ 0 & 1 & -0.5 \end{pmatrix} \rightarrow \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} -0.18806 - 0.39281i \\ -0.18806 + 0.39281i \\ -1.22387 \end{pmatrix}$$

The real part of each eigenvalue is negative, indicating that the equilibrium is stable at these parameters. The imaginary parts indicate that this equilibrium is a spiral node.

Stability Analysis: Delayed

If we include the delay term, our equilibrium remains the same, so $r(t - k) = r(t)$ at the equilibrium. However, the stability of the system at the equilibrium is more difficult to determine. Rather than taking the Jacobian (like for non-delay systems) and forming the characteristic equation from the determinant, we need to “separate” the delay term first. To accomplish this, we use the following form for our characteristic equation:

$$\left| -\lambda I_l + J_0 + \sum_{i=1}^n J_{\tau_i} e^{-\lambda \tau_i} \right| = 0$$

Where λ is the solution set of eigenvalues, I_l is the identity matrix for a system of l equations, J_{τ_i} is the Jacobian matrix (with respect to some delay τ_i) evaluated at the equilibrium, and n is the number of delays in the system. This gives us the characteristic equation of:

$$\left| \begin{pmatrix} -1 - \lambda & 0 & -0.18213e^{-k\lambda} \\ 1 & -0.1 - \lambda & 0 \\ 0 & 1 & -0.5 - \lambda \end{pmatrix} \right| = 0 \quad \longrightarrow$$

$$\lambda^3 + (1.6)\lambda^2 + (0.65)\lambda + (0.05) + (0.18213)e^{-\lambda k} = 0$$

We can now examine the stability for delays of different values for k . Using MATLAB, I created a program to map the perimeter of the rectangle bound by the line $x = 0$, $x = 4$, $y = -3$, and $y = 3$ under the transformation of the characteristic equation with $\lambda = x + iy$ (Appendix, Figure i). When separated in to real and imaginary parts, the characteristic equation is rather messy so it will not be shown here. The results are shown in Figures 1 and 2 below.

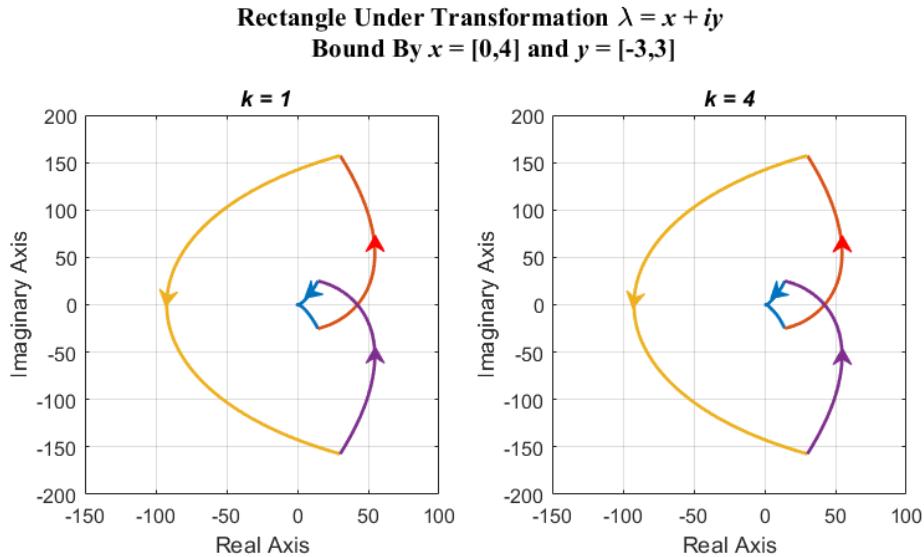


Figure 1 Transformation Under Characteristic Equation

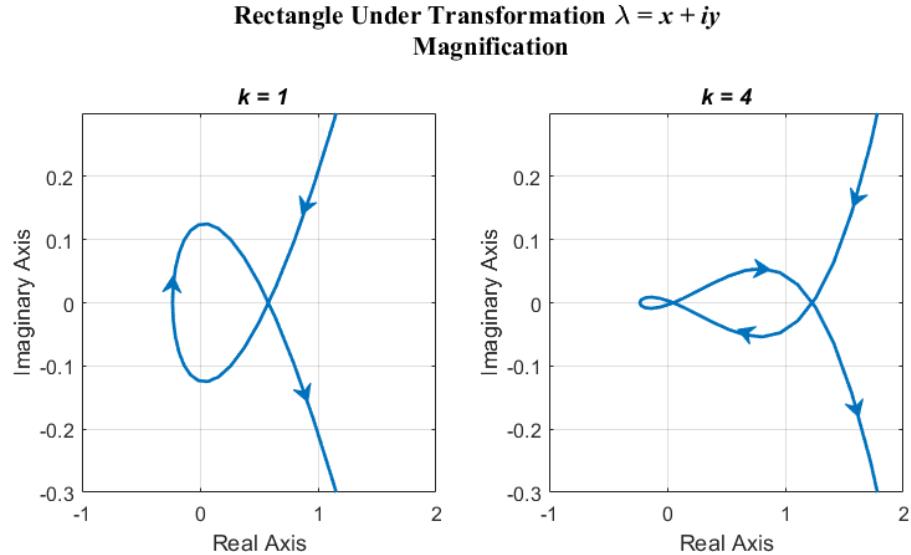


Figure 2 Differences Between $k = 1$ and $k = 4$

Examining these graphs, we can evaluate the winding number for each to determine how many eigenvalues lie inside the bounded region. For $k = 1$, the winding number is 0. We can see from figure 1 that one counter-clockwise loop about the origin is made (+1), but the magnification in figure 2 shows that an additional loop appears going clockwise (-1). These add to zero, indicating that there are no eigenvalues contained in the region under this value of k . If we assume that this region is representative of all eigenvalues with positive real parts, then we can say that the system's eigenvalues likely have negative real parts, and the system is thus stable at the equilibrium we defined earlier.

For $k = 4$, the winding number is 2. We have a similar loop in the clockwise direction (+1), and the magnification shows that the second loop is also in the clockwise direction (+1). Since these add to 2, there are 2 eigenvalues in the region we defined. This indicates that the system has 2 eigenvalues with positive real parts, and the equilibrium is likely unstable (or a saddle).

We can simulate the solutions to this model using MATLAB's function `dde23()` (Appendix, figure ii). We assumed values of zero for each function for $t \leq 0$. Figure 3 shows the results of the solutions for $k = 1$ and $k = 4$. Note that different historical values for the system resulted in similar long term behavior, indicating that the system is robust.

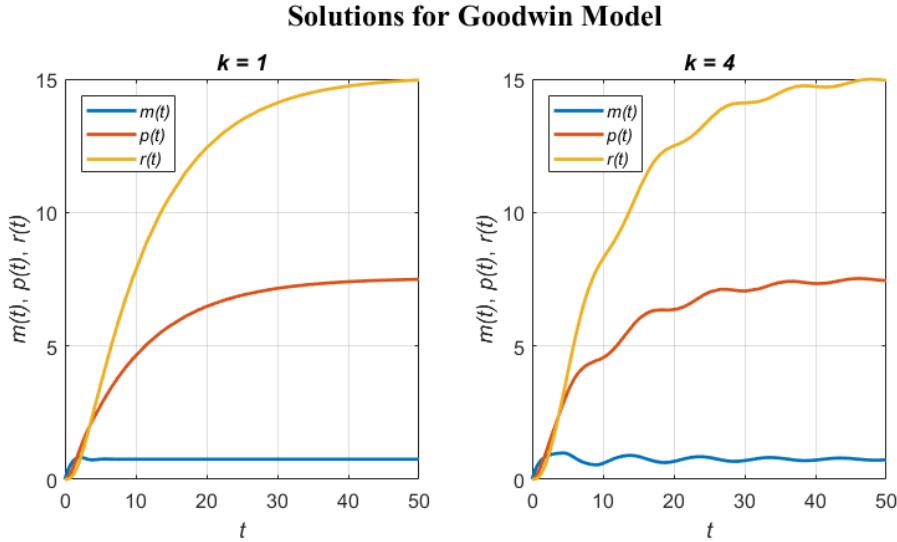


Figure 3 Differences Between $k = 1$ and $k = 4$

We can see here that the $k = 1$ curves reach an equilibrium around $t = 50$, and level off to smooth curves, indicating a stable equilibria. On the other hand, the $k = 4$ curves oscillate about the equilibria for some time before smoothing out.

Hopf Bifurcation

We can find a Hopf bifurcation in this system by substituting $\lambda = i\omega$ into our characteristic equation, then setting both the real and imaginary parts equal to zero. MATLAB's `vpasolve()` function was used to find the Hopf bifurcation values of $\omega = 0.28680$ and $k = 3.72894$.

Appendix

```
1  function delay_rec(k,p,q)
2
3  % Maps a rectangle bound by the lines x = [0,p] and y = [-q,q]
4  % k = time delay value
5  % p = x_max
6  % q = y_max
7
8  h=p/100;
9  l=q/50;
10
11 x = zeros(401,1);
12 y = x;
13 a = x;
14 b = x;
15
16 x(1) = 0;
17 y(1) = q;
18 a(1) = 3.0*x(1)*y(1)^2-0.182131947240132*exp(-k*x(1))*(cos(-y(1)*k))-...
19 0.65*x(1)-1.6*x(1)^2-1.0*x(1)^3+1.6*y(1)^2- 0.05;
20 b(1) = -y(1)*0.65-x(1)*y(1)*3.2-0.182131947240132*exp(-k*x(1))...
21 *sin(-y(1)*k)-x(1)^2*y(1)*3.0+y(1)^3*1.0;
22
23 for i=2:401
24
25 if i < 102
26     x(i) = 0;
27     y(i) = y(i-1)-l;
28 elseif i < 202
29     x(i) = x(i-1)+h;
30     y(i) = -q;
31 elseif i < 302
32     x(i) = p;
33     y(i) = y(i-1)+l;
34 elseif i < 402
35     x(i) = x(i-1)-h;
36     y(i) = q;
37 end
38 a(i) = 3.0*x(i)*y(i)^2-0.182131947240132*exp(-k*x(i))*...
39 (cos(-y(i)*k))-0.65*x(i)-1.6*x(i)^2-1.0*x(i)^3+1.6*y(i)^2-0.05;
40 b(i) = -y(i)*0.65-x(i)*y(i)*3.2-0.182131947240132*exp(-k*x(i))*...
41 sin(-y(i)*k)- x(i)^2*y(i)*3.0+y(i)^3*1.0;
42 end
43
44 plot(a(1:101),b(1:101)); hold on
45 plot(a(101:201),b(101:201));
46 plot(a(201:301),b(201:301));
47 plot(a(301:401),b(301:401));
```

Figure i MATLAB Code for Transformation via Characteristic Equation

```

1  function v = lags( t,y,z )
2
3 -    b1 = 1;
4 -    b2 = 0.1;
5 -    b3 = 0.5;
6
7 -    ylag1 = z(:,1);
8
9 -    v = zeros(3,1);
10 -   v(1) = 1/(1+ylag1(1)^4)-b1*y(1);
11 -   v(2) = y(1)-b2*y(2);
12 -   v(3) = y(2)-b3*y(3);
13
14 - end
1      % Delay Differential Equation Solutions for Goodwin Model
2
3 - history = zeros(3,1);
4
5 - subplot(1,2,1)
6 - k=1;
7 - sol = dde23('lags',k,history,[0,50]);
8 - plot(sol.x,sol.y)
9 - grid on
10 - legend('{itm(t)}','{itp(t)}','{itr(t)}');
11 - title('{itk = 1}');
12 - xlabel('{itt}');
13 - ylabel('{itm(t)}, {itp(t)}, {itr(t)}');
14
15 - subplot(1,2,2)
16 - k=4;
17 - sol = dde23('lags',k,history,[0,50]);
18 - plot(sol.x,sol.y)
19 - grid on
20 - legend('{itm(t)}','{itp(t)}','{itr(t)}');
21 - title('{itk = 4}');
22 - xlabel('{itt}');
23 - ylabel('{itm(t)}, {itp(t)}, {itr(t)}');

```

Figure ii MATLAB Code for *dde23()* Execution