Physics 2G03 Term Project
The Goodwin Oscillator Model of A Simple Negative Feedback Loop In Gene Expression

Name: Tessa Klettl
ID#: 1234408
Date of Submission: Wednesday December 16th 2015

Introduction

The expression of genes in a cell is governed by an incredibly complex system of regulatory frameworks. The two stages in protein production can be broadly described as transcription and translation, and these steps are tightly controlled at every stage by regulatory mechanisms. Transcription is the “reading” of the information encoded in DNA; this information is stored in a single-stranded nucleic acid sequence called messenger RNA, or mRNA, which then leaves the nucleus. In the translation step, specialised proteins called ribosomes use the information carried by mRNA to assemble proteins one amino acid at a time. When all the mRNA has been read and processed, the protein sequence is complete.

Transcription or translation can be regulated in a host of ways. Some genes are constitutively expressed; this means that unless they receive a signal to “turn off”, they are constantly being transcribed into mRNA. Other genes are not transcribed unless they receive a signal to “turn on”. These signals can come in the form of proteins that bind to the DNA, called transcription factors (TFs) or they can be other regions of DNA. Alternately, a protein signal may interact with one of the enzymes required for transcription, and thereby affect transcription indirectly. Translation of mRNA into a protein sequence is mostly regulated by proteins called enzymes, and small molecules called second messengers. These can enhance or inhibit the necessary translation reactions. Of course, indirect regulation of translation can occur when the transcription of genes for these enzymes is regulated. These mechanisms aren't necessarily like an on/off switch either; many regulatory elements work to amplify or dampen transcription and/or translation.

Dr Brian Goodwin was a Canadian scientist with a background in both mathematics and biology. By combining these two areas of expertise, he helped to found the field of theoretical biology. In 1965 he published a paper detailing a model that came to be know as the Goodwin oscillator.[1] It uses a series of ordinary differential equations (ODEs) to model a simple negative feedback loop; a gene codes for mRNA, which is translated into an enzyme, which activates an inhibitor protein. This activated inhibitor then represses expression of the gene at the beginning of the pathway by binding to and inactivating the TF that activates transcription. Essentially, the expression of the gene inhibits the gene's expression. This was a landmark model because the idea that regulation of genetic transcription is responsible for levels of proteins in the cell was still in its infancy, and an oscillatory model of genetic regulation would describe the periodicity of cell processes such as cellular “clocks” that had been observed experimentally. The Goodwin oscillator can be found below:

$$\frac{dX}{dt} = \frac{v_0}{1+(Z/K_m)^p} - k_1 X$$
$$\frac{dY}{dt} = v_1 X - k_2 Y \qquad (1)$$
$$\frac{dZ}{dt} = v_2 Y - k_3 Z$$

Where X = concentration of mRNA, Y = concentration of enzyme, Z = concentration of inhibitor, and d/dt is the derivative of these with respect to time. The k_0, k_1 and k_2 parameters represent the rates of degredation of their respective components, and the v_0, v_1 and v_2 parameters are the rates of transcription, translation and catalysis respectively.[2] This takes into account the reality that the cellular environment is a constant balancing act, in which components are simultaneously being broken down and manufactured. Where the equilibrium in this tug-of-war lies determines the overall effect on the cell and its behaviour.

Km is the Michaelis constant, a measure of the strength of the bond between the inhibitor and the TF; the greater the strength, the smaller the Km.[3] Therefore, a high affinity between the two leads to a larger denominator in the first ODE, and a smaller change in concentration of mRNA.

The final parameter is P, the Hill coefficient.[2] This is a measure of the cooperativity of the inhibitor's repression. Cooperativity is an interesting behaviour that some regulatory proteins, such as inhibitors, can display. This is when the initial binding between inhibitor and TF causes a change in the three-dimensional structure of the TF in such a way that the activation energy for subsequent inhibitors to bind to that TF is greatly reduced. An example of cooperative behaviour is the binding of oxygen to haemoglobin, which is capable of carrying 4 molecules of oxygen. When the first oxygen binds, it changes the shape of haemoglobin so that the remaining 3 molecules bind much more readily.[3] In this case, it is a beneficial system that increases oxygen saturation in the lungs. In Goodwin's oscillator, the Hill coefficient is a measure of this kind of cooperativity. If it is $P > 1$, cooperativity is positive, which is the case for the haemoglobin example: the oxygen *increases* haemoglobin's affinity for further oxygen. If $P < 1$, cooperativity is negative, which means that the binding of one molecule *decreases* the protein's affinity for binding more of that molecule. If $P = 1$, then binding is independent. In 1968, Griffith proved that for $P > 8$ and critical small values of k_n, a single limit cycle for each set of parameters will result, whereas for $P \leq 8$, oscillations are degenerate.[4] In essence, this means that small disturbances cause the system to return to an oscillation with the same amplitude around the same point, as long as the Hill coefficient $P > 8$ and the degredation constants are small enough.

This project aimed to create a computer simulation for system (*1*), and use the simulation to explore the effects that changing parameters would have on the behaviour of the system.

Methods

To find solutions for the set of ODEs in (*1*), the classical Runge-Kutta method was used. This method results in an approximation of the solution with an accuracy of $O(\delta t^5)$. Given timesteps for which to solve, the method uses the known derivatives of functions X, Y & Z as well as initial conditions to give solutions for X, Y & Z at various time points. The equations follow below.

Let $\frac{dX}{dt} = f(t, X)$, and let $X(t_0) = X_0$, and let $h = \delta t$, and let $n =$ iteration number

$$
\begin{aligned}
X_{n+1} &= X_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k4) \\
k_1 &= f(t_n, X_n) \\
k_2 &= f\left(t_n + \frac{h}{2},\ X_n + \frac{h}{2}k_1\right) \\
k_3 &= f\left(t_n + \frac{h}{2},\ X_n + \frac{h}{2}k_2\right) \\
k_4 &= f(t_n + h,\ y_n + hk_3)
\end{aligned}
\qquad (2)
$$

In order to apply the equations in (*2*), the program MyProject was written. All relevant files are provided in Appendix 1. Two functions were written, DeriveConcs and RungeKutta. DeriveConcs provides the definition of the rates of change of X, Y & Z as given in (*1*). RungeKutta calls DeriveConcs for use in the equations in (2) to create an array containing the concentrations of mRNA, enzyme, inhibitor and the time for which those concentrations are calculated. The main program in GoodwinMain.c scans in input for the initial concentrations and the desired timestep, calls RungeKutta to evaluate at those conditions, and then prints the returned array. The destination of the printed output is specified on the command line. For example, in Unix, command (*3*) stores the output in the file `output.dat`, which can then be called with the plotting function `plot.py` in Appendix 2.

```
./MyProject > output.dat
```
(*3*)

All parameters that may be changed can be accessed in the header file Goodwin.h as definitions. One can change the initial concentration, the rate of creation, or the rate of degredation of X, Y or Z. The Hill coefficient and Km can also be adjusted, as well as the number and size of timesteps. All one need do is change the desired value in the final column after the appropriate definition, save the header file, and remake MyProject.

Results

Verification and Validation

First, to verify that MyProject ran smoothly and gave expected results, a set of control values for the parameters were established. These are physically reasonable numbers that are expected to produce the oscillation, according to Griffith's analysis,[4] and are reported in Table 1. Additionally, the starting concentrations of all components were set to be 0.

	k_1	k_2	k_3	v_0	v_1	v_2	Km	P
Control Value	0.1	0.1	0.1	0.5	0.5	0.5	0.5	8.1

Table 1: Control values for the program parameters.

To verify that (*1*) was accurately modeled with MyProject, it was run with the control parameters and the output was plotted and is shown in Figure 1.

As expected, the concentrations of each component settle into an oscillation about a value, indicated as the mean of the last half of the timesteps. To ensure that this behaviour agrees with Griffith's analysis, P values of 7.9 and 8.0 were tested. Both of these conditions were expected to cause a degeneration of the oscillation, which would be reflected in the amplitude of the oscillations decreasing over time. To investigate this, I chose to increase the time scale and focus on the activity of one component, the inhibitor, so that changes in amplitude could be visualised more easily. The results were plotted in Figures 2, 3 and 4.

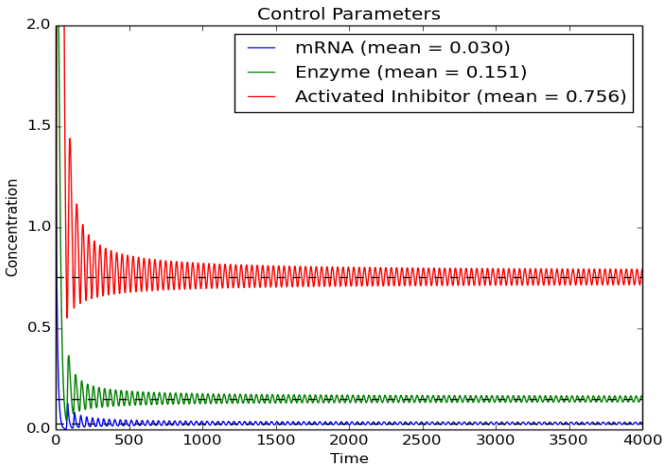

Figure 1. MyProject evaluated at control parameters. The equilibrium point is the mean, calculated over the final half of the timesteps. Axes are in arbitrary units, au.

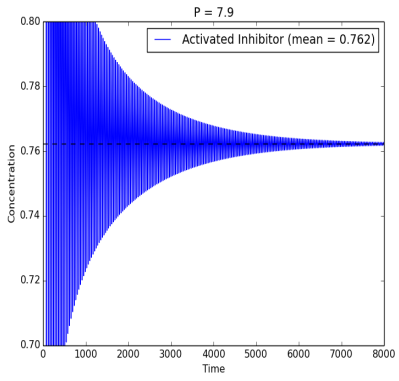

Figure 2. Inhibitor concentration vs time, in arbitrary units, when P = 7.9. Equilibrium of degenerate oscillations approximated as the mean, calculated using 6000≤t≤8000.

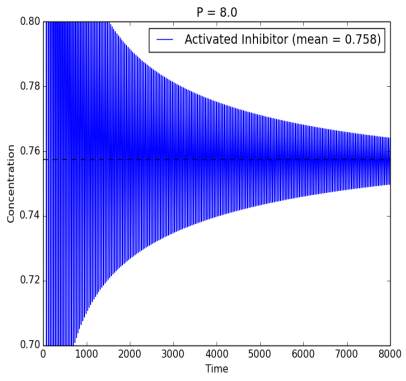

Figure 3. Inhibitor concentration vs time, in arbitrary units, when P = 8.0. Equilibrium of degenerate oscillations approximated as the mean, calculated using 6000≤t≤8000.

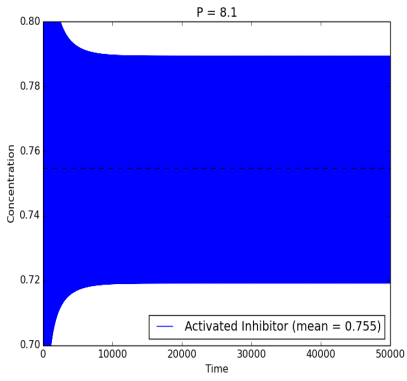

Figure 4. Inhibitor concentration vs time, in arbitrary units, when P = 8.1. Equilibrium of degenerate oscillations approximated as the mean, calculated using 37500≤t≤50000.

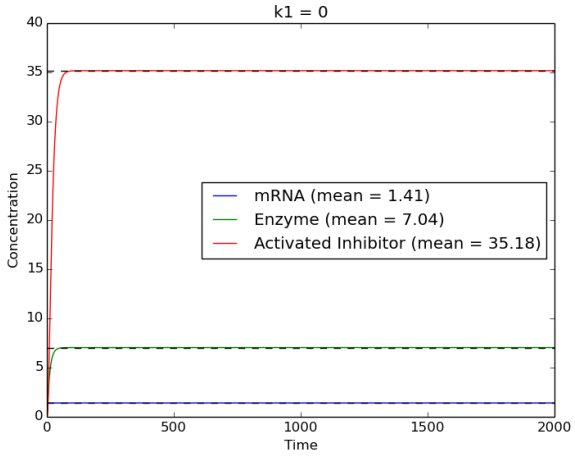

Figure 4. The concentration vs time plot in arbitrary units for when k1 = 0.The time limit behaviour (max/min) was approximated as the mean, calculated using 1000≤t≤2000.

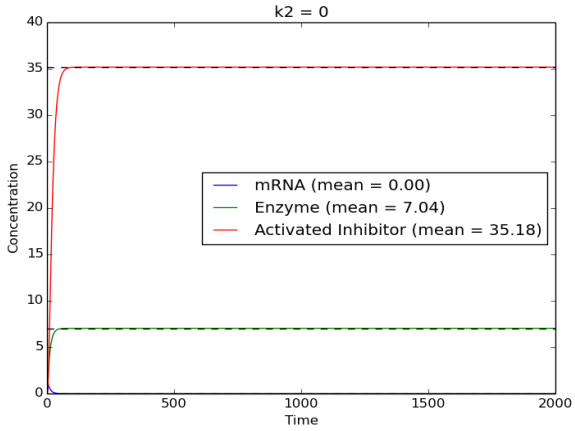

Figure 5. The concentration vs time plot in arbitrary units for when k2 = 0. The time limit behaviour (max/min) was approximated as the mean, calculated using the 1000≤t≤2000.

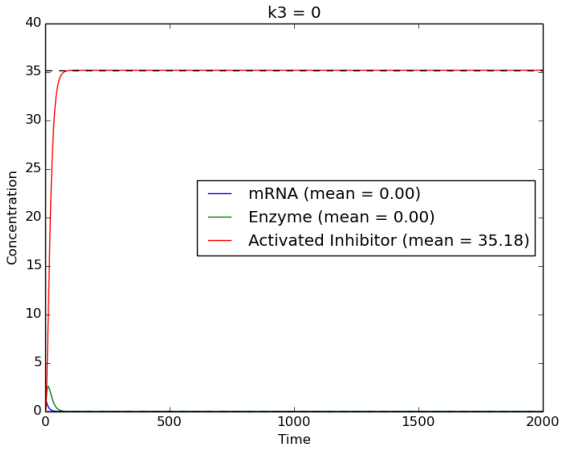

Figure 6. The concentration vs time plot in arbitrary units for when k3 = 0. The time limit behaviour (max/min) was approximated as the mean, calculated using 1000≤t≤2000.

These results conform fully to the oscillation condition that P must be greater than 8. Even with the time scale increased from 8000 units to 50,000 units, no degeneration of the oscillation can be observed when P > 8.0. However, when P = 8.0 and P = 7.9, the oscillations can clearly be seen collapsing about the equilibrium point. This disturbance can also be noted as small changes in the mean over the final quarter, which is a crude approximation for the equilibrium point. An interesting relationship can be seen as well; the greater the deviation from 0.755, the quicker the oscillations collapse, as the further away from the oscillation conditon P > 8 the system grows.

To further validate that the model behaves in a manner consistent with physical biology, each degredation constant was set to 0 in turn. The plots of each outcome are found in Figures 4, 5 and 6. If k1 = 0, we would expect that each component would increase to a maximum value and maintain it; if mRNA is not being degraded while still being formed, the translation and catalysis steps would proceed unhindered to a maximum that is dependent on the inhibitor parameters. When k2=0, we would expect to see mRNA decrease to 0, while enzyme and inhibitor increase to a maximum; with no degredation of enzyme, the initial mRNA transcription before the inhibitor begins acting results in enough translation to keep the levels of enzyme and inhibitor high. Finally, when k3 = 0, we would expect to see declines in both mRNA and enzyme, while inhibitor reaches a maximum.

Figures 3-6 show precisely the expected patterns, supporting that equation (*3*) has been appropriately modeled and is describing a real physical system. Another test was performed to verify that the model produced physically reasonable results. In this case, each formation rate constant was set to 0 in turn. When v_0=0, we would expect to find no increases in any concentration, because transcription, which is the rate of which v_0 represents, is the first step in this 3 step control system, where each subsequent component requires all prior components to be able to begin forming. If v_1 = 0, that indicates that the rate of translation is 0, therefore we would expect to see mRNA increase to a maximum value, and all other components remain 0. The third case, when v_2=0, represents the catalysis of the inhibitor protein. Therefore, inhibitor levels should stay at 0, while mRNA and enzyme levels increase to a maximum. The results are shown in Figures 7, 8 & 9.

Once again, the results follow our predictions of concentration behaviours. The effects of the k and v parameters appear to be in good agreement with their corresponding physical rate constants, and therefore support the validity of this model as a feedback loop mechanism.

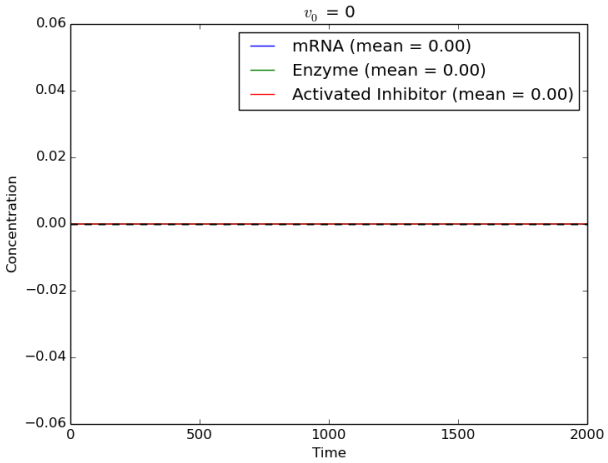

Figure 7. The concentration vs time plot in arbitrary units for when v_0 = 0. The time limit behaviour (max/min) was approximated as the mean, calculated using 1000≤t≤2000.

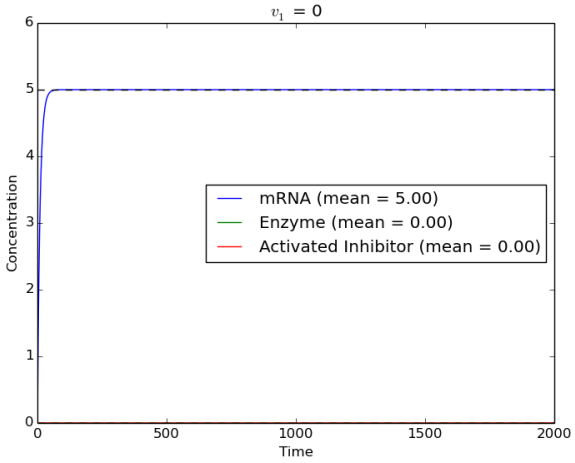

Figure 8. The concentration vs time plot in arbitrary units for when v_1 = 0. The time limit behaviour (max/min) was approximated as the mean, calculated using 1000≤t≤2000.

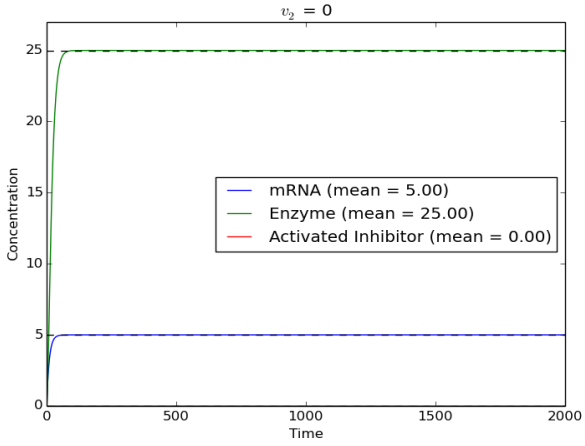

Figure 9. The concentration vs time plot in arbitrary units for when v_2 = 0. The time limit behaviour (max/min) was approximated as the mean, calculated using 1000≤t≤2000.

Experimentation

It would be interesting to see how changing the initial conditions changes the resulting oscillatory behaviour, while still keeping P=8.1 as in the initial conditions. To investigate possible effects on the equilibrium point, amplitude, or maximum concentrations, this was implemented using MyProject, and the results plotted in Figures 11, 12 and 13. The control parameters were re-plotted, using the same time scale to make visual inspection easier.

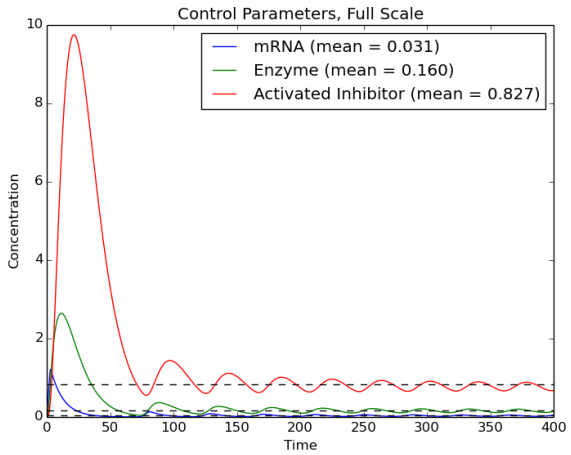

Figure 11. Control parameters showing full peaks, in concentration vs time (arb). The equilibrium point was approximated with the mean, calculated between 300≤t≤400.

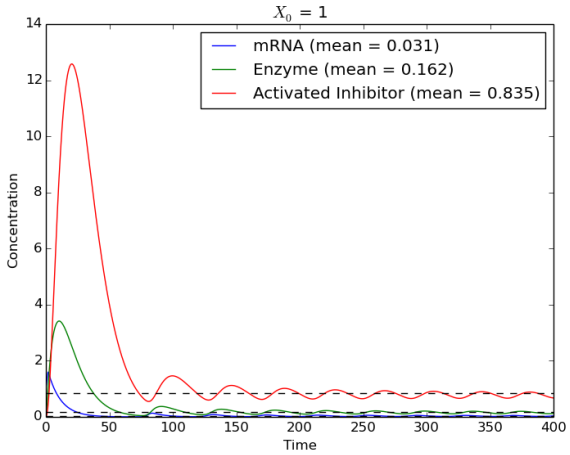

Figure 10. Function of concentration vs time in arbitrary units when X_0 = 1. The equilibrium point was approximated with the mean, calculated between 300≤t≤400.

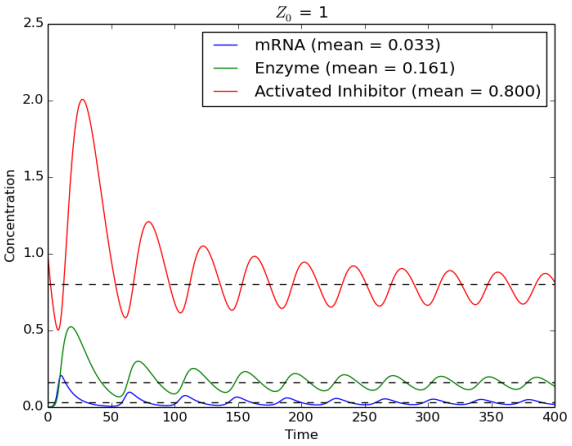

Figure 12. Function of concentration vs time in arbitrary units when Z_0 = 1. The equilibrium point was approximated with the mean, calculated between 300≤t≤400.

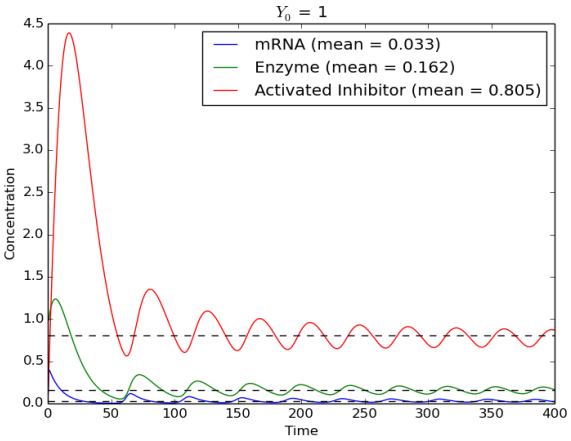

Figure 13. Function of concentration vs time in arbitrary units when Y_0 = 1. The equilibrium point was approximated with the mean, calculated between 300≤t≤400.

These indicate that the chief effect of changing intial parameters in isolation is not changing the period or the the behaviour towards infinity. While there are small differences in the mean, these were tested with larger timescales and found to be a symptom of the small timescale used, rather than an indication of a change in the equilibrium point. What is affected are the amplitudes of the initial peaks, or in other words, the size of the initial perturbation. Additional testing revealed that varying the initial

concentrations in tandem had the same effect: the peaks of initial activity were different, but all systems returned to a stable oscillation about the same equilibrium points, with apparently equal periods and amplitudes. In a cell, this means that the negative feedback mechanism is a self-regulator that is able to respond to fluctuations in concentrations of any of the components, and return them to stable periodic fluctuations with the same properties.

The next avenue of investigation was the effect of varying the degradation constant. Experimentation showed that there was a critical ratio of *P/Km* that resulted in the ability to manipulate the degradation constants to a greater degree without dampening the oscillations. A search through the literature suggested the new control values of $Km = 1$ and $P = 10$ to test this.[4,5] Changing these values while keeping the other parameter controls from Table 1 constant confirmed that a limit cycle of the same overall form as Figure 1 was once again produced. The new equilibrium points for mRNA, enzyme and inhibitor concentration were found to be 0.071, 0.354 and 1.771 respectively. It was interesting to note these increases; the initial *P/Km* ratio was 16.5, while the new ratio was 10. Recall from the introduction that Km is inversely proportional to TF-inhibitor affinity, and that P is proportional to inhibitor cooperativity. The observation that that the smaller *P/Km* ratio results in higher concentrations of pathway components makes sense, as low affinity, even with higher cooperativity, will result in decreased transcription inhibition. The *P* value as an exponent, however, suggests that there is a critical ratio beyond which the degree of cooperativity can overcome low affinity, and this would be an interesting avenue for further testing.

Values from $0.04 < k_n < 2.4$ were tested with the new control parameters and confirmed to produce limit cycles, and some clear patterns emerged. When all three degredation constants were increased simultaneously, that increasing the degradation constants increased the values of the equilibrium points and the amplitude of the oscillations, while decreasing the oscillation period for all concentrations. However, when just one of three degredation constants was increased, a different pattern emerged. Representative plots are shown in Figures 14-17.

In this pathway, the concentrations are interdependent. Z depends on Y, which depends on X. The negative feedback occurs because X depends on Z. Interestingly, Figures 15-18 seem to show that degradation impacts travel linearly along the pathway. When the degradation of one component is increased, the components that depend on it show decreased concentration, as given by the equilibrium point. However, the component as well as those before it in the pathway show an increased concentration, when compared to the new control means calculated when $P = 10$ and $Km = 1$.

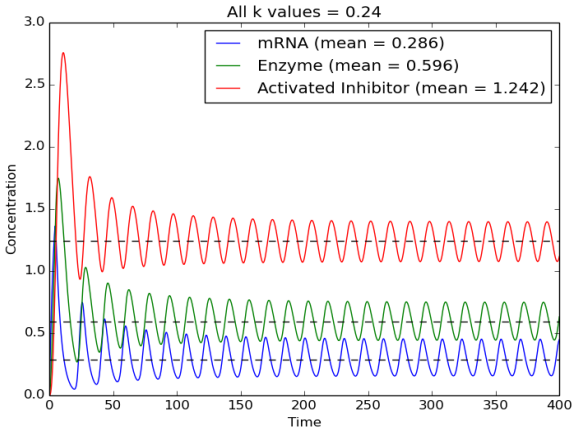

Figure 14. Plot of concentration vs time in arbitrary units when all degredation constants = 0.24. The equilibrium point was approximated with the mean, calculated between 300≤t≤400.

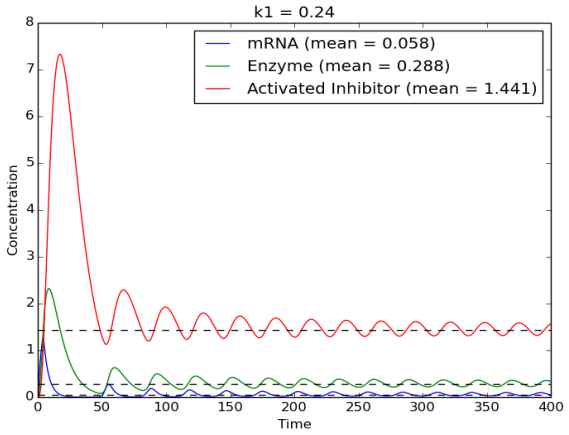

Figure 15. Plot of concentration vs time in arbitrary units when k1 = 0.24. The equilibrium point was approximated with the mean, calculated between 300≤t≤400.

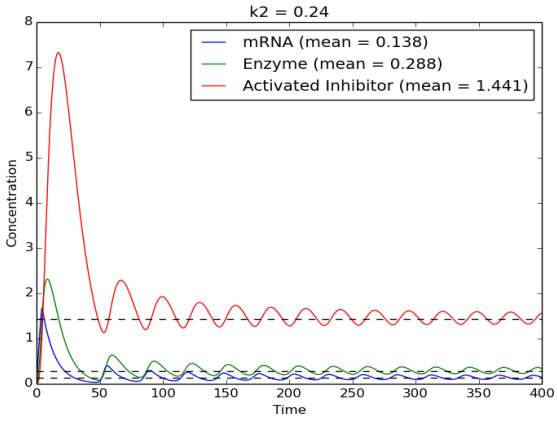

Figure 16. Plot of concentration vs time in arbitrary units when k2 = 0.24. The equilibrium point was approximated with the mean, calculated between 300≤t≤400.

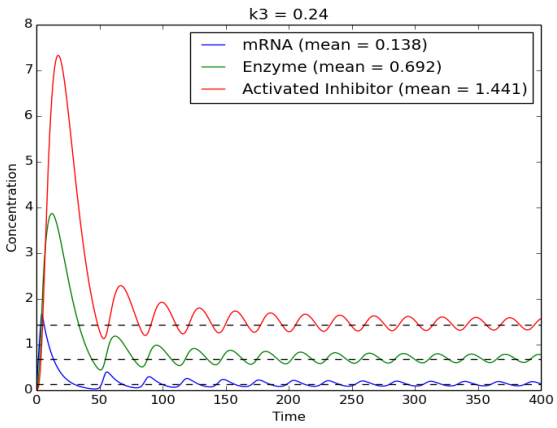

Figure 17. Plot of concentration vs time in arbitrary units when k3 = 0.24. The equilibrium point was approximated with the mean, calculated between 300≤t≤400.

Additionally, when a single component is degraded more quickly, the period remains the same, independent of which component it is. However, when all three components degrade faster, the period is shorter, and the amplitude greater, than when any one component degrades by that same linear coefficient.

In a cellular environment, these effects make sense. There is a necessary lag in the peak concentrations of mRNA, enzyme and inhibitor, because each component must be manufactured before the next can begin to be created. This is evident in all the figures in this report, as a phase shift in the oscillations. Increasing the rate of change of one component will cause its concentration vs time plot to have a steeper slope, and due to this lag, a larger difference in concentration will occur before the feedback loop ripples through the pathway to begin to reverse the direction of change. The period is also shorter because the more drastic changes propagate faster through the pathway, as concentrations become high or low enough to change the centre of the tug-of-war referred to in the introduction.

Conclusion

The program MyProject, based on the classical Goodwin oscillator model of genetic self-repression, was tested in each of its parameters to verify that the program was functional, and that the Runge-Kutta integration method chosen was sufficiently accurate to produce expected results. After this was verified, MyProject was used to test the effects of changing initial concentrations of mRNA, the enzyme encoded by that mRNA, and the inhibitor activated by the enzyme. It was found that the feedback mechanism is robust enough to return concentrations to oscillations about the same equilibrium point, even with fluctuating initial conditions. This indicates that the simple regulatory mechanism modeled is self-sustaining, at least within the parameters studied. Additionally, the effect of varying the degradation constant k_n for each component was studied. It was found that within the parameters studied, values between 0.04 and 0.24 produced stable oscillations. It was also found that the period of the oscillations decreases proportionally to k_n, while the amplitude and equilibrium point increase proportionally to this constant.

Future research avenues would include studying more variations among the other parameters, such as rates of formation, and adding features to the MyProject program such as modifications to the Goodwin model that have been made in the years since its initial publication to better reflect physical evidence. A criticism made at its publication was that the Hill coefficient of $P = 8$ was unrealistically high, and therefore that the system was not a useful model for real-world cell conditions.[4] Since Griffith's paper, it has been found that by replacing the linear degradation rate constants *k1*, *k2* and *k3* with Michaelian kinetics equations such as in (*4*), that limit cycles can be generated with P values greater than 4, which is much more realistic in scope.[6]

$$\text{In} \quad \frac{dY}{dt} = v_1 X - k_2 Y \quad ,$$

$$k_2 Y \quad \text{is replaced with} \quad \frac{k_2 Y}{K_D + Y} \quad \text{to give:} \qquad (4)$$

$$\frac{dy}{dt} = v_1 X - \frac{k_2 Y}{K_D + Y}$$

References

1. Goodwin BC. Oscillatory behavior in enzymatic control processes. *Adv Enzyme Regul*. 1965;3:425-437. doi:10.1016/0065-2571(65)90067-1.
2. Wang Y, Hori Y, Hara S, Doyle FJ. The collective oscillation period of inter-coupled Goodwin oscillators. March 2012.
3. Lodish H, Berk A, Kaiser CA, et al. *Molecular Cell Biology*. 7th ed. New York: W.H. Freeman; 2013.
4. Griffith JS. Mathematics of cellular control processes I. Negative feedback to one gene. *J Theor Biol*. 1968;20(2):202-208. doi:10.1016/0022-5193(68)90189-6.
5. Gonze D, Abou-Jaoudé W. The Goodwin model: behind the Hill function. *PLoS One*. 2013;8(8):e69573. doi:10.1371/journal.pone.0069573.
6. Gonze D, Bernard S, Waltermann C, Kramer A, Herzel H. Spontaneous synchronization of coupled circadian oscillators. *Biophys J*. 2005;89(1):120-129. doi:10.1529/biophysj.104.058388.

Appendix 1

Goodwin.h

```
typedef struct { float X,Y,Z; } conc;

conc DeriveConcs( conc A );

conc RungeKutta( conc A, float h);

// Initial concentrations of X, Y, Z (mRNA conc., enzyme conc., inhibitor conc.)
#define X0 0
#define Y0 0
#define Z0 0

// Number of timesteps (number of iterations)
#define NTS 20000

// Size of timesteps.
#define TS 0.1

// Rate of mRNA production (transcription: input > 0)
#define V0 0.5
// Rate of mRNA degredation (input > 0)
#define K1 0.1
// Rate of enzyme production (translation: input > 0)
#define V1 0.5
// Rate of enzyme degredation (input > 0)
#define K2 0.1
// Rate of inhibitor activation (catalysis: input > 0)
#define V2 0.5
// Rate of inhibitor degredation (input > 0)
#define K3 0.24

// Hill constant. P > 8 leads to limit cycle.
#define P 10.0

// Michaelis constant. Affinity of inhibitor for gene. Input > 0.
#define Km 1.0
```

RungeKutta.c

```c
#include <stdio.h>
#include <math.h>
#include "Goodwin.h"

/*Given concs(t=0), derivatives of X,Y & Z, find concs(t=i)*/
conc RungeKutta(conc A,float h){
  conc result,k1,k2,k3,k4,temp;
  k1 = DeriveConcs(A);
  temp.X = A.X + h/2*k1.X;
  temp.Y = A.Y + h/2*k1.Y;
  temp.Z = A.Z + h/2*k1.Z;
  k2 = DeriveConcs(temp);
  temp.X = A.X + h/2*k2.X;
  temp.Y = A.Y + h/2*k2.Y;
  temp.Z = A.Z + h/2*k2.Z;
  k3 = DeriveConcs(temp);
  temp.X = A.X + h*k3.X;
  temp.Y = A.Y + h*k3.Y;
  temp.Z = A.Z + h*k3.Z;
  k4 = DeriveConcs(temp);

  result.X = A.X + (k1.X + 2*k2.X + 2*k3.X + k4.X)*h/6;
  result.Y = A.Y + (k1.Y + 2*k2.Y + 2*k3.Y + k4.Y)*h/6;
  result.Z = A.Z + (k1.Z + 2*k2.Z + 2*k3.Z + k4.Z)*h/6;

return result;
}
```

DeriveConcs.c

```c
#include <stdio.h>
#include <math.h>
#include "Goodwin.h"

conc DeriveConcs(conc A){
    conc result;
    result.X = V0/(1+pow((A.Z / Km),P)) - K1*A.X;
    result.Y = V1*A.X - K2*A.Y;
    result.Z = V2*A.Y - K3*A.Z;
return result; // Make this return something that is a conc
}
```

GoodwinMain.c

```c
#include <stdio.h>
#include <math.h>
#include "Goodwin.h"

int main(){

const int maxn = NTS;
int n;
float h;    // h = timestep
conc result[maxn];

result[0].X = X0;
result[0].Y = Y0;
result[0].Z = Z0;
h = TS;

// Check timestep input
if(h<0){
        printf("The timestep size %g is not valid, must be >0.\n",h);
        return (-1);
}

// Check initial concentration input
if(result[0].X<0){
        printf("The mRNA concentration %g is not valid, must be >0.\n",result[0].X);
        return (-1);
}
if(result[0].Y<0){
        printf("The enzyme concentration %g is not valid, must be >0.\n",result[0].Y);
        return (-1);
}
if(result[0].Z<0){
        printf("The inhibitor concentration %g is not valid, must be >0.\n",result[0].Z);
        return (-1);
}

// Check rate input
if(V0 < 0 || V1 < 0 ||  V2 < 0 || K1 < 0 || K2 < 0 || K3 < 0){
        printf("All rates must be non-negative.\n");
        return (-1);
}

// Check Km
if(Km<0){
        printf("The Km value %g is not valid, must be >0.\n",Km);
        return (-1);
}

// All inputs are good, perform the calcultaions and print output
for(n=1;n<=maxn;n++){
    result[n] = RungeKutta(result[n-1],h);
        printf("%f %f %f %f\n",result[n].X, result[n].Y, result[n].Z, n*h);
}


return 0;
}
```

Appendix 2

plot.py – A representative sample

```
#!/usr/bin/python

from matplotlib.pyplot import *
from numpy import *

output = genfromtxt('output.dat')
xval = output[:,0]
yval = output[:,1]
zval = output[:,2]
time = output[:,3]

xmean = mean(xval[xval.size/4:])
ymean = mean(yval[yval.size/4:])
zmean = mean(zval[zval.size/4:])
plot(time,xval,label="mRNA (mean = %4.3f)" % xmean)
plot(time,yval,label="Enzyme (mean = %4.3f)" % ymean)
plot(time,zval,label="Activated Inhibitor (mean = %4.3f)" % zmean)

axhline(xmean, color='k', linestyle='--')
axhline(ymean, color='k', linestyle='--')
axhline(zmean, color='k', linestyle='--')
print xmean, ymean, zmean
xlabel('Time')
ylabel('Concentration')
title('k3 = 0.24')
#ylim(1.4,1.45)
xlim(0,400)

legend()
#legend(loc = 'center right')
#legend(loc = 'lower right')

#gcf().set_size_inches(25,5)
#show()
#savefig("k3_024")
```