ChBE 7770
Prelim 1
3/25/19
Franklin Gong

1.) (a)

Given the following 4 elementary steps and their rate constants

$$\begin{array}{l|c}
G_j + R_X^o \leftrightarrow \left(G_j\colon R_X\right)_C & k_+/k_- \qquad (1) \\
\left(G_j\colon R_X\right)_C \rightarrow \left(G_j\colon R_X\right)_O & k_I \qquad (2) \\
\left(G_j\colon R_X\right)_O \rightarrow R_X + G_j & k_A \qquad (3) \\
\left(G_j\colon R_X\right)_O \rightarrow m_i + R_X + G_j & k_{E,j} \qquad (4)
\end{array}$$

Where G_j is the concentration of free gene j, R_X^o is the concentration of free RNAP, $\left(G_j\colon R_X\right)_C$ is the concentration of the inactive RNAP:Gene j complex, and $\left(G_j\colon R_X\right)_O$ is the concentration of the active RNAP:Gene j complex, we can use a material balance around the closed and open forms of the RNAP:Gene j complex at steady state to find transient concentrations for these gene bound forms in terms of the rate constants and the concentrations of free RNAP and free gene j.

$$\frac{d}{dt}\left(G_j\colon R_X\right)_C = k_+\left(G_j\right)\left(R_X^o\right) - \left(k_- + k_I\right)\left(G_j\colon R_X\right)_C = 0 \text{ @SS}$$
$$\frac{d}{dt}\left(G_j\colon R_X\right)_O = k_I\left(G_j\colon R_X\right)_C - \left(k_A + k_{E,j}\right)\left(G_j\colon R_X\right)_O = 0 \text{ @SS}$$

$$\left(G_j\colon R_X\right)_C = \frac{k_+}{k_- + k_I}\left(G_j\right)\left(R_X^o\right)$$
$$\left(G_j\colon R_X\right)_C = K_{X,j}^{-1}\left(G_j\right)\left(R_X^o\right) \qquad (5)$$

$$\left(G_j\colon R_X\right)_O = \frac{k_I}{k_A + k_{E,j}}\left(G_j\colon R_X\right)_C$$
$$\left(G_j\colon R_X\right)_O = \tau_{X,j}^{-1}\left(G_j\colon R_X\right)_C \qquad (6)$$

Plugging in eqn. 5 to eqn. 6 gives eqn. 7 below.

$$\left(G_j\colon R_X\right)_O = K_{X,j}^{-1}\tau_{X,j}^{-1}\left(G_j\right)\left(R_X^o\right) \qquad (7)$$

The total RNAP balance is given in equation 8

$$R_{X,T} = R_X^o + \left(G_j\!:R_X\right)_C + \left(G_j\!:R_X\right)_O + \sum_{i=1,j}^{N}\{(G_i\!:R_X)_C + (G_i\!:R_X)_O\} \tag{8}$$

Where $R_{X,T}$ is the total amount of RNAP in the system and the summation is the amount of RNAP complexed to genes besides gene j. Plugging equations 5 and 7 into equation 8 gives an expression for the total amount of RNAP in terms of parameter ratios, gene concentrations, and free RNAP concentrations. Because the rate law equations and material balances apply generally to all the genes in the system, this is done for the summation as well.

$$R_{X,T} = R_X^o + K_{X,j}^{-1}\left(G_j\right)\left(R_X^o\right) + K_{X,j}^{-1}\tau_{X,j}^{-1}\left(G_j\right)\left(R_X^o\right) + \sum_{i=1,j}^{N}\left\{K_{X,i}^{-1}(G_i)\left(R_X^o\right) + K_{X,i}^{-1}\tau_{X,i}^{-1}(G_i)\left(R_X^o\right)\right\}$$

Solving for R_X^o

$$R_X^o = \frac{R_{X,T}}{1 + \left(K_{X,j}^{-1}\right)\left(G_j\right)\left(1 + \tau_{X,j}^{-1}\right) + \sum_{i=1,j}^{N}\{\left(K_{X,i}^{-1}\right)(G_i)\left(1 + \tau_{X,i}^{-1}\right)\}}$$

$$R_X^o = \frac{\left(R_{X,T}\right)\left(K_{X,j}\right)\left(\tau_{X,j}\right)}{\left(K_{X,j}\right)\left(\tau_{X,j}\right) + \left(K_{X,j}\right)\left(G_j\right)\left(1 + \tau_{X,j}\right) + \sum_{i=1,j}^{N}\left\{\frac{\left(K_{X,j}\right)\left(\tau_{X,j}\right)}{\left(K_{X,i}\right)\left(\tau_{X,i}\right)}(G_i)\left(1 + \tau_{X,i}\right)\right\}}$$

$$R_X^o = \frac{\left(R_{X,T}\right)\left(K_{X,j}\right)\left(\tau_{X,j}\right)}{\left(K_{X,j}\right)\left(\tau_{X,j}\right) + \left(K_{X,j}\right)\left(G_j\right)\left(1 + \tau_{X,j}\right) + \mathcal{E}_j} \tag{9}$$

Plugging eqn. 9 into eqn. 7 gives us an expression for the concentration of the open complex in terms of total RNAP. This can then be applied to our proportionality expression shown in eqn. 10 to give the final kinetic limit of transcription.

$$\left(G_j\!:R_X\right)_O = \frac{\left(R_{X,T}\right)\left(G_j\right)}{\left(K_{X,j}\right)\left(\tau_{X,j}\right) + \left(K_{X,j}\right)\left(G_j\right)\left(1 + \tau_{X,j}\right) + \mathcal{E}_j}$$

$$r_{X,j} = k_{E,j}\left(G_j\!:R_X\right)_O \tag{10}$$

$$\boxed{r_{X,j} = k_{E,j}R_{X,T}\left(\frac{\left(G_j\right)}{\left(K_{X,j}\right)\left(\tau_{X,j}\right) + \left(K_{X,j}\right)\left(G_j\right)\left(1 + \tau_{X,j}\right) + \mathcal{E}_j}\right)}$$

(b)
For a multigene system to approximate to a one gene system, $\mathcal{E}_j$ would have to be very low, which means that the ratio of RNAP bound to gene i vs gene j would have to be close to zero. For this to be the case, the concentration of gene i, G_i, in the system could be very low i.e. these genes aren't being expressed. Also, changes in the rate constants for the elementary binding could disfavor gene i being transcribed either by slow binding of RNAP to gene i, $k_+ \downarrow$, or RNAP failing to be active most of the time, $k_I \downarrow$. This could be the result of specific promotors in the system not being present or active inhibition of gene binding.

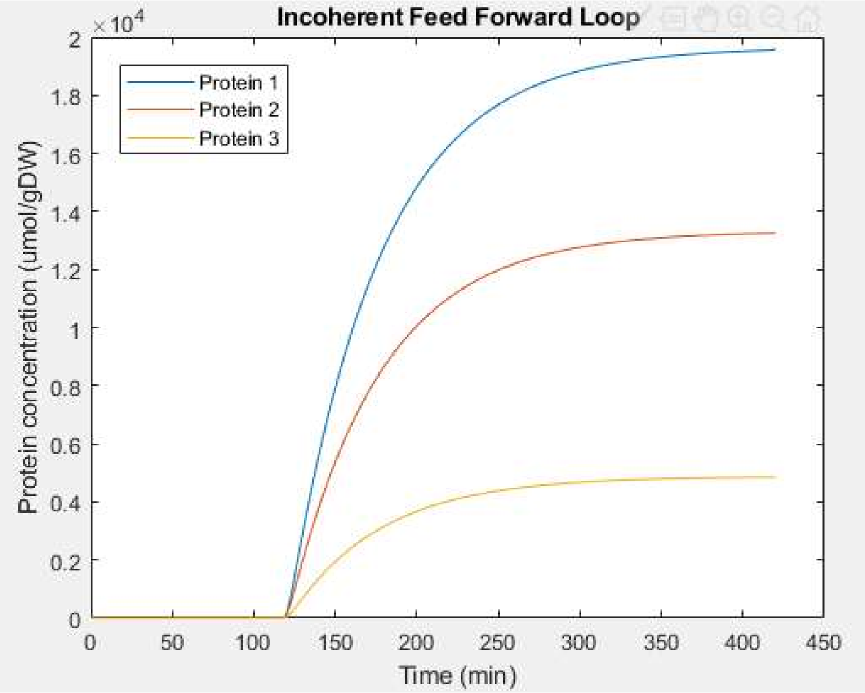
2.) (a)

The solution begins by deriving material balances around each of the 3 mRNA transcripts and the translated proteins themselves and using this to generate a system of differential equations which can be solved numerically to yield the final answer. This solution was done in MATLAB using the ODE45 equation solver.

This was done by first initializing an ODE system and defining the biophysical constants given by the prompt of the question. Each constant was given in terms of nmol/gDW for units of concentrations and minutes for units of time. The lengths of the gene transcripts and their complementary proteins were used to determine the time constants for the system. It was assumed that the rate of abortion was much less than the rate of elongation for both transcription and translation. Because the saturation constants and gene/ribosome concentrations were already given, we were then able to calculate the kinetic rates of transcription and translation and modified them with binding functions and weights based on the information provided in tables 1 and 2 in the prompt. In order to simulate activation, the weight and binding function for the respective interaction were both included in the numerator and denominator of the transcription rule. To simulate inhibition, the weight and binding function for the interaction were placed in the denominator only. The differential equation matrices were then set up and a new file for running the system was created.

In the new running file, the inducer concentrations over time were set up and the ode45 function was called. The plot of each of the protein concentrations vs time is shown in the figure below with steady state assumed to take ~60 minutes to reach, another 60 minutes for Phase 1 from 60 to 120 minutes, and the inducer added and maintained at 10 mM for another 300 minutes for Phase 2 from 120 to 420 minutes.

The parameters and systems are stored in the file “prelim1problem2system.m” and the running code is stored in the file “Prelim_1_Problem_2.m”.



(b)
Perturbations in both the positive and negative direction for each model parameter were used to determine the scaled state sensitivity coefficients as a function of time. To do this, a 10% increase and decrease was made to each model parameter and the partial derivative was approximated by using a central difference taken from average concentration values for each of the three proteins on three 20 minute windows, one during Phase one (60-80), one during the early Phase 2 (120-140), and one during the late Phase 2 (400-420). This approximate derivative was then scaled using a scaling term evaluated along the unperturbed trajectory over the same time frames.

Central difference scheme:

$$\frac{\partial x_i}{\partial p_j} = \frac{x(p + \Delta p) - x(p - \Delta p)}{2\Delta p}$$

This process was performed manually by storing the forward and reverse perturbations in separate arrays before calculating the coefficient array and manually storing that in a separate file ("Prelim1Problem2bStore.m").

(c)
Using SVD on the sensitivity arrays, it was the determined that the model species that was the most important was Protein 1 followed by protein 2 then protein 3. This SVD analysis was done by combining the sensitivity data from part b into 3 separate arrays using the trapezoidal rule to approximate the integral. Then the SVD function on MATLAB was used to simplify the data and generate the 2 orthogonal and 1 singular matrices. From the U matrix, the importance-order was

determined. This order makes sense as Protein 1 regulates the other proteins, meaning changes in P1 concentration will lead to rapid shifts in the concentrations of the other two. Next is P2 which regulates the expression of P3. Finally comes P3 which does not regulate any expression.

3.)
(a)
For G, RNAP, G*, NTP, mRNA, Pi, NMP, rib, rib*, AAtRNA, GTP, tRNA, GDP, protein, AA, ATP, AMP down the rows and for v1-v6 and b1-b9 along the columns in that order.

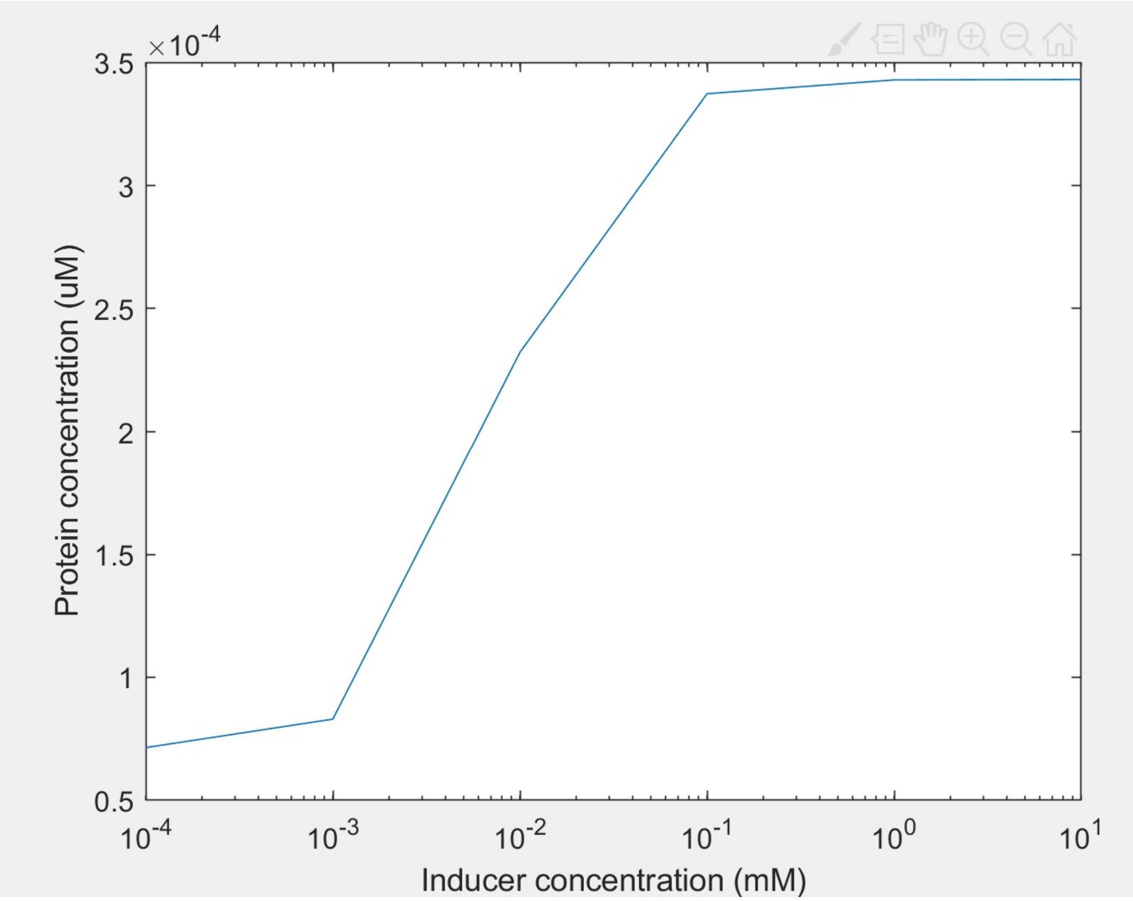
$$S = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -n & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2n & 0 & 0 & 2a & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & n & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & a & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

From the kinetic limit expressions,

$$r_x = k_{E,j}R_{X,T}\left(\frac{(G_j)}{(K_{X,j})(\tau_{X,j}) + (K_{X,j})(G_j)(1+\tau_{X,j})}\right)\left(\frac{W_1 + W_2\left(\frac{I^n}{I^n + K^n}\right)}{1 + W_1 + W_2\left(\frac{I^n}{I^n + K^n}\right)}\right)$$

$$r_L = k_{E,j}R_L\left(\frac{(m_j)}{(K_L)(\tau_L) + (K_L)(m_j)(1+\tau_L)}\right)$$

(b)
This system was solved at several different inducer concentrations using the linear programming function in MATLAB. The resulting semilog plot of this is shown in the figure below. The solution was obtained by first solving the kinetic rates of transcription at multiple inducer concentrations and using the mRNA degradation rate to find the steady-state mRNA concentration. This was then used to determine the kinetic rate of translation and these values were used as the bounds for the linear programming problem. All other reaction bounds were considered to be arbitrary for this reaction scheme. The solution file for this is stored in "Prelim1Problem3b.m"



(c)

The most sensitive and optimizable fluxes are those which are not fully known or characterized. Thus, any exchange fluxes that deal with species with “a” or “n” factors are the most sensitive. These species would be protein and phosphate which correspond to external fluxes b3 and b9.