Bistability in the lactose regulatory system of *E. Coli*: A discrete Markov chain approach to study the stochastically triggered induction

Akshay Rajhans

<rajhans@seas.upenn.edu>

A course project towards the ESE 680:003 course on Systems Biology Spring 2007

I. ABSTRACT:

The lactose regulatory metabolism observed in *Escherichia Coli* is one of the most extensively studied systems in the world of Systems Biology. Because of the positive feedback regulation present in this system, many interesting behaviors such as multiple equilibria (bistability) and limit cycles, hysteresis, and possibility of oscillations are observed. Various studies are being done on this system, right from the sixties, and various models have been proposed. Most of treated the concentrations of the reacting substances as the variables, and built ordinary differential equations as per the dynamics of the system. They all provided good theoretical foundation for analyzing the system. This was a purely deterministic approach to look at this system. However, if we are working in the domain of lower concentrations, the assumption that these concentrations are continuous variables is no longer valid.

A more recent school of thought proposed by Munsky and Khammash [2] in the Finite State Projection method treats the individual molecule counts as the discrete variables, on which to build the dynamics. Unlike [2], the purpose of this work is not to save the computation time for large populations and/or solve the dynamics analytically. The purpose of this work is indeed to understand the dynamics, if discrete approach is used.

The work by Julius et al [1] and others try to model the inherent stochasticity in the network. They propose a stochastic hybrid system to model the network, and also design a hybrid controller for the system.

As explainend in [1], the bistability behavior in the lac system consists of the two stable states, namely the low (uninduced) and the high (induced) states, in the lac system. Even if the dynamics are maintained such that the system stays in the uninduced state forever, the inherent stochasticity enables occasional molecules to jump into the induced state. The scope of this report is to study this behavior, namely the stochastically-driven induction during the bistable behavior. Since the uninduced state generally works in the domain of low concentrations, discrete modeling of the reacting chemicals as the individual molecule counts is a valid approach.

The following sections of this report describe a bit more in-depth background, the various methods used to understand the system, the results obtained and the related discussion.

II. INTRODUCTION:

The simplified model of the lac system as proposed in [1], and shown here in figure 1, would be a good starting point for this discussion. As seen in the diagram, *Thio-methyl galactosidase (TMG)* works as an inducer for the mRNA, which generates the genes that encode the enzymes β -galactosidase and permease. Permease in turn, enhances the inward flow of external TMG into the cell of E. Coli. After entering inside the cell, the

external TMG adds to the already present concentration of *TMG*. This completes the positive feedback loop of the system.



Figure 1: The lactose network with gratuitous external inducer (TMG), explained in [1]

The ordinary differential equations characterizing the dynamics of the lac system have been proposed in [1], and have been listed here as Equation 1. The variables M, B, T, P and T_e represent the molar concentrations of *mRNA*, β -galactosidase, Permease, TMG and external TMG respectively.

$$\frac{dM}{dt} = \alpha_M \frac{1 + K_1 (e^{-\mu \tau_M} T(t - \tau_M))^n}{K + K_1 (e^{-\mu \tau_M} T(t - \tau_M))^n} + \Gamma_0 - \tilde{\gamma}_M M, \quad (1a)$$

$$\frac{dB}{dt} = \alpha_B e^{-\mu\tau_B} M(t - \tau_B) - \tilde{\gamma}_B B, \qquad (1b)$$

$$\frac{dT}{dt} = \alpha_L P \frac{T_e}{K_{T_e} + T_e} - \beta_L P \frac{T}{K_{L_1} + T} - \tilde{\gamma}_L T, \qquad (1c)$$

$$\frac{dP}{dt} = \alpha_P e^{-\mu(\tau_P + \tau_B)} M(t - \tau_P - \tau_B) - \tilde{\gamma}_P P.$$
(1d)

Equation 1: The ODEs in [1] that model the dynamics of the lac system

These concentration equations need to be discretized i.e. converted to molecule count before deploying the discrete approach. The disctretization calculation can be explained as follows: There are 6.023×10^{23} molecules in one mole concentration of a substance. This equals Avogadro's number. These are the number of molecules in unit volume of the substance. The volume of a typical E. Coli cell is assumed to be 10^{16} liters. Further, the concentrations measured in the experiments done in [1] were in mM. So, the total product comes out to be 6.023×10^4 molecules equivalent to 1mM concentration, for this particular setting. The equation form for this conversion constant C_N, as given in [1] has been given here as Equation 2.

$$C_N = 10^{-16} \, 1.6.023 \cdot 10^{23} \frac{\text{molecules}}{\text{mole}} \cdot 10^{-3} \frac{\text{M}}{\text{mM}},$$
$$= 6.023 \cdot 10^4 \frac{\text{molecules}}{\text{mM}}.$$

Equation 2: The discretizing conversion constant C_N, [1]

Since the enzyme β -galactosidase does not play any role in the dynamics, and is merely a measure of the mRNA activity, it has not been considered. The equations (1a), (1c) and (1d) however, have been converted to the discrete molecule counts by using C_N.

For details about the meanings and the values of the other constants in equations (1a), (1c) and (1d), please refer to [1].

The rates obtained in the discrete counterparts of these equations were treated as if they were the rates of Poisson processes. In particular, the positive and the negative parts of the rates were separated and treated as if they signify different Poisson events. Hence, we have now six different Poisson processes, namely increase and decrease in *mRNA*, *TMG* and *Permease* molecule counts respectively.

III. METHODS APPLIED:

(a) Simulations using discrete Markov chain model:

As proposed in [2], each configuration of the molecule counts of each different reactant could be considered as a different state. From each such state, the system could possibly jump to six possible neighboring states, where the mRNA, or TMG or Permease molecule count differs from that of the state at hand by +/-1. The possible 'jumps' are nothing but the Poisson events occurring with the rates mentioned in section II above. From these Poisson rates, the probabilities of jumping into a particular neighboring state can be calculated as $(1 - e^{-\lambda_j t}) \cdot \left\{\prod_{i \neq j} (e^{-\lambda_i t})\right\}$, where, j is the index of the jump that actually

happens, given that all other indices *i* represent jumps that do not occur. If the time step *t* is chosen small enough, we can safely assume that the probability of two events (jumps) occurring within that small interval is negligibly small as compared to that of one event occurring.

The rates, and in turn, the probabilities of these jumps are state dependent. Hence, a different value of this six-tuple of the rates needs to be calculated for each state.

The number of molecules, and in turn, the possible number of states could be potentially infinite. Hence, we use the notion of finite state projection proposed in [2]. Here, the size of the state space under consideration would be the hypercube formed by the origin and the upper limits on the molecules of mRNA, TMG and Permease that could possibly be present in the uninduced state. This size was derived from the experimental results in [1], however, was kept somewhat flexible to get the exact idea about the behavior. All the remaining states are abstractly treated as the sink state.

Once the probability six-tuple is known for all the states, then a suitable starting state could be assigned, and different simulations could be run. The idea is to map the sum of all six probabilities and its difference from 1 as seven different segments on the number line between 0 and 1. During every time step, a random number with uniform distribution is generated. Depending upon which of the segments it lies on, the corresponding action is taken (i.e. one of the six possible jumps, or stay-where-you-are).

The Markov property assumption made here implies that these jumps are memoryless. Once you make a jump, the probability of the next jump is not affected by what you did before. It depends only on the state you are in.

(b) Solving without simulation, for the expected exit time:

If and when the size of the finite state projection is fixed, we have the fixed number of states (plus a sink state), say N. We also have the transition probabilities between the neighboring states. From these, we can generate the transition probability matrix of size (N x N). Each i^{th} row in this matrix could be made to correspond to the individual probabilities where of the states where it can go next. This is a standard practice in any machine learning algorithm that uses Markov chain model, and could be found in any good reference, for example [3]. Another practice is to assign the starting probabilities to the states. We can either assume a single starting state or can even assign a suitable probability distribution over the different states. Once the transition probability matrix and the starting probability vector are set up this way, then during next time step iteration, the probability distribution of being in that state is just a pre-multiplication by the transition probability matrix to the vector. This pre-multiplication by the transition probability matrix can be done iteratively until we get the desired distribution over the states.

In case of this setting, the probability of staying in the sink state could be assigned equal to 1, and all probabilities of any exit from there could be set to 0. Then, we can easily compute the number of iterations of the pre-multiplication needed to achieve the required probability of being in the sink state, say 99%. From this data, the expected value of the induction time can be computed by summing the addition in the probability times the iteration number, multiplied by the time step increment, used between two iterations. So,

Expected induction time = $\sum_{t=1}^{T} t \cdot p(being in the induced state, during tth iteration),$

where, T is the number of iterations needed to reach 99% probability of being in the induced state.

IV. RESULTS:

1. Generation of simulated trajectories:

Here is a sample of 20 different simulations in 2D. The three colored graphs represent the corresponding molecule counts during that iteration. The time step was kept at 0.01 min.



Figure 2: A sample 2D graph showing 20 differrent simulations

The black colored stars indicate the entry into the induced state. The magenta colored star indicates the total shutdown of the system (i.e. all molecule counts reaching zero).

The dimension of the finite state was kept at 400 molecules of TMG, 15 molecules of *Permease*, and 3 molecules of *mRNA*. The initial state was assigned to be (100,10,1) in the (*TMG*, *Permease*, *mRNA*) space.

Here is another set of simulations, this time shown in 3D. The random color and line pattern has been used to differentiate between different traces.



Figure 3: A sample 3D plot showing 20 different simulations

This time, the finites state limits were set to be (900, 30, 3), while the starting state was assigned to be (300, 15, 1) in the (*TMG*, *Permease*, *mRNA*) space.

The temporal aspect of how things evolve cannot unfortunately be seen in this graph. However, this turns out to be a better visualization of the dynamics within the state space. Particularly so, because the neighboring states can actually be seen as neighbors in this plot, this plot is definitely better than the 2D counterpart.

The red stars plotted denote the exits into the induced state. On the other hand, the simulations that reach the origin are the traces that undergo complete shutdown.



2. Solving for the expected exit time from the low state:

Figure 4: Sample screenshot for visualizing the diffusion of the probability mass over time

Here is a screenshot of the visualization results for solving the time. Figures 1 through 3 in this screenshot display the probability mass as it evolves over time for mRNA count equal to 0, 1 and 2 respectively. The figure 4 shows a histogram of where the probability mass is located. The states are mapped into a straight line, for the purpose of getting a histogram.

Due to the memory limitations, visualizations for sizes above $(200 \times 10 \times 3)$ are not feasible.

All the four plots get updated in real time, and provide an excellent visualization tool for understanding the diffusion and the migration of the probability mass over time.

The numbers being printed on the prompt in the background represent the probability expressed in percentage of being in the induced state, during each iteration.

Here's another visualization, which was started from zero mRNA counts. The mRNAs are created during the process, but as expected, the whole process of induction is a lot slower than for the case of one mRNA molecule at the starting time step.



Figure 5: Another sample visualization: Reaction started with no mRNA present

V. DISCTUSSION:

- The simulations and the visualizations generated as a part of this project are indeed very helpful for understanding the dynamics and the behavior of the lac regulation system.
- The parameters, the size of the finite state space, and the initial conditions can be changed, and every time we get the appropriate visualization of the dynamics. This is an effective and a fun way of understanding the system, and give as much satisfaction as solving the dynamics equations analytically.
- The expected exit time can be calculated without having to run simulations.

- These simulations are extremely fast, than the simulations for the continuous or the hybrid counterparts. If the concentration levels are small enough, the gains are much higher if we are to use the discrete approach.
- The stochasticity introduced by [1] is indeed observed in the visualizations. It definitely explains the system behavior.
- The results obtained during this project could definitely be extended towards learning the deinduction behavior, and the combined behavior of the induction and de-induction

VI. ACKNOWLEDGEMENTS:

Agung, for the concept, and for the invaluable time you spared for me throughout...

Adam, for constantly giving new ideas, new directions and new inputs...

Thank you both.

VII. REFERENCES:

[1] A. Agung Julius, Adam Halasz, M. Selman Sakar, Harvey Rubin, Vijay Kumar, George J. Pappas, "Controlling biological systems: the lactose regulation system of Escherichia coli"

[2] Brian Munsky and Mustafa Khammash, "The finite state projection algorithm for the solution of the chemical master equation", *THE JOURNAL OF CHEMICAL PHYSICS*, 124, 044104 (2006)

[3] Andrew W. Moore, "Hidden Markov Models" <u>http://www.autonlab.org/tutorials/hmm14.pdf</u> <u>http://www.cs.cmu.edu/~awm/tutorials</u>