Systems Bioengineering 3
Homework 12

Greg Kiar

November 18, 2014

1. (a) For the following evaluation of average time, we make use of the fact that the transcript follows an exponential trend.

$$E(T) = t_{avg} = \int_0^\infty te^{-\alpha t}dt$$
$$= \alpha \left[\frac{-t}{\alpha} - \int_0^\infty \frac{1}{\alpha} e^{-\alpha t} dt \right]$$
$$= \alpha \left[\frac{-t}{\alpha} e^{-\alpha t} + \frac{-1}{\alpha^2} e^{-\alpha t} \right]_{0,t}$$
$$t_{avg} = \frac{1}{\alpha} = 10\text{min}$$

At equilibrium, $\beta = \alpha n$, $\therefore$ $\mu_n = 10$ at equilibrium. The variance is 10 since for independent stochastic events $\sigma_n^2 = \mu_n$.

(b) We can see that the ODE of this system is: $\dot{X}(t) = \beta - \alpha X(t)$. Solving this model using the equation:

$$\dot{X} = \frac{d}{dX} ln(X(t))$$
$$0 = \frac{d}{dX} \left(ln(\tfrac{\beta}{\alpha} - n) \right)$$
$$n = \frac{\beta}{\alpha} = 10$$

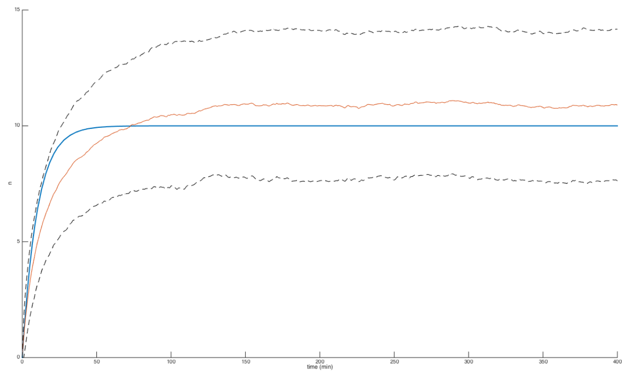
(c) Source code for this part was provided by the TAs. It has been modified slightly and is attached in an appendix.

(d) When starting the simulation at $n = 10$ and running it for $10,000$ min, the mean and variance were $\mu = 10.1287$ and $\sigma^2 = 9.8604$, respectively. These values are near but not exactly expected values. This makes sense because the discrete stochastic system has a finitely small timestep. If the timesteps were infinitely small or the trial infinitely long, the values would approach the theoretical values.

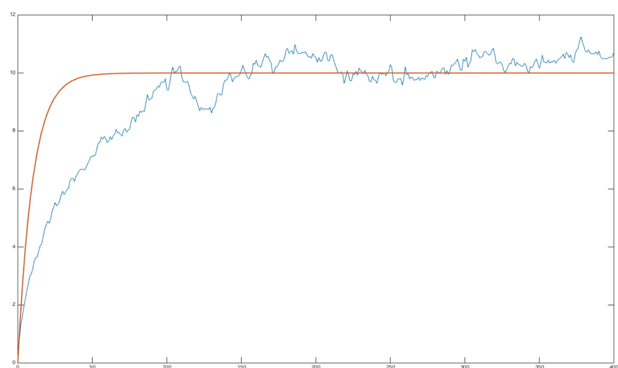
2. (a) Solving $\dot{X}(t) = \beta - \alpha X(t)$ for the case prior to equilibrium allows us to use the Laplace transform. The solution of this system becomes

$$X(t) = \frac{\beta}{\alpha}(1 - e^{-\alpha t})$$

(b) Shown below is the solution for 1000 randomly generated trajectories each for $100min$.



(c) It can be seen by the figure below, a plot of the variance, that the variance follows the same exponential trend as the random variable n. From this, we can set an approximate equation for the variance to be $\sigma^2 = \frac{\beta}{\alpha}(1 - e^{-\alpha t})$. From this we can see that the variance is at a maximum as $t \to \infty$.



3. (a) Similarly to the equation found in 2a), we can solve $\dot{X}(t) = \frac{\beta}{b} - \alpha X(t)$, where $b = 5$ here. The result for $X(t)$ is as follows:

$$X(t) = \frac{\beta}{b\alpha}(1 - e^{-\alpha t})$$

We can see from this result that the burst size, b, affects the forward rate of the system. The forward rate$= \frac{\beta}{b}$.

(b) Simulating the stochastic and ODE models of this system for $10,000 min$ with the inital condition $X(0) = n(0) = 2$, the equilibrium value, the discrete model produced the values shown below.

Discrete	$\mu = 12.0428$	$\sigma^2 = 30.3180$

(c) For the stochastic system with larger burst sizes, the mean and variance both increase. When testing this for different burst sizes, an approximately linear relationship is observed for both the mean and variance values. This relationship does not change as β or α increase and decrease simultaneously to one another. The approximate expressions are as follows:

$$\mu = \frac{\beta}{\alpha}$$
$$\sigma^2 = \frac{2\beta}{b\alpha}$$

Appendices

A Trajectories

```
function [T, N] = get_trajectory(n_0, t_max, burst)
% GET_TRAJECTORY - Computes a single Gillespie trajectory.
% Outputs:
% T - transition times
% N - counts at each transition time
% Inputs:
% n_0 - initial number of molecules.
% t_max - length of time of the simulation.

global a b
t = 0;
j = 1;
T(1) = 0;
N(1) = n_0;
while t < t_max
    j = j+1;
    dt = exprnd(b + a*N(j-1));
    t = t + dt;
    T(j) = t;
    if rand < b/(b+a*N(j-1))
        N(j) = N(j-1) + burst;
    else
        N(j) = N(j-1) - 1;
        if N(j) <= 0
            N(j) = 0;
        end
    end

end
end
```

B Example main script

```
clear
global a b
a = .1; % alpha
b = 1; % beta

n_0 = 0; % Initial count
nt = 1000; % Number of trajectories
```

```
t_max = 400; %Minutes for simulation (Approx).

G = cell(nt,2);
figure(1);
clf
hold on
for i = 1:nt
    [T, N] = get_trajectory(n_0, t_max,1);
    G{i,1} = T;
    G{i,2} = N;
    [T_int(i,:), N_int(i,:)] = const_intervals(T, N, t_max, 1);
end

for j = 1:size(N_int,2)
    mu_n(j) = mean(N_int(:,j));
    var_n(j) = var(N_int(:,j));
end
t = T_int(1,:);
[tout, xout] = ode15s(@xoden, [0 t_max], n_0);
plot(tout, xout, 'LineWidth', 2);
plot(t, mu_n, t, mu_n+sqrt(var_n), 'k--', t, mu_n-sqrt(var_n), 'k--');
xlabel('time (min)'); ylabel('n');
hold off

figure
plot(t, var_n)
hold on
```