SIMULATION STUDIES OF A FLUID QUEUING SYSTEM

BY

YUAN YAO

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Adviser:

Assistant Professor Mohamed Ali Belabbas

Abstract

This thesis analyzes an example of a resource allocating fluid queuing system. Fluid queuing models are widely used these days in describing the performance of network switches, routers and so on. Some works focus on the description of fluid queue in terms of probability. However, in this thesis, we focus on finding the optimal control policy using simulation. Stochastic differential equations play an important role in the problem formulation and simulation. We prove that strict mathematical expression of optimal control is hard to come up with when the controller is part of the stochastic differential equation. Thus, simulation is used to find optimal control for an example system, which is defined in the thesis.

Acknowledgment

First of all I would like to express my sincere gratitude to my advisor, Prof. Mohamed Ali Belabbas, who enlightened me with his knowledge, patience and insight. I also would like to thank the University of Illinois for offering me the opportunity to study and research in this inspiring academic environment. My thanks also go to my fellow friends: Jifei Xu, Xuandong Xu, Lida Zhu, Tao Yang, Tongtong Ye and Hui Xue. Last but not least, I would like to thank my parents, Aihua Liu and Hua Yao, for their unconditional love and support.

Contents

1. Introduction

Much literature about fluid queue systems is related to network analysis. According to [1], the performance evaluation of telecommunication and computer systems can be generalized to an infinite buffer fluid queue driven by a Markovian queue. There are several famous Markovian queue models, such as M/M/1, M/M/K/L, and M/PH/1/L. Moreover, much research has focused on the property and behavior of birth-death processes, [2]-[6] for example. However, these papers tend to go through the flow equation quite quickly and jump into the probability analysis. In this thesis, the system we are going to simulate is not only a queuing system; it is a queuing system with an input control which decides which queue the input is going to. The possibility analysis in the steady state is not enough for us to find the optimal control to minimize the cost. Brockett et al. [7] provide another way to analyze the fluid queuing system. We will use the stochastic differential equations first to describe the system, and then simulate it. Other papers about stochastic fluid queuing systems are [8],[9] and [10].

Let us move on to the system we are going to deal with in this thesis. A general model of these systems contains multiple queues and one input. The time intervals between the arrivals of the input are exponentially distributed with a rate λ. Each queue has its own processing rate μ_i. There is also a controller u, and it is used to decide which queue the arriving input is allocated to. Given this information, a stochastic differential equation can be established, and simulation showing the states of each queue over time, in other words the length of the queues over time, can be performed. Then we will choose a cost function which could be a function of u, the controller, and the length of the queues. The goal is to find a control policy which minimizes the cost function. We will try to use the knowledge of stochastic differential equations to prove that with the presence of the controller, u, in the differential equations, it is hard to solve for the optimal control explicitly. Then we will make a guess as to what the optimal controller would look like and use the simulation method to test the guess.

This thesis is structured as follows. In chapter 2, some basic mathematical background about stochastic differential equations will be provided. We will also mention some basic ideas in dynamic programming to find the optimal control. In chapter 3, we will discuss the model we are going to simulate in detail; we will prove that the optimal control is hard to find using math derivation. An assumption of the optimal control will be made in this section as well. In chapter 4, we will present three sets of simulation results to prove the assumption we made in the previous section.

2. Mathematical Preliminaries

2.1 Stochastic differential equation

A Poisson counter is a simple process, but it is very powerful in describing stochastic processes. A Poisson counter is a non-decreasing process which takes on values of positive integers and the jump time between jumps is exponentially distributed according to some parameter λ. Combining the idea of Poisson counter with differential equations, we get a stochastic differential equation.

Consider the following stochastic differential equation:

$$dx(t) = f(x)dt + g(x)dN_t \tag{2.1}$$

where dN_t is a Poisson counter. Intuitively, the solution of (2.1) is:

(a) On an interval where no jump is happening, *x(t)* is the solution to $\frac{dx}{dt} = f(x,t)$.

(b) If there is a jump at time t_1, then $x(t_1^+) = x(t_1^-) + g(x(t_1^-), t_1)$.

That is the solution to a stochastic differential equation in the sense of Ito. And equation (2.1) is also called a Poisson driven stochastic differential equation (PDSDE).

For simple PDSDE, it is quite straightforward to write down the solution in the sense of Ito. However, when the equations become complicated and more Poisson counters get involved in the question, this solution gives us very little information about the behavior of the whole system. In order to solve this problem, the expectation rule is introduced so that we can find the expectation of x at the time$t = \tau$.

Consider a general PDSDE:

$$dx = f(x)dt + \sum_{i=1}^{n} g_i(x)dN_i \tag{2.2}$$

If ψ is a differentiable function, then

$$d\psi(x,t) = \frac{\partial\psi}{\partial t}dt + \langle\frac{d\psi}{\partial x}, f(t,x)\rangle + \sum_{i}^{n}(\psi\big(t, x + g_i(x)\big) - \psi(t,x))dN_i \tag{2.3}$$

For example, given

$$dx = -xdt + xdN$$

Let $\psi(x) = x^2$; then, applying the Ito rule for ψ gives:

$$d\psi = -2x^2dt + ((x+x)^2 - x^2)dN = -2x^2dt + 3x^2dN$$

With the application of Ito rule, the expectation of $x(t)$ is easy to find:

$$d\mathbb{E}x = \mathbb{E}(f(x,t))dt + \sum_{i}^{n} \mathbb{E}g_i\big(t, x(t)\big)\,\lambda_i dt \tag{2.4}$$

where λ_i is the rate of the corresponding Poisson counter dN_i.

Furthermore with the expectation of *x(t)* and the Ito rule, the probability density function of *x(t)* can be found, which is called the Fokker-Plank equation. But we are not going to use it in this thesis.

2.2 Optimization

For a system which can be described as a PDSDE with a controller *u*, such as

$$dx = (f(x,t))dt + (g(x,u,t))dN \tag{2.5}$$

With a cost function

$$J_t = \sum_{i=t}^{n} [c(x_i, u_i, i)] \tag{2.6}$$

We want to find the controller $u_1, u_2 \dots \dots, u_n$ such that the expectation of cost is minimized.

$$\inf_{u_1,u_2,u_3,\dots\dots u_n} \mathbb{E} \sum_{t=0}^{n} [c(x_t, u_t, t)] \tag{2.7}$$

Given the PDSDE and the cost function, we can easily calculate the cost for a certain control we select by simulation using computers. We cannot guarantee that the control we have is optimal. We can also predict that for a system like (2.5) which has n jumps during a certain time period t_n, there will be infinite controls out there. Even for a system where *u* can only take a value between -1 and 1 (the kind of system we are going to simulate), there will still be 2^{n}different possible controllers. It is extremely hard to find the optimal *u* through simulations. For deterministic systems, knowledge of dynamic programming can be used to find the optimal control *u* recursively.

Consider a discrete-time system

$$\mathrm{x}_{\mathrm{t+1}} = f(x_t, u_t, t) \tag{2.8}$$

with a given initial condition x_0 and a cost function

$$J = \sum_{t=0}^{t=n-1} c(x_t, u_t, t) + J_n(x_n) \tag{2.9}$$

where $J_n(x_n)$ is considered as the terminal cost. Define $V(t, x)$ the optimal cost starting at time *t* from the state *x*. We call function *V* the value function.

$$V(x, t) = \inf_{u_t, u_{t+1, \ldots\ldots} u_n} \sum_{s=t}^{t=n-1} c(x_s, u_s, s) + J_n(x_n) \tag{2.10}$$

$$V(x, t) = \inf_{u_t} \left[c(x_t, u_t, t) + \inf_{u_{t+1}, u_{t+2} \ldots\ldots u_n} \left[\sum_{s=t+1}^{n} c(x_s, u_{s,s}) + J_n(x_n) \right] \right] \tag{2.11}$$

After observation we can see that the second term of equation (2.11) is independent of u_t. Thus we can write the value function in a recursive form:

$$V(t, x) = \inf_{u} [c(x, u, t) + V(a(x, u, t), t + 1)] \tag{2.12}$$

for $t < n$ and a boundary condition $V(n, x) = J_n(x)$. This equation is called the Bellman equation.

3. Problem Formulation

3.1 Model and simulation basics

The model we consider in this thesis is based on stochastic differential equations.

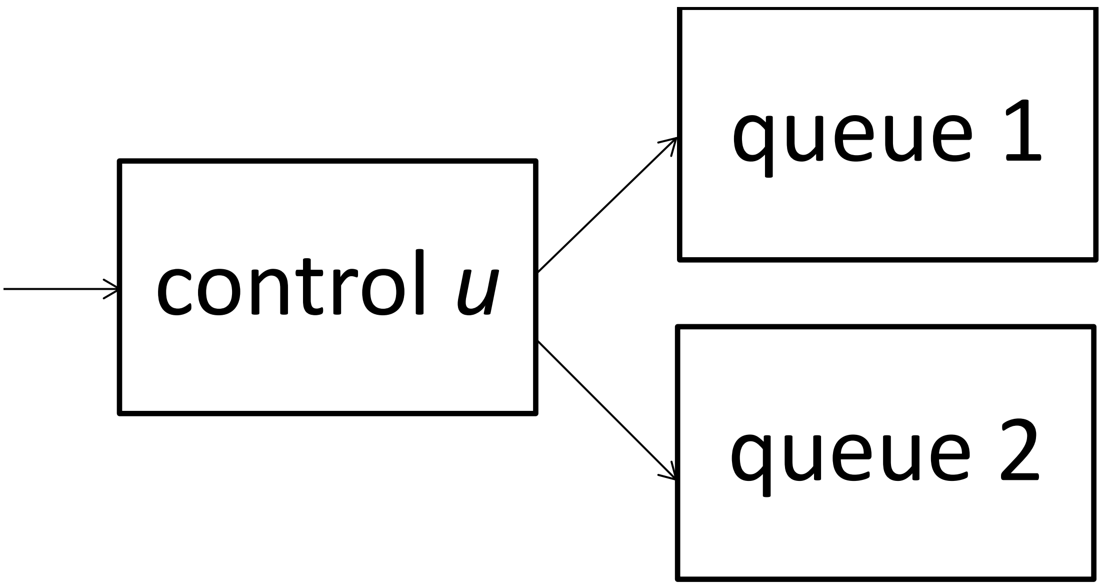

Figure 3.1 Simple model of the system we are going to deal with. Suppose we have a packet come into the system with a rate λ. We have controller u to decide which queue this packet is going to. For example, given u $\in [-1, 1]$, if u=1 the packet goes to queue 1 and if u=-1 the packet goes to queue 2. Queue 1 and queue 2 have processing rates μ_1 and μ_2 respectively.

Consider the case in Figure 3.1 The coming package will go through a controller u and the controller will allocate the package to either queue 1 or queue 2 based on the current state. The behavior of the system can be written in PDSDE form:

$$dx_1(t) = f_1(x)dt + g_1(u, x_1, x_2)dN \tag{3.1}$$

$$dx_2(t) = f_2(x)dt + g_2(u, x_1, x_2)dN \tag{3.2}$$

where x_1 and x_2 represent the length of queue 1 and queue 2 respectively. $N(t)$ is a Poisson counter with parameter λ. The cost function is a function of u, x_1 and x_2.

Writing the system in PDSDE allows us to simulate the length of queues over time using Euler integration. We can use the computer to simulate the continuous process discretely by choosing very small time step Δt.

To do the simulation:

1. An array Q is created using random number generator. Each entry of Q records the jump time of the Poisson counter *N(t)*.
2. Between the jumps, we find the change of the length of the queues as follows:

$$\Delta x_1(t) = f_1(x)\Delta t \text{ if } x_1 > 0, 0 \text{ otherwise} \tag{3.3}$$

$$\Delta x_2(t) = f_2(x)\Delta t \text{ if } x_2 > 0, 0 \text{ otherwise} \tag{3.4}$$

3. At the jumps, the controller u is decided based on the current state, x_1 and x_2. In this case:

$$\Delta x_1(t) = 1 \text{ if } u = 1, 0 \text{ otherwise} \tag{3.5}$$

$$\Delta x_2(t) = 1 \text{ if } u = -1, 0 \text{ otherwise} \tag{3.6}$$

4. Doing the simulation for a period of time can calculate the overall cost using the cost function. This simulation will be done several times with the same jump time under different control policies, and a curve showing cost vs. policies will be created.

The major problem here is to quantitatively distinguish different policies so that we can create a plot according to different policies. If we think it through, the policies are the criteria to switch the controller, in this case to change *u* to 1 or to -1 and vice versa. Intuitively, if the cost function depends on the length of the queues, the policies should also depend on the length of the queues. Thus an assumption has been made here that the optimal policy which minimizes the cost function can be found as:

$$u = 1 \text{ if } \alpha x_1 > x_2 \tag{3.7}$$

$$u = -1 \text{ if } \alpha x_1 \leq x_2 \tag{3.8}$$

Different policies have different α values, and we can visualize the cost vs. different policies quite easily.

3.2 Mathematical approach of an example of a simple model

Consider the following system:

$$dx_1 = -\mu_1 I_{x_1} dt + \frac{u+1}{2} dN \tag{3.9}$$

$$dx_2 = -\mu_2 I_{x_2} dt + \frac{-u+1}{2} dN \tag{3.10}$$

where I_{x_1} and I_{x_2} are the indicator functions, which equal to 1 when x_1 or x_2 is greater than zero and zero otherwise. When *u=1* the input goes to $x_1$and when *u=-1* the input goes to x_2 and *dN* is a Poisson counter with parameter λ. We also define the cost function as:

$$J_t = \int_0^\infty c(x_1, x_2) \tag{3.11}$$

where

$$c(x_1, x_2) = x_1^2 + x_2^2 \tag{3.12}$$

The expectation of the cost function can be expressed as the sum of the expectation of $x_1^2$and x_2^2. We use the Ito rule and expectation rule to find $\mathbb{E}x_1^2$ and $\mathbb{E}x_2^2$.

Chose $\psi = x_1^2$

$$dx_1^2 = -2x_1\mu_1 I_{x_1} + \frac{u+1}{2} dN \tag{3.13}$$

We can also find this relation for x_2:

$$dx_2^2 = -2x_2\mu_2 I_{x_2} + \frac{-u+1}{2} dN \tag{3.14}$$

Taking the expectation on both sides,

$$d\mathbb{E}x_1^2 = -2\mathbb{E}(x_1 I_{x_1})\mu_1 dt + \frac{\mathbb{E}(u)+1}{2} \lambda dt \tag{3.15}$$

Note that $\mathbb{E}\left(x_1 I_{x_1}\right) = \mathbb{E}(x_1)$. The next step is to find $\mathbb{E}(x_1)$. Taking the expectation on both sides of the original PDSDE, we have:

$$d\mathbb{E}x_1 = -\mathbb{E}(I_{x_1})\mu_1 dt + \frac{\mathbb{E}(u)+1}{2} \lambda dt \tag{3.16}$$

We have a term which contains $\mathbb{E}(I_{x_1})$. The expectation of the indicator function is the probability of $x_1 > 0$, *P(x>0).* There is another term that shows up in both the expectation of x_1 and expectation of x_1^2: $\mathbb{E}(u)$. $\mathbb{E}(u)$is not something we can solve directly, since *u* is the controller we need to modify. Thus an explicit expression of the controller *u* is almost impossible to find.

3.3 Simulation details

Given the information we have from sections 3.1 and 3.2, we need to simplify the problem a little bit. We are going to find the relationship between the optimal α, the ratio between x_1 and x_2 when we switch the policies, and the ratio between the processing rates, $\frac{\mu_1}{\mu_2}$. The simulation will be done using different cost functions.

We also need to consider some constraints in this problem. First of all, both the processing rates, μ_1 and μ_2, should be slower than the arriving rate λ, because, in practice, if we have a system with a higher processing rate than the arriving rate, there will be almost no cost for storage. However, this constraint may not apply to the system where the control is part of the cost function. For example, consider the following cost function:

$$\mathrm{c}(\mathrm{x}, \mathrm{u}, \mathrm{t}) = x_1^2 + x_2^2 + 10(1 + \mathrm{u})^2 \tag{3.17}$$

This can be considered to mean that there is a difference in the transportation cost by choosing different storages (queues).

The second constraint is that the sum of μ_1 and μ_2 should be greater than or equal to the arriving rate λ. If $\mu_2 + \mu_2 < \lambda$, the expectation of both x_1 and x_2 will go to infinity as the time goes to infinity. Intuitively, for a system like this, the optimal control will be the controller which sends the input to the shorter queue regardless of the processing rate. This is not an exciting model to talk about, and it will not be considered in this thesis. Also I am going to simulate a little bit of this kind of model to show that the policy, α more specifically, is a constant.

4. Simulations and Results

We will deal with a system that looks exactly like Figure 3.1. The cost function is set to be:

$$c(x_1, x_2) = x_1^2 + x_2^2 \tag{4.1}$$

The behavior of the queues obeys the following PDSDE:

$$dx_1 = -\mu_1 I_{x_1} dt + \frac{u+1}{2} dN \tag{4.2}$$

$$dx_2 = -\mu_2 I_{x_2} dt + \frac{-u+1}{2} dN \tag{4.3}$$

where I_{x_1} and I_{x_2} are the indicator functions, which take the value of one if x_1 or x_2 is greater than zero, zero otherwise. dN is a Poisson counter with parameter λ, and *u* is the controller which takes the values -1 and 1. When *u* takes the value 1 the input goes to queue 1 and when *u* takes the value -1 the input goes to queue 2. We assume that the processing rate of queue 1 is greater than queue 2. It is obvious that the expectation of x_1 is greater than that of x_2. Thus we set a policy such that:

IF $\alpha x_1 > x_2$

u takes the value u=1

ELSE

u takes the value u=-1

END

where α takes the value from 0 to 1, α ∈ (0,1].

4.1 Simulation set 1

In this simulation set, we are going to show that if the combined processing rate of both queues is smaller than the arriving rate, the policy variable α will be a constant.

We will set the arriving rate $\lambda = 1$ and we will do the trials listed in Table 4.1.

Table 4.1 Set 1 Simulation Profile

	μ_1	μ_2
Trial 1	0.4	0.4
Trial 2	0.5	0.3
Trial 3	0.6	0.2
Trial 4	0.7	0.1

For all the trials, the run time has been set to 5000 and the time step is set to 0.01. We will track the cost for different α which takes the value from 0 to 2 with a step size of 0.01.

For the first trial we get a path of x_1 and x_2 that looks like Figure 4.1 and Figure 4.2. It shows that the length of the queue goes to infinity as the time goes to infinity. This implies that there the queues are not able to solve all the work that comes into the queue. This is definitely not a desirable system in real life. The paths of x_1 and x_2 have the same patterns of Figure 4.1 and Figure 4.2 for the other three trials. We are not going to show all the figures in this thesis.

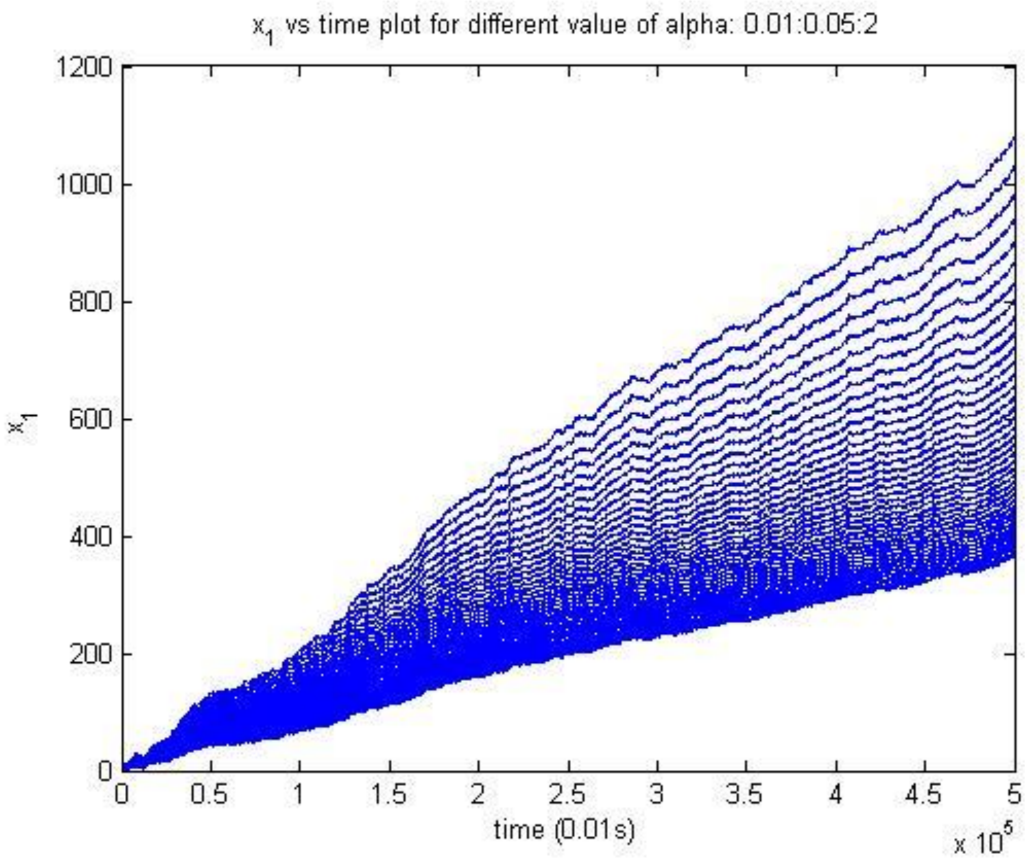

Figure 4.1 The length of queue 1 versus time for trial 1. Different curves show different behavior with different policies. The policy parameter α takes the value from 0.01 to 2 with the step size 0.05. The uppermost curve corresponds to $\alpha = 0.01$ and the lowest curve corresponds to $\alpha = 2$.

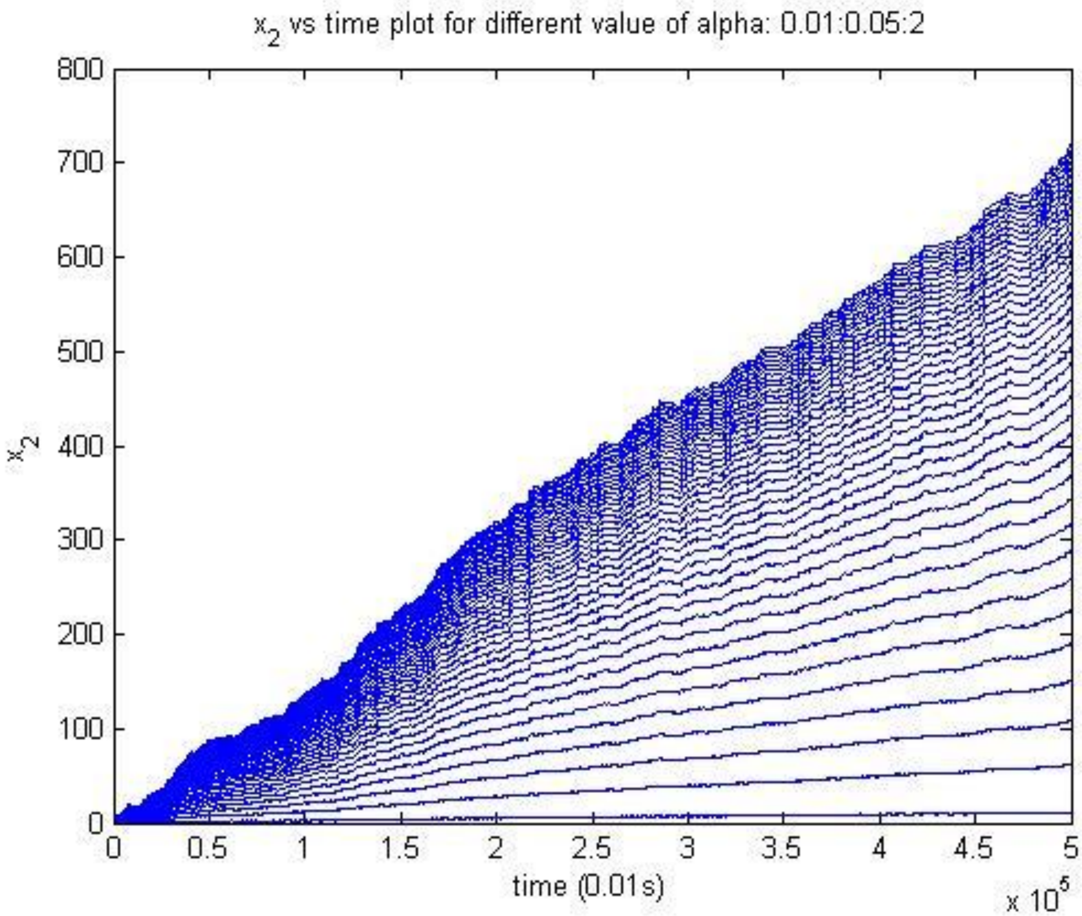

Figure 4.2 The length of queue 2 versus time for trial 1 with the same simulation specification as Figure 4.1. The uppermost curve corresponds to $\alpha = 2$ and the lowest curve corresponds to $\alpha = 0.01$.

However, we still want to show what kind of policy gives us the lowest cost for this kind of system. Figure 4.3 shows the relationships between the policy α and the cost for different trials.

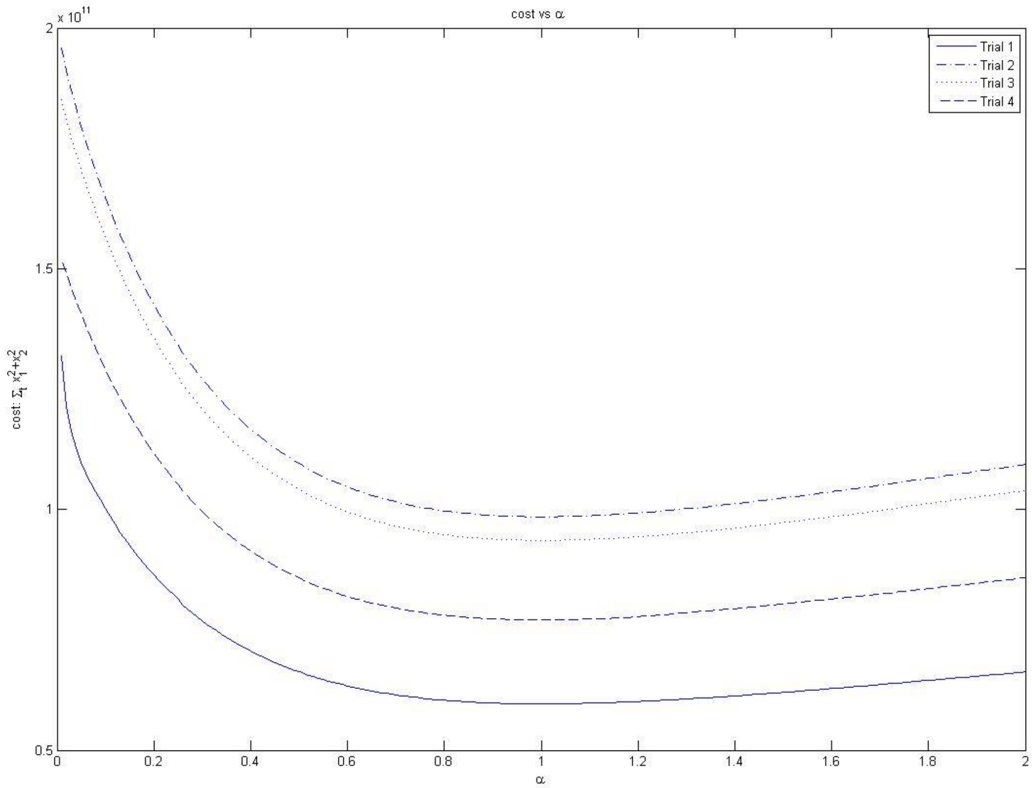

Figure 4.3 Graph of the overall cost $\Sigma_t(x_1^2 + x_2^2)$ versus the policy parameter α for different trials (see Table 4.1).

From Figure 4.3 we can tell that the lowest value of cost is achieved at $\alpha = 1$ for all four trials. We can easily come up with the conclusion that for a system where the sum of the processing rates is lower than the arriving rate, $\mu_1 + \mu_2 < \lambda$, the optimal control is achieved at $\alpha = 1$, which is independent of the processing rates μ_1 and μ_2.

4.2 Simulation set 2

In this simulation set, we are going to do something more exciting; we will set the combined processing rate equal to the arriving rate. We will do the trials listed in Table 4.2.

Table 4.2 Set 2 Simulation Profile

	μ_1	μ_2
Trial 1	0.55	0.45
Trial 2	0.6	0.4
Trial 3	0.65	0.35
Trial 4	0.7	0.3
Trial 5	0.75	0.25
Trial 6	0.8	0.2
Trial 7	0.85	0.15
Trial 8	0.9	0.1
Trial 9	0.95	0.05

For all the trials, the run time has been set to 5000 and the time step is set to 0.01. We will track the cost for different α which takes the value from 0 to 1 with a step size of 0.01. Since the processing rate of queue 1 is always greater than or equal to the processing rate of queue 2, we assume the length of queue 1 is greater than or equal to the length of queue 2.

As in the previous simulation set, we are going to show how the queue length changes over time. For trial 1, it is shown in Figure 4.4 (for x_1) and Figure 4.5 (for x_2). We can tell from the graphs that the queue length is fluctuating around some point as time goes on. Although the expectation of the queue length is changed along different policies, we can at least conclude that the queue length will not explode as time goes to infinity.

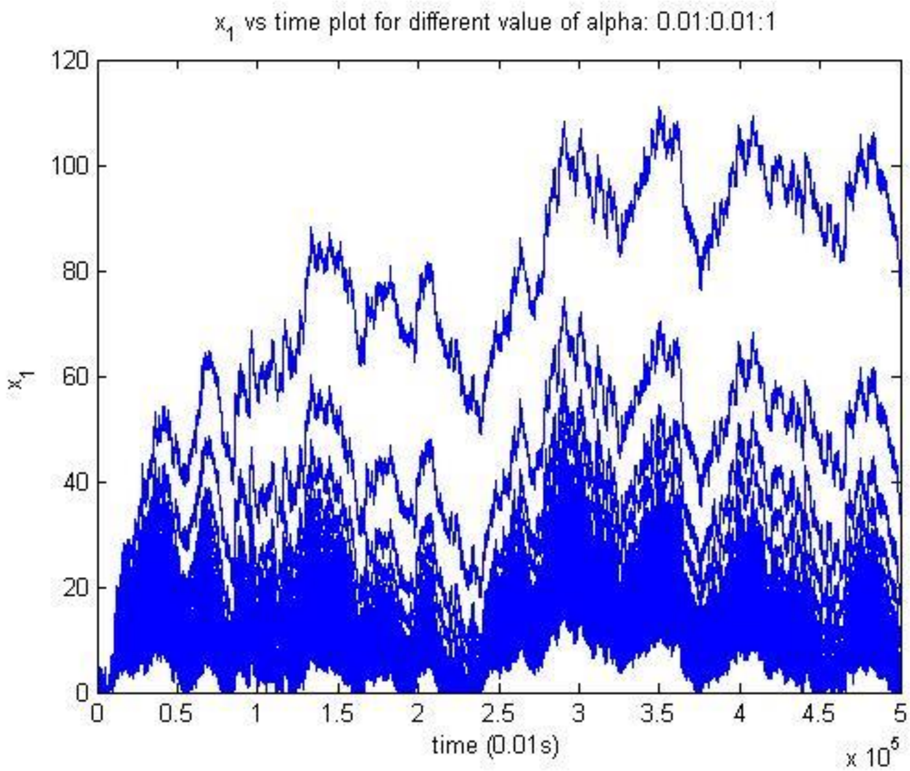

Figure 4.4 Length of queue 1 vs time for different α, $\lambda = 1$ $\mu_1 = 0.55$ and $\mu_2 = 0,45$.

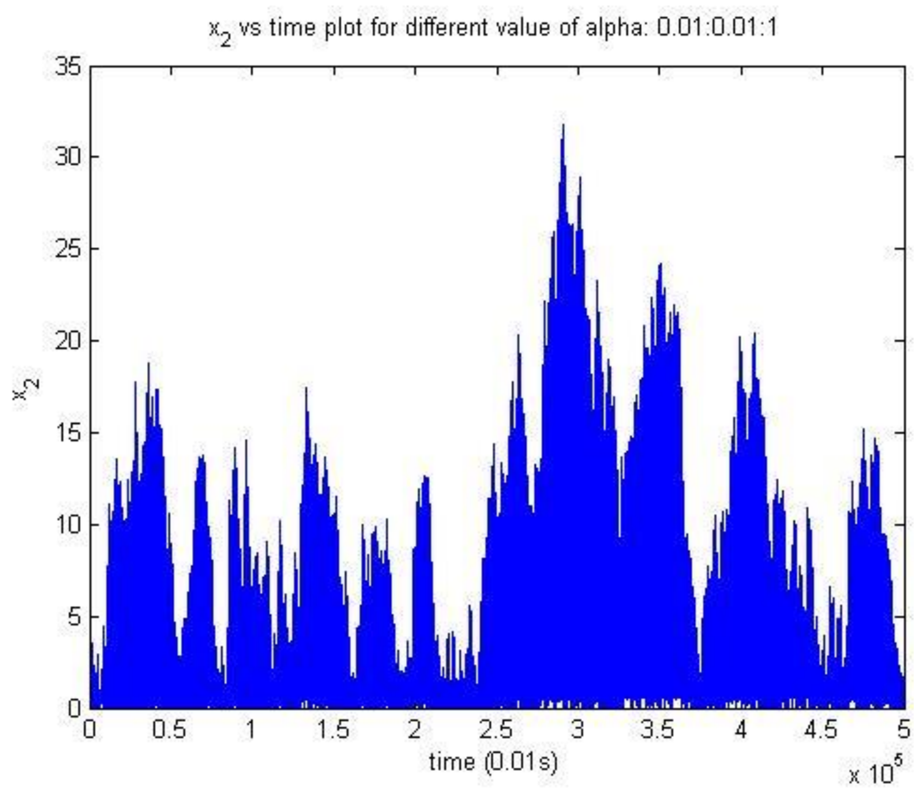

Figure 4.5 Length of queue 2 vs time for different α, $\lambda = 1$ $\mu_1 = 0.55$ and $\mu_2 = 0,45$.

Now we will show how the optimal control strategy changes as the ratio between the processing rates of the queues changes. Figure 4.6 is a graph of cost versus α for different trials. We can read the α value for the lowest cost from the graph. The simulation is down five times for each trial and the α reads as shown in Table 4.3.

Table 4.3 Optimal α Readings for 5 Simulations of Set 2

	1st read	2nd read	3rd read	4th read	5th read	average	Ratio $\frac{\mu_1}{\mu_2}$
Trial 1	1	0.99	1	1	0.99	1	1.22
Trial 2	0.94	0.99	0.98	0.98	0.99	0.98	1.5
Trial 3	0.99	0.91	0.99	1	0.98	0.97	1.86
Trial 4	0.91	0.95	0.77	0.89	0.99	0.90	2.33
Trial 5	0.71	0.99	0.77	0.98	0.84	0.84	3
Trial 6	0.85	0.98	0.77	0.66	0.95	0.84	4
Trial 7	0.83	0.68	0.67	0.98	0.76	0.78	5.67
Trial 8	0.72	0.92	0.98	0.91	0.88	0.88	9
Trial 9	0.92	0.94	0.57	0.57	0.76	0.75	19

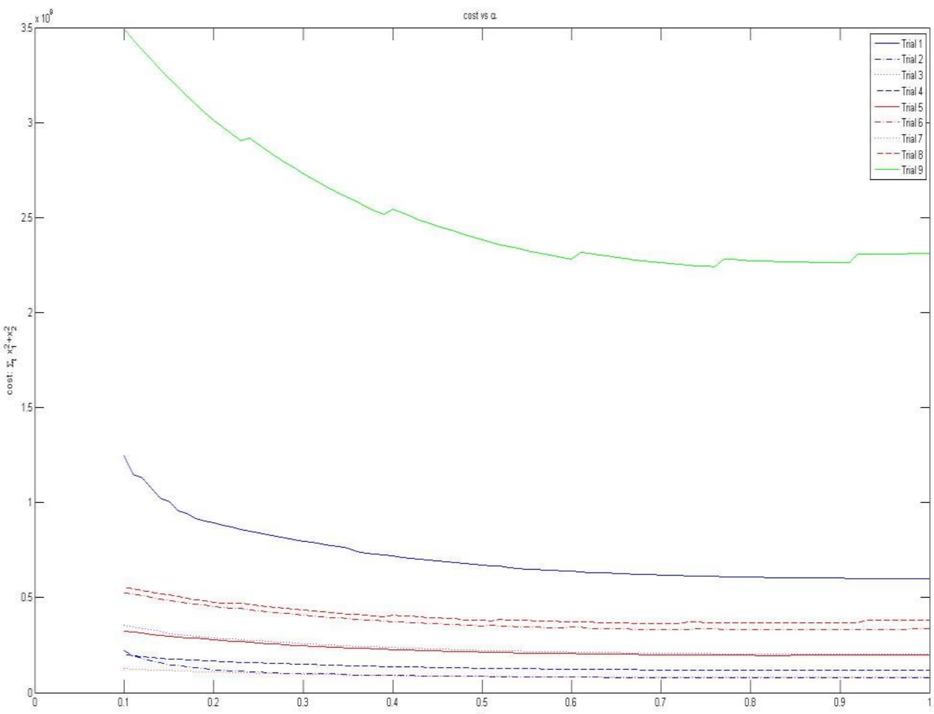

Figure 4.6 Graph of the overall cost $\Sigma_t(x_1^2 + x_2^2)$ versus the policy parameter α for different trials (see Table 4.2). The graph is plotted beginning with $\alpha = 0.1$ because the cost for those small α values becomes so large that the rest of the graph is hard to read.

From Table 4.3 we can tell that the optimal α changes a lot even for a certain ratio of μ_1 and μ_2. However, a general decreasing trend can be observed as the ratio of μ_1 and μ_2 increases. Figure 4.7 can be plotted based on the data we have in Table 4.3. Figure 4.7 also shows a linear fitting of the curve.

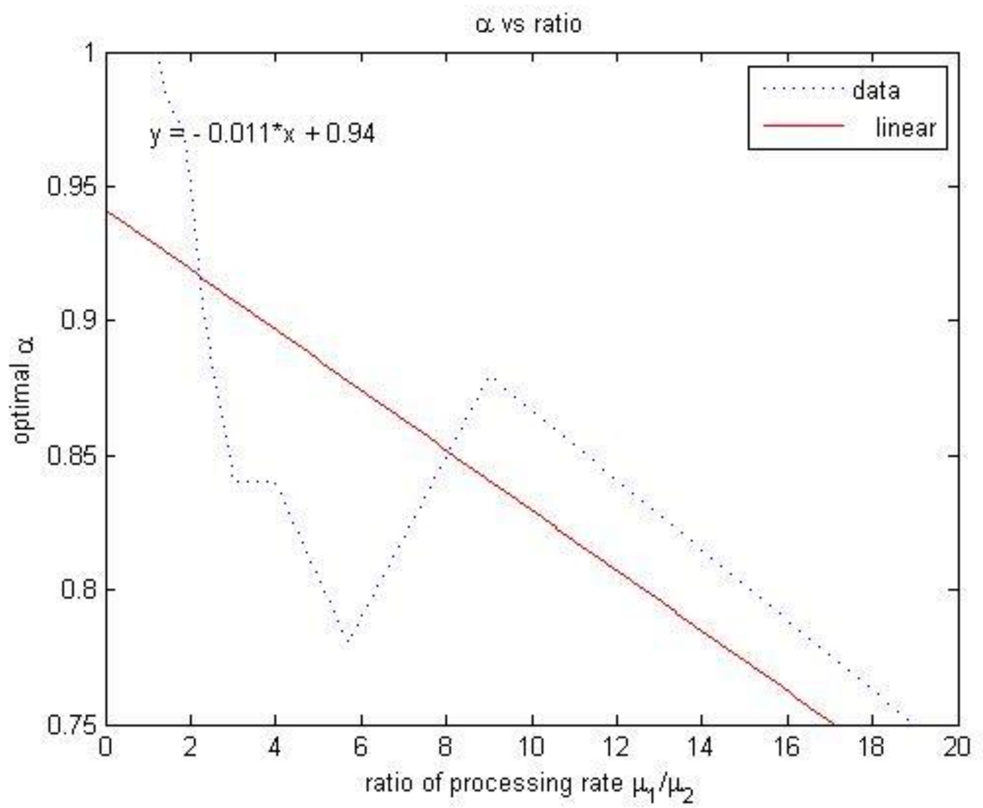

Figure 4.7 Curve of α versus the ratio with a linear fitting.

We can see from the curve that the first seven trials behave closer to a linear curve compared to the last two trials of data. The abnormal behavior of data in trial 6 most likely is because of the stochastic property of simulation.

4.3 Simulation set 3

In this simulation we will set the combination processing rate greater than the arriving rate, and as before we will set the $\mu_1 + \mu_2$ to be constant. We are going to do the sets of trials listed in Table 4.4.

Table 4.4 Set 3 Simulation Profile

	μ_1	μ_2
Trial 1	1.05	0.95
Trial 2	1.10	0.90
Trial 3	1.15	0.85
Trial 4	1.20	0.80
Trial 5	1.25	0.75
Trial 6	1.30	0.70
Trial 7	1.35	0.65
Trial 8	1.40	0.60
Trial 9	1.45	0.55

As before, Figure 4.8 and Figure 4.9 show how the length of the queue changes over time. We can see that most of the time the queue length stays between 0 and 1.

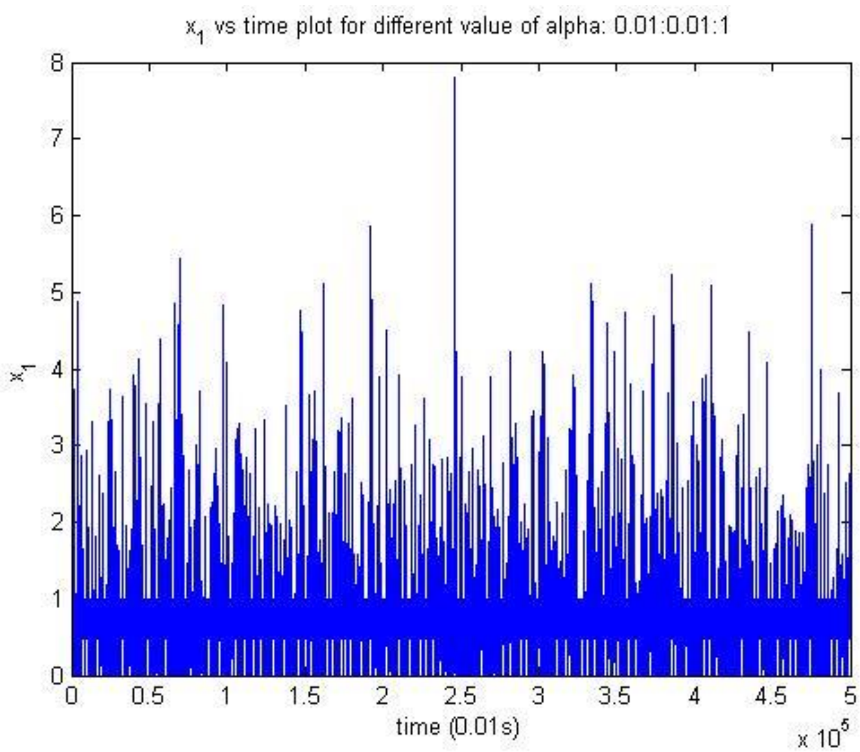

Figure 4.8 Length of queue 1 vs. time for different α, $\lambda = 1$ $\mu_1 = 1.05$ and $\mu_2 = 0.95$.

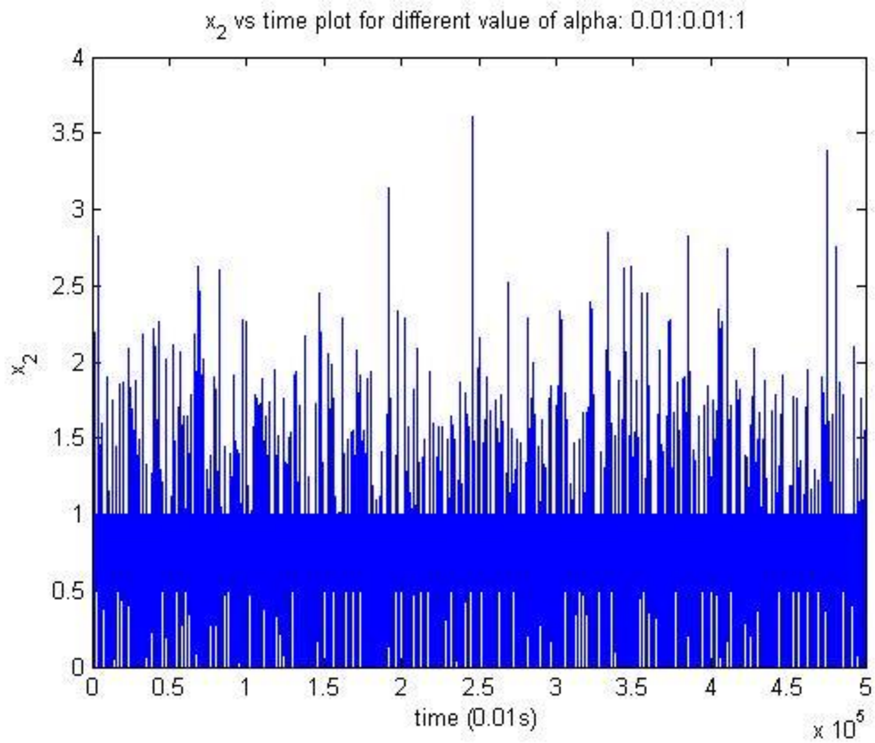

Figure 4.9 Length of queue 1 vs. time for different α, $\lambda = 1$ $\mu_1 = 1.05$ and $\mu_2 = 0.95$.

In Figure 4.10 we put all the nine trials together to see how the policy affects the overall cost of the system for different processing rate combinations.

As with simulation set 2, we will do this simulation 5 times and read the α values corresponding to the lowest costs.

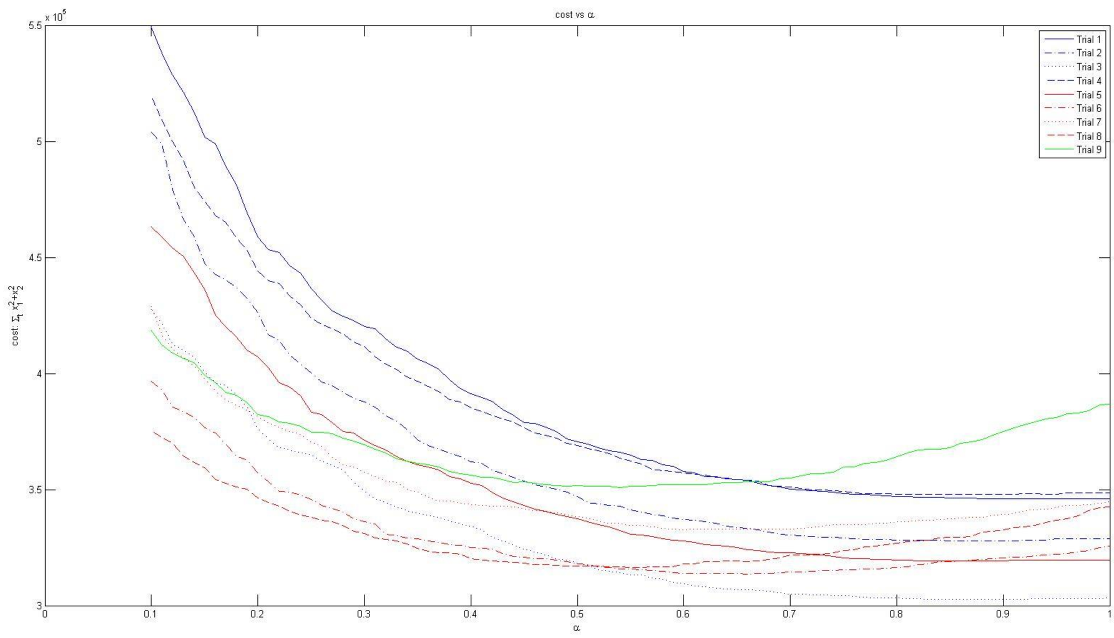

Figure 4.10 Overall cost $\Sigma_t(x_1^2 + x_2^2)$ versus the policy parameter α for different trials (see Table 4.4). Since we are simulating a stochastic system, the curve show here is just one example to give a demonstration of how the curve α versus cost looks.

Table 4.5 Optimal α Readings for 5 Simulations of Set 3

	1st read	2nd read	3rd read	4th read	5th read	average	Ratio $\frac{\mu_1}{\mu_2}$
Trial 1	0.97	0.97	0.94	0.94	0.95	0.954	1.105
Trial 2	0.88	0.89	0.87	0.84	0.92	0.88	1.22
Trial 3	0.89	0.89	0.86	0.91	0.81	0.872	1.353
Trial 4	0.88	0.90	0.83	0.86	0.88	0.87	1.5
Trial 5	0.89	0.84	0.87	0.87	0.88	0.87	1.667
Trial 6	0.66	0.68	0.66	0.62	0.63	0.65	1.857
Trial 7	0.60	0.65	0.70	0.63	0.63	0.642	2.077
Trial 8	0.56	0.57	0.60	0.56	0.57	0.572	2.333
Trial 9	0.53	0.54	0.53	0.55	0.52	0.534	2.6364

Figure 4.11 is based on Table 4.5. The linear function we get is $\alpha = -0.29 * \left(\frac{\mu_1}{\mu_2}\right) + 1.3$. This time the linear fitting looks more convincing compared to the fitting we did in the previous chapter. If $\frac{\mu_1}{\mu_2} = 1$, it means queue 1 and queue 2 process at the same rate. Intuitively, the optimal cost point is achieved when the input is sent to the shorter queue, which means $\alpha = 1$. The solution our linear equation gives us is 1.01, which is 1% off, and it can be considered quite accurate. Moreover, let us try $\frac{\mu_1}{\mu_2}$ =3; using our equation we get $\alpha = 0.43$ as our optimal control policy. For simulation we set $\mu_1 = 1.5$ and $\mu_2 = 0.5$.

After 5 simulations we get an average value of optimal $\alpha = 0.45$. The error is about 5%. This is still quite an accurate estimation considering the system is a stochastic process. However, if we try $\frac{\mu_1}{\mu_2} = 4$, the estimated value by the linear approximation is 0.14 but the simulated result gives us a value of α around 0.37, which means this linear approximation is only applicable for a certain range.

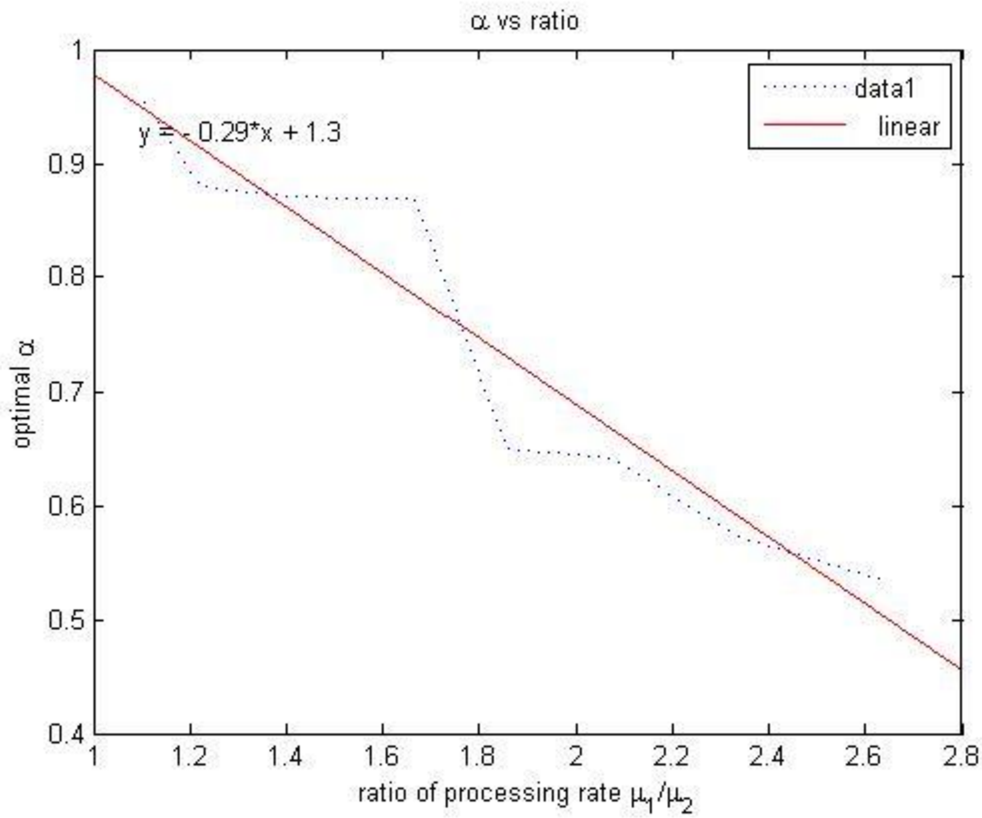

Figure 4.11 Curve of α versus the ratio with a linear fitting of the data from Table 5.

5. Conclusion and Future Work

We have proved that even for a very simple resource allocating fluid queuing system (described in chapter 3). It is hard to explicitly write down a mathematical expression for a controller to minimize certain cost functions. We decided to find an expression of the controller by simulating the system. We choose the control switch policy to be a function of α. To keep the problem simple we use $\alpha x_1 < x_2$ as our control policy. Then we assume that the optimal control policy parameter α might have a linear relationship with the ratio of the processing rates of the two different queues. A simple example analysis shows that there is a linear relationship. However, this linear relationship only exists for some range of $\frac{\mu_1}{\mu_2}$ when the combined processing rate is greater than input rate ($\mu_1 + \mu_2 > \lambda$).

There is a lot we can do to improve the result we get here. From the mathematical perspective, a strict mathematical proof to show the linear relationship between α and cost can be made. On the other hand, because it is a stochastic system, we always want to do as many simulations as possible to eliminate the random factors during the simulation. Moreover, more simulations can be done. In this thesis, we only simulate one specific model with one specific cost function. More simulations can be done to find the optimal control policy for a more general model.

References

[1] B. Sericola and B. Tuffin "A fluid queue driven by a Markovian queue," *Queueing Systems*, vol. 31, pp. 253–264, July 1999.

[2] I. Adan and J. Resing, "Simple analysis of a fluid queue driven by an M/M/1 queue," *Queueing Systems*, vol. 22, pp. 171–174, March 1996.

[3] D. Mitra, "Stochastic theory of a fluid model of producers and consumers coupled by a buffer," *Advances in Applied Probability*, vol. 20, no. 3, pp. 646-676, September 1988.

[4] T.E. Stern and A.I. Elwalid, "Analysis of separable Markov-modulated rate models for information-handling systems," *Advances in Applied Probability*, vol. 23 no. 1, pp. 105–139, March 1991.

[5] E.A. van Doorn and W.R.W. Scheinhardt, "A fluid queue driven by an infinite state birth-death process," *Teletraffic Science and Engineering*, vol. 2 pp. 465-475, June 1997.

[6] J. Virtamo and I. Norros, "Fluid queue driven by an M/M/1 queue," *Queueing Systems*, vol. 16, pp. 373-386, 1994.

[7] R. W. Brockett, W. B. Gong, and Y. Guo, "Stochastic analysis for fluid queueing systems," in *Proceedings of the 38th IEEE Conference*, 1999, pp. 3077–3082.

[8] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *Bell System Technical Journal*, vol. 61, pp. 1871-1894, October 1982.

[9] P. Bremaud, *Point Processes and Queues Martingale Dynamics*. Springer, 1981.

[10] A. F. Marins-Neto and E. Wong, "A Martingale approach to queues," *Mathematical Programming Study*, vol. 6, pp. 97-110, 1976.