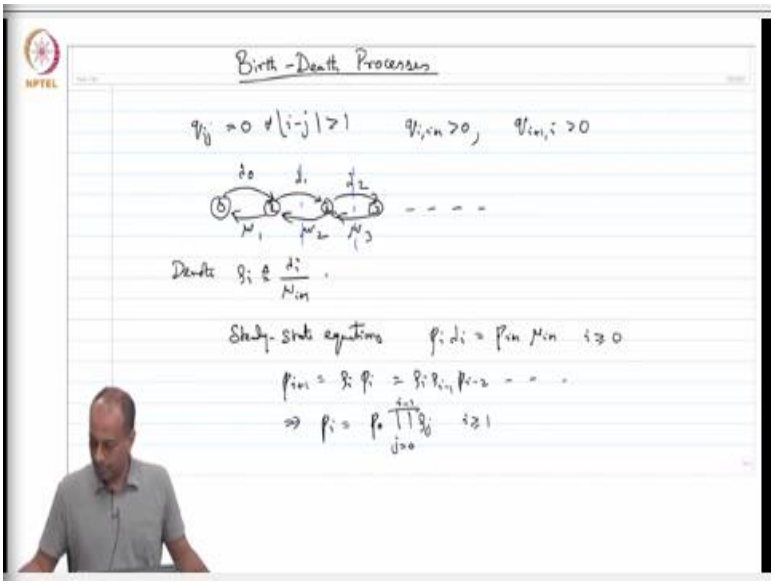
Stochastic Modeling and the Theory of Queues
Prof. Krishna Jagannathan
Department of Electrical Engineering
Indian Institute of Technology-Madras

Lecture-74
The Birth-Death Continuous Time Markov Chains

(Refer Slide Time: 00:16)

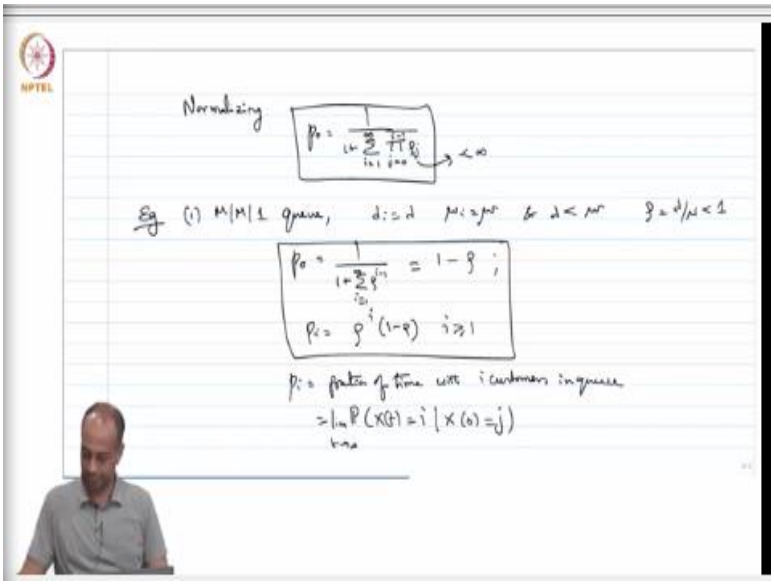


Welcome back, today we will discuss the topic of birth death processes in continuous time. We have already studied birth-death Markov chains in discrete time; we will now discuss birth-death CTMC's. So, for a birth-death CTMC we have q ij has the property that q ij = 0 for i - j when absolute value of i - j greater than 1 q ij = 0. So, q i, i + 1 is positive and q i + 1 i is positive and q ij = 0 for absolute value of i - j greater than 1.

So, if your state space is 0, 1, 2 etcetera if it is the non negative integers, you have the following structure. You have q 0 1 let us call q 0 1 = lambda 0 is to be mu 1, lambda 1 mu 2 and so on, mu 3 dot, dot, dot. If, so I am of course drawing the q i j's, you can also draw if you want the time sampled version with lambda delta and all that or you can draw the embedded chain and the transition rates you can notate it any other way.

This is probably the simplest where I have drawn all the transition rates q ij which are non zero. Now for this kind of a birth-death process, we can basically denote rho i as lambda i over mu i + 1. And if you write out the balance equations or steady state equations, just like in the discrete case you will notice that there is a automatic balance across each of these transitions. So, basically you will have equations that look like P i lambda i = P i + 1 mu i + 1, which this is for i greater than or equal to 0. Therefore P i is equal to, so I can write P i + 1 = rho i P i, which again I can iterate as rho i, rho i – 1, P i - 2 and so on. From this I get P i = P 0 product j = 0 through i - 1 rho j.

(Refer Slide Time: 03:51)



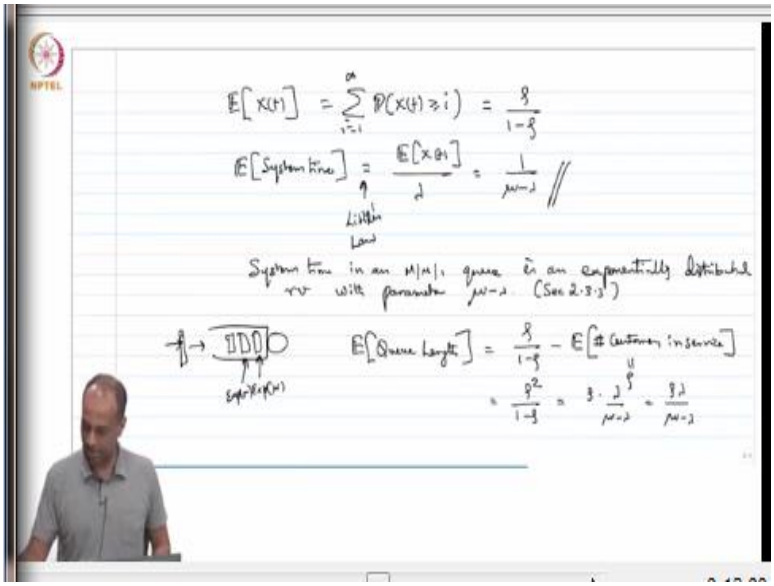
And normalizing, we get P 0 = 1 over 1 + sum over i = 1 through infinity product j = 0 to i - 1 rho j. So, this is P 0, from here you can just go back, you can just plug this back into that equation and get all the P i's. Assuming of course that the denominator is finite then you will get a non zero I mean strictly positive P i's and that is a steady state probabilities for this birth-death process.

So, here the lambda i the forward lambda i's are the birth rates at state i and mu i is the death rate at state i and lambda i is the birth rate at state i. And you can solve this is a very simple CTMC to solve. And now a lot of very important queuing systems Markovian queuing systems fall under this birth-death category. The first example is an M/M/1 queue which is very familiar to us, where each lambda i = lambda and each mu i = mu and lambda is assumed to be less than mu.

In this case you get P 0 = 1 over 1 + sum over rho power i - 1, i = 1 to infinity which is of course because we have assumed rho is less than 1, rho = lambda over mu which is less than 1 this will just be 1 - rho, this is a geometric series. And then we can calculate P i to be equal to rho to the i times 1 - rho for i greater than or equal to 1. So, these are the so 1 - rho is the probability that there are 0 customers in the M M 1 queue. And rho to the i times 1 - rho is the probability that there are i customers in the M M 1queue.

So, you can P i has the interpretation of either the fraction of time as we know fraction of time with i customers in queue which is also equal to the probability that X t = i probability that the i customers given X 0 = anything you want, it could be any j in the limit t tending to infinity. So, no matter where you start the probability that you have i customers in the queue as t becomes large is in fact P i which is rho to the i times 1 - rho.

(Refer Slide Time: 07:18)



We can also easily calculate expected X t which is the expected number of customers in the system which is just you can just take since this is a non negative random variable you can just take probability X greater than equal to i, i = 1 to infinity. If you just use this other geometric sum you get rho over 1 - rho, rho over 1 - rho is the expected system occupancy of an M/M/1 queue.

And if you look at expected system time with what is the total expected time spent by a customer in the system which will be equal to expected total number of customers in the system divided by lambda and this is by Little's law, which if you work it out comes out to be 1 over mu – lambda. And again we have taken lambda to be strictly less than mu, so on this is what it is. In fact, so this should not be surprising if you go back to your study of Poisson processes in fact we know we can find out the system time in an M/M/1 queue is an exponentially distributed random variable with parameter mu - lambda.

And this is something we have already encountered in one of the examples in the chapter on Poisson processes, let me just tell you it is in section 2.3.3. Essentially what happens is that if you have a M/M/1 queue, you have all these customers. So, you have to wait for, so each of these guys the service time is an independent exponential of sum rate mu. So, if I am an incoming customer who is just coming into the system, what is my waiting time or what is my system time? The system time is the total time I have to wait.

Let us for the sake of argument, let us say that this is FCFS first come fist serve. So, I have to wait for all these guys in front of me, which is all exponential mu. So, I have to wait for a certain random number of exponential mu random variables to finish and then I have my own service random variable which is exponential mu and then I am done serving. So, and how many people are ahead of me? When I enter the system I see some i customers in front of me with probability P i which I know to be rho to the i times 1 – rho, which is like a geometric distribution.

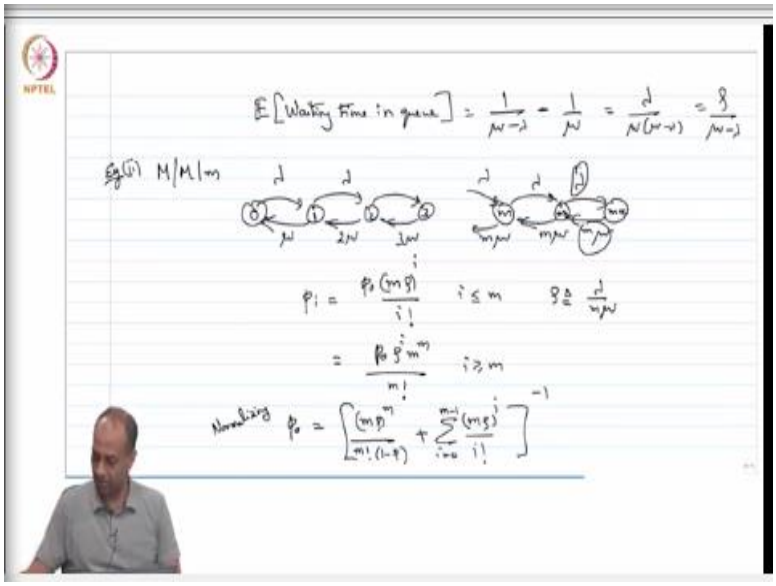
This rho to the i times 1 - rho is a geometric distribution offset by 1 perhaps. So, you have a geometric sum of exponential nu random variables, we already know that from undergrad probability we know that geometric sum of exponentials is a exponential random variable. Using that we can show that the system time in an M/M/1 queue is an exponential random variable with rate mu times 1 - rho with parameter mu times 1 - rho or which is mu - lambda.

So, it is not surprising that the expected time is 1 over mu – lambda, it is the system time is exponential with parameter mu – lambda, not only is the expected time mu - lambda, the system time random variable is exponentially distributed with parameter mu - lambda. And the argument

is what I just said, you are waiting for a geometric number of exponentials to finish. So, that should not be too surprising. Now you can also calculate expected Q length etcetera. Expected Q length is 1 over 1 – rho, which is just, so basically you take rho over 1 - rho the expected number of customers in server in the service, which is rho, this is turn out, it will be rho.

So, this will just work out to be let me just see, this will be rho square over 1 - rho, is that correct? Yeah, rho square over 1 - rho, this is. So, rho times rho by 1 - rho which is lambda over mu - lambda that is equal to rho lambda over mu – lambda. And if expected queue length is this much.

(Refer Slide Time: 13:01)



And also expected waiting time in queue is just total expected system time which is 1 over mu - lambda minus that customer's time in service which is equal to lambda over mu times mu - lambda which is equal to rho by mu - lambda. Which makes sense because if you divide the expected queue length by lambda by little you should get the expected waiting time in queue. So, that also makes sense, the sanity check, so that is good.

So, it is the M/M/1 queue is now we really fully understand it. You can also do other things like you can do example 2 M/M/ m. So, here the Markov chain, so you have m servers now. So, the arrival rates are all lambda, this is of course a birth-death chain. So, when you have 1 customer,

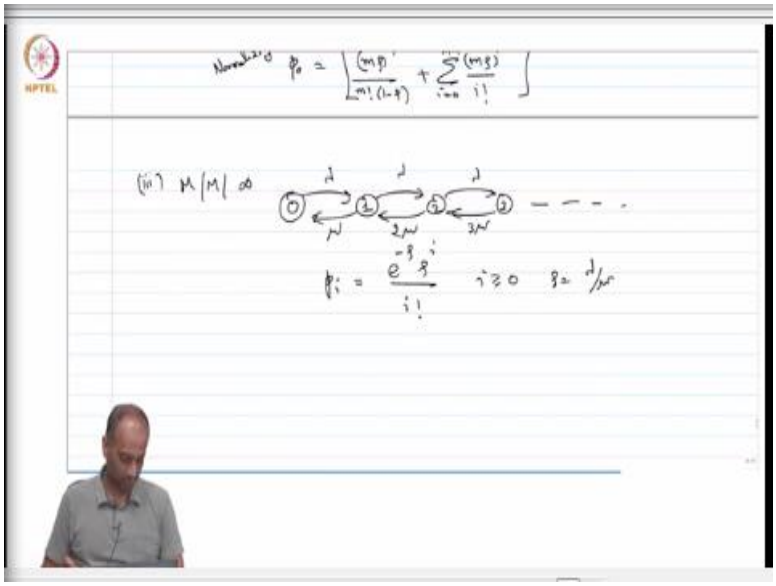
the service rate is mu and you have 2 customers the service rate is 2 mu, 3 mu and so on till M have m mu. But beyond that you have only m mu because there are only m servers.

And queuing only begins after there are m customers in the system. There are m servers, so whenever there is less than or equal to m customers in the system they will all be in service and beyond m, m + 1 onwards they will queue up. So, you can draw a Markov chain like this, a Markov process like this plus this should be m + 2 and so on. So, this is also a birth-death chain except the mu i's are different till m mu.

So, if you just work this out, if you just do the birth-death process calculation, you get P i = P 0 m rho to the i over i factorial for i less than or equal to m, where rho is now defined to be equal to lambda over m mu, m mu is the total server capacity. So, I am now defining lambda to be the ratio of that to that, not lambda over mu. So, if you just look at if you just write out the balance equation, this is what you get. And you get P i = P 0 rho to the i m to the m over m factorial for i greater than or equal to m.

And you can now solve P 0 by normalizing, so if you work out P 0 by normalizing get P 0 = m rho to the m by m factorial times 1 - rho plus whole to the -1 some mess, it does not simplify in any beautiful way or anything. It is what it is but these are all strictly positive numbers, so you have the steady state probability of there being i customers in this M/M/m system. From which again you can calculate the expected number of customers in the system time expected, waiting time and all that. It will all be 1 big mess as you can see but you can calculate it.

(Refer Slide Time: 17:16)



And as a particularly nice case is if you have M/M infinity, which means you have infinitely many servers, this corresponds to, so there is no waiting at all in the system. So, you have lambda mu, lambda 2 mu, lambda 3 mu and so on. So, for this situation you will get P you can show that P i will be equal to e power -lambda over mu it is e power – rho, rho to the i by i factorial for i greater than or equal to 0 and rho is equal to lambda over mu. So, it is a Poisson distributed in steady state there are Poisson number of customers in an M/M infinity queue, where the Poisson parameter is rho.

In fact this should not be too surprising, if you go back and look at your expression for the M g infinity queue which we did when we studied non-homogeneous Poisson processes, we got a similar expression. So, for the M/M infinity which is just a special case of M g infinity which we modeled using non-homogeneous Poisson processes. We are getting the same answer; it is just a special case of something we have already studied. So, all this is very nice, this is all very simple calculations but they are important Markovian queuing systems.