Foundations of Cyber Physical Systems

Prof. Soumyajit Dey

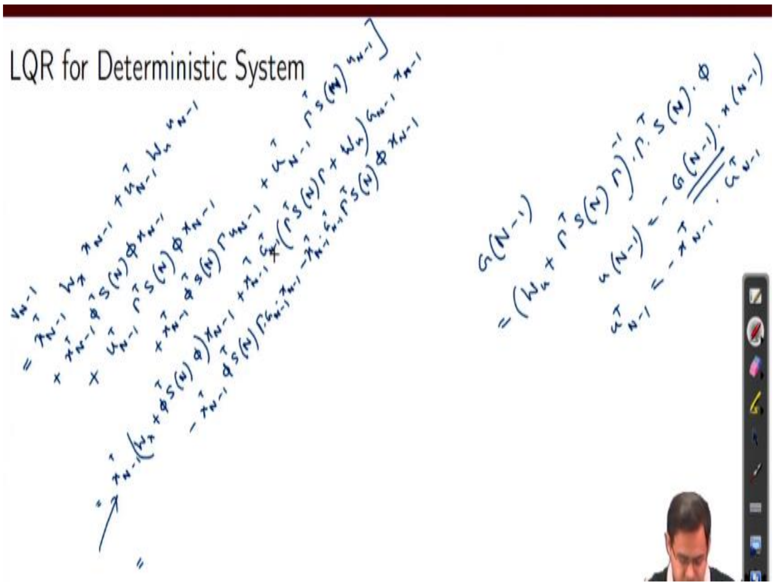
Department of Computer Science and Engineering

Indian Institute of Technology – Kharagpur

Lecture - 45

Quadratic Program Based Safe Controller Design - Continued

(Refer Slide Time: 00:25)



LQR for Deterministic System

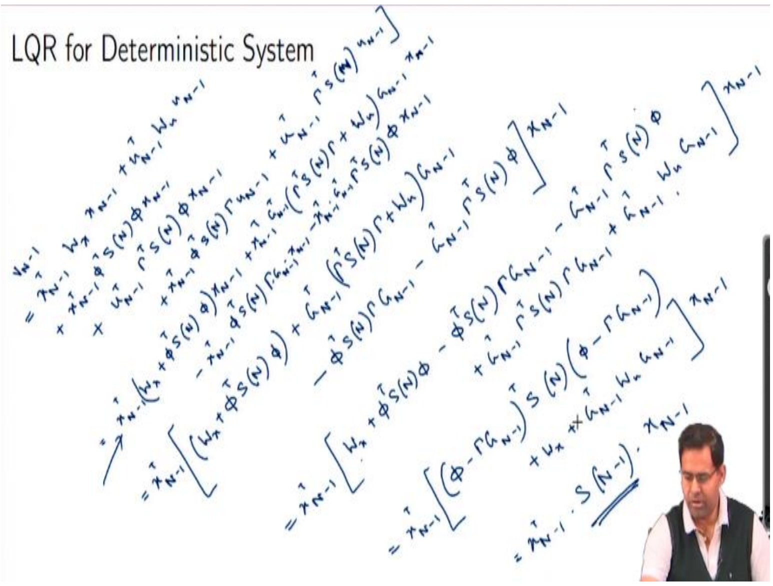
$G(N-1) = (W_u + \Gamma^T S(N) \Gamma)^{-1} \cdot \Gamma^T S(N) \cdot \Phi$
$u(N-1) = -G(N-1) \cdot x(N-1)$
$u^T_{N-1} = -x^T_{N-1} \cdot G^T_{N-1}$

Welcome back to this lecture series on Foundations of Cyber Physical Systems. So we will start right from where we left it off in the previous lecture. So we have derived the N minus one th step but we want to figure out what is the general way to compute the LQR gain of the controller in all the steps right. So let us proceed what we if you remember we have already given you an expression for this intermediate cost function for the last step N S minus V_{N-1}. So in that one let us try to apply this equation of G right. So for doing that fine let us if you remember that was one function where we had a significant number of terms right so fine let us let us then first write down that function so that was my bigger expression and from this expression we made things a bit easier by clubbing some of these terms so this was my V_N right now you see we have this G(N) which is kind of linking of my U(N-1) equal to minus G(N-1) into x right. So wherever we have U that this expression can be used up right.

So fine let us apply that now if we do that this first term will be staying right and let us see where we will be having the changes. So what will happen is this will be changing right. So this will be a sign change and then well I had U transpose N here right. So that would become U transpose N minus 1 here so that would become like G(N-1). So I am skipping the brackets for space here right times x(N-1) right. So let us carry out in a similar way what all changes fine. So we will also find some change here. So that is again. So I believe this was U transpose right if we just check yeah it was U transpose N-1.

So that would mean U transpose N-1. So that would give me minus x transpose times G transpose right. So fine then let us write in that way x transpose N-1 G transpose N-1 followed by whatever was there right and similarly in the first two terms here there you have us right. So those will get changed and two minus signs will come which will eventually cancel each other. So fine then just write them off and here we have the transpose so fine that is how it will look like right yeah.

(Refer Slide Time: 08:44)



LQR for Deterministic System

$$V_{N-1} = x_{N-1}^T W_x x_{N-1} + u_{N-1}^T W_u u_{N-1} + x_{N-1}^T \Phi^T S(N) \Phi x_{N-1} + u_{N-1}^T \Gamma^T S(N) \Phi x_{N-1} + x_{N-1}^T \Phi^T S(N) \Gamma u_{N-1} + u_{N-1}^T \Gamma^T S(N) \Gamma u_{N-1}]$$

$$= x_{N-1}^T (W_x + \Phi^T S(N) \Phi) x_{N-1} + x_{N-1}^T G_{N-1}^T (\Gamma^T S(N) \Gamma + W_u) G_{N-1} x_{N-1} - x_{N-1}^T \Phi^T S(N) \Gamma G_{N-1} x_{N-1} - x_{N-1}^T G_{N-1}^T \Gamma^T S(N) \Phi x_{N-1}$$

$$= x_{N-1}^T \Big[(W_x + \Phi^T S(N) \Phi) + G_{N-1}^T (\Gamma^T S(N) \Gamma + W_u) G_{N-1} - \Phi^T S(N) \Gamma G_{N-1} - G_{N-1}^T \Gamma^T S(N) \Phi \Big] x_{N-1}$$

$$= x_{N-1}^T \Big[W_x + \Phi^T S(N) \Phi - \Phi^T S(N) \Gamma G_{N-1} - G_{N-1}^T \Gamma^T S(N) \Phi + G_{N-1}^T \Gamma^T S(N) \Gamma G_{N-1} + G_{N-1}^T W_u G_{N-1} \Big] x_{N-1}$$

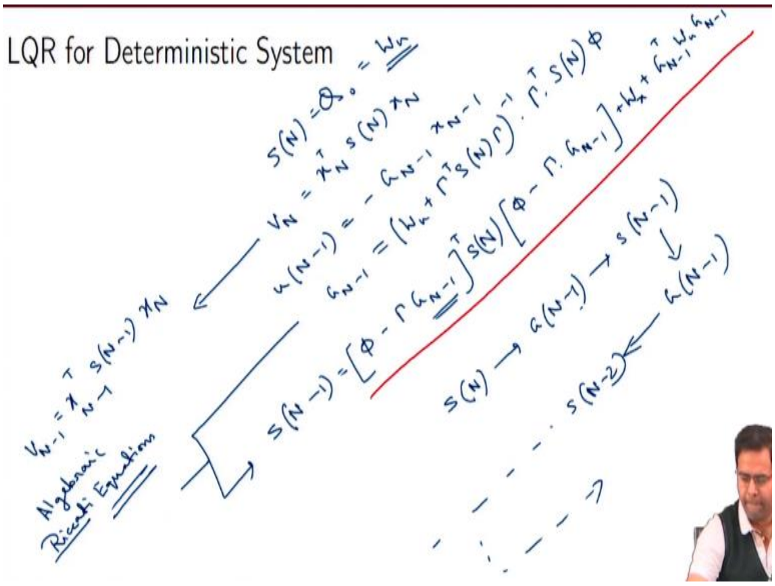
$$= x_{N-1}^T \Big[(\Phi - \Gamma G_{N-1})^T S(N) (\Phi - \Gamma G_{N-1}) + W_x + G_{N-1}^T W_u G_{N-1} \Big] x_{N-1}$$

$$= x_{N-1}^T \cdot \underline{\underline{S(N-1)}} \cdot x_{N-1}$$

So let us simplify further and let us see how things turn up now. So now you see as a result of this what what what am I having I am now having everything in a quadratic form of x minus 1 x(N-1) right. So that is that is the great thing about it. So I can take x transpose N-1 common here yeah. So that is it right. So I have an entire expression which is kind of in quadratic form of x(N-1) right. So that is nice.

So let us see what we can do about this. So as it happens this can be a bit simplified. So let us see what we can we can check out here yeah I mean basically we can just do this multiplication here right and distribute the terms over gamma right. Now the question is why would I do that because I want to create another specific form here which would be something like this. Let us so you see you can take certain things common here and you can write a nice form this thing in transpose S(N). So this is quadratic over phi minus gamma G(N-1) okay and whatever remains. So you have six terms four of them can be expressed in this nice quadratic form and then you have this remaining this and this. So in effect what you have is this right I can say this times S(N-1) times x(N-1). So if you recall we have been expressing this V function this partial V function in this form like quadratic form over x with the weightage function being some some S term right. So as you can see that we can again write x(N) V_{N-1} in a quadratic over x with this intermediate weightage being attributed to what we call as now S(N-1) because it is Nth step N 1th minus one th step. We had V_N equal to x(N) S(N) x(N) transpose S(N) x(N). Similarly we can write V_{N-1} as x(N-1) transpose S(N-1) and x(N-1). In effect what we have done we have discovered that well what will be the form of S(N-1) in general expressed in terms of this system parameter.

(Refer Slide Time: 15:24)



LQR for Deterministic System

$S(N) = Q_0 = \underline{\underline{W_u}}$

$V_N = x_N^T S(N) x_N$

$u(N-1) = -G_{N-1} x_{N-1}$

$G_{N-1} = (W_u + \Gamma^T S(N) \Gamma)^{-1} \cdot \Gamma^T S(N) \Phi$

$S(N-1) = [\Phi - \Gamma \underline{\underline{G_{N-1}}}]^T S(N) [\Phi - \Gamma \cdot G_{N-1}] + W_x + G_{N-1}^T W_u G_{N-1}$

$V_{N-1} = x_{N-1}^T S(N-1) x_N$

Algebraic Riccati Equations

$S(N) \rightarrow G(N-1) \rightarrow S(N-1)$

$\downarrow$

$G(N-1) \rightarrow S(N-2) \ldots$

So what can we say now we can say that we have got two equations one is I derived x(N-1) in terms of the others but now using that I can just write what is S(N) in general right. So fine let us write. So yeah let us write both the expressions whatever we had. So first we had this expression for G(N) right and even before that we had this expression over S(N) right.

So let us write all of them together here. So what we did was we took S(N) to be Q naught W_U right. That was our. So that means S N is known to me by system parameter and then we said that well let the function V denote the partial sum. So that means in the last step there is no sum at all what it remains is just the last states weightage right. So and that was this thing right yeah and then we are now kind of summing up the overall discussion here what we did was we computed well we are able to see that well with all these what would be my gain in the last step.

What should be my control inputs generating function in the last step and that came out nicely as this N-1 x(N-1) where we had an expression for N-1 thing as well this is my last step control input and given that I know all the parameters here I know W_U S(N) and every stuff here right. So I can evaluate G(N-1) question is how do I compute the previous ones.

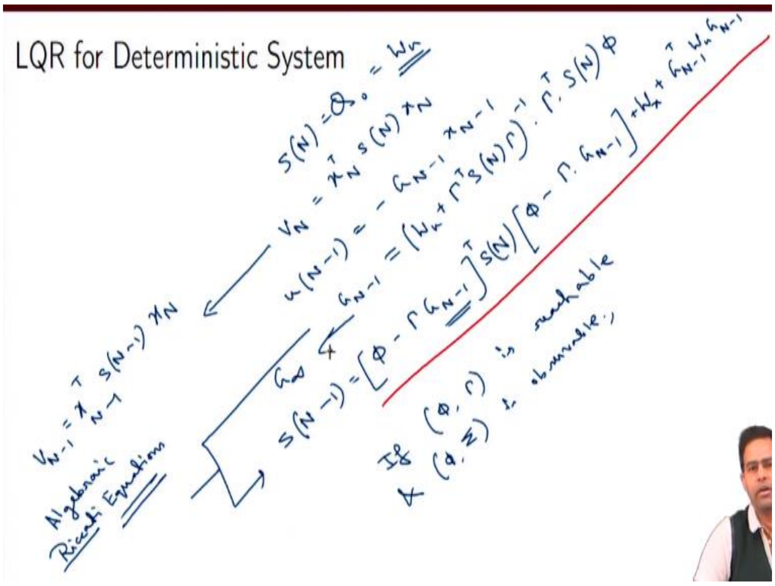
So for that what we finally proved was well for all the previous ones just like I had this so for N N-1 we were able to compute this this form and that gave me this nice expression that well this S N minus one how does it really look like right. So we figured out that S N minus one look like this. So that is what came around right we got an expression for S N minus one right.

So it is phi minus gamma G(N-1) S(N) phi minus gamma G(N-1). Again I mean here is a transpose here. So it is in quadratic and plus W_x plus G transpose N-1 W_U G(N-1) right. Now how does this help well you see this is these are now we have got a nice recurrent form right. That means if this is S(N-1) the same expression holds for S(N-2) just by change the parameters of G by N-2 similarly you can compute S N-3 by changing the parameters of G to N-3 right. But the question is how do I go about this. So what is happening is these two equations now together define a recurrence if you see. So you start with S(N) which you knew from S(N) you computed G(N-1) you apply G(N-1) here to compute S(N-1).

Now you apply this value of S(N-1) here right. If you put S(N-1) here what do you get you get G(N-2) here right. So from this you get G(N-2) you apply G(N-2) here you get S(N-2). So that is how you can now keep on evaluating all these terms right and so in that way you get all the control.

So you can just compute keep on computing these values right. That means you can compute all these LQR gains right and given that you can compute all these LQR gains you just apply U equal to minus G times x that relation to compute all the actual control inputs right. Now when you are solving this recurrence it would eventually stabilize right. So when this is stabilized then you you get a solution to this recurrence by the way this has a nice name this system of equations has a nice name this is known as is known as algebraic Riccati equation based on the name of the inventor.

(Refer Slide Time: 21:52)



LQR for Deterministic System

$S(N) = Q_0 = W_x$

$V_N = x_N^T S(N) x_N$

$u(N-1) = -G_{N-1} x_{N-1}$

$G_{N-1} = (W_u + \Gamma^T S(N) \Gamma)^{-1} \cdot \Gamma^T S(N) \Phi$

$S(N-1) = [\Phi - \Gamma G_{N-1}]^T S(N) [\Phi - \Gamma G_{N-1}] + W_x + G_{N-1}^T W_u G_{N-1}$

$V_{N-1} = x_{N-1}^T S(N-1) x_N$

Algebraic Riccati Equations

G_∞

If (Φ, Γ) is reachable & (Φ, Σ) is observable,

Now it also happens to be the case based on certain con conditions we can actually argue that there exists a nonnegative unique solution to this set of algebraic Riccati equations but for the time being let us.

Let us not get into all these conditions what happens is if you are doing this LQR gain control design in matlab I mean you can understand we decide we discussed this theory to give you give you an idea about how the optimal control problem is actually formulated and solved. But in matlab what they will do is they will if you give the system parameters I mean there is a formula for LQR you just apply it it will autofill in itself the matlab routine we will solve this recurrence it will check this observability of the system etcetera.

That means there are under certain conditions under which this recurrence is going to give you an eventual stable solution right and based on that it will give you for the system of equation a unique symmetric answer okay.

So and that answer is what we call as your LQR gain okay. This J this G I mean it will eventually stabilize to some G infinity let us call it.

So and you can actually try this for any standard system you take the system take our cruise controller systems and instead of using the matlab routine you can actually try solving this reoccurrence over some iterations and you can just plot the gain value and you can check that if it stabilizes okay. You will see that after some iterations none of G and S are changing so that means they have stabilized right and that is what matlab is going to in independently do it for you. If you use those I will give yet another functions. So fine with this maybe we will end this lecture. Thank you for your attention.