Foundations of Cyber Physical Systems

Prof. Soumyajit Dey
Department of Computer Science and Engineering
Indian Institute of Technology – Kharagpur

Lecture - 44

Quadratic Program Based Safe Controller Design - Continued

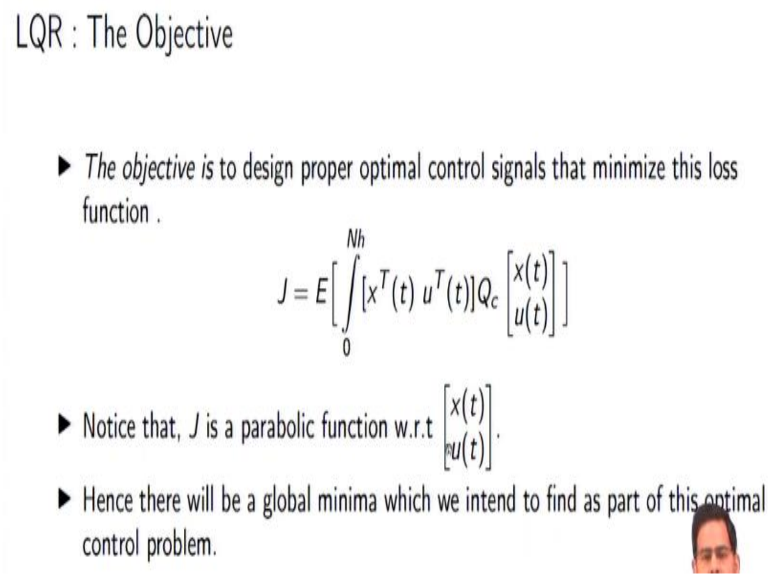
(Refer Slide Time: 00:32)

Course Organization

Topic	Week	Hours
CPS : Motivational examples and compute platforms	1	2.5
Real time sensing and communication for CPS	2	2.5
Real time task scheduling for CPS	3	2.5
Dynamical system modeling, stability, controller design	4	2.5
Delay-aware Design; Platform effect on Stability/Performance	5	2.5
Hybrid Automata based modeling of CPS	6	2.5
Reachability analysis	7	2.5
Lyapunov Stability, Barrier Functions	8	2.5
Quadratic Program based *safe* Controller Design	9	2.5
Neural Network (NN) Based controllers in CPS	10	2.5
State Estimation using Kalman Filters (KF)	11	2.5
Attack Detection and Mitigation in CPS	12	2.5

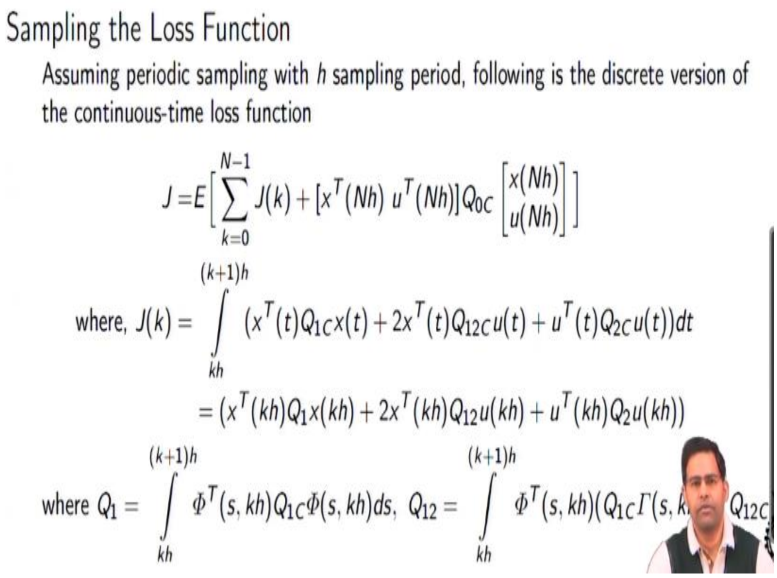
Hello and welcome back to this lecture series on Foundations of Cyber Physical Systems. So fine we will start from where we left in the last lecture. So we were in week nine we were discussing about this LQR design quadratic which is one of the quadratic program based techniques for controller design.

(Refer Slide Time: 00:50)



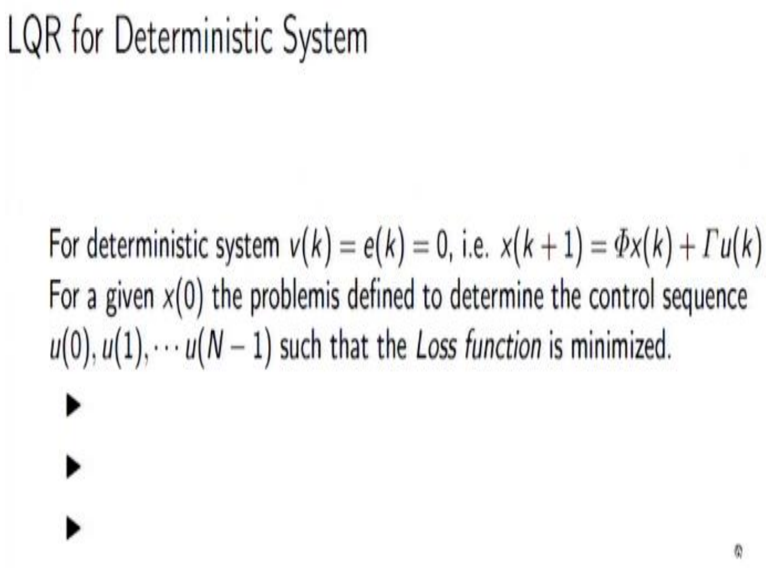
So fine I believe we have already set up the objective function here and we have already observed that this would be a parabolic function with respect to x and u together as the state right.

(Refer Slide Time: 01:02)



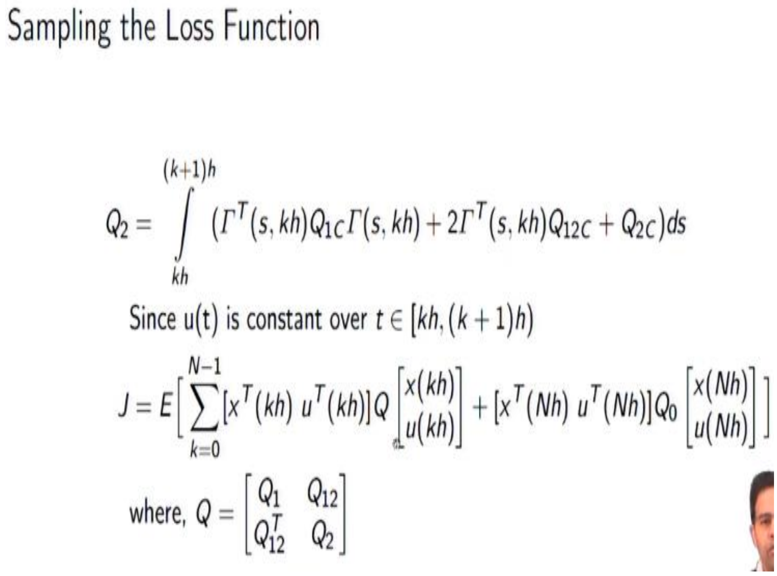
And hence what we want to do is we want to figure out the sequence of control inputs which is going to minimize this function.

(Refer Slide Time: 01:06)



And not only that we have also observed that we can apply this principle of optimality here. That means what we will do is we will solve the method in a by unrolling back from from the Nth step to the 0th step because we will assume that the control inputs we are trying to figure out they will be such that they will be optimal with respect to the state where a decision is taken okay, so that all the future decisions continued to be optimal like that. So fine what we will do is we will we will try to figure out the last control input let us say the n minus one th assuming that all the previous control inputs have already been optimal and then like I said we will unroll back in time. So let us see how we approach this optimization problem.

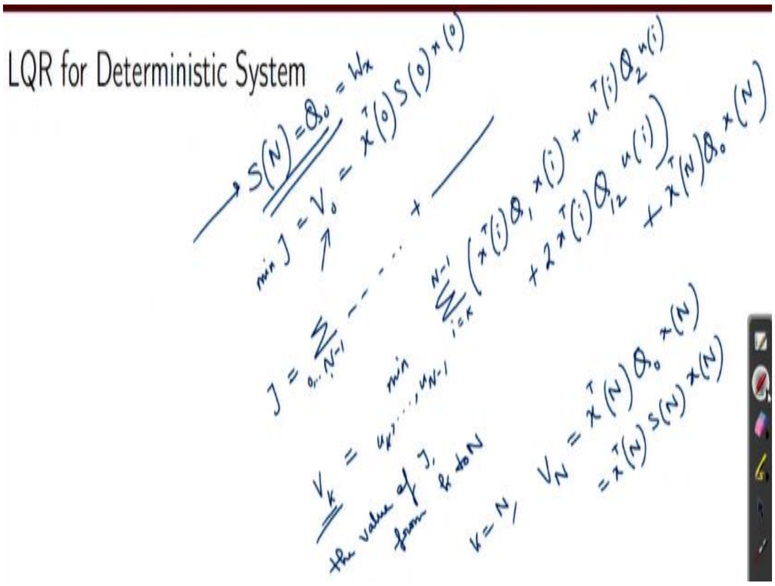
(Refer Slide Time: 02:06)



So what we will do so just to recall if you remember we have already set up this cost function here okay and for this cost function I have a quadratic form over x and u right and the summation over that and finally the last state also appears as a quadratic term. And here we have all these components of Q written down like here and we have already defined them earlier.

So we have already defined them earlier like this Q_1, Q_{12} like that fine so most of this treatment has been taken from Armstrong's book on digital controller implementation. So if you want to study this thing in detail that would be the reference to go for okay. So fine let us let us go ahead with this problem solving right here.

(Refer Slide Time: 03:00)



So what we will do is we will first assume some component S(N) here that means with some value of a function S at the Nth step okay and let that be some so that is Q_0 which is a kind of let let us say that is equal to some parameter W_x here okay. Now what we do is we define the minimum value of the loss something like this. So let the minimum value of the loss be this for the initial state. So if you see we are bringing in many notations. We have defined something called S(N) which is a sequence S 1s 2s 3 like that up to S(N) okay and we are saying that well let that be equal to Q naught which is defined earlier and let that be equal to some W_x here okay and then we are trying to define this loss function in a in a different way using this variable V naught. So basically we are connecting the value of J with the number of steps we are considering in this loss function okay.

So let us let us understand what it means so if you recall what really was J. So J was something like this summation right, J was this summation over an Nth step right where you are trying to figure out all these 0 to N-1 control inputs and finally that N step value right. So what we do is let us consider that function, so technically J was like expectation of that summation if we recall.

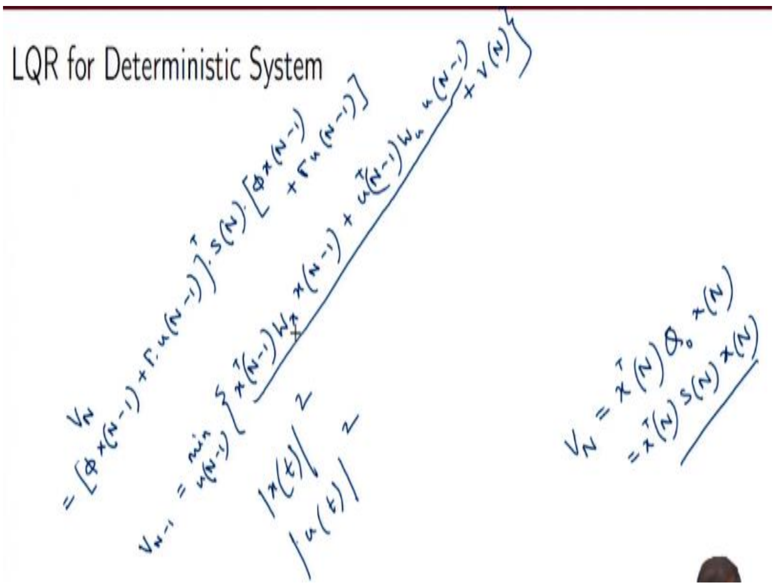
Now going inside that let us define something let us try to calculate what is the partial sum starting from the kth step up to the Nth step. So that means we are trying to figure out this control inputs U(k), U(k+1) up to U(N-1). So we are just writing the usual function here. So this was our original function right.

So just consider we are considering I mean we are taking it not from 0 to N minus 1 but from some kth step to N th minus 1 step okay. So then we start calling it as something like V_k okay and of course you have the Nth step value. So this is it right. So this is what is it right. Since this is the. So that means if I say set k to N and we have V_N right and V_N would not have this summation at all so V_N would be like this x transpose N this final weighted value Q naught x(N) right.

Now we have taken that to be S(N) right here right. So we will have V_N as x transpose N, S(N), x(N). So we are using this relation right here okay. So now let us try to do something let us let us mind our objective we want to figure out what is the last optimal control input right assuming all the previous decisions have already been optimal.

So what we need to do is this is what we have V at the Nth step we want to figure out u n minus one so we need to unroll this thing and create suitable values of x and related them basically with suitable values of x from the N minus 1th state right.

(Refer Slide Time: 09:14)



So if we just try to do that here. So we continue from here so what I am doing is I am replacing x(N) by its recursive definition in terms of x(N - 1). So this is what we have here right. So now let us try to do something similar and figure out well what is this V_{N-1}. So I hope this is clear right I mean all we have done is we have unrolled these values of x N and written them in terms of x(N - 1). Now if we try to derive what is V_{N-1}. Let us understand how it really should be right.

so V_N is just the last prime thing right there is there is nothing as part of the summation part of the original j but now when I talk about V_{N-1} it will at least have the last part of the summation right with the N minus 1 th step followed by this expression of V_N that is the last steps value right. So if you remember the form of j right. So just to recall you can just look over j was something like this that was the summation right so we just have V N with this value right now let us write V_{N-1}, V_N, V_{N-1} will have one term from here followed by the term for V_N right so let us see so it will just have the last element of the summation. So that would mean the quadratic terms of x and u in the N minus 1 th step and you have the weight for let us call this as W_x the weight for x like we defined earlier. So this is the minimization, essentially we are just considering the last step plus there should be V_N right.

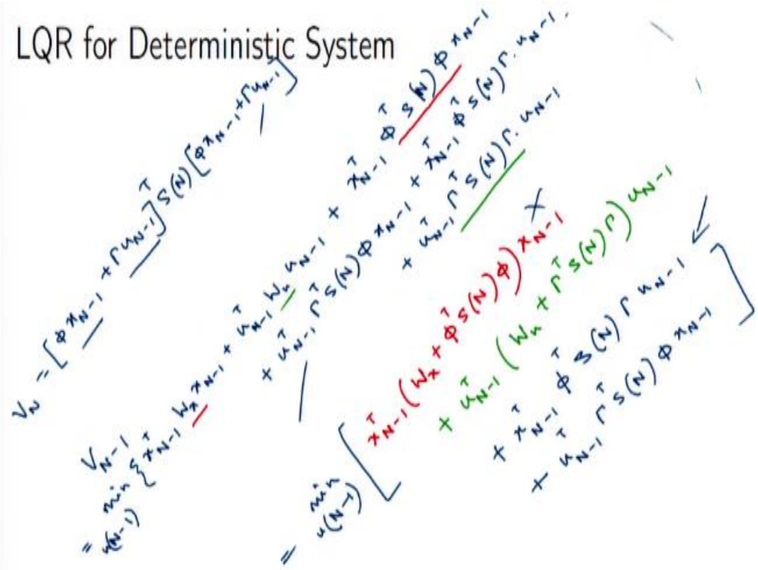
Of course I mean I need to have this inside the minimization because V_N is the last term so this is coming as the last term in the summation followed by V_N which is the last term. So that is what you have right. Now let us understand what we want to minimize. We want to minimize the this values of x(t) and u(t) squares right because we do not want the control to be abrupt and we want the state to steer away too much from the equilibrium which is at zero right. So well let us figure out what we can do.

The idea is since I want to minimize just these square terms that means excess we want to minimize the terms like these ones right because that has been my objective right. Now in the process we have this covariance terms creeping in right. So what I mean is fine I hope this is understood. So we can just delete these parts. So if you see in our original this this was the nice looking thing right but if you go back here you have terms like this right. So you have terms like this you have terms like this these are the ones I want to minimize and then the covariance terms we have I mean they also appear right but what we can do is we can ignore these covariance terms of x and u in our subsequent derivations and we will consider some distribution considering that for some distribution the covariance of x and u are zero.

So let us understand what I am just saying. So this is the loss function we are saying that well under some distributions we can assume this covariances between x and u used to be zero righ.t

I mean they are independent I mean of course that depends but overall let us understand what I want to do. I want to minimize the terms like x square and the terms like u square right. So that is why I will focus on these terms of the loss function and I will see that if I minimize them then also in a way I am minimizing the sum but you one may argue that there exist this covariance but we will we will ignore that for this part of the derivation here.

(Refer Slide Time: 15:42)



So the way we will ignore is like I said that we will assume that these values of x and u they are following some distribution where this covariance happens to be zero I mean there is no covariance relation between x and u. So fine let us now proceed over that. So I just wrote V_{N-1} equal to something plus V_N and we also expanded V_N earlier so let us put all of them together and write the formula now. So I am using subscript instead of bracket here for the sake of simplicity otherwise there would be too many of them. This is the weight parameter for the x's okay. So if you see we are kind of simplifying the derivation formulations now so instead of writing those Q 1s, Q 2s right. We in order to relate that well this is about x we are just writing the way W_x okay so similarly W_u.

Now if you remember here we had V_N and previous to that we had expanded V_N right and our V_N expansion let me just rewrite it was something like this right the phi x N minus one plus gamma u(N-1) right and that is transposed then you have S(N) and then again phi x(N-1) plus gamma u(N-1) right. So pretty much that is what it was so we need to carry out this multiplication and then write the V_N so that is the deal right fine let us do that so this would really mean that well this this will be transposed this will be transposed and then they will get multiplied right. So I am just writing this transpose followed by multiplication step right here.

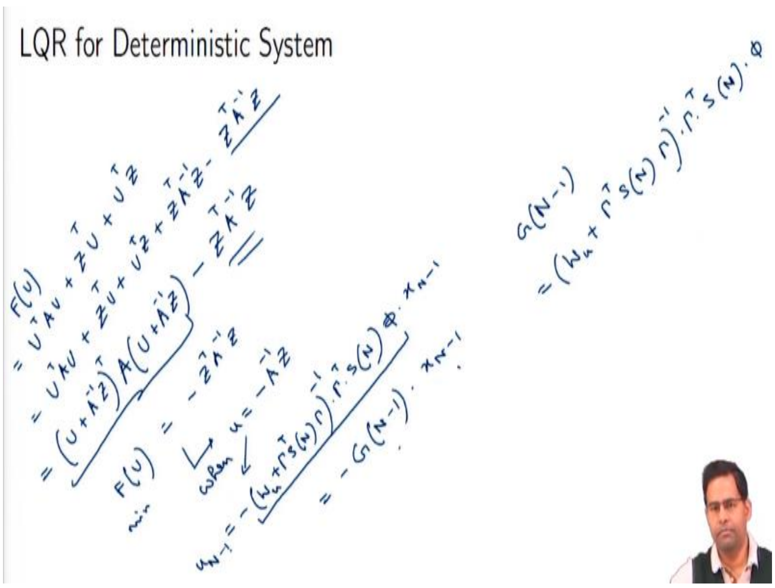
So this will go forward right so x transpose N minus 1 phi will be transposed then S(N) will be there right and then I have phi then x(N-1) plus so I am expecting four terms out of here right.

So you will have u transpose n minus one gamma transpose S(N), phi x N minus 1 right and then again take this and this term needs to be multiplied now with the other two. So that is what we get right. So fine we can write this in a modified way something like this. So we can actually combine certain terms here for example you can I think you can combine because they are all in the quadratic form of x right. So you can combine these two and similarly you get you let us identify the quadratic terms of u so you can combine this and this right. So let us do that then yes yeah.

So yeah but of course there will be more terms so let us not put this bracket here and instead let us just nicely write down the other terms right but the good thing is we have successfully written the quadratic terms of x and u right. So fine now let us just consider these terms here right the terms relating x and u right we have done we have created the ones for quadratic in x and quadratic in u and here are the other terms. So we have taken care of this one here and then the one that is remaining is this one yeah.

So those are four terms right. Now let us observe something interesting we are trying to minimize over u right. Now if you see the first term it cannot have any role for that minimization so this does not have any u at all right so fine let us consider the other three terms and let us try to rewrite this minimization in a in a well-known form. Let us let us do that.

(Refer Slide Time: 23:14)



So what we are doing is we are considering these three terms right and we will try to write them up. So we write a function F of U which is of this form U transpose A U plus Z transpose U plus U transpose Z where of course let us see what are these. So if you see U transpose A U by that we mean this term and then you have Z transpose U that is the second one and U transpose Z is this last term right. So that would really mean something like this right let me show.

So this is done right we do not need to bother about this because this does not have U but what we have effectively done here is we have considered this as U transpose A U so this is my A right and this is my Z transpose U right.

So Z let us see what Z is. Z is practically this sorry this is my let us see yeah. So A is this W U plus gamma transpose S(N) gamma and Z is well this is U transpose Z right. So U times Z so essentially Z is this your phi transpose S(N) phi sorry gamma transpose S(N), phi, x(N-1). So that is that is your Z right. So if that is your Z then you take the transpose of that you you see that its airing up right so this thing if I assume this to be Z if I take Z transpose that is naturally coming as right so x(N-1) transpose then phi and getting transposed S(N) is like assumed to be a symmetric one by its form like we the way we have taken it right so S(N) is fine and then you have gamma transpose again a transpose of that becomes gamma so that is Z transpose here right.

So this is what you have right. So now let us do some modifications here. So what we can do is U so we I just rewrote the same thing but along with that I will just add and subtract one term. question is why? The reason I do this is because as it seems that now I can take these four terms together apart from these these four terms and I can again create a nice looking quadratic form. So that is what I see right. Now since this is a quadratic term right this is non-negative right so that only means that well this function F(U) can be minimal with a value which is equal to this so F(U) min would be this and this will happen when this U is equal to minus inverse Z okay. So what does this mean now we can write that well what does this mean U is U N minus one. And what is the inverse?

So this is my A right. So W_U plus gamma transpose S(N) gamma. So fine that is that is my A right so inverse of that times Z right. So what is Z. So this is my definition of Z here right. So if I take it up gamma, S(N), phi. In fact, I do not need this bracket also here yeah. So this is what we get and fine we can write this as some function G of N minus one which one are we writing well this much right this much is G(N-1) okay and that gets multiplied by x(N-1) so you see here is your control right.

So just like we are habituated to a form like U equal to minus k x you have U(N-1) equal to minus something times U(N-1) and we are in keeping with the step N minus one we are writing it as N minus one of G the N minus one th step value of the gain and the gain value comes up to be this based on our definition here right. So this is my gain matrix then so as we can see this method tells me that this function will be minimized because that minimization of the function was our objective right and if I if I minimize this function it is a loss function because it tells me that then the energy of the system is minimized with minimum value of x almost near to the stable state and minimum expense of energy for control right.

So what I have is G(N-1) equals it is a long derivation but it is just for your understanding of course I am not going to ask you to do this derivation as part of an exam question but you want to understand how a typical quadratic problem is solved this is a classic example.

So that is why we are actually taking the pain of moving through all these optimizations involved in solving this quadratic program. So that is it G(N-1) is given by gamma U, I mean W_U plus gamma transpose S(N) gamma whole inverse gamma transpose S(N) phi. So that is your N minus 1th step gain and you have a definitive way to calculate it given the parameters of the system.

Do I have all the parameters? Well if you inspect you have S(N) right but S(N) is defined as Q naught right. So that is not a problem. But the question is that is only the Nth minus 1th step now do I have to obtain expressions for all these steps I mean how do I really go about it. So let us figure that out. So what we will do is now we will use this result in the original definition of V_{N-1} that we were writing and let us see how I mean where does it take us.

So fine maybe we end this lecture here and we have already consumed my time right and so maybe we will start from right here in the next lecture. Thank you.