Optimal Control, Guidance and Estimation

Prof. Radhakant Padhi

Department of Aerospace Engineering

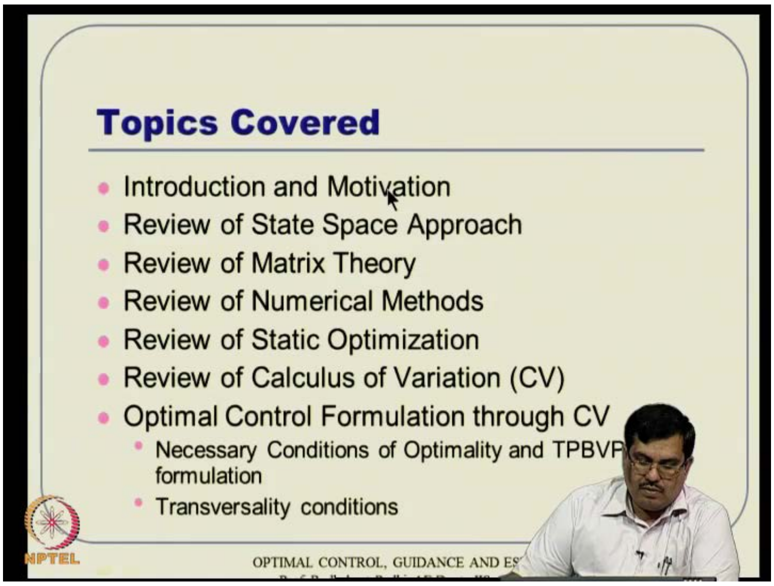
Indian Institute of Science, Bangalore

Lecture #39

Take Home Material: Summary – I

Alright, hello everybody; let us continue our lecture. We are almost at the end of the course really and before you wind up this course, I thought it is good to have a kind of some, some, some sort of a summary of all that we discussed because we have talked many, many things in this, in this course and before you wind up, it is actually good to see everything in some sort of a together, since actually it kind of an overview should have and even if you do not remember all the details, I think, at least this much you should remember at the end of the course actually.

(Refer Slide Time: 00:49)

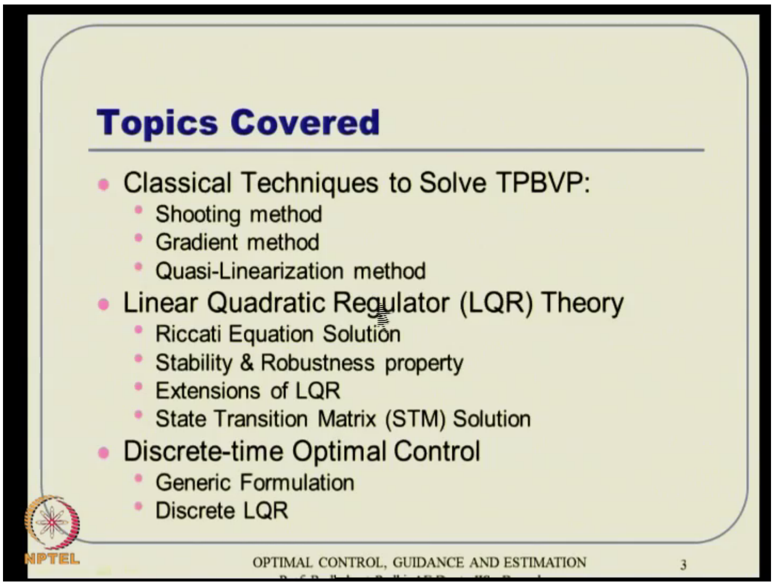


So, let us get started quickly and (( )) if you, if you remember the topics, that we covered in this course, we started with a small introduction and motivation. Then, quickly went through this review of state space approach and then, followed by review of matrix theory, as well. Then, we also had a review of numerical methods and followed by static optimization; then, we went on to calculus of variations and then started optimal control actually.

So, essentially, those of you who did not have too much of background, I thought it is still possible to kind of catch up with and that is, this review will actually help you there. And these two lectures, I will actually directly start with static optimization because the course is on optimal control. So, all the other things are where there is no time to review again, I think we will directly start with a little bit concept of static optimization.

Then, we, I mean, optimal control, we derive these necessary conditions of optimality and we can quickly see, that actually it leads to this 2 point boundary value formulation. We also had a generic transversality conditions and then talked various things about that actually.

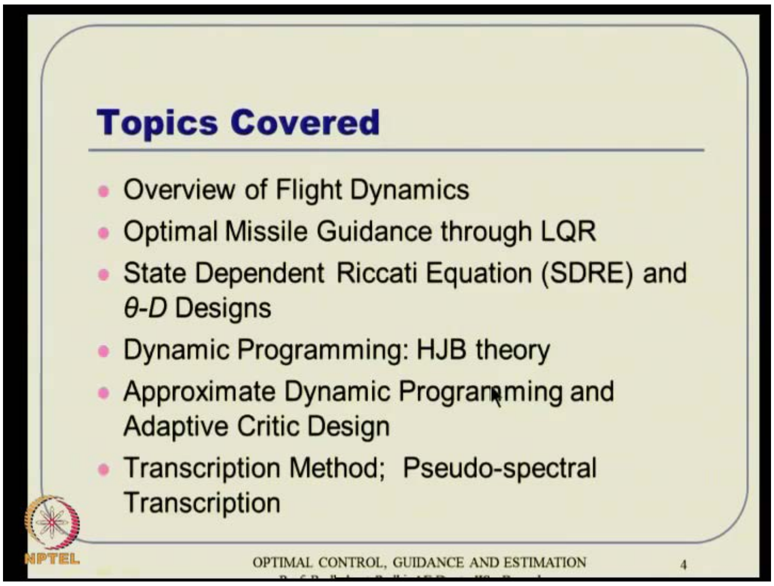
(Refer Slide Time: 01:52)



And went on to solve this classical techniques to solve this numerical problem, I mean 2 point boundary value problem. Essentially, we talk shooting method, gradient method and quasi-linearization method.

That was followed by this, this very standard LQR theory, which we have, we devoted few lectures to study this in, in entire detail. We, we studied Riccati equation solution, stability and robustness properties, extensions of LQR in various, various ways and then, state transition matrix approach solution. And then followed by, it is, it is the application in missile guidance and all that actually. Then, we went on to a discrete time optimal control of, first we had generic formulation, then discrete LQR as well.

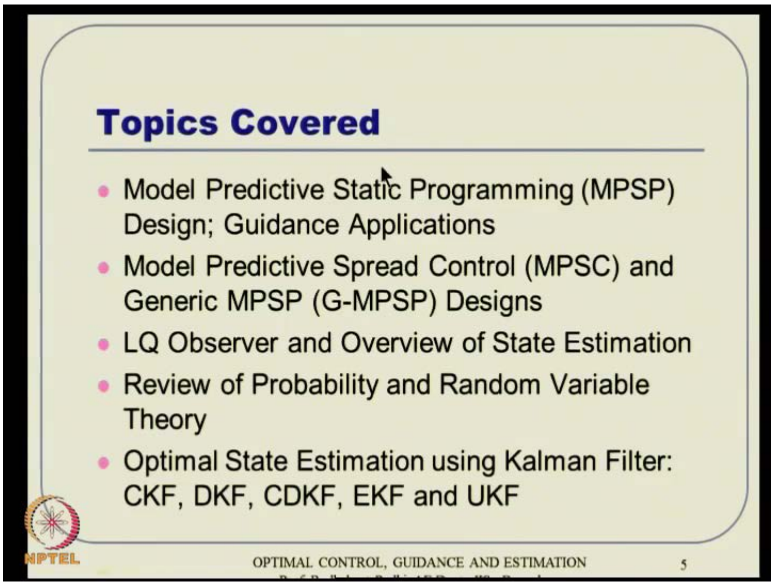
(Refer Slide Time: 02:31)



On the way, we had this overview of flight dynamics and then also talked about optimal missile guidance through LQR. And then, this we went on to non-linear formulations and started with these extensions of LQR idea in the SDRE frame work, followed by theta D design.

Then, we went on to dynamic programming approach, which are HJB theory, then approximate dynamic programming followed by adaptive critic design. Then, transcription method, pseudo-spectral transcription, especially we had a class actually.

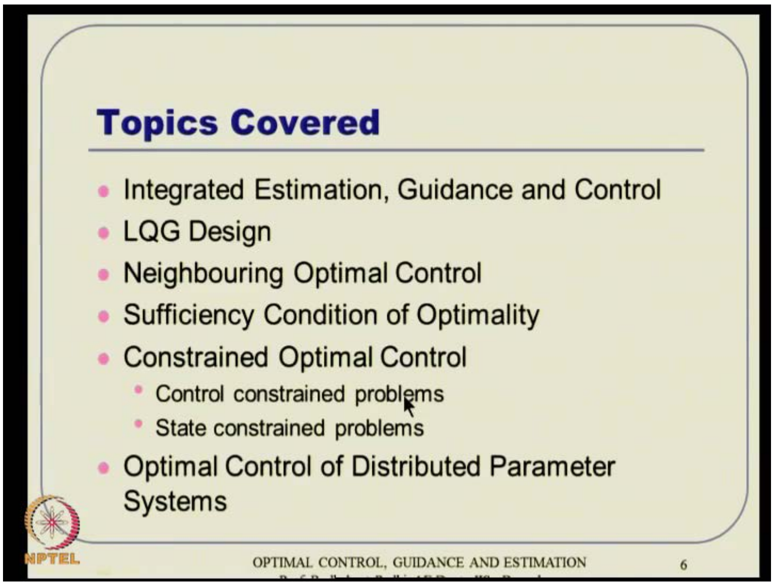
(Refer Slide Time: 02:59)



That was followed by this model predictive static programming and we also had this various guidance applications of this technique. And then, we extended, that to this spread control idea with control parameterizations and all, and then we also had a (( )) of very recent developments, which talks about generic MPSP actually.

And then, we also discussed LQ observer and then, had an overview of state estimation, that was followed by detailed kind of discussion about review of probability theory, random variable also and then, we went on to derive the Kalman filter using various frameworks. First, in continuous time framework, CKF; then discrete time frame work, DKF, the continuous discrete Kalman filter, I mean. So, where system dynamic will continues, but measurement is discrete and then, we extended these ideas. So, something called extended Kalman filter, which is heavily used and then, again we discussed both in continuous time as well as continuous discrete framework. And before you wind up, we also discussed something called uncentered Kalman filter and then, had a, had a glimpse of how to implement it really, actually.

(Refer Slide Time: 04:06)



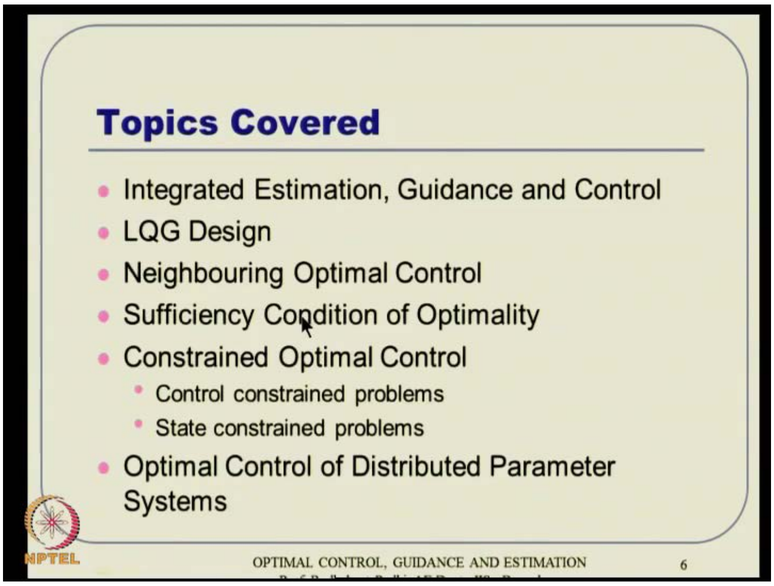
Then, we also discussed integrated estimation, guidance and control followed by this so called linear quadratic Guassian design, where you design the control from LQR, whereas states come from Kalman filter.

And then, we also talked little bit on neighboring optimal control, sufficiency conditions of optimality and then, we had about three lectures on constrained optimal control, where we discuss about control constrained problems, as well as, state constrained problems.

And before you wind up, we also add two lectures on optimal control application to distributed parameter system.

So, overall, if you see, the course topics are very wide and starting from here, starting from introduction, reviews of various materials to the basic to, to classical topics to numerical solutions, advanced topics to various applications, in flight control, all that actually.

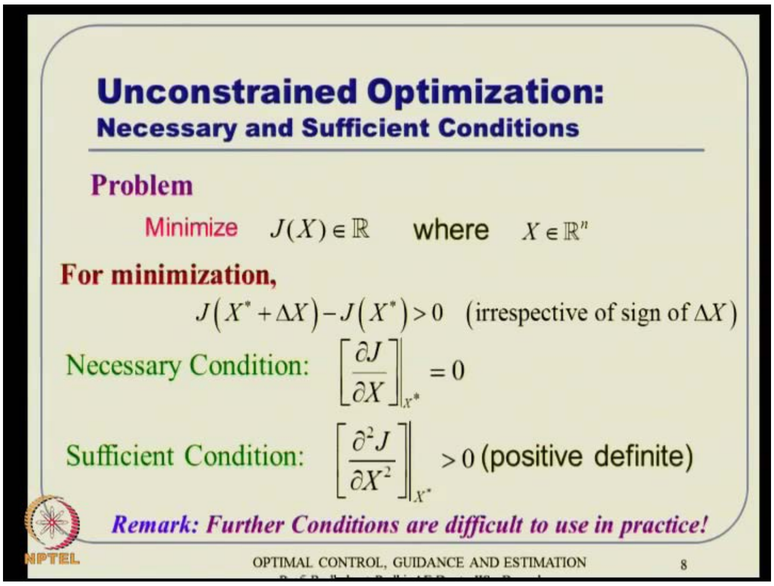
(Refer Slide Time: 04:58)



So, I thought towards end, it is, I will, I will just go one by one, slowly, and then try to kind of summarize all what all we have discussed. But also, on the way, I will not go to too much into aerospace applications, especially this flight dynamic guidance and all that actually.

I will confirmations to review of only the tricks and techniques basically. I will not follow any example, I will also not follow any, any applications per say basically.

(Refer Slide Time: 05:29)

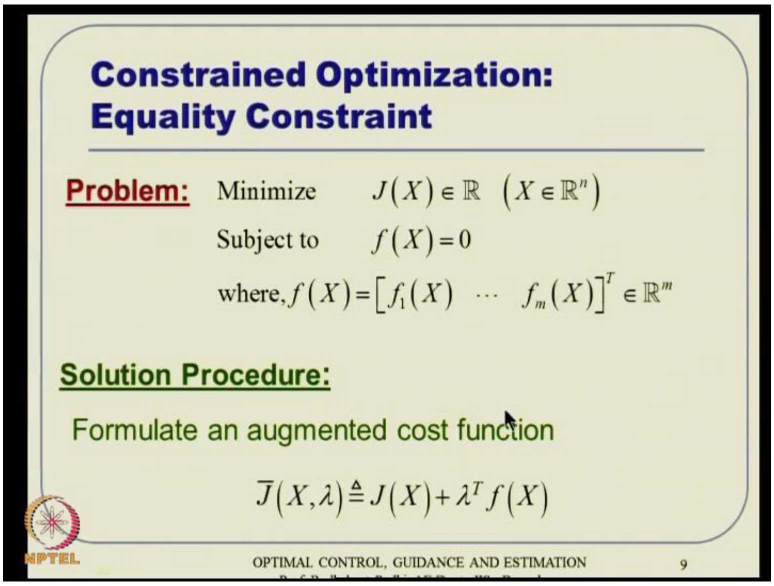


Alright, so let us get started. This is, is how we started the course on two lectures on, on static optimization and first, we discussed something like a scalar cost function of vector variables.

So, you want to minimize this J of X, X is a vector, but J is scalar and to, to have this condition met, what we observe is that J of X star plus delta X minus J of X star has to be always greater than 0 irrespective of the sign of delta X, so direction of delta X actually. It does not matter, which direction you go, but still in these relationships, if it satisfies, then X star is a minimum point actually and for that we, we put the Taylors series expansion for that and then, first term cancels out. The remaining term was sign sensitive, we want to make it insensitive thing like that. So, that led to the necessary condition, that first gradient should be 0 and followed by sufficiency condition, that the assigned matrix evaluated at X star should be positive, definite also.

And also observes, their further conditions are like 3rd derivative, 4th derivative. Also, things are possible, but they are also difficult to use in practice, as I mean, unless the problem is a scalar problem really. So, we did not bother so much on those aspects actually.

(Refer Slide Time: 06:36)



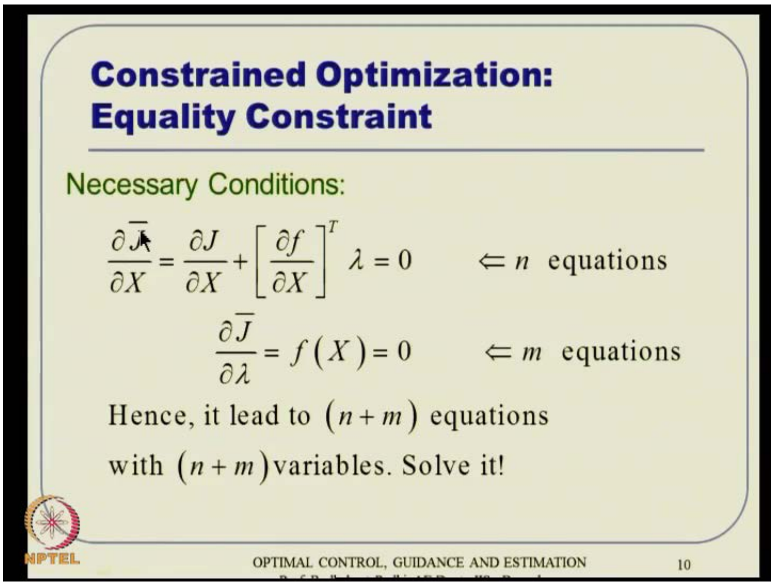
Then, you went for this constraint optimal for mean optimization problem and we started with this equality constraint.

So, the cost function was scalar, X was still a vector and this set of constraints, equality constraints, was actually m constant, m dimensional variable actually. So, the solution procedure turns out to be like that, J is to J bar is J of X plus lambda transpose times f of X actually because that is how it is.

And then, this comes from this, this Lagrange's existence theorem sort of thing, which tells, that ok, the existing set of Lagrange's variables lambda again, same dimension is f and if you, if you construct an augmented cost function like this and then it is equivalent to kind of, see this problem in, in n plus m dimension, the dimension becomes more, but essentially J bar becomes a free optimization problem in the, in the variables X and lambda together.

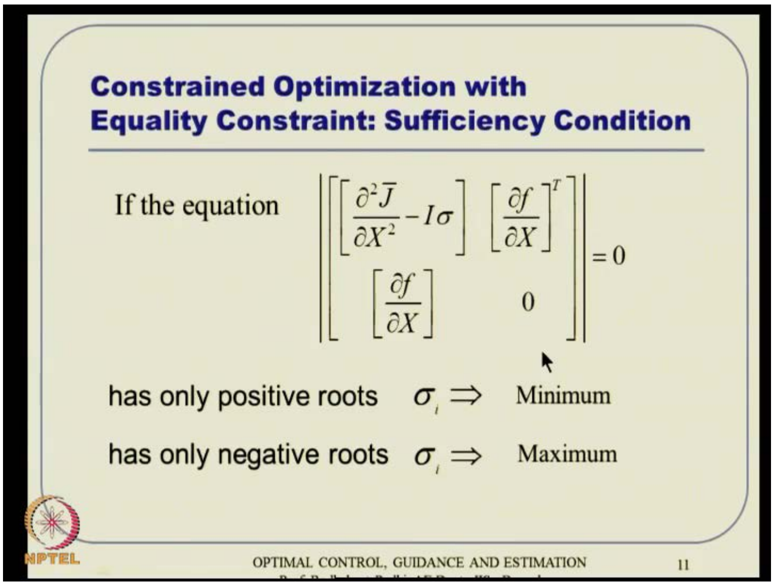
So, then, you can go back and apply these necessary condition in that setting and then came up with this idea, that del J bar by del X has to be 0 and del J bar by del lambda should also become 0 simultaneously.

(Refer Slide Time: 07:35)



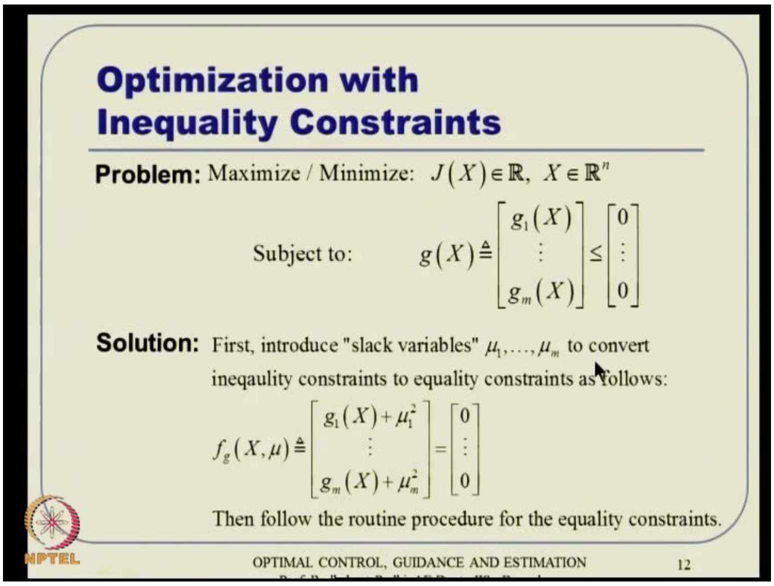
So, this leads to these n plus m equations with n plus m variables also. So, it is possible to solve together. After we solve, we discard lambda and take, take X star, whatever X star comes from their actually.

(Refer Slide Time: 07:54)



Then, we have a sufficiency condition kind of check, which if we want do, now you formulated this kind of a matrix, then take determinant, make it equal to 0 and then, solve for roots and if this equation has only positive roots, then it leads to a minimum and if it, it has only negative roots, it leads to be maximum problem actually.

(Refer Slide Time: 08:14)

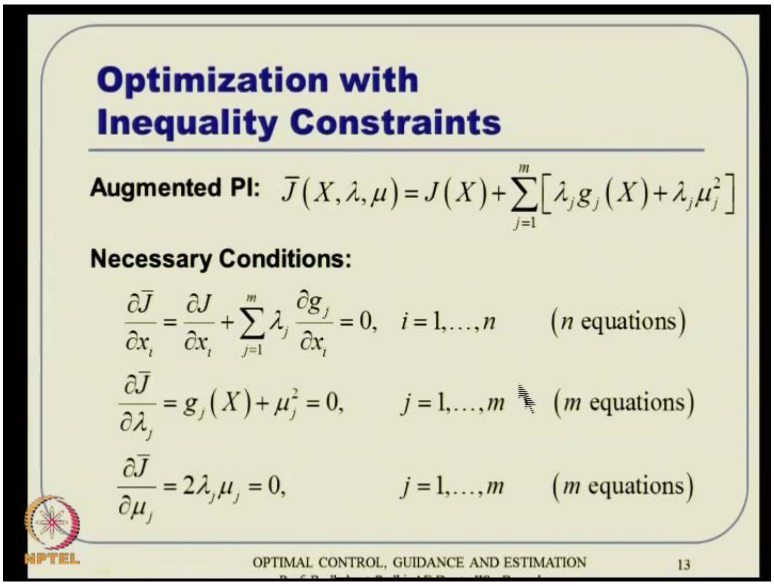


Alright, so next we followed up with, with this inequality constraints sort of thing. So, we had this same problem, I mean, the cost function is still a scalar, but it is now subjected to m inequality conditions actually.

In general, it can be less than equal to 0, but if it, even if it is positive, then you can flip the sign and put then everything, everything like less than equal to 0 basically.

So, then, the idea was to introduce a set of slack variables mu one to mu m and convert these inequality constraints to set of equality constraints actually. However, this, because it is inequality constraints, we do not know how far they are from 0. So, we really do not know the numbers for these actually, mu 1, mu 2 up to mu m, where we do not know the number. So, we just know that there exist m numbers like that for which these, these inequality constraints can be converted into equality constraints.

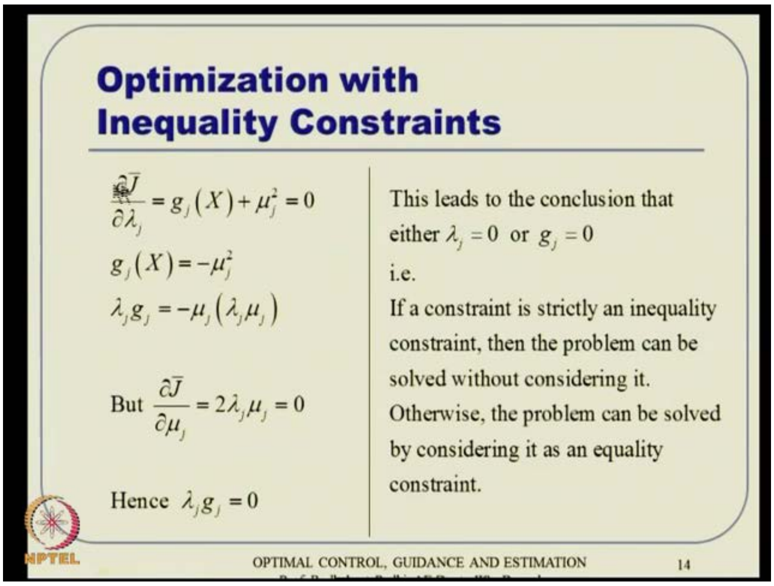
(Refer Slide Time: 09:05)



But then, there was the necessity of finding out conditions and all. So, then, we augmented, that, that J bar becomes not only a function of X and lambda, now it becomes function of X lambda and mu also. So, then, the necessary conditions expands to that del J bar by del x i del J bar by del lambda j and then del J bar del mu j also basically.

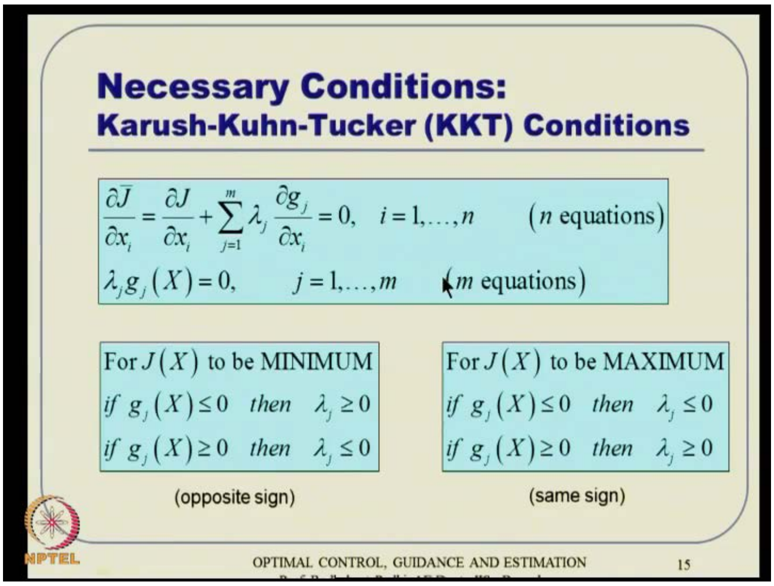
So, dimensionally problem, dimensionality of the problem becomes n plus 2 m now basically.

(Refer Slide Time: 9:29)



So, when try, when tried and then tried to analyze all these conditions and ultimately turns out, that it lambda j g j is equal to 0, that means, either lambda j has to be 0 or g j has to be 0.

(Refer Slide Time: 09:42)

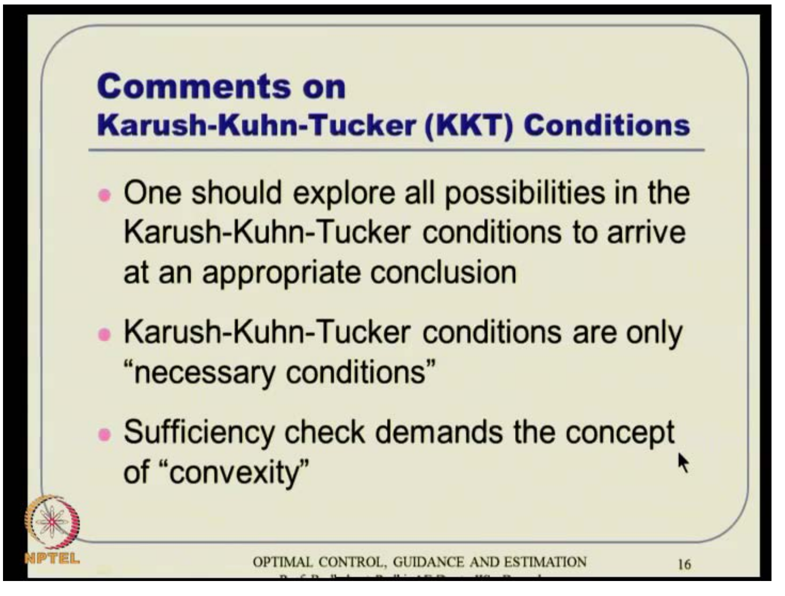


And then, we, we went back and then tried to analyze all these conditions and then essentially it gives us these so called KKT conditions, that is, Karush-Kuhn-Tucker conditions, which tell us, that these necessary conditions are something like this. This equation has to be satisfied, this is also satisfied.

However, you, now to, now we have to analyze the, the various possibilities and for J of X has to be minimum, then these conditions have to be satisfied actually. Similarly, for J of X has to be maximum, then these conditions have to be satisfied. That means, this, these have to be opposite sign and this has to be same sign actually, g and lambda, basically that way. And we also had examples on the way to demonstrate how it is possible, how it is possible to apply analytically and all that.

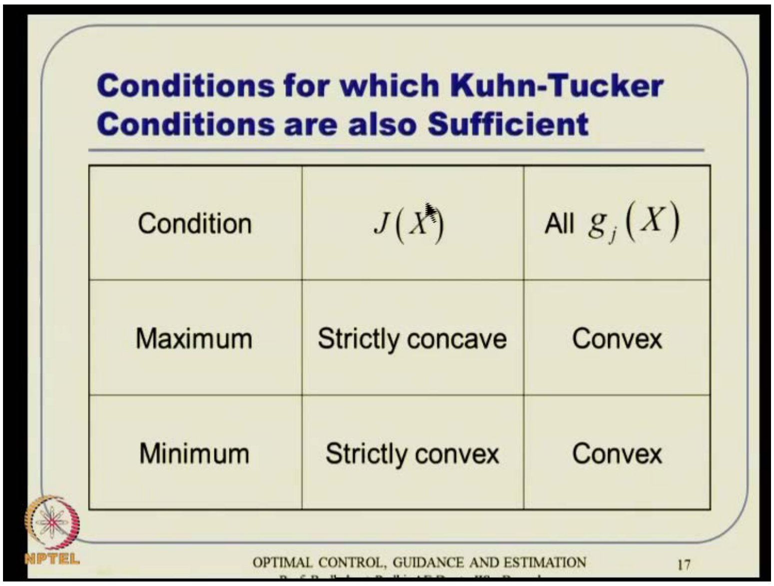
So, if you are then, I mean, curious or I hope you have gone through the lecture already, but in case you have not gone through, then you can see that lecture also. So, this is what it is.

(Refer Slide Time: 10:31)



I also explored all possibilities in KKT conditions to, to arrive at appropriate conditions. Then, also remember, that KKT conditions are only necessary conditions and sufficiency check demands the, the concept of convexity as well.

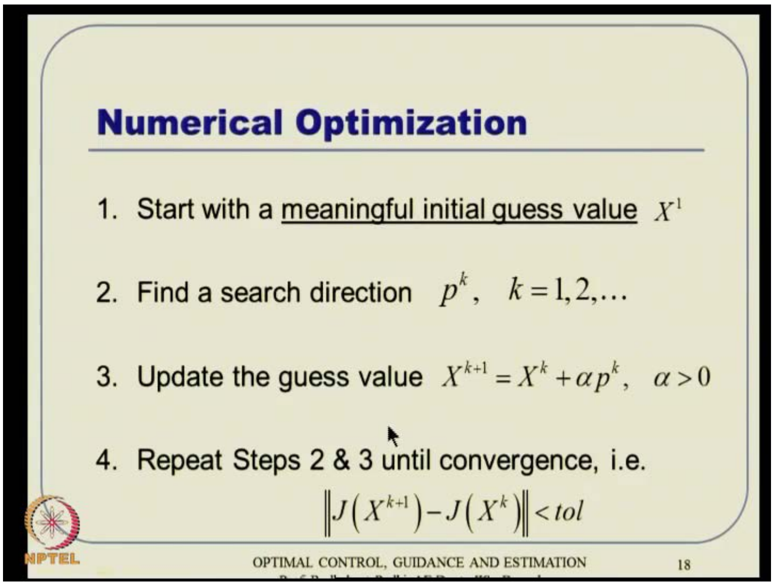
(Refer Slide Time: 10:47)



And we summarize the results, that if J of X is strictly convex, sorry, strictly concave, whereas g j happens to be convex, then it leads to a maximum case, otherwise if it is strictly convex, both are and this is convex, then it leads to a minimum case actually. Alright, so that, that kind of a thing.

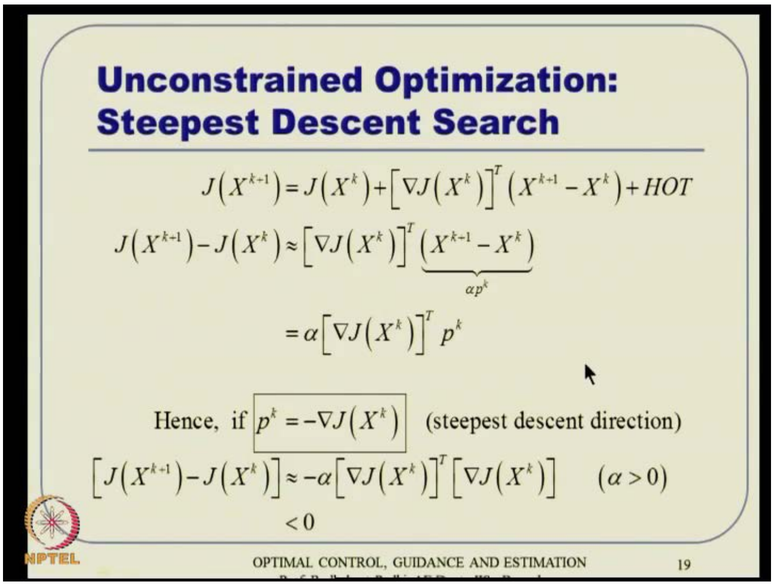
Then, we proceeded further, that we, we observe, that all these type of analysis is, is too much to do for, for practical problem and if the problem becomes more and more difficult, these conditions, analytically applying them becomes quite difficult actually. So, there was a necessity of a kind of go through a little bit of numerical solution approach as well. So, we studied a couple of techniques as well.

(Refer Slide Time: 11:24)



The first one, the philosophy remains same. So, the, the philosophy is something like this, you start with a meaningful initial guess and then, find a search direction. Some direction has to be found out, then update the guess value properly in that particular direction for some step and repeat the procedure until there is convergence there, means, you do not see any further improvement in this in this sense actually. So, this is the generic procedure for, for all numerical optimization.

(Refer Slide Time: 11:52)

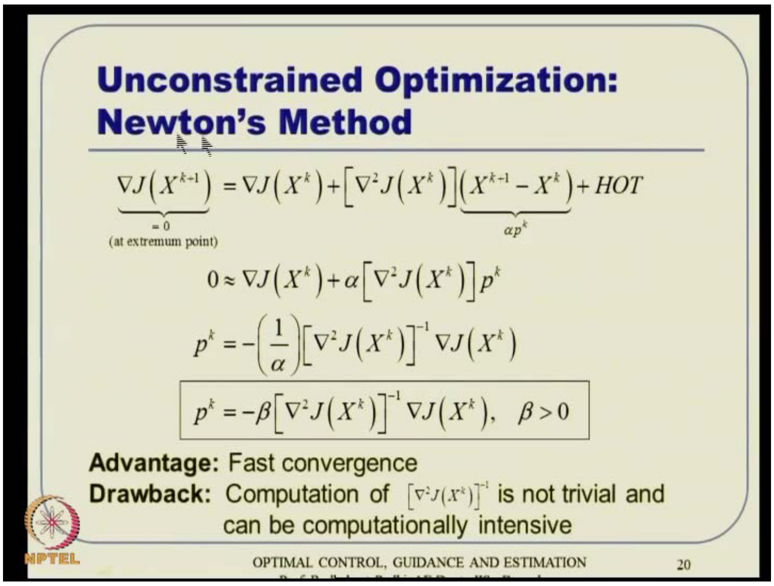


And then we saw, that the very first thing, that comes to mind is steepest descent search and steepest, I mean, this is very intuitive as well, the, all that you do is, this J of X k plus 1 is, you expand about J of X k in Taylor series.

So, then, neglect the higher order terms, keep only the 1st order expansion of the gradient value, I mean, this, this term. Then, it turns out, that this J of X k plus 1 minus J of X k, this term is approximately equal to that actually. So, if you consider, that as alpha times p k, then p k if you really want to, kind of, has a, have a meaning. I mean, decrease in the value of J of X k plus 1 compare to J of X k, then this quantity has to be always negative, I mean, negative into some, some quadratic terms sort of thing. So, it makes sense to, kind of, rho p k is nothing but a negative gradient basically.

So, then, it is guaranteed to have some sort of a quadratic term like these. So, it is guaranteed to be less than 0 basically. So, you will keep on decreasing basically, but this property happens to be very local. So, there was a necessity of fine tuning more and then there is concept like line search and then concepts like conjugate gradient methods and think like that actually.

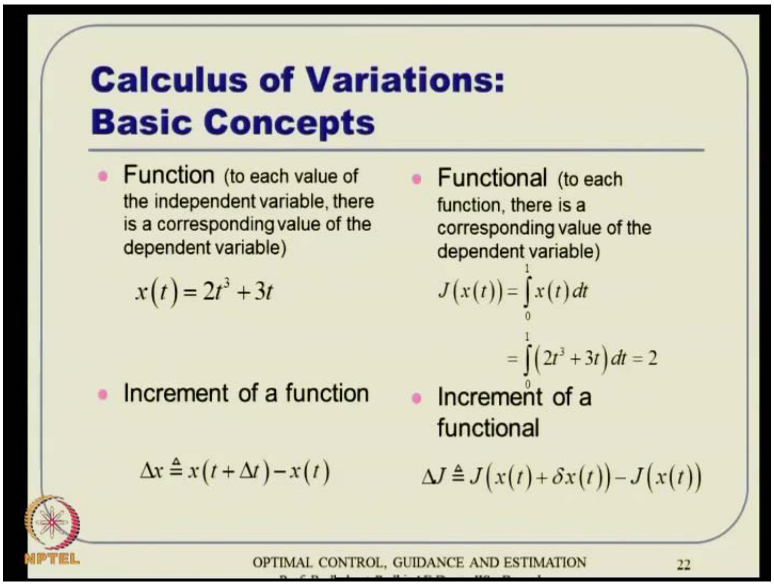
(Refer Slide Time: 12:59)



So, another method was something like Newton's method and Newton's method, little bit kind of takes a help of an, help of an analysis results with, tells us that, that on the optimum value, then gradient of, that (( )) grad J has to be 0. So, if you assume, that X k was not optimum, but X k plus 1 is to be optimum, then gradient at that point, actually X k plus 1 is to be 0.

So, now observing, that suppose you want to expand this (( )) this vector valuate function, now in Taylors' series, then it turns out to be something like that. Now, it gives us a platform to keep some sort of a like a semi quadratic term basically. So, we proceed of most of this similar sense, but instead of, I mean, this p k become sort of, function of gradient only, like here, here it was negative gradient, but here it turns out to be some sort of a gradient function, but I mean negative gradient. But this also helps actually this, this (( )) matrix inverse and all that, actually. So, advantage is, it leads to fast convergence, but drawback is this, this (( )) matrix inverse is not trivial to computing in general and it, it may need little bit more computation actually.

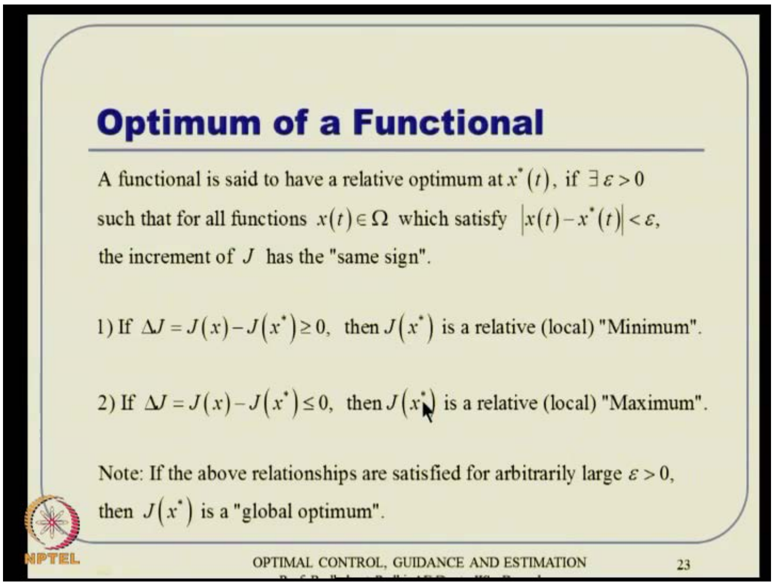
(Refer Slide Time: 14:18)



So, with that background, we, we followed proceeded to calculus as a variation and then, calculus of variations. First, we had some, some concepts like that. What we mean by function in, in general function space and what we mean by functional in calculus of variation actually?

Essentially, it is a kind of a function of function, but that a J of x of t, where t is the independent variable, whereas J, whereas X is a dependent variable of t and J is a function of X of t. Basically, that kind of things are called functional, but here if you simply take X of t, but t is an independent variable, then it is a function necessary, essentially.

(Refer Slide Time: 14:49)

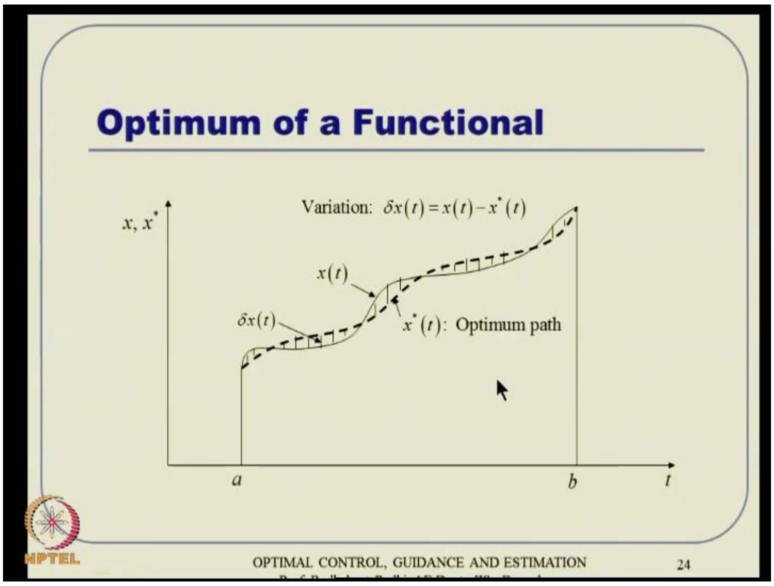


So, we studying various concepts and things like that about that and we finally, land up with an observation, that if you really want to claim, that some function is of, some functional is optimum, then what is the condition necessary basically?

So, it turns out, that if I define gradient of J something like J of X minus J of X star, which is greater than equal to 0, it has to be greater than equal to 0, then J of X star is a, essentially a relative minimum actually.

If gradient of J happens to be always negative, then J of X star is actually relative maximum. So, what does it mean? Now, you remember, that X is not just a value, but its, itself is a function actually and then, what you are looking for? That increment of J has to have same sign, l is, I mean, if as long as x of t lies in the neighborhood of x star of t essentially and if that neighborhood condition is relaxed, that means, epsilon can be very high in a bit like that, so arbitarily large epsilon. Then, then you tell, J of X star is nothing but a global optimal value basically.

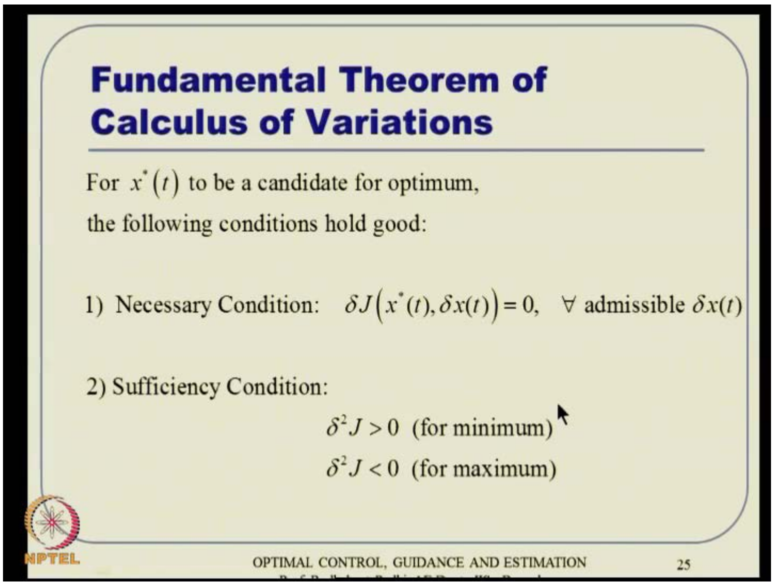
(Refer Slide Time: 15:53)



So, pictorially, it turns out to be something like this. If we tell that, that dotted line is actually optimum path, then around that we can draw any other curve and the difference between them is nothing but the variation actually.

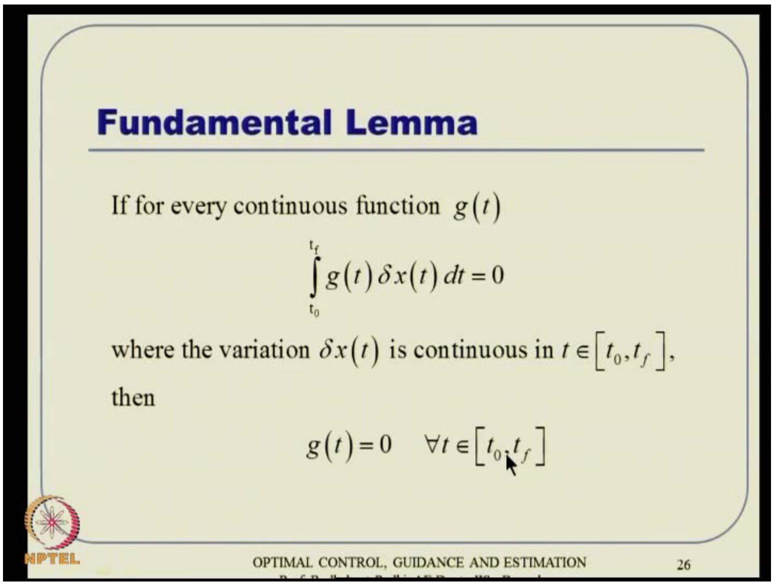
So, what we are telling her, that J of X means, you, J of X evaluated on the solid line path and J of X star is evaluated on the dotted line path actually and irrespective of what, whatever variation you want to take around the optimum value, if these condition is always satisfied, then J of X star is actually relative minimum, otherwise if this condition is satisfied, it is a, is a relative or are local maximum actually.

(Refer Slide Time: 16:35)



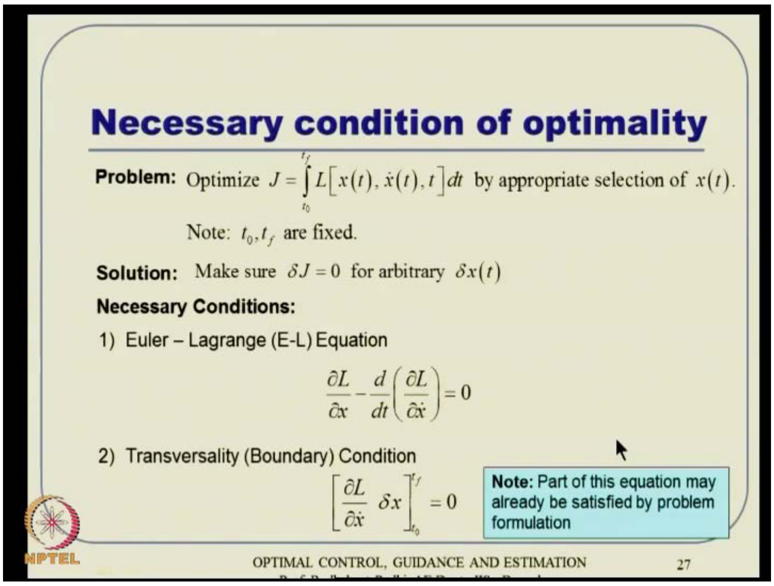
Then, we (( )), I mean, then, then it was, there was necessity of deriving necessary and sufficiency conditions for optimality and then, you observe, that very close to static optimization in calculus variation also turns out, that the first gradient, this first variation has to be equal to 0 for all admissible variations of X actually. And sufficiently (()) condition turns, that the 2nd variation has to be positive for minimum, our second variation has to be negative for maximum, actually.

(Refer Slide Time: 17:04)



Then, there was a fundamental lemma based on which we derive lot of necessary conditions and all that actually. And that can, that then tell us the, that for, if, if for every continuous function g of t this is valid, this integral is valid, where this variation is continuous in, in this time interval t naught to t f, then g of t has to be identically 0 throughout the interval and we actually gave a small proof for that as well, actually (()), alright.

(Refer Slide Time: 17:34)

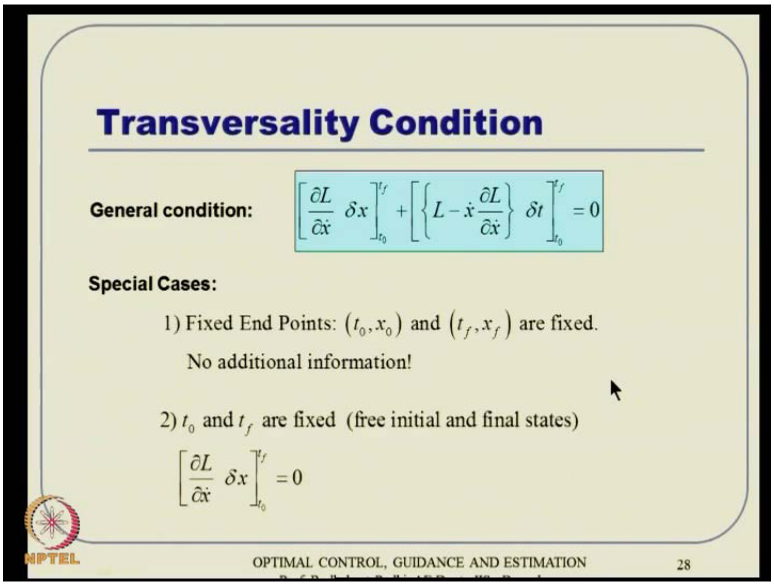


So, based on that observation we formulated some sort of variational optimization problem now, which tells us something like this, we are interested to optimize this, this functional by appropriate selection of this X of t and t 0 t f is started with fixed value, things like that. So, all that it was necessary is somehow find out conditions for 1st variation equal to 0 basically. So, we went ahead and analyzed that , in that took the first variation and things like that and excited this theorem, that if for every continuous function g of t this integral is 0, then there is no another way, g of t has to be 0 actually.

So, so using these fundamental lemma, it turns out, that we can derive some, something like Euler Lagrange equation, which terms out to be something like, that del L by del x minus d by d t times d L by del x dot equal to 0basically.

Then, this is an associated transversality or boundary condition as well, which turns out to be something like these actually. So, that the derivation is there in the next part of the lecture, you can, you can, on you can revisit that lecture and then find out it also basically.

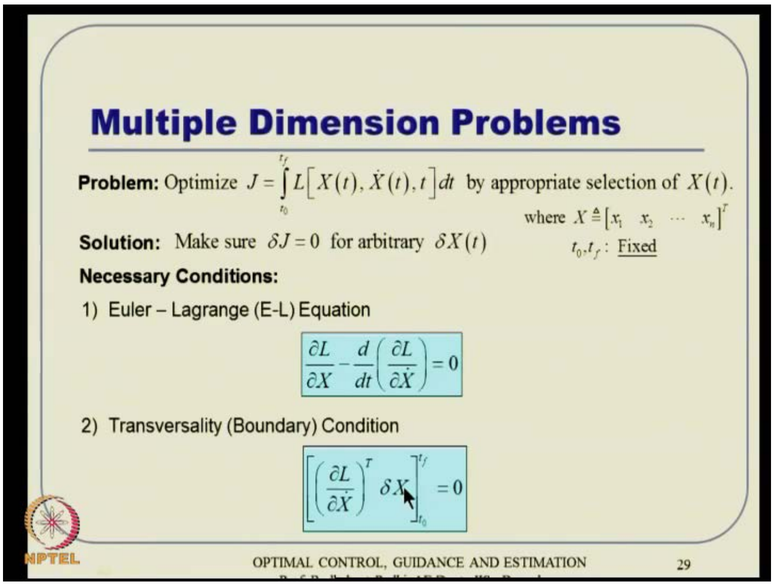
(Refer Slide Time: 18:45)



Then, we also observe on the way, that there is a generic transversality conditions, which turns out to be something like this and then, using these transversality condition in various cases, their boundary conditions can be derived. And suppose it has fixed end point, that means, t naught, X naught, and if X f are both fixed, then it does not give any additional information. However, if t naught and t f are fixed, but X naught and X f are free, then you have to use these conditions and again, depending on whatever, whatever constraints we have, then we have to neglect that. Whatever, whatever is free, the coefficient has to be 0 actually.

So, there we will get another boundary condition actually that way, alright.

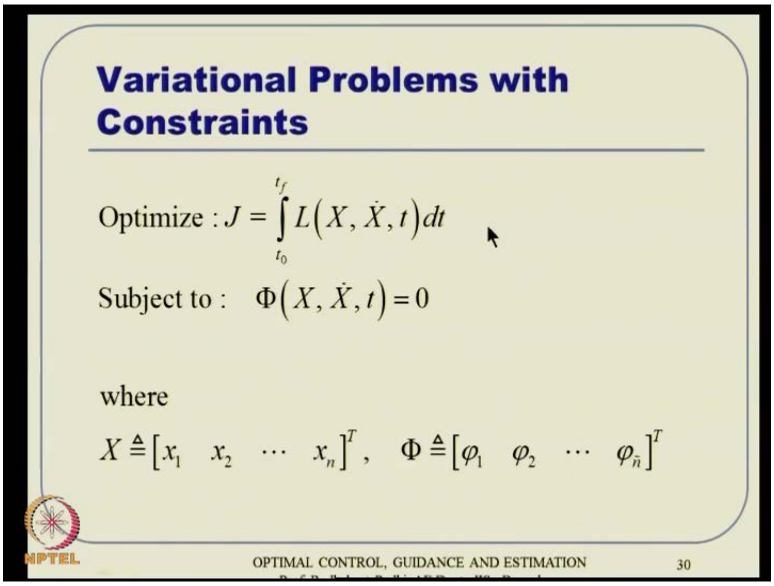
(Refer Slide Time: 19:21)



So, this is, this we extend, this, this concept to multiple dimensional problems, where X of t is actually a vector value now, it is an end component actually and the idea was again to minimize, that first variation is 0 for arbitrary (( )) delta X and then it turns out, the necessary conditions turns out, that it turns del L by del X minus d by d t of del L by del X out to be 0. But remember, these are now vector equations actually, component by component you write n equations actually.

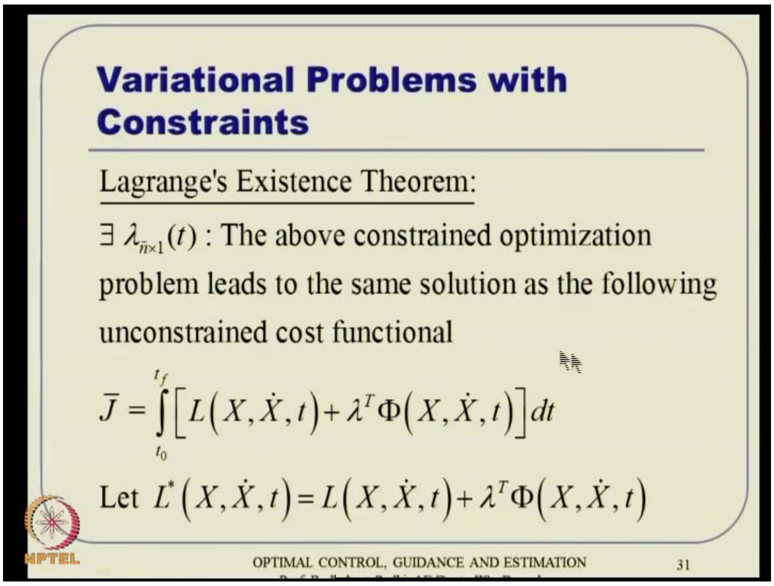
Now, similarly transversality conditions also turns out to, turns out to be nearly same is, I mean, t naught t f are fixed actually, so that this term does not, does not come actually. t f and t 0 are fixed, then this variation of t f is, is 0 and variation of t naught is also 0. So, this entire term will turns out be that way, alright.

(Refer Slide Time: 20:12)



Now, we are not interested in only free optimization. So, we slowly want to introduce state constraints and, and things like that. So, we also studied this, this constraint optimization problem, where an algebraic constraint comes along with the cost functional also.

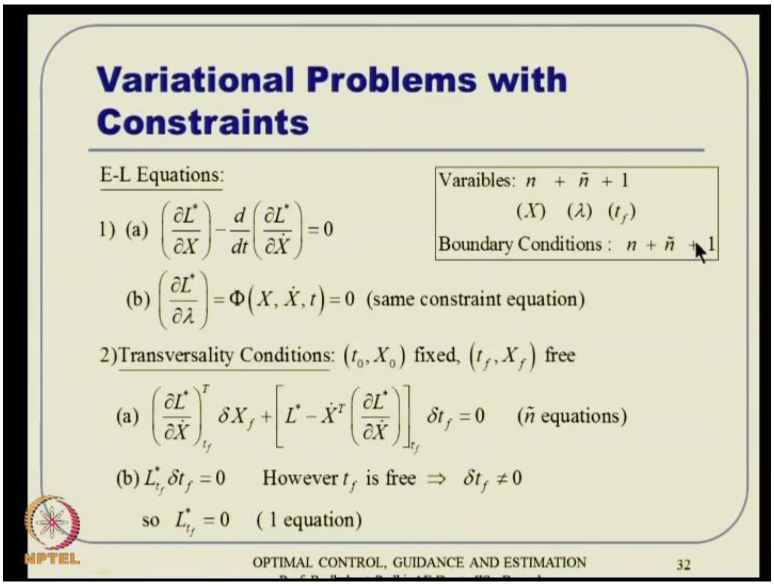
(Refer Slide Time: 20:29)



So, then, we told there is similar Lagrange existence theorem also holds good, which actually tells, that we can construct J bar, something like this, where lambda comes inside the integral. That means lambda is actually no, no more a constant value, it is a time varying variable actually.

And then, you can actually go ahead and apply the, the EL equations and things like that and then come up with these sorts of relations actually.

(Refer Slide Time: 20:51)



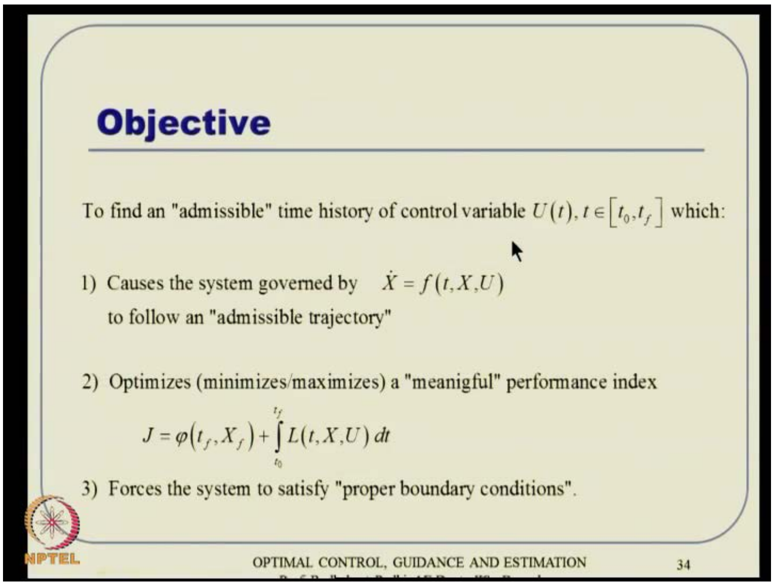
So, this tells us that state equation is valid and then, the associated with that, the lambda equation is also valid and that we will see it little later also actually. I mean, we have seen how do, we kind of use that lambda dot equation and all that actually, that way.

The transversality condition in vector dimension is also similar and then you can, you can use this EL condition along with the transversality condition to arrive at some, some optimal control or optimal trajectory essentially basically.

But the problem dimension is now something like this. So, X is n dimensional, lambda is n tilde dimensional and t f, if it is free, f is to be a one more dimensional actually. So, it is actually n plus n tilde plus 1 dimensional problem that we are talking about actually. Then, that was the kind of calculus of variation summary sort of thing and more details obviously, are there in the detailed lecture, I mean, detailed lectures.

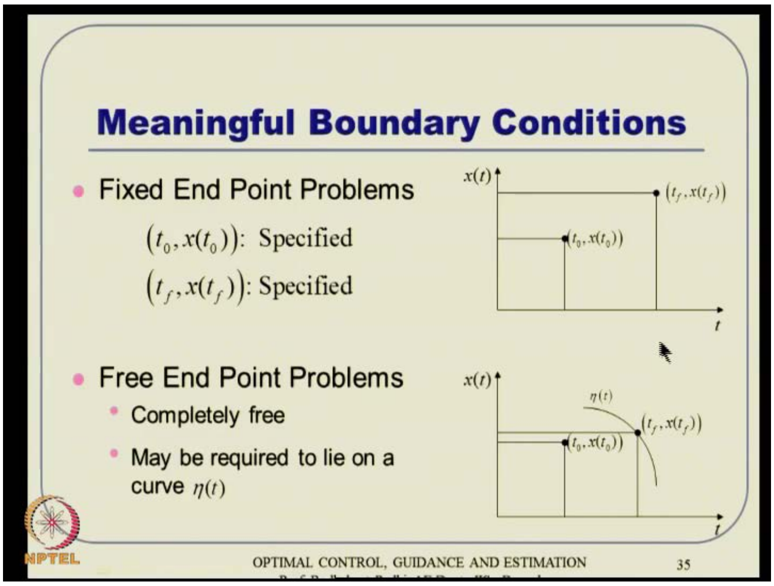
But then, after that we, we wanted to kind of use those concepts in optimal control problems actually.

(Refer Slide Time: 21:59)



So, the objective was something like this, the, to find an admissible time history of the control variable U of t varies from t naught to t f, which causes the system governed by this non-linear system dynamics to follow an admissible trajectory. And on the way, it, it should also optimize a meaningful performance index of this form, which is typically called a kind of (( )) problems, where it is fairly generic concept, I mean, can account for many, many systems actually, many practical problems and also should force the system to satisfy appropriate boundary conditions as well.

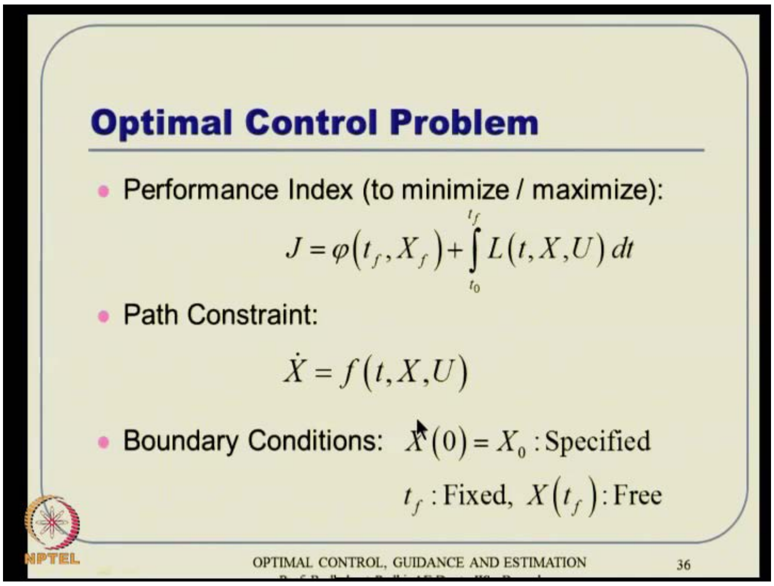
(Refer Slide Time: 22:33)



So, these are the objectives we started with and then we also observed what are meaningful boundary conditions for, say, can have fixed end point problems, where both the things are fixed or you can have free end point problems, either both are completely free, or it, or the most of the time it also turns out, that the final condition is free. But it is kind of constrained to lie on a curve actually, especially lies an orbit transfer problem for satellite and all that, it does not meet or where you join. As soon as you join the orbit and then have the orbital condition, then that is the orbit that you are going to follow from there onwards actually.

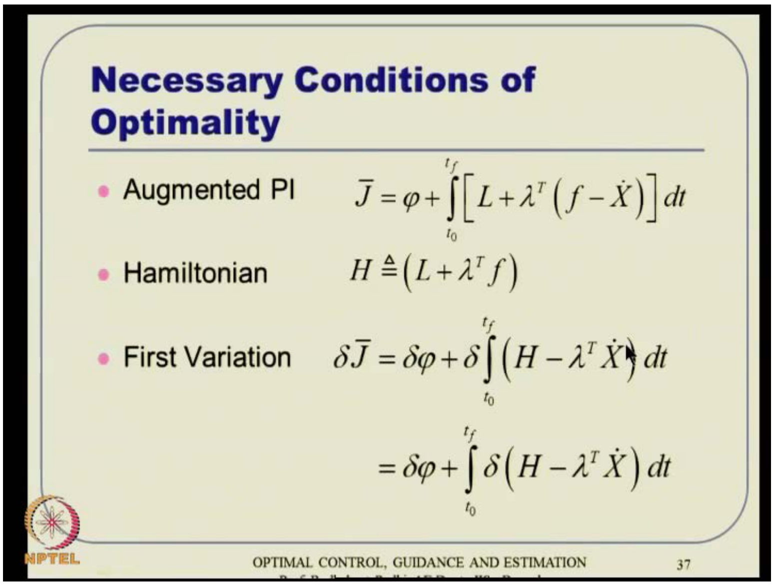
So, that kind of things require, that free end point problems of formulation, where it is not completely free, but the final point is required to lie on a curve actually.

(Refer Slide Time: 23; 18)



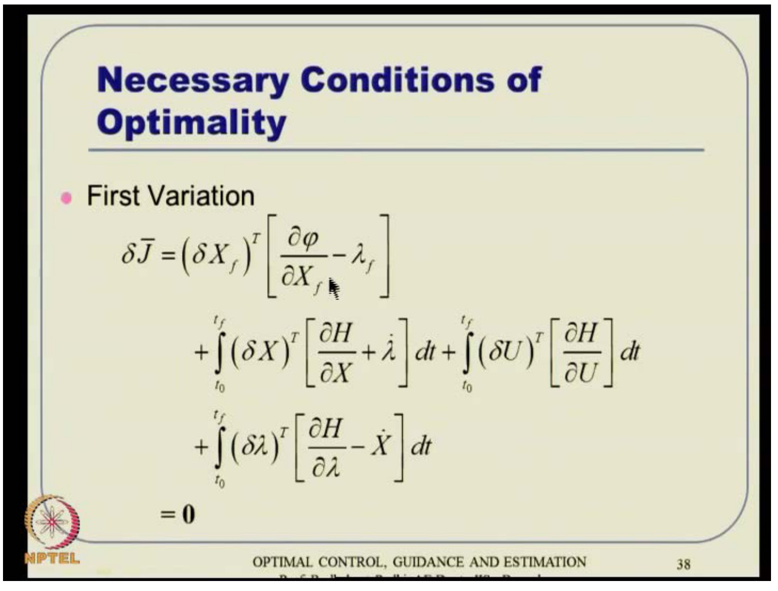
And went ahead and formally stated the problem, like performance index is some, something like this from and we had a path constraint in the form of system dynamics. Now, in, in state phase form and we have some boundary conditions, where initial state conditions are given and t f is fixed and X of t f is free sort of thing.

(Refer Slide Time: 23:36)



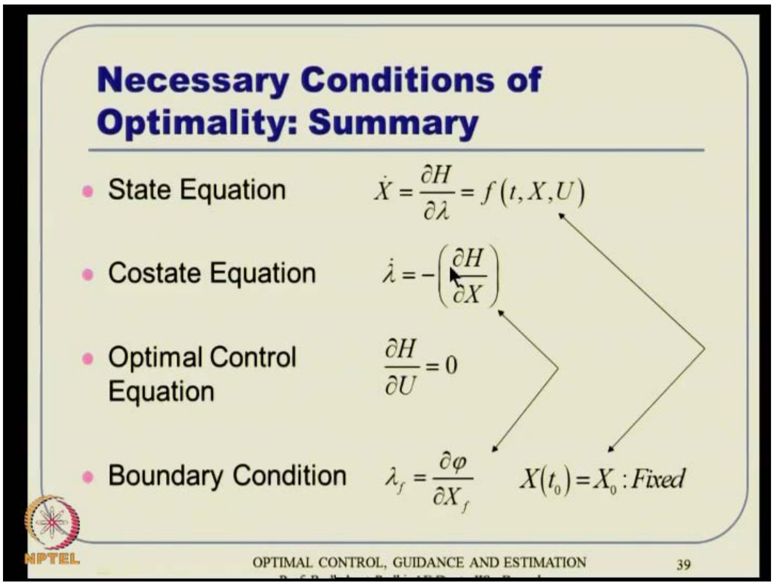
So, then we formulated an augmented cost function, then took the, define the Hamiltonian, L plus lambda transpose f, only that part of it, and then analyze the first variation, which has to be equal to 0 basically.

(Refer Slide Time: 23:51)



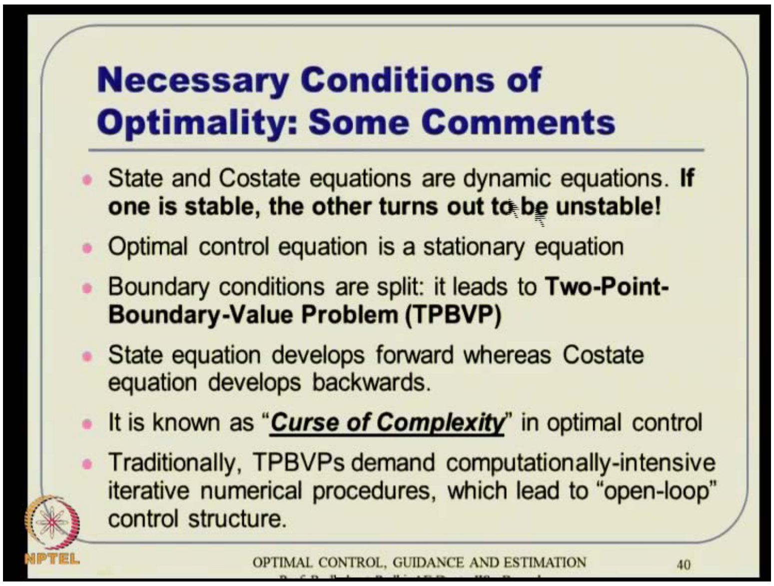
The first variation turns out of this form and then exciting the fundamental lemma of the calculus of variation, all the coefficients needs to be 0, which gave us a bunch of conditions, which tells us, that something like state equation, costate equation, optimal control equation and boundary conditions.

(Refer Slide Time: 23:59)



So, state equation and is X dot is f of t, X, U, where lambda dot is minus del H by del X, all sort of things actually. Optimal control equation is del H by del U and, and boundary conditions are this form; so far so good. We also observed, that these 2 equations are together, that means, state equation (( )) forward and costate equation, that actually there is final boundary condition here, so the costate equation (( )) backward actually.

(Refer Slide Time: 24:32)

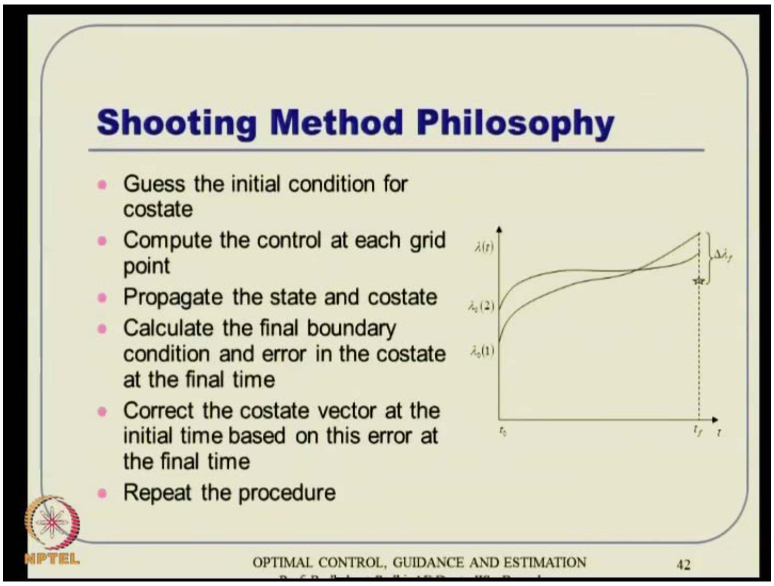


And then these lead to. So, many observations like this the state and costate equations are dynamic equations and unfortunately if it turns out that if one is stable the other is unstable actually whereas, optimal control equation is a stationary equation; that means, just a algebraic set of equations actually as the more difficult thing of the critical observation is the boundary conditions are split actually.

Now, even though it is a 2 n dimensional problem the boundary conditions partly it is given at initial time and partly it is given at the final time the, is the whole, whole problem essentially. So this essentially known as Curse of Complexity and then it is also known as two-point-boundary-value problem, was demand some sort of computationally-intensive iterative numerical procedures and it leads to this, this open loop control structure as well as actually. So, that is the type of analysis leads, it leads to the difficulty of the problems (( )) actually.

Now, what the layers? There are classical numerical methods, people have attempted it and we have actually discussed three methods as part of the lecture and quickly I will review two methods essential here.

(Refer Slide Time: 25:31)



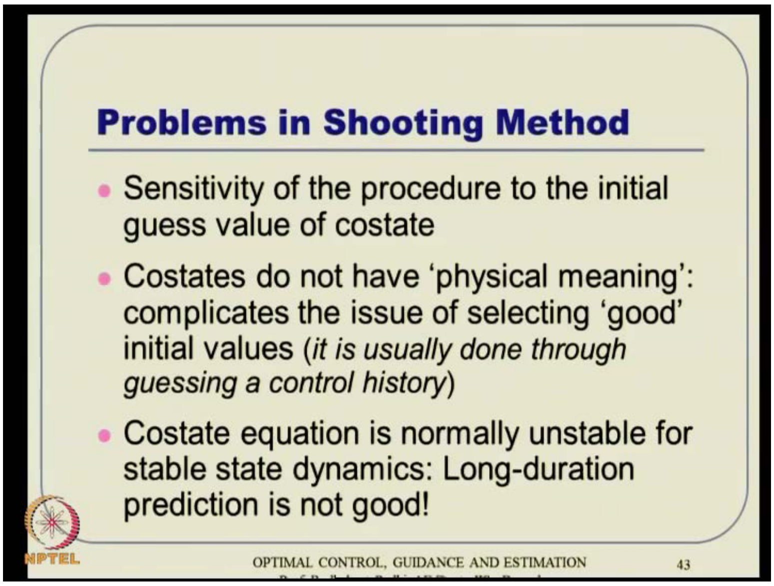
So, what, one we talked is shooting method philosophy, shooting method and shooting method philosophy, I, we initially, we, we observe, that these equations, the set of equation, that difficult is the boundary condition are not available at the same time. So, because the state, initial condition is available, so I will guess the initial condition for costate also, so that these two equations can be propagated together forward actually.

So, that is, what, what we have done here initially, you come up with some sort of guess for lambda is 0 and then, next 0 lambda 0 to be taken together, you can integrate the equations forward. On the way, you keep solving for an optimal control as well and then keep marching ahead and then, finally, land up with some value of lambda f, which is not the decide value. So, obviously, there is an error and utilizing that error we have to kind of update this lambda 0 value.

So, this is the initial case, this the faster, after fast iteration, things like that. So, you update it and then repeat the procedure and land up will include some, somewhere else actually, which, which is hopefully little more better than what we initially guessed actually. So, then, we keep on repeating the procedure and till we get the solution.

Initially, the problem in this method is high sensitivity with respect to the initial value of costate. Remember, for a, for stabilizing control, I mean, sorry, for stable plant for costate, equation is unstable. So, it, you have a little bit inaccuracy in the initial condition value. Then, it actually leads to lot of inaccuracy in the final value and eventually, the entire procedure may breakdown. It may, it may not even converge actually.

(Refer Slide Time: 27:04)

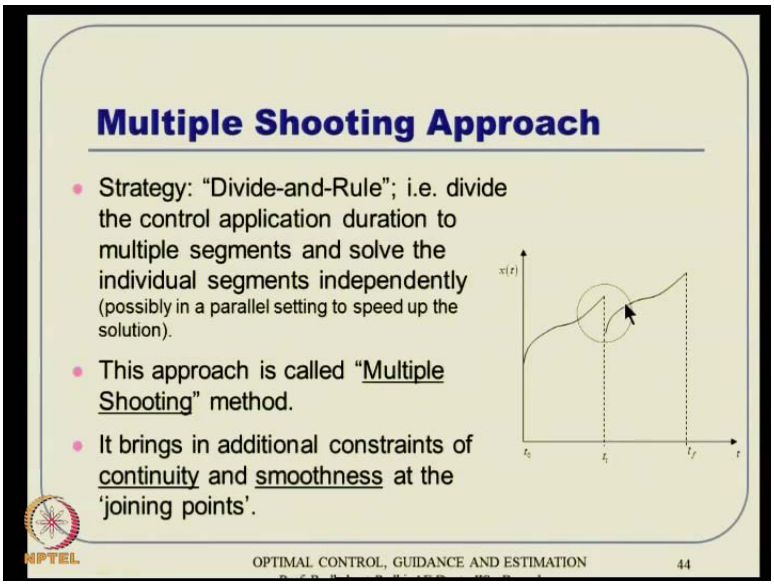


And this, this effort of guessing this costate value is even more difficult because costate do not have any physical meaning there, not in, nothing like distance, velocity and everything, I mean, current, voltage, like that there, nothing like that. So, we do not know what kind of numbers we need to choose actually. So, that is why, it is usually done through guessing some set of a control history.

So, we guess a control history form t naught t f (( )) X naught. So, we go there with that control history find out lambda f from the boundary condition and then integrate the costate equation backwards to go, to come up with some sort of a lambda 0 value.

And so, this is the critical drawback of this method, that costate equation is typically integrated in the forward direction and hence, we are actually integrating an unstable equation in the forward direction, which is their, which is not good and hence, long duration prediction is not good actually.

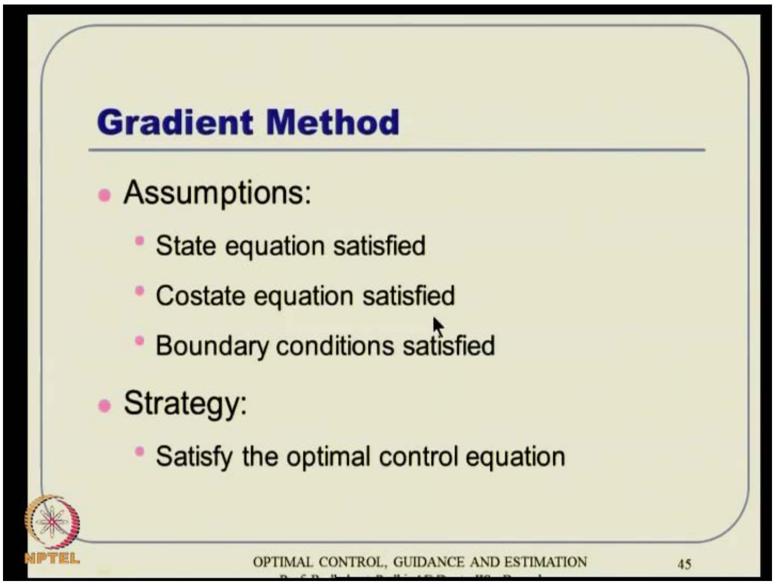
(Refer Slide Time: 27:57)



So, that is why, people will thought about divide and rule policy, something called multiple sorting and all that actually. So, divide the path, breakup path into 2 parts, 3 parts in the, and things like that, I mean, finite number of segment really and then, you can actually integrate this, this problems in, in segment wide sense, but also remember while doing, that the continuity and smoothness of the joining point have also be ensured. That means, you are a, we have to introduce more and more constraints to kind of ensure that actually.

This approach is called in multiple sorting and generalization of that is, is something like direct transcription. And we also add a glimpse of what is called direct transcription method as part of this course actually.

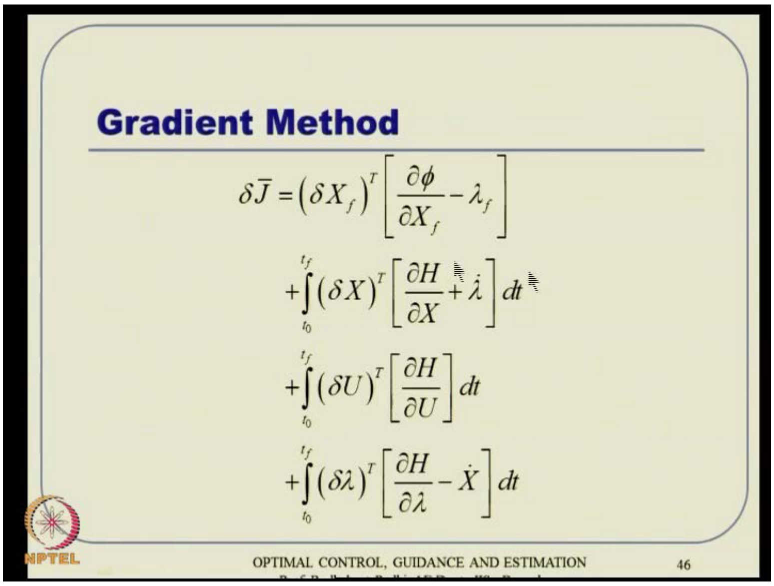
(Refer Slide Time: 28:36)



Now, coming back to the gradient method, which is another idea actually, what we tell is, we start with guess, guess history of the control values from t naught to t f and with respect to that guess history, we assume, that if you use that guess history in the state equations, then whatever trajectory you get there, we assume, that, that is a kind of closed optimal in the following sense. There, the state equations is satisfied, costate is also satisfied, the boundary conditions also satisfied, whereas the costate equation is the, sorry, the optimal control equation is not satisfied. And remember, out of all these things, whatever conditions we have, everything is need to be satisfied, then only you can talk about kind of solution being optimal, otherwise not optimal actually.

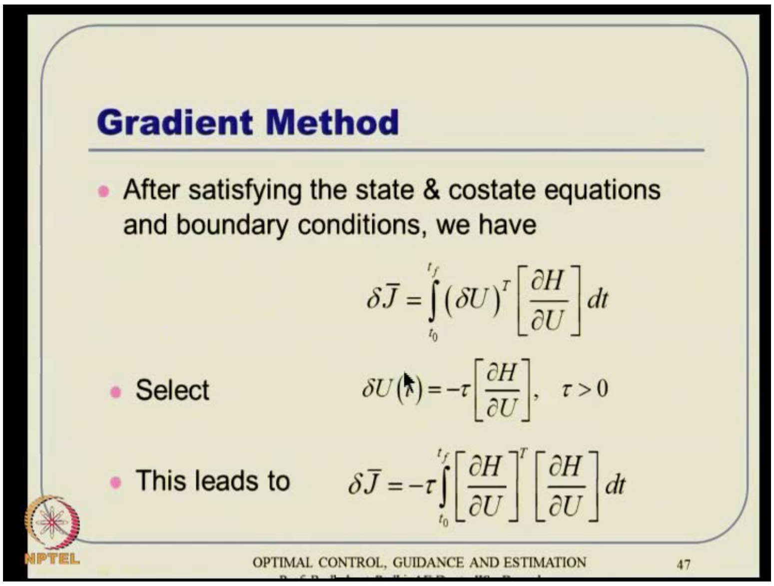
So, here, we are assuming, that the optimal control equation is actually, kind of, getting not satisfied, everything else is getting satisfied. Then, the idea is how do you iterate on the control history, so that, that equation will also get satisfied without (( )) all these conditions actually.

(Refer Slide Time: 29:37)



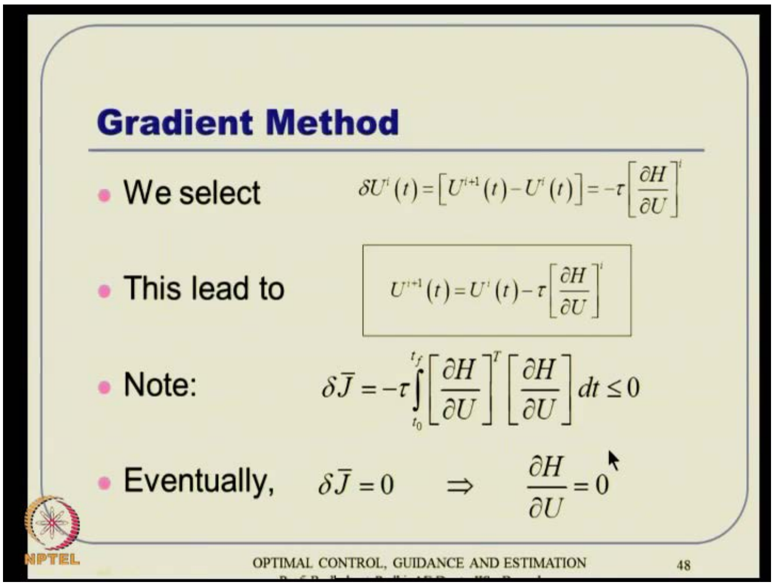
So, then we analyzed, if you just talk about 1st variation of J bar, this turns out to be something like this and you are assuming, that I mean, you are assuming, that this equation satisfied, boundary condition satisfied, so this is 0; costate equation satisfied, this is also 0 and this state equation satisfied, this is also 0. So, we land up with only this term actually.

(Refer Slide Time: 29:54)



That is why, del J bar turns to be like this and hence, if you really want to make it, kind of negative, decreasing value and all that, then you take del U is nothing but minus tow times del H by del U. So, this term becomes some sort of a quadratic term like this actually. So, this is guaranteed to decrease actually. So, thus, thus, the whole idea of gradient method, which is also inspired from this, this steepest descent method in static optimization, actually.

(Refer Slide Time: 30:21)

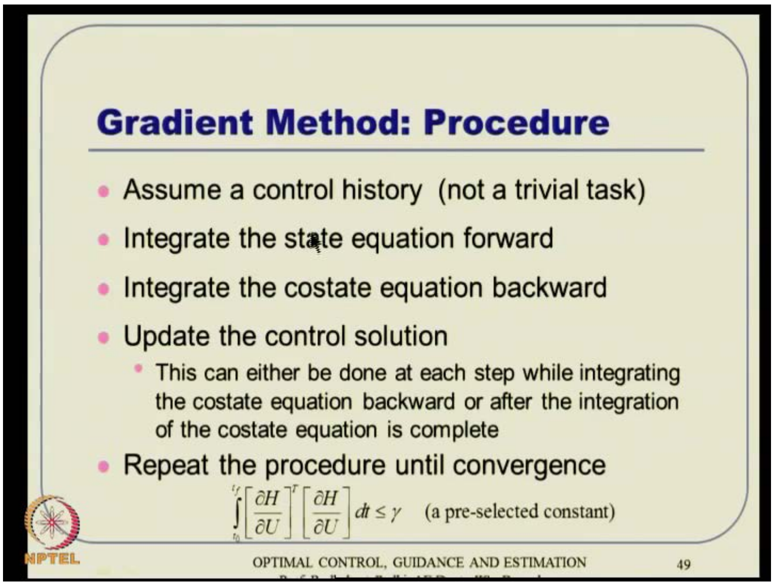


So, then idea is we select this del U i something like this and all over from t naught to t f and then, this leads to this, this kind of update actually, from, from i to i plus 1 actually, alright.

And eventually, if it keeps on decreasing, then eventually, this del J bar is going to be 0 and when del J bar going to be 0, then only one it can happen is, del H by del U equal to 0 because all these are quadratic term and integration of quadratic terms means, unless this value itself is 0, this cannot be 0 actually.

So, by ensuring this 0, the all ensuring the del H by del U is 0 and that is nothing but the optimal control equation actually. So, that leads to this, this gradient method idea basically.

(Refer Slide Time: 31:03)



So, this is what is summary summarized here. We assume a control history, which is again, assuming a control history is typical, not, not trivial, we have to really look at the problem and then, according to the problem you have to select it judiciously actually.

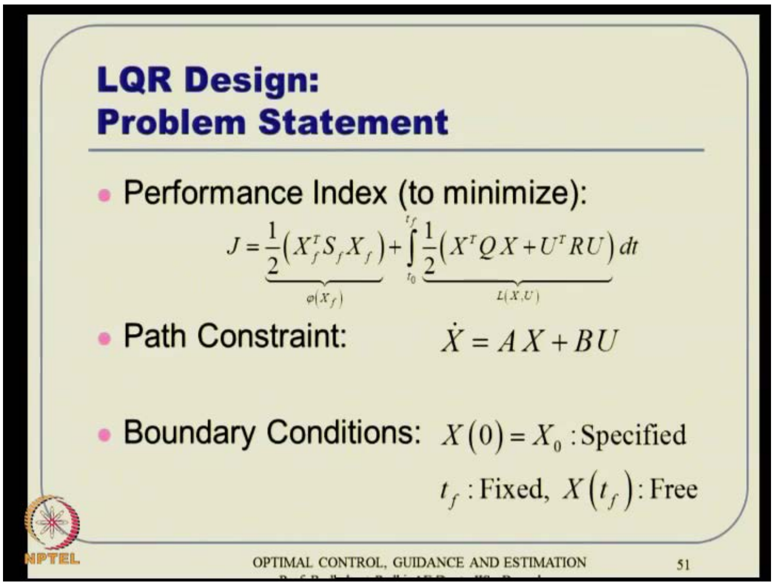
Then, first, using that control history, integrate the state equation forward. Then, at the end, evaluate the boundary condition for lambda and then, using that boundary condition, you can integrate the costate equation backward and update the control solution. Either, this can be done, both 2 ways, actually it can be done, either we done at each step while integrating the costate equation backward or after integrating the entire costate equation from t f to t naught and then, you can update the control history, all the control history from t naught to t f at one go, basically.

So, this procedure needs to be repeated until convergence and that is what the gradient method is actually. Actually, on the way we have given some examples also, including some sort of a missile turning examples and all that, which is a little more realistic, I mean, problem actually.

Alright, then there was a quasi linearization method also in that particular lecture. So, if you are interested, you can go back and study that as well.

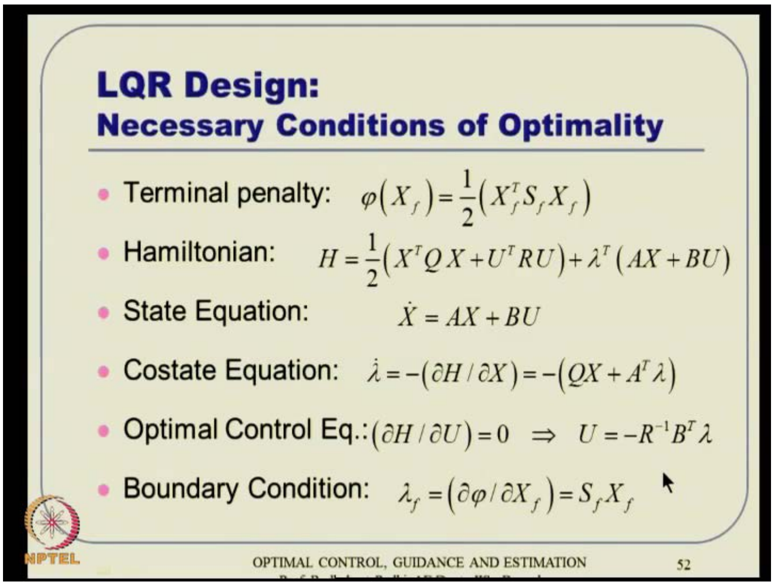
Then, we migrated to this, this linear quadratic regulator theory, which is there in, nineteen, which is developed around 1940s and 50s and it is one of the branch of optimal control theory, which is heavily used in practice also, so, and this essentially leads to this state feedback form of control design, even though the system dynamics has to be linear and all that, actually. So, that is what it, it gains significance in that point of view.

(Refer Slide Time: 32:37)



So, the performance index was selected to be quadratic. There was a penalty, terminal penalty and there was a path penalty, all of which is quadratic path constraints, leads to be linear and boundary conditions are something like this actually.

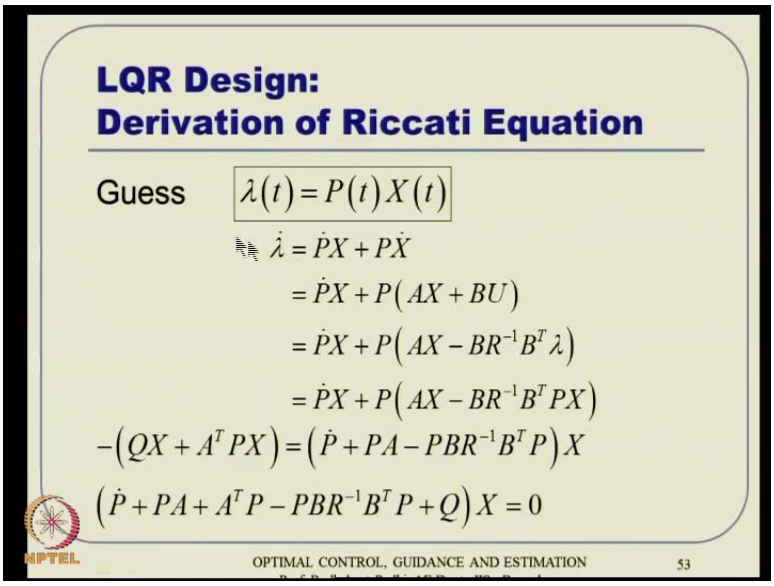
(Refer Slide Time: 32:53)



So, essentially, once you analyze this necessary condition and we observe that phi of X f is this one; the L f is this one. So, Hamiltonian is l plus lambda transpose f. So, one part is this and lambda transpose f is nothing but this part actually. X dot is f of X plus B U, that is what Hamiltonian is and then, using this necessary conditions (()), it turns out the state equation is nothing but X dot is A X plus BU. Costate equation is minus of a X plus A transpose lambda. Optimal control is del H by del U equal 0, which essentially leads to U is nothing but minus R inverse transpose lambda and boundary condition is S lambda f is S f X f.

Now, these are the conditions, but how do you use this actually.

(Refer Slide Time: 33:42)

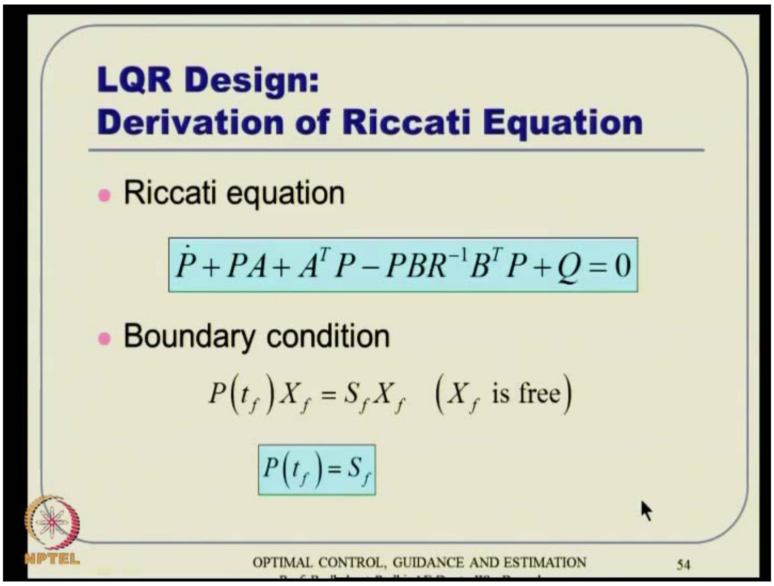


So, observing the fact, that t equal to t f lambda is a linear function of X f, we thought why not we guess something lambda of t to X of t, but you also gave solid, good mathematical reasons, why it leads to be in that form only, basically I mean. So, there was some uniqueness theorem also, one idea is there, but if there is a solution, then that has to be only solution for LQR problem and also, there are other ideas like we have vector space methods. If you see (()) and all that, it has to be lambda of t each of, each component of lambda t has to be kind of linear functional of all these things actually, of X.

So, this is, it has to be written in this form only, but once you write it, then you can analyze lambda dot and then, you can use X dot is A X plus B U. State equation U is minus R inverse B transpose lambda optimal control equation and then, lambda is nothing but P times X coming from here, whereas lambda dot is this equation coming from costate equation.

So, once you put everything and then take everything together to one side, turns out to be like that. So, X is arbitrary, it cannot be 0. So, then the coefficient has to be 0, that leads to this famous Riccati equation with boundary condition, that P of X f, this boundary condition is there from this lambda f is nothing but P of X f, but this relationship P of X f is nothing but s f X f. So, P of X f has to be S f, actually this condition.

(Refer Slide Time: 34:55)

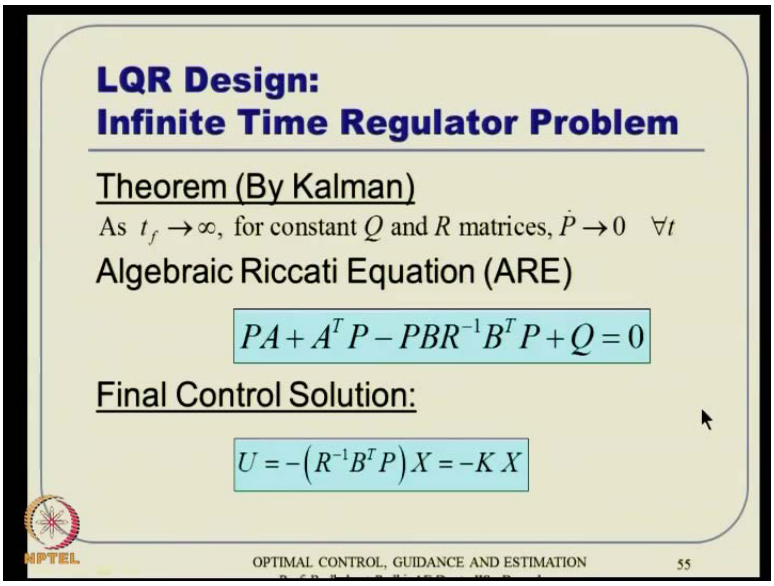


So, you think, that you look at that t equal to t f, we have boundary condition for free, irrespective of whatever is the initial condition of X, it does not matter actually.

So, in invariable, you know matter where the initial condition of state, we, we can actually start with the boundary condition of Riccati matrix P and then, use this Riccati matrix equation, differential equation to integrate it backwards from t f to t naught stores, that values and then use it online is lambda equal to times X and then this control is nothing but minus R inverse transverse lambda, lambda is B times X actually.

So, that gives us some sort of an R inverse B transverse times P is nothing but the, but the gain matrix k basically.

(Refer Slide Time: 35:41)



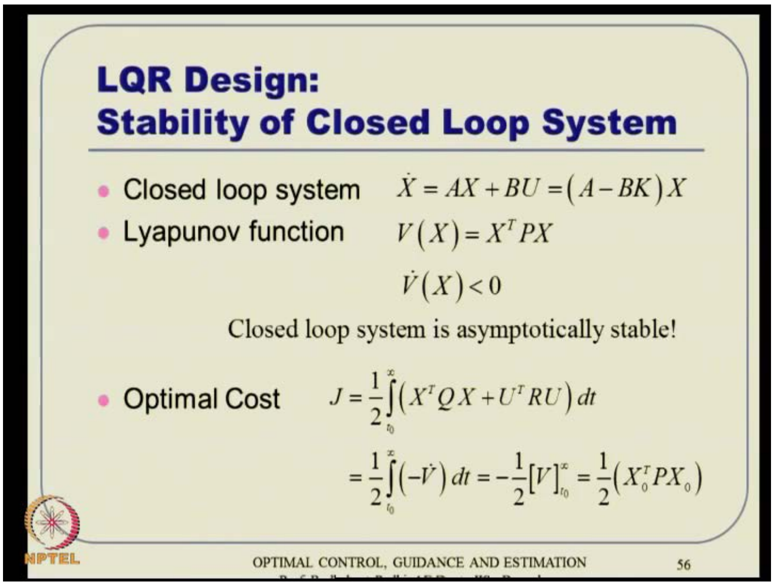
So, this is what is written here, U is nothing but minus R inverse B transpose P, this is, this is gain, and then essentially, it leads to this, U equal to minus K X form, which is in state feedback thing, state feedback form, but also, we observe on the way, that if t f goes to infinity, Q and R are constant matrices, then P dot is identically 0 for all time.

So, X acting, that it turns out, that this, this is 0, so that means, it results in the algebraic Riccati equation actually, which is very commonly used all over the actually that way.

So, we do not have to really solve for some, some differential equations, store the matrices offline and all that actually on warded and use it online and all that, you essentially, you can just solve 1 equation and if you happen to be a difficult equation we can solve it offline still, and then keep, keeps one matrix in, in memory and then, sort using this and then term this gain, can be computed like that and I have the gain, the, I mean the gain matrix is available.

So, the state control happens to be a state feedback form and that too, a linear feedback from actually.

(Refer Slide Time: 36:49)

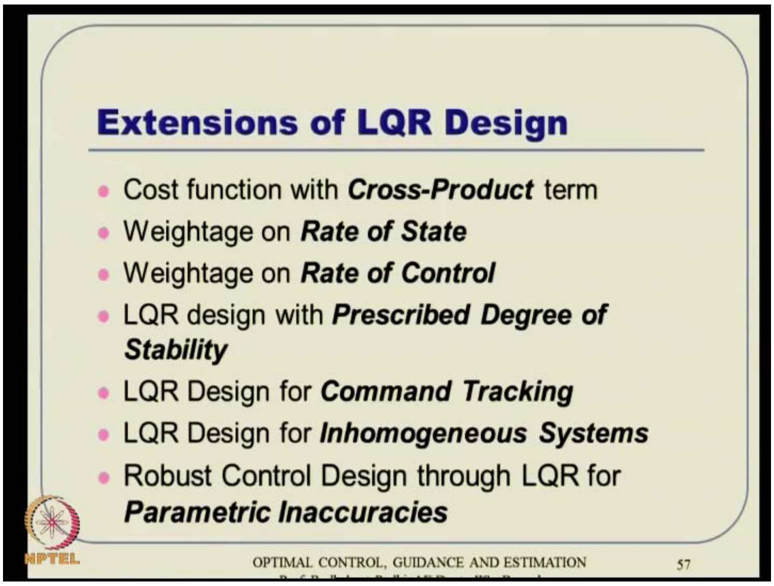


Then, we also kind of analyze the stability of the closed loop behavior. So, we had noticed, that closed loop can be written something like this and selected Lyapunov function of the P, X transverse PX and P happens to be, remember this a non-linear equation, it can have multiple solution also, but you are interested in that particular solution, which is positive definite actually. So, once we have taken that equation and put it f, then this becomes a positive definite and all that it was necessary is to, so that v dot X is negative definite and we are able to solve that actually, so that it tells us, that the closed loop system is essentially asymptotically stable.

And because it is also readily unwanted and all that, it turns out, that it is globally asymptotically stable, as in, all I mean, if you see the solution nature, we really see, that for constant R V and constant K, it is a solution, nature will tell. There is, it is actually globally exponential stable also basically.

Now, to find out the optimal cost, we have to some analysis of this cost function, especially when t goes to infinity and it turns out, that the same thing can be written as something like this, negative of v dot here, so it can be evaluated. The derivative, that integral will go, it can be evaluated like that, but V of infinity is nothing but X of infinity coming here. So, X of, X of infinity is 0, that what goes and you land up with only X naught actually. The moment we know X naught, we can actually say what the optimal cost for value can be actually.

(Refer Slide Time: 38:13)



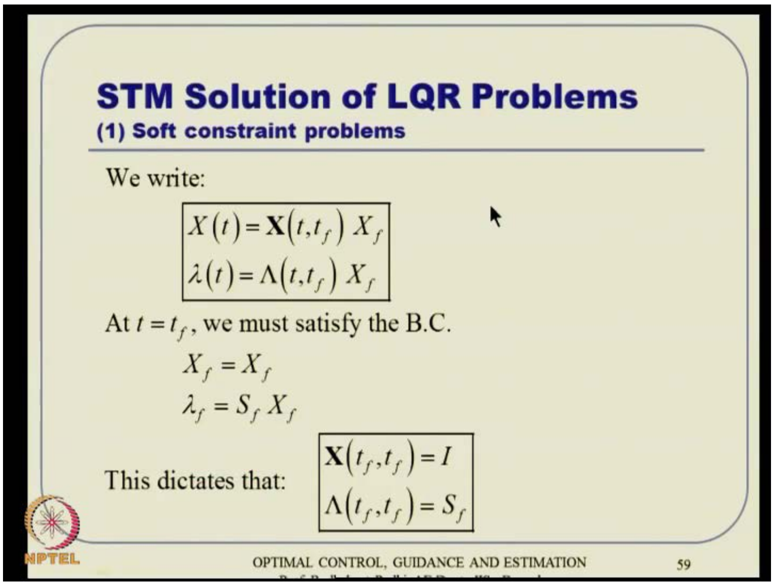
Then, we, we extended this LQR design to several, several, the concepts and ideas and all that. First, we did some cross product term and then tell how to actually reformulate the problems, so that cross product terms can be also solved. And we also went to this (( )) rate of state constrained or weight as a rate of state formulation and essentially leads to this cross product on formulation again. So, it was, it was possible to solve.

We also showed weightage on rate of control, if it there what to do essentially, lead to this P i sort of control essentially. And then, we have this LQR design with prescribed degree of stability also. So, that is how it is. And then LQR design for command tracking. So, this prescribed degree of stability tells, that in the closed loop, the Eigen values should, should lie left side, left of certain, certain vertical line, which is alpha away from the imaginary axis basically.

So, it is possible to do by formulating the cost, cost function in such a way, that X transpose q X plus U transpose R U that term multiplied that with e to the power 2 alpha t and then we formulate the problem and then it can possible to solve that all the Eigen value of the closed loop system will be remaining left side, left of a vertical line, which is alpha away from the imaginary exits actually. So, that is the prescribed degree of stability.

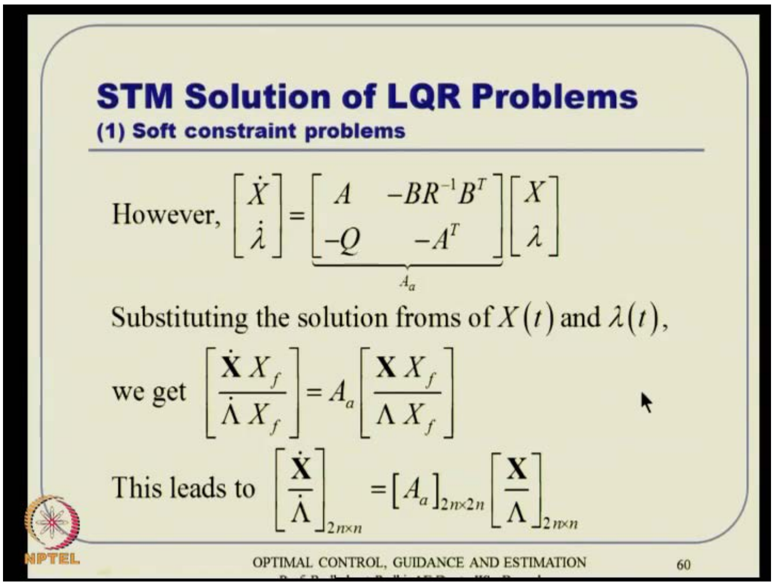
Then, we also show how to extend that command tracking problems and also show how to extend that for inhomogeneous systems. That means, X dot is X plus V U plus C, if you have something like that what to do about that. And also show how slightly different formulation, which, which talks about robust control through LQR design, especially for parametric inaccuracies and also again lands up with this, this integral control feedback sort of ideas actually, alright.

(Refer Slide Time: 40:11)



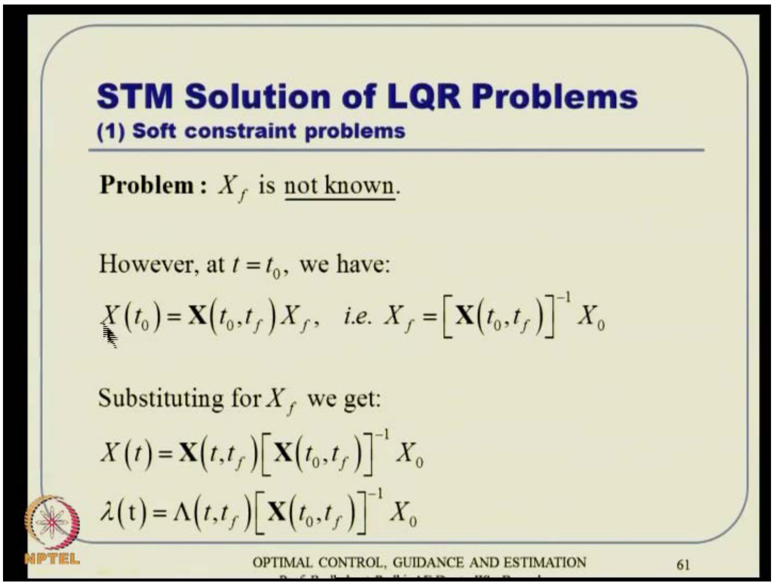
So, after that we again went ahead and extended that for a state transition matrix approach. I mean, we look the problem and try to kind of solve it using this state transition matrix approach ideas, where we write this X of t is something like state transition matrix times X f now, and lambda of t is also this state transition matrix for lambda times X f and t equal to t f, this condition has to be satisfied. So, we get the boundary conditions at t f actually.

(Refer Slide Time: 40:27)



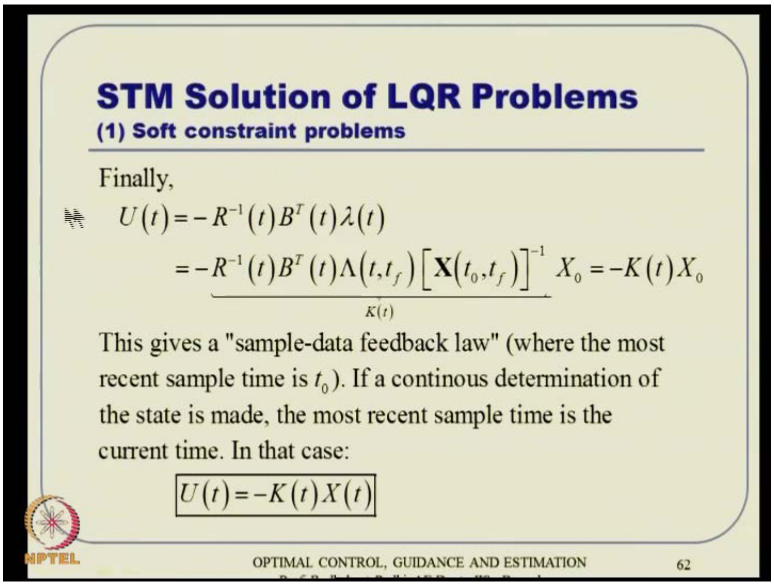
Then, it turns out, that the X dot and lambda dot can be written something like this and hence, substitution, this expression we can sometime write like that. So, these state transition matrices will satisfy this kind of equation essentially. So, we have these equations and we have these boundary conditions essentially.

(Refer Slide Time: 40:42)



So, we can solve that basically and also notice, that X of t naught can be, can be extracted from X f that way and substitute back here. So, X of t and lambda t, we call lambda of t becomes like this.

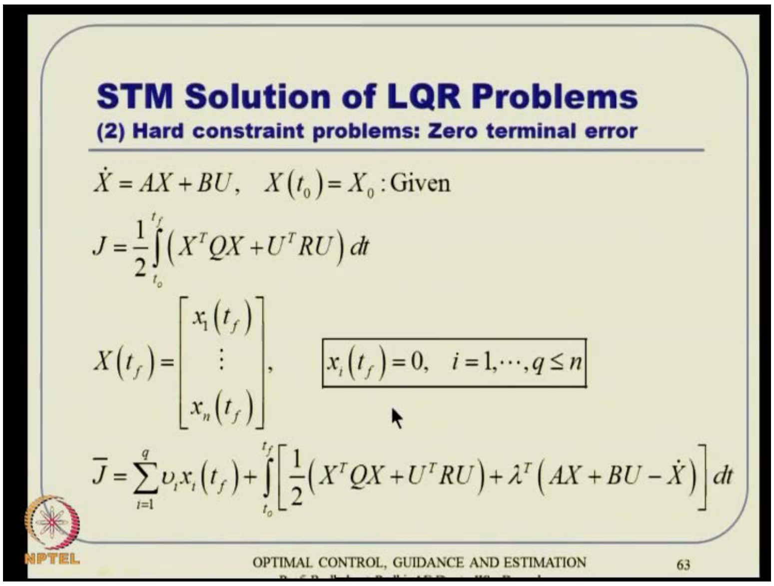
(Refer Slide Time: 40:56)



And then finally, U of t can conduct this form and this can be again written something like K of t times X naught actually. Now, we can tell my t naught is t, then it will essentially lead up with this, this U of t essentially.

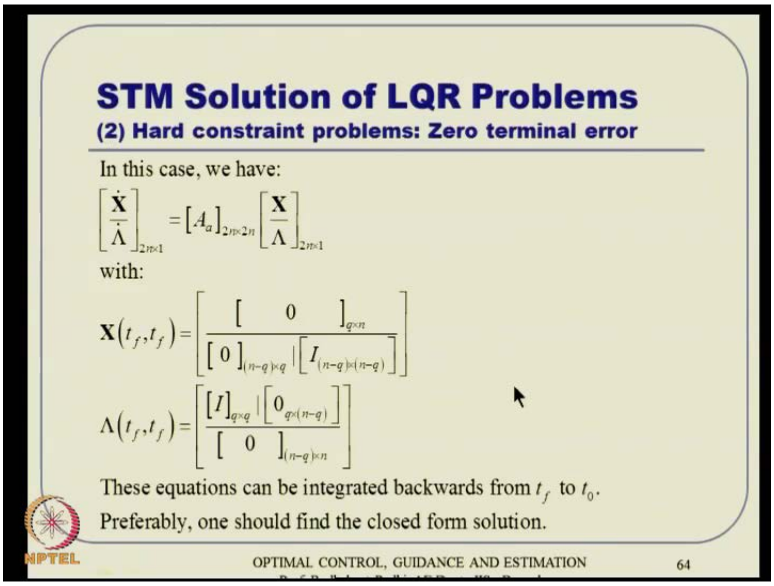
So, that is sample-data-feedback law sort of things actually. So, U of t essentially can be computed again as minus of K of t times X of t times.

(Refer Slide Time: 41:17)



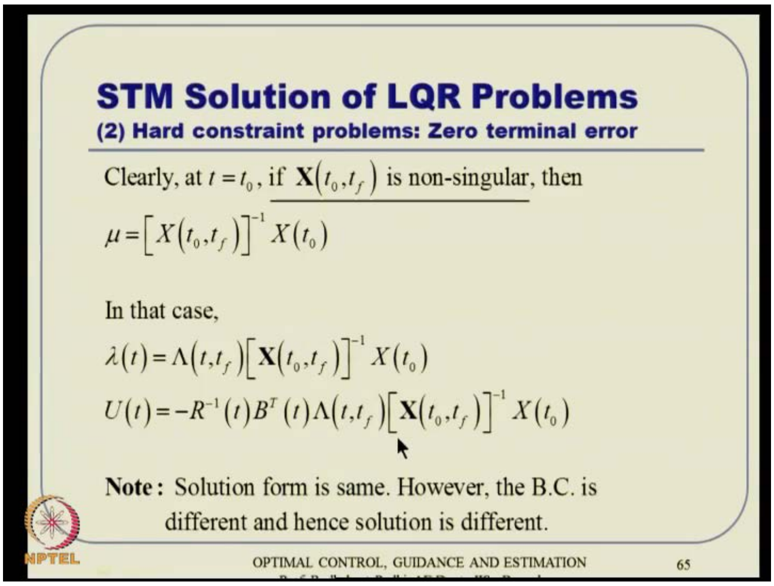
Then, this we extended that to hard constraint problem also and told ok, here the X f was free, but here X f is hard constraint sort of things. That means, X i of t f has to be equal to 0 for, for i one to q. So, for, for a set of states, part of the states should be identical equal to 0, that is, the hard constraint actually.

(Refer Slide Time: 41:49)



So, then, we have to formulate this, this new also basically. I mean, this, this part will come as additional part of it and then we analyze this in a fairly similar way. So, again it landed up with, with, with state transition matrices being something like this and boundary conditions now are different. The boundary conditions turns out to be something like this actually, that means, as long as t is away from t f, things are ok, but t, when t approaches towards t f, remember this matrix is actually a singular matrix actually.

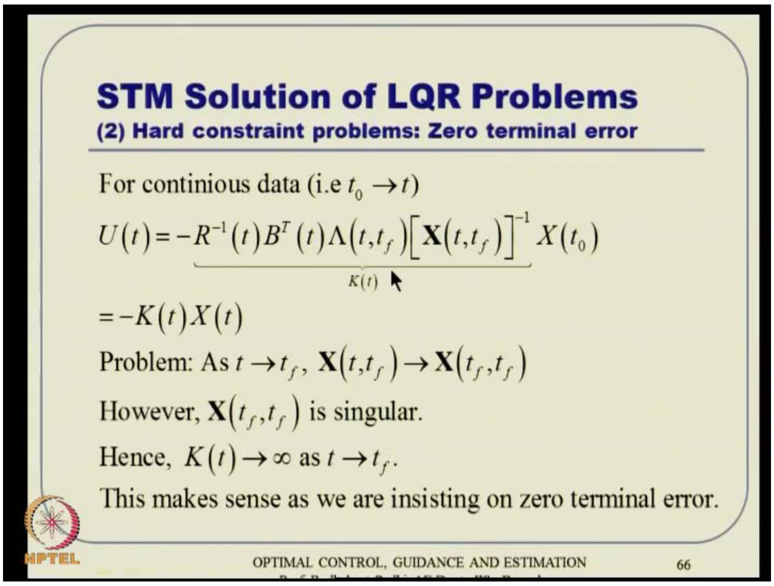
(Refer Slide Time: 42:13)



Then, because of that, then there was a, I mean, if you see this control expression, there is an inverse of that and singular matrix inverse is obviously infinity and all that actually.

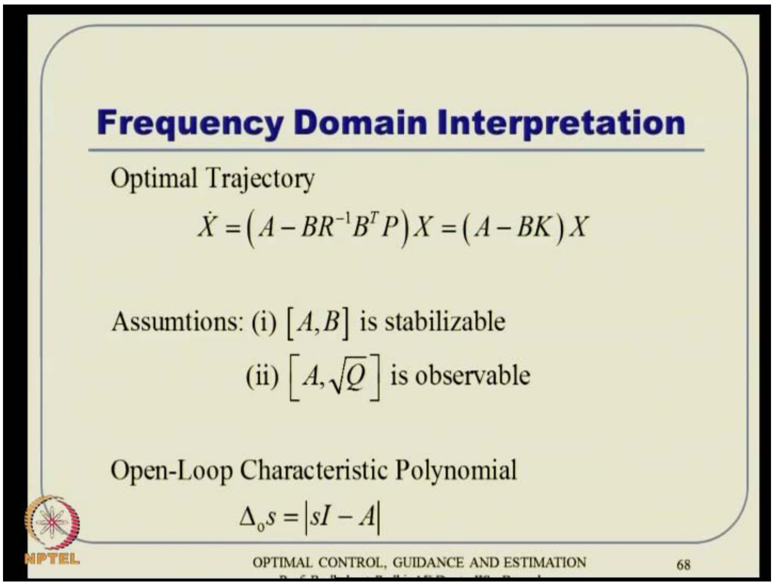
So, we have a problem in this, this singularity of the control actually. So, after some time the control will, which guaranteed to go to saturation, that there U time, you can apply the saturated value or something, but then not a very good formulation actually.

(Refer Slide Time: 42:35)



So, we solve that actually that time and you also had little bit glimpse of frequency domain interpretation of LQR.

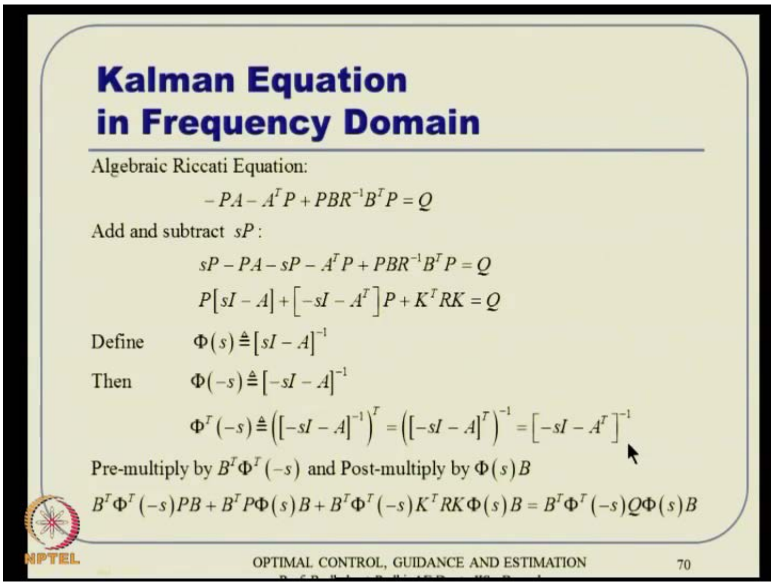
(Refer Slide Time: 42:43)



So, we thought, optimal trajectory is dictated by this close loop, close loop system dynamics and then, we assume, that A, B is stablizable, whereas A square root of q is observable and they have the, we define this Open-Loop characteristic polynomial is something like this and then we analyzed the close loop characteristic polynomial, highly turns out to be given in this form actually.

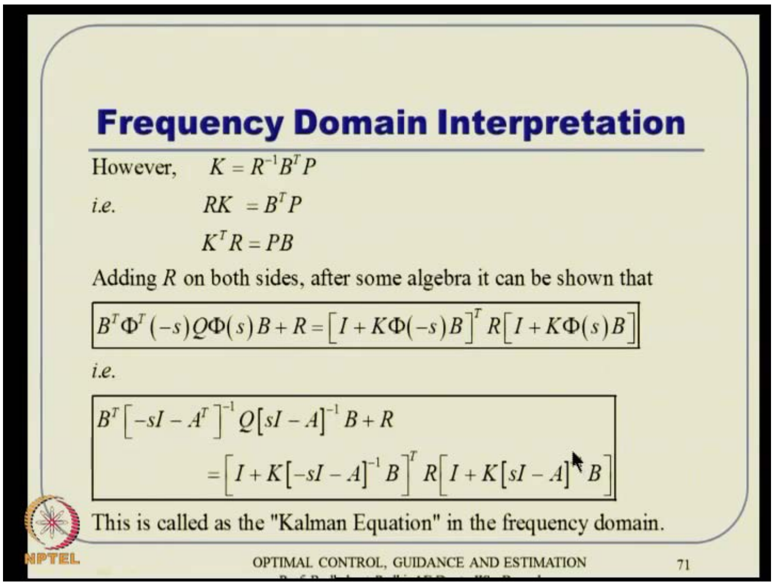
So, this particular thing, I plus k times s I minus A, s I minus A inverse time, this part called return difference matrix and then, this loop gain matrix is something like this actually.

(Refer Slide Time: 43:16)



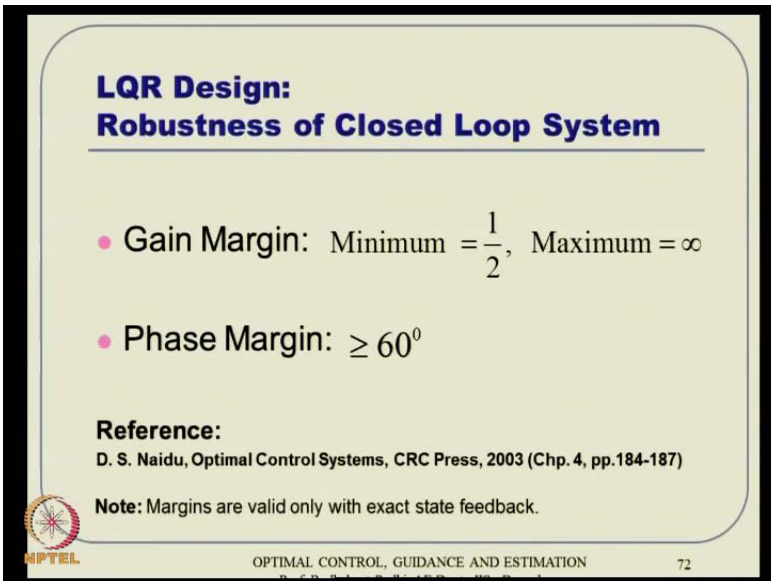
The algebraic Riccati equation, if you start from there as it is, then you can add and subtract this s P terms, this and this, lot of these algebra and then define this polynomial phi of s sI minus A inverse and then carried out, and of this pre-multiply and post-multiply operation and all that.

(Refer Slide Time: 43:33)



Essentially, it will, we landed up, we some, some equation like this, which talks less, which is called Kalman equation in frequency domain actually. As possible to use this equation to derive the, the gain matrix values actually.

(Refer Slide Time: 43:48)

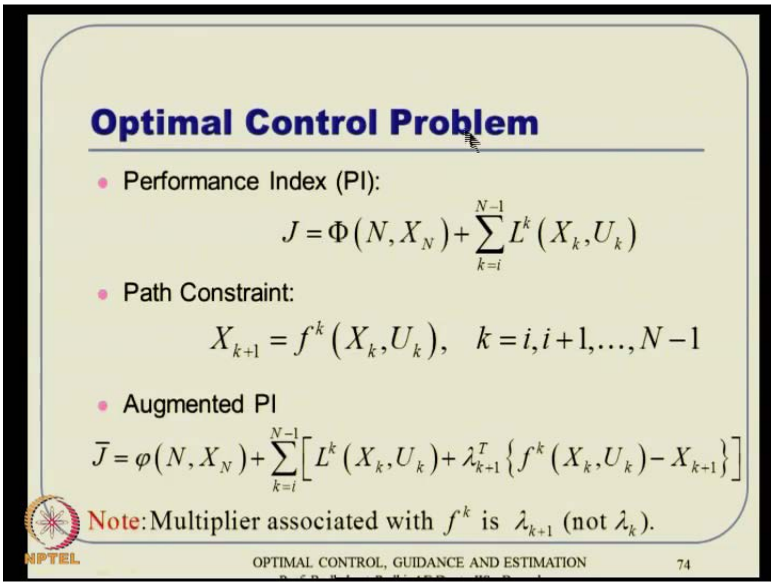


So, given examples as well it turns out, that using this analysis further, as it turns out, that the gain margin is minimum half and maximum can be s I as infinity and the phase margin is always greater than equal to 60 degree in the LQR design, which is very good.

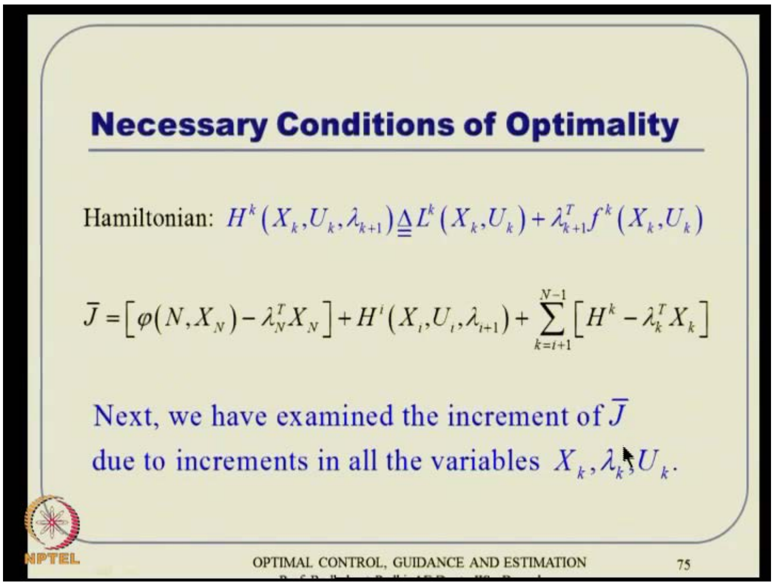
But also remember, these margins are valid only with exact state feedback. The moment you put something, something like a observed state or estimated state feedback, then these are no more valid actually.

Before you wind up the LQR topic, we also discussed something like discrete time LQR and discrete time optimal control theory in general first, before going there.

(Refer Slide Time: 44:23)



(Refer Slide Time: 44:31)

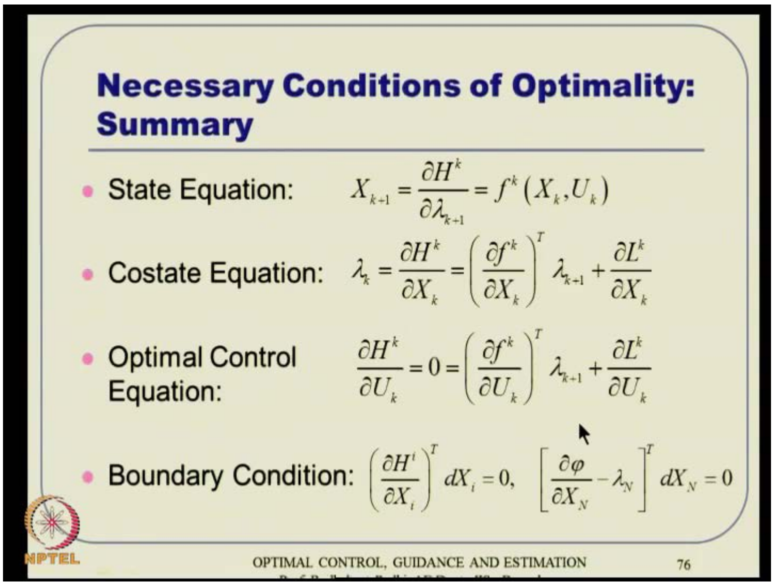


So, the performance index for all given in a discrete time and path constrained to, also given in a discrete time, augmented cost function is also like this, something like that, that we had Hamiltonian described as a very standard with. The only difference here of course, is the association of lambda k plus 1 here, it is not lambda t.

We assume, that if you see the continuous time, we expect, that here should be lambda k, but here it turns out, that start with lambda k plus 1 and not lambda k basically.

So, we had this Hamiltonian formulation that way and then, again this J bar formulated that way and we have to, we have examined the increment of J bar due to increments in the variables of X k lambda k and U k also.

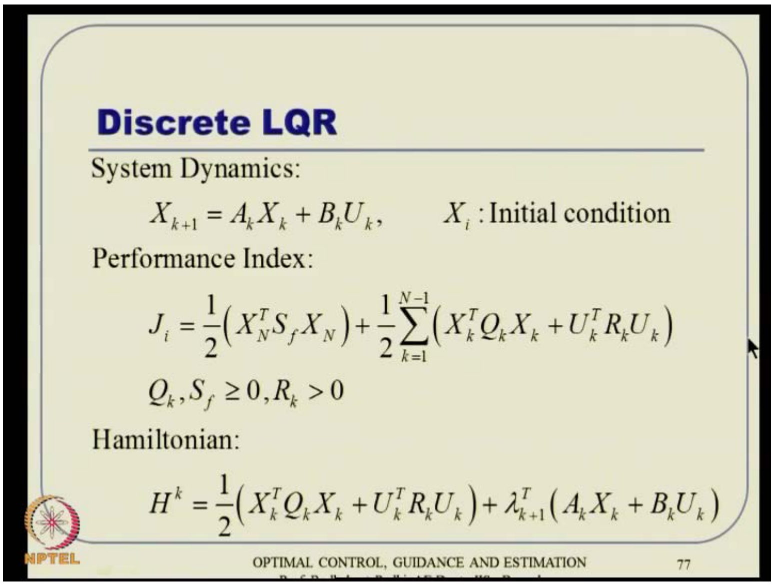
(Refer Slide Time: 45:09)



And then, ultimately land up with these state, costate, optimal control and boundary conditions equations as well; very, very similar to what we know before. The only difference is this, it is wherever lambda k was there, we have wherever, we expect, that lambda k will be there in the right hand side, turns out to be lambda k plus 1 actually.

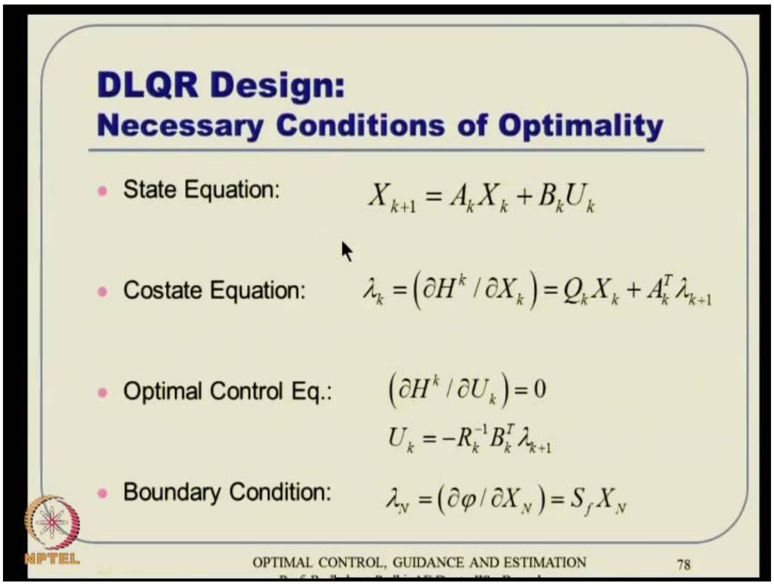
So, to know the U k, we actually need to know lambda, lambda k plus 1, each other that way and for otherwise, it is fairly similar. This state equation doubles (( )) forward, costate equation doubles (( )) backward. Remember, lambda k is a function of lambda k plus 1 like that and optimal control equation, stationary equation has to be solved to get the, the control value k actually. The boundary conditions were also given like that.

(Refer Slide Time: 45:48)



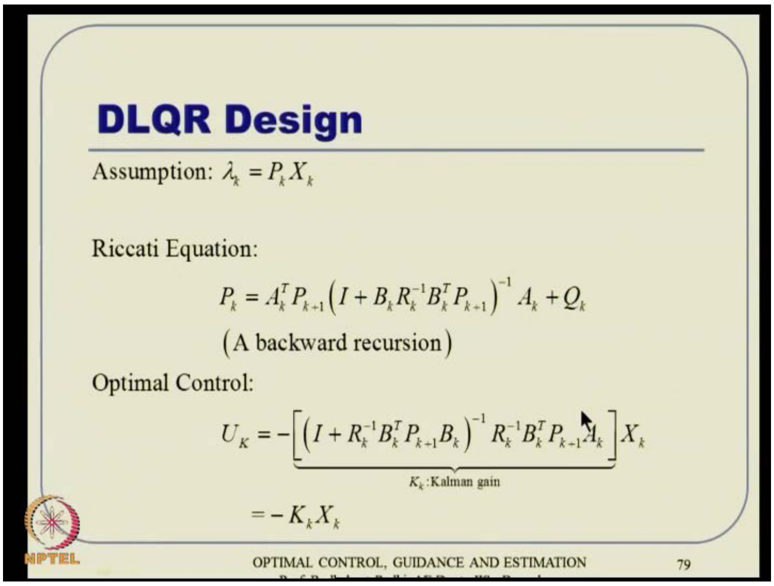
So, now, coming to discrete LQR, so we this, we have a linear form of system dynamics and discrete setting. We know particular advantage of taking, whether the time invariant of time varying, so we consider time varying in general and then performance index is, is quadratic as well.

(Refer Slide Time: 46:07)



The Hamiltonian and all we can define and then, it lends up will land up with state equation, costate equation, optimal control boundary conditions all that actually in the discrete frame work. So, again, the state equation, same costate equation like these.

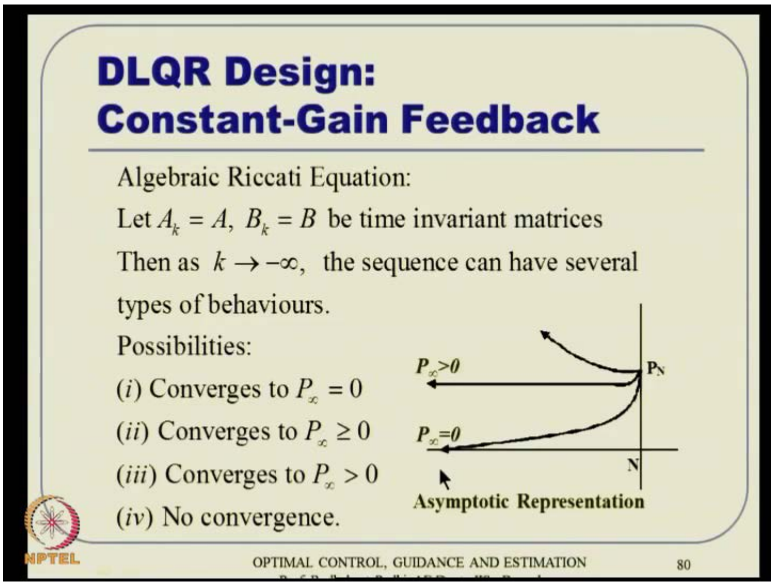
(Refer Slide Time: 46:29)



But lambda k happens to be a function of lambda k plus 1 now and U K is also a function of lambda k plus 1. So, with that observation we can, we can try to apply these and then try to derive this equivalent Riccati equation and all that. So, you are, again we have seen lambda k is nothing but P k times X k and that entirely derive this, this Riccati equation, which is again a backward recursion actually.

Then, optimal control equation can also be analyzed and it turns out, that it can be given this form where their entire big matrix, what you see here, is nothing but the Kalman gain actually. So, U k is minus k times X k.

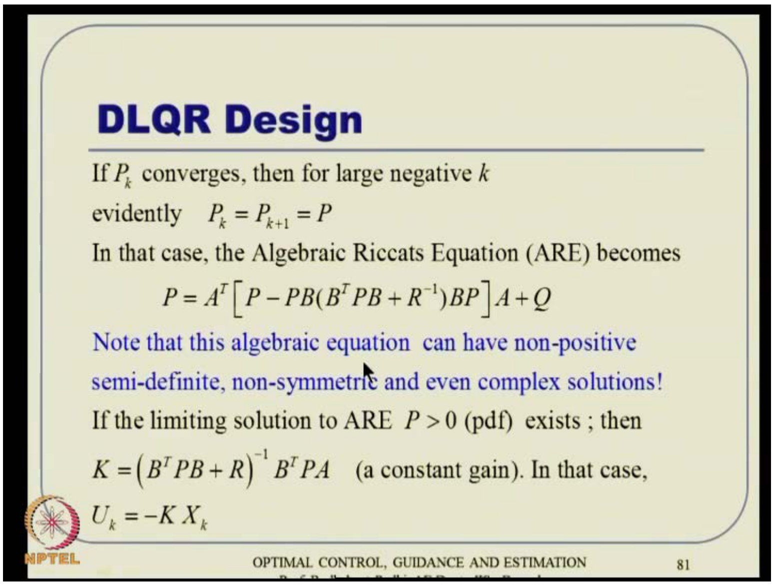
(Refer Slide Time: 46:55)



So, this is the DLQR design path, but it essentially turns out, that there is no, not much of guarantee assurance sort of things here. So, there are possibilities actually.

So, the possibilities, like when you start with P N and try to integrate backward actually, it can actually converges to P infinity 0 or P infinity some finite value, which is greater than 0 and converges to some P infinity, strictly greater than 0 or the no convergence at all actually.

(Refer Slide Time: 47:22)

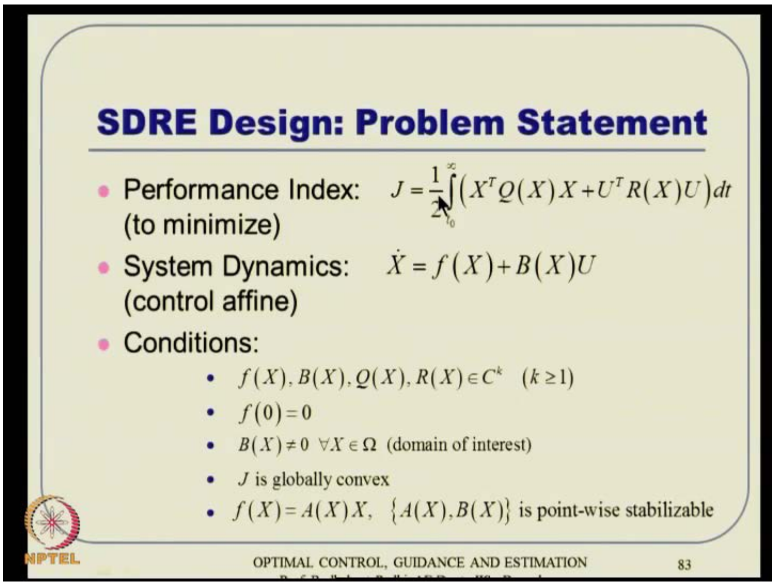


So, depending on the situation view to, to analyze or take that particular value, which is (( )) actually. So, essentially, if you tell, that on the steady state I have a constant matrix, that means, P k equal P k plus 1 equal to P. So, in that case, algebraic, algebraic Riccati equation becomes this way. And this algebraic Riccati equation can have no positive, semi-definite, non-symmetric and even complex solutions actually.

So, if the limiting solution of ARE exists and is not, not guaranteed to exists, but if you take these, then k happens to be like these actually, which is actually nothing but a constant k. In that case, we can write U k is minus k times X k, where k happens to be a constant gain actually.

So, these are discrete time, but again, because of advancement of computer technology and all that, now people do not have discrete time, do not give that much importance to discrete time formulation actually. So, continuous time result can be implemented with very high, I mean, frequency update then is as good as applying that with, I mean, discrete time theory at it will becoming more or less kind of absolute actually, so that too much importance was not necessity actually in that sense.

(Refer Slide Time: 48:48)

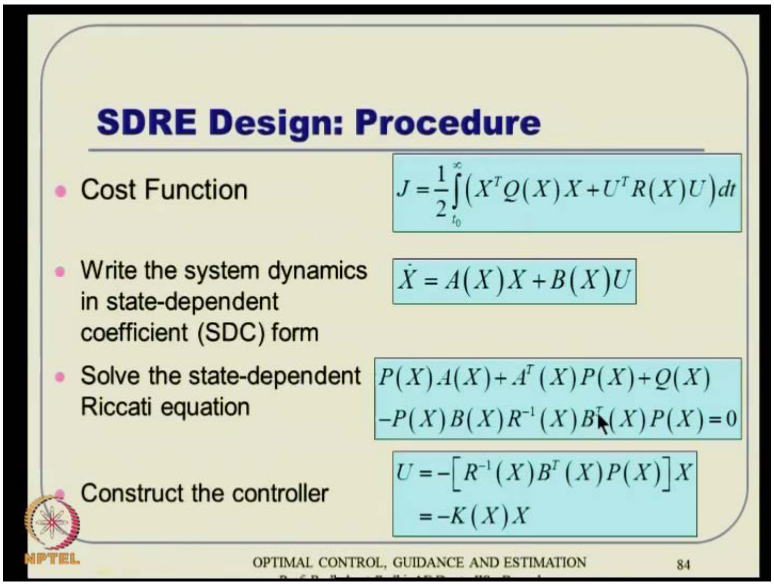


But anyway coming back to that, the further extension we, we discussed about something called state dependent Riccati equation design, which is nothing but SDRE design sort of thing and this is actually a slight extension of LQR idea for, for a particular class of non-linear problems and where the performance index is still given as quadratic looking form. It may not be really a quadratic, remember q is a function of X now and R is the function of X also, but as long as it is written like that it feels as it it is quadratic actually.

And system dynamic happens to be in this form, which is control affine; that means, linear in the control variable, whereas it can be non-linear adjusted variable actually. And then, the condition necessary are something like this, f of X, B of X, Q of X, R of X has to be class C k is at least 1. They are all smooth function, the 1st gradient vector at least is continuous actually and f of 0 has to be 0, that is a critical observation in this form. And B of X has to be non-negative all the time in the domain of a interest, J has to be globally converse. Remember Q and R are not, not really positive, semi-definite, positive, definite matrices.

So, we cannot guarantee that , but condition is irrespective of whatever they are, they are to be given, entire function has to be globally convergence actually. And then, one critical observation is, whatever this f of X, it has to be written in this form, which is like linear looking form sort of thing, is not really linearization, remember that. The Taylor series (( )) not valid, we simply write it some, some looking from actually. In other words, if it is X minus X Q, we can write, always write is 1 minus X square whole multiplied by X, actually that kind of thing.

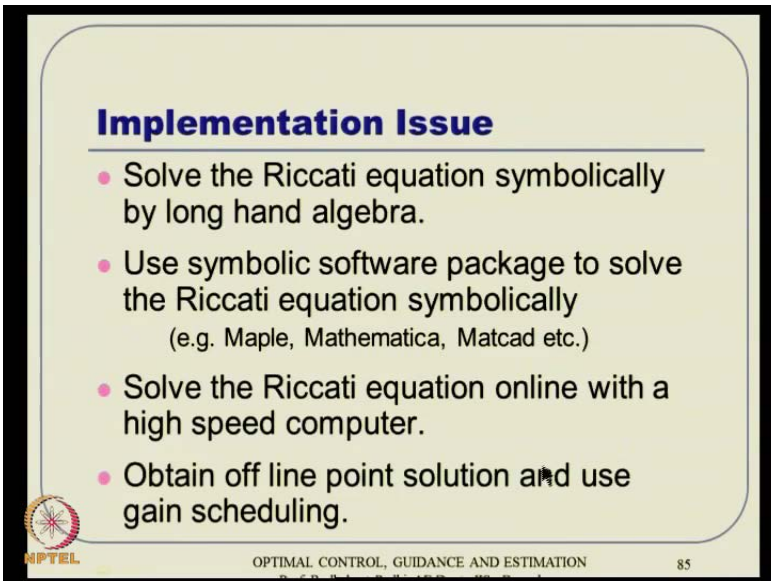
(Refer Slide Time: 50:19)



So, if we write it that way and assume that A of X, B of X is point-wise stabilizable at all points actually, then what happens? The cross function appears to be linear, I mean, quadratic, it feels like quadratic. If you, if you close your eye for Q X and R X for a second, that it feels quadratic and the system dynamics is, is written in this, this so called state dependent coefficient form. That means, this f of X is written in A of X times X. So, now, it, it also appears to be linear, not really linear, but appears to be linear.

So, now, linear looking state equation and quadratic looking cost function. So, the idea here is, we can treat for a moment that at, at any particular grid point, it is actually a quadratic regulator problem and hence, we go back to the LQR theory, entire locate this, we solve this required equation at every grid point online. And then, whatever P matrix comes out of that, the positive definite P matrix, I can use it in this formulation and compute again k for, remember k is function of X actually, because every grid point it has to take a different value actually. So, (( )) essentially comes up with a non-linear controller, but a linear looking structure essentially basically.

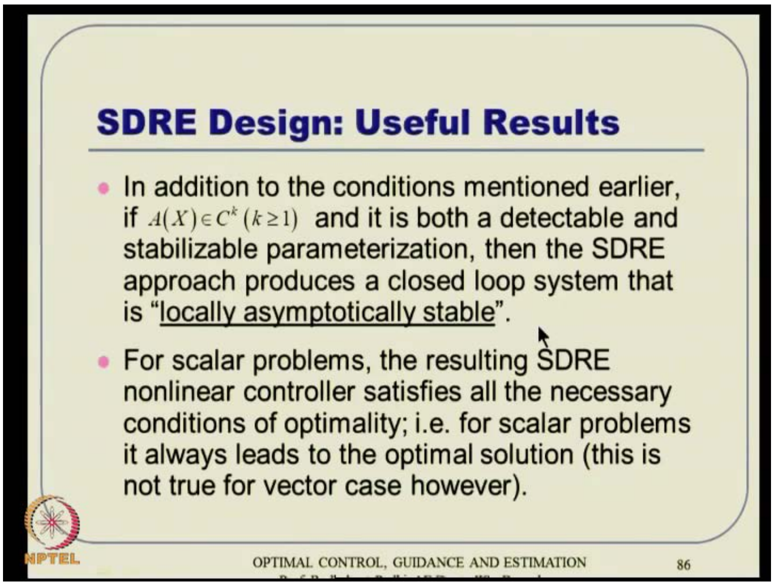
(Refer Slide Time: 51:22)



And then, there are implementation issues because we are demanding online solution of Riccati equation, how do you do that? So, first of all if, if it is possible, then Riccati equation can always be solved by long hand algebra, especially if it is a two-dimensional problem, sometime these 2-D guidance problems and all, the people end on it. So, they analyze the Riccati equation, kind of extensively in the symbolic form itself, and come up with some sort of a symbolic solution, this is the best thing if it is possible.

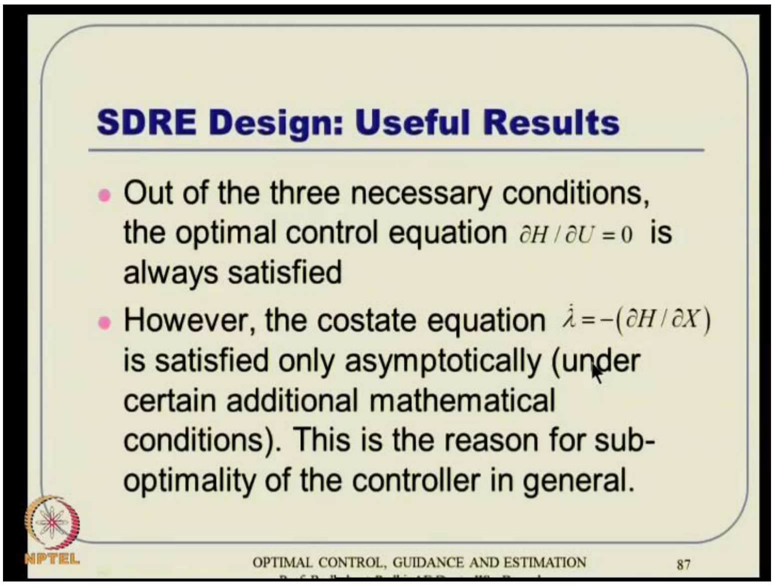
Otherwise, the next basis, probably use some sort of a symbolic software package to solve the Riccati equation symbolically, the variety of software available, something like Maple, Mathematica, Matcad, things like that, that can be used and then, solve the Riccati equation. Otherwise if it is, that also is not that, then you, to solve that Riccati equation online with a very high speed computer and that is no, no more a distant reality always. It is actually a reality because now people use, even for aerospace applications people use power PC and things like that. But then, if for ground phase applications, where you can actually implement some sort of a fast processor and all in with relative is actually, that is not a very big deal, we can, we can solve online also. If that, that also is not possible for whatever reason, then you can always get some sort of offline solution and go for gain scheduling ideas actually. Then, it is, here is a result, not only it is just intuitive and hence elegant also for implementation, but (( )) of some sort of nice results actually, nice analytical theoretical guarantee actually.

(Refer Slide Time: 52:42)



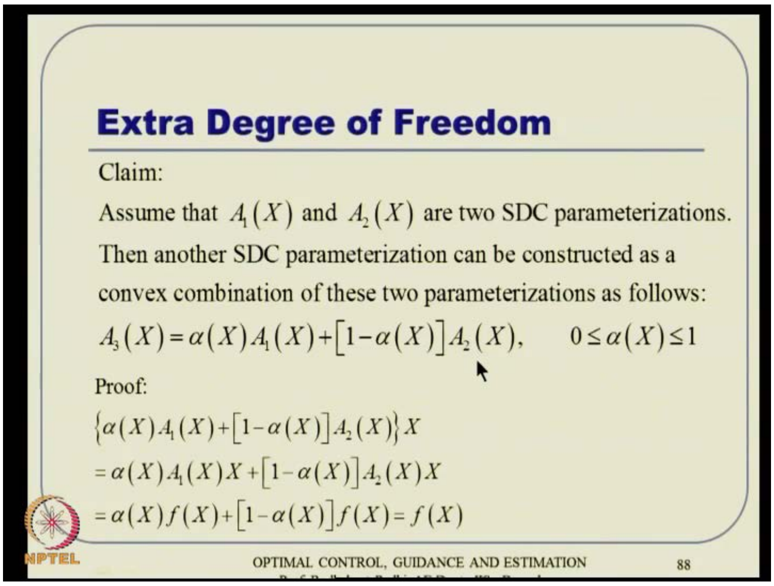
So, one of that, it tells, that, LQR solution is always guaranteed to be locally asymptotically stable and for scalar problem, it is guaranteed to the optimal, that is what these theorems tell.

(Refer Slide Time: 53:02)



And then, why it is sub-optimal? It is not necessary optimal because of the fact, that out of the necessary conditions, state equation is anyway satisfied. In addition to that optimal control equation is also always satisfied. The one that is not satisfied is the costate equation actually for this costate equation will also get satisfied asymptotically. As the control starts getting applied to the system, it will get, I mean, it will start satisfied actually.

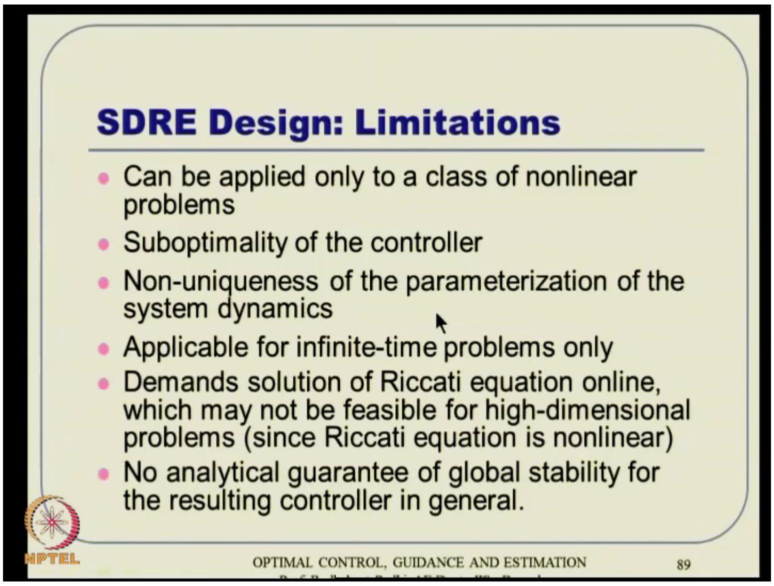
(Refer Slide Time: 53:28)



Then, it also, this frame work gives some sort of extra degree of freedom, which tells us, that remember this writing, this SDC form is non-unique. So, people can write in a variety of ways, but still you can come up with, somebody comes with A 1 and somebody comes with A 2 and both are 2 valid SDC parameterizations, remember that. Then, somebody (( )) can little more smarter and tell, well, wait a second, I can actually formulate A 3, which is a convex combination of these A 1 and A 2 in thus, that form and then, it give some additional adjustment tool or tuning tool alpha, which I can actually adjust if I am, if I am able to adjust it in a, in a good way. Then, I will get a much better solution using A 3 instead of either A 1 or A 2 and in fact, we have given some two-dimensional example, when we were discussing this, this SDRE lecture actually.

So, you can do some sort of an offline optimization formulation and then come up with a proper value of alpha, so that this A 3, which is derived out of that some convex combination like that is a valid, is a valid SDC parameterization, which will give much better results compared to either A 1 or A 2 actually. So, that kind of things also available and then, whether A 3 is a valid parameterization or not, this, this is very easy to see it. So, that is actually a valid parameterization because once you, once you plug in this formula, it actually based on same f of X. Obviously, this A 3 is, is a valid parameterization as well.

(Refer Slide Time: 54:52)



Now, there are certain limitations of this design also we solved. Then, 1st thing is, it can only be applied to a class of non-linear problems. That means, the system dynamics has to be control affine especially and, and quadratic cost function has to be taken, so that, that confines to the class of problems where this thing can be there. But remember, is not a major restriction in the sense that many practical problems do naturally fall into that frame work actually.

Alright, then it has to sub optimality of the controller. So, it, it actually leads to kind of a suboptimal control, all the necessary conditions of optimality gets satisfied only towards end. That means, only asymptotically the costate equation will get satisfied. So, that way, it is a suboptimal actually.

Then, we have non-uniqueness of parameterization was a major difficulty. That means, how do you write this f of X equal to A of X times X, that, that becomes A R, that becomes lot of open questions there basically.

And then, we have this application, it is the entire design is applicable only for infinite time problems because Riccati equation has to be solved, we cannot be a dynamic equation, it has to be algebraic equation. Algebraic equation demands, that the t f has to be infinity actually. So, that is applicable only for infinite time problems basically.

Then, also see, that the, if actually demands the solution of Riccati equation online, which may not be feasible for high dimensional problems especially. So, if the, if your system dynamics says something like 50 states and 100 states and all that, which is not, I mean, some problems arise, that way especially this, this structural control problems are, let us say vibration control problems like that actually, if you talk about that, always say in general, this, this distributed parameter system, infinite dimensional system were large number of states are required actually that way. So, those kinds of problems it may not be possible to solve this Riccati equation online actually

Now, no analytical guarantee of global stability is available even though local stability is, that global stability is there actually. So, that is, what we, we discussed up to SDRE and then, this is where I will stop this lecture, but the next lecture will continue this review and then, if then, so very quickly of many other things, that we have discussed as well. Alright, thanks a lot for this time.