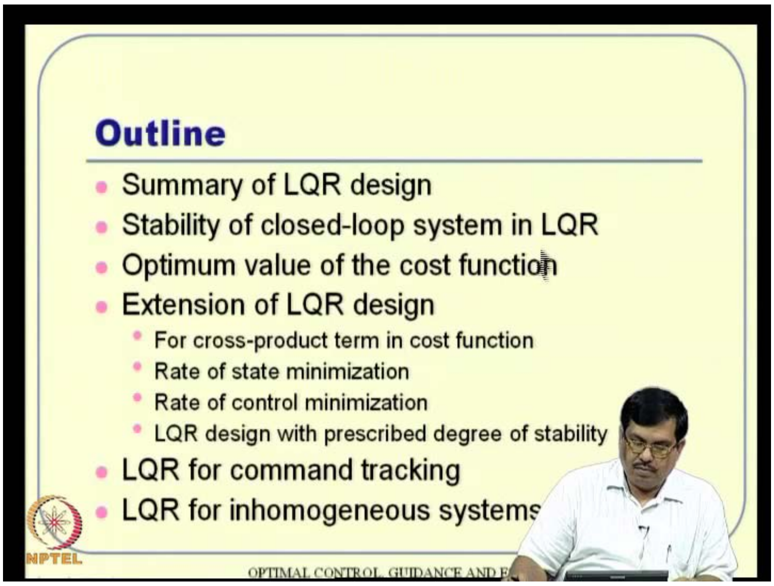
Turbomachinery Aerodyanmics
Prof. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Science, Bangalore

Module No. # 05
Lecture No. # 11
Optimal Control, Guidance and Estimation
Linear Quadratic Regulator (LQR) - II

Hello everybody, we will continue with our lecture series for Optimal Control, Guidance and Estimation. Last lecture we have seen LQR - Linear Quadratic Regulator design development and as I told in the last class, we will continue several lectures on this topic. This is one of the very popular optimal control techniques as well as a technique, which is already implemented in industry and all that; it is so important, so we will give the importance in about three four lectures actually.

(Refer Slide Time: 00:47)

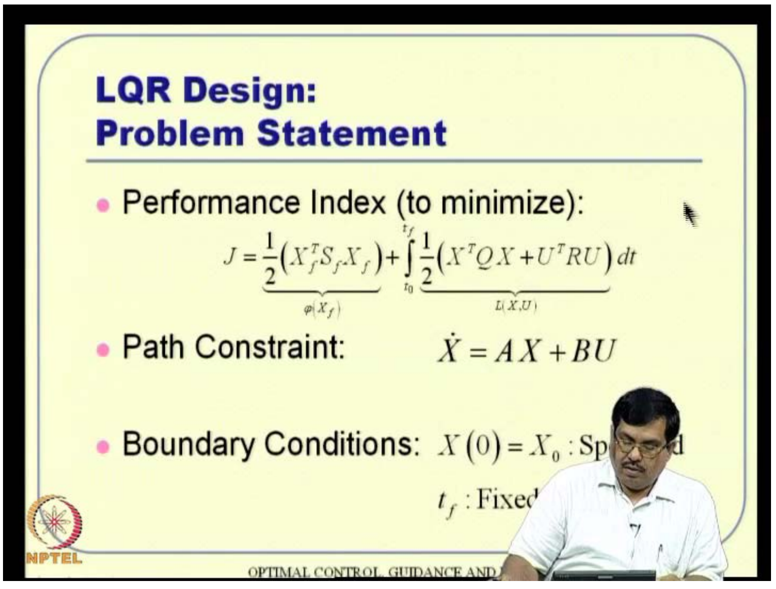


So, this particular lecture, I want to cover some of these, first very brief summary of this LQR design what we discussed in the last class. Then, we will formally prove the stability nature of the closed loop system with a LQR control, so that will be through the optimum theory and all that, so but formal proof we will give actually.

Then as a bi-product, we will also be able to say what is the optimum value of the cost function in an infinite time problem and that is that is kind of closely link to each other basically, we will see that. Then, we will have some couple of extensions of this LQR design and you can see the literature, there are quite a few extensions possible actually, but we are not going to talk everything, but we will give a few extensions which will kind of give you confidence that you can given a problem, you can probably read, understand and probably develop yourself as well actually.

So, for those things will come, I mean this four five examples that will give all extensions that will discuss here, first term is the cross product term in cost function and that is something like X transpose U sort of thing, if you if you have that kind of thing, then what you do. And then, there is a rate of state rate of state minimization, that means you want to minimize X dot, similarly rate of control minimization, you want to minimize U dot term and if there is also a extension, which talks about some prescribed degree of stability, how do you execute that? So, this kind of things we will we will discuss in this particular lecture, we will also follow it up with the formulation for command tracking as well as all formulation for inhomogeneous system. That means, if you have X dot equal to X plus V plus C then, this called inhomogeneous system, because when X and U are 0, X dot is also 0, I mean a sorry when X and U are 0, then X dot is still C basically. So, that kind of things are called inhomogeneous system and all that, we will see the all this extensions in this class actually.

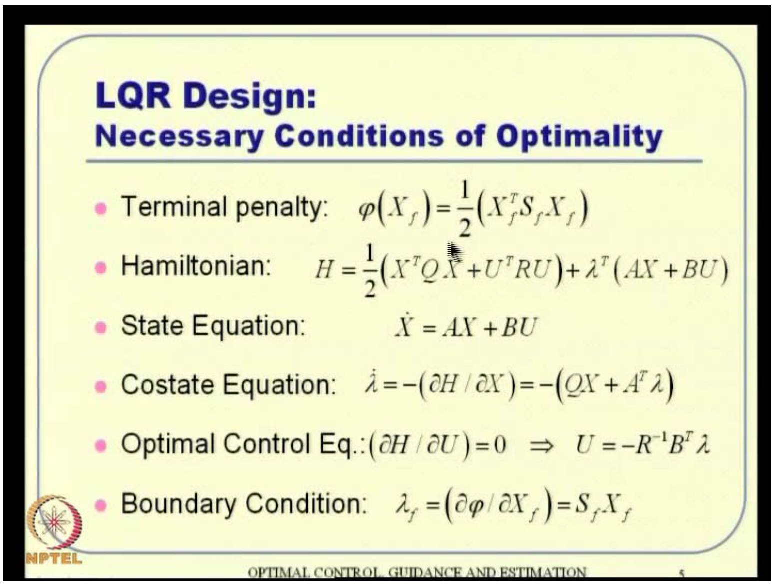
(Refer Slide Time: 02:53)



So, first a brief summary of this LQR design what we discussed in the last class. The performance index in this LQR formulation is something like this, you have half X f transpose S of X f, which is the total thing is phi of X f plus integral t naught to t f per half of this thing and then X transpose Q X plus U transpose R U. Several things we discussed last class that R is to U positive definite, Q X is to U positive semi definite, S f is to U positive semi definite like that and there are some guide lines about the selecting of X of Q and R, all those things we discussed. And this also a fundamental requirement of A and B pair has to be controllable as well as A and square root of Q that pair has to be detectable or in general observable actually.

So, this is the perform index that we want to minimize subject to the path constraint, X dot equal to A X plus B U is a system dynamic linear systems and then followed by boundary conditions, where X of 0 is X 0 which is seem to be specified, t f is fixed and X of X of t f is free, so those kinds of formulations we discussed in the last class. So, now went back and then started with this Hamiltonian definition phi f phi f x f definition like that.

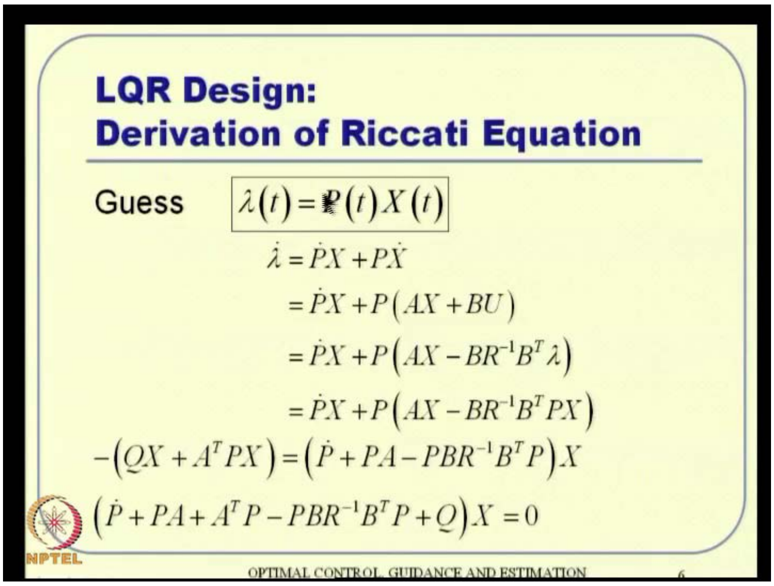
(Refer Slide Time: 03:58)



So, phi of X f is very naturally like this and Hamiltonian is l plus lambda transpose f and l turns out to be this one and f turns out to be this one (Refer Slide Time: 04:10). So, l plus lambda transpose A X plus B U, so we are ready now to apply our necessary conditions of optimality, which is state equation, which is X dot equal to l del X by del lambda which is nothing but, A X plus B U, f of X U basically. So, X dot equal A X plus B U and then we have this costae equation, lambda dot equal to minus del H by del X, which is minus Q X plus A transpose lambda.

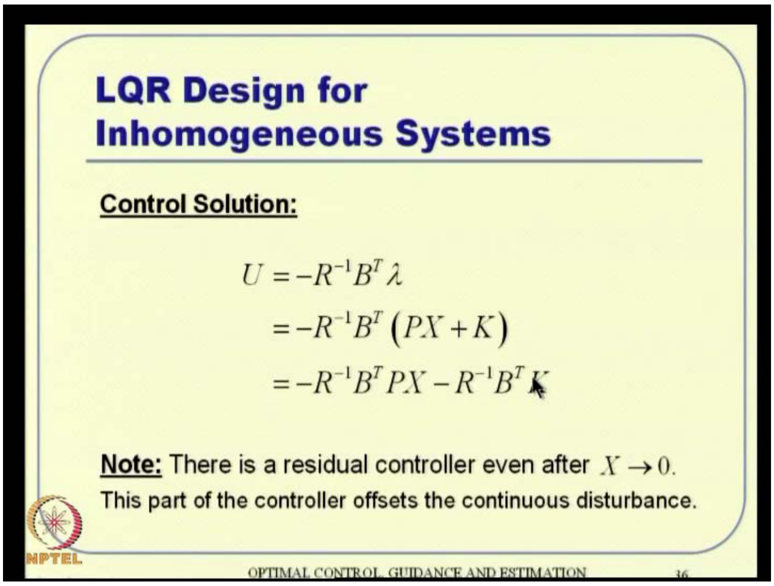
Then you have optimal control equation, which is nothing but, del H by del U equal to 0 which leads to U equal to minus R inverse B transpose lambda, we all derive this actually last class. Then finally, the boundary condition turns out to be lambda f equal to del f by del X f and phi of X f is this one. So, we can do this partial derivative and tell lambda f is to be S f X f. Then motivated by this I mean this expression, where that t equal to t f lambda equal lambda is a linear function of X f, so and as well as some other argument.

(Refer Slide Time: 05:14)



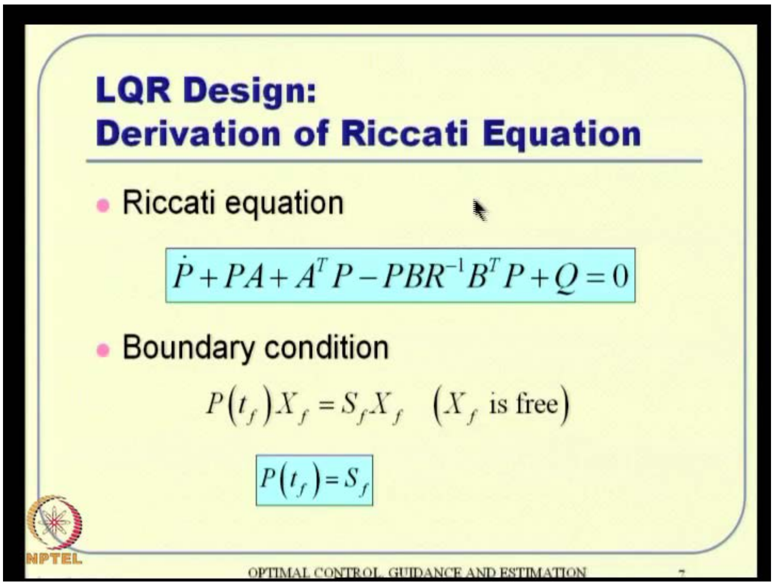
So, we happened I mean we assume that, how about how about lambda of t is something like linear function of X of t at every point of time, that linear dependency can be different; that means, P of t can be actually time varying function, time varying matrix rather, but it is at every point of time, we still have this linear relationship actually. Then we went back and it applied these conditions, whatever conditions we have, state equation and costae equation and optimal control equation and we started with differentiating this expression what we have.

(Refer Slide Time: 05:48)



So, lambda dot turns out to be like that, X dot we substituted as system dynamic, then U be substituted as this expression, where lambda is again substituted as P times X actually. So, we have then lambda dot was nothing but, this expression minus Q X minus A transpose lambda, where lambda is nothing but, again P times X. So, we putting all together and bringing everything to left hand side and assuming that X is not equal to 0 and this expression is valid for infinite combinations of X and all that, so you can tell that coefficient has to be 0.

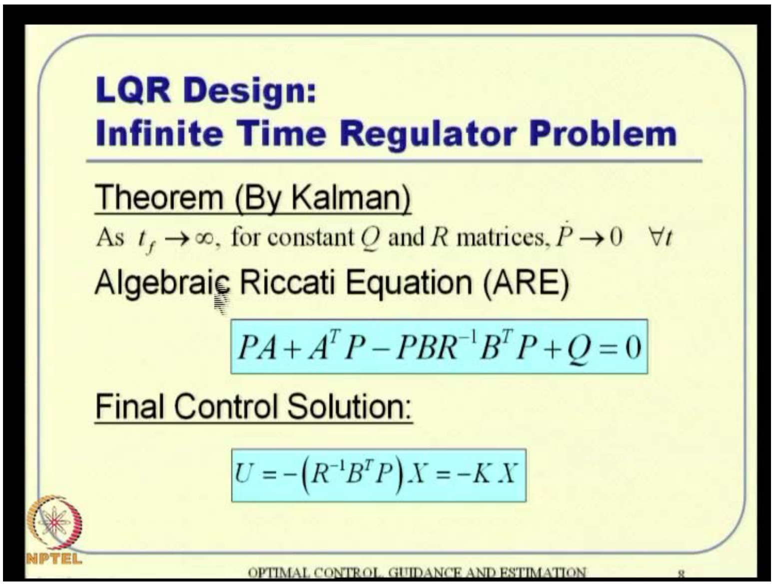
(Refer Slide Time: 06:13)



So, this essentially leads to this Riccati equation, which is essentially is a differential equation, differential matrix equation rather, with a boundary condition popping of from this boundary condition. So, you have lambda f is P f X f which is equal to S f of X f, so P f turns out to be something like S f actually. So, what we told is we start with this boundary condition and we have a differential equation for this Riccati matrix anyway, so you can back propagate it from t f to t naught and then from P f T for all t belong to t naught to t f is available. Once P f t is available, then we can calculate lambda of t and once lambda of t is available, you can calculate this U of T, so U is available for lambda.

So, on the on the way we also observe that by Kalman's theorem, when t f goes infinity for constant Q and R matrices, P dot happens to be 0 for all t; so that means, this equation what you have as a differential equation, it based on to an algebraic equation and this is what is popularly used all over the place and by default, if t f is not mentioned then it is assumed that, t f goes to infinity.

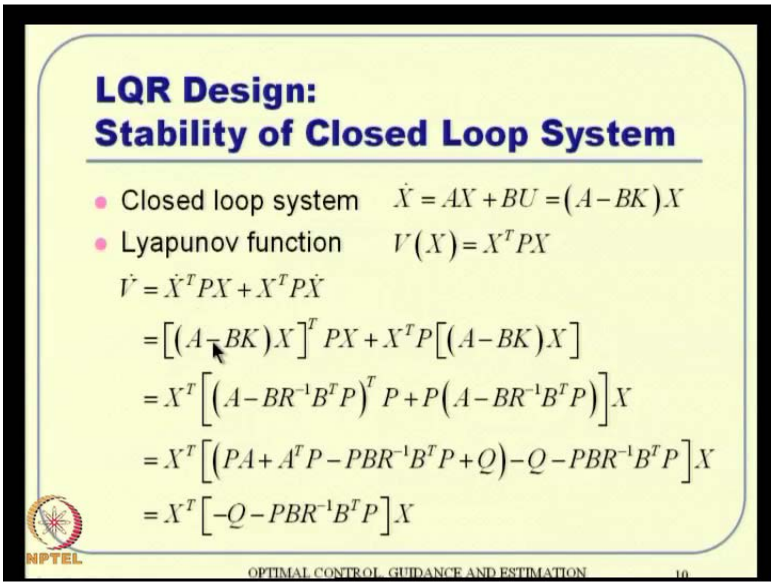
(Refer Slide Time: 06:59)



So, this is the way to solve, but this we don't need to propagate anything, but still it is a non-linear matrix equation, which needs to be solved actually, but there are efficient algorithms available because this equation is heavily studied. So, once you get a solution either this way or that way at P f T, then we have this control ready which is U equal to minus R inverse B transpose lambda, where lambda is P times X. So, combining this P to the side, you can tell that there is a gain matrix R inverse B transpose P, and hence U equal to minus K X, where K is nothing but, R inverse B transpose P, so this is the whole summary of the LQR design actually, alright.

So, with this idea or with this knowledge, we will proceed further and first we saw stability of the closed loop system in LQR actually. Remember, when I am not talking anything about finite term, infinite term this are all we will do discussions based on infinite time formulation actually.

(Refer Slide Time: 08:11)



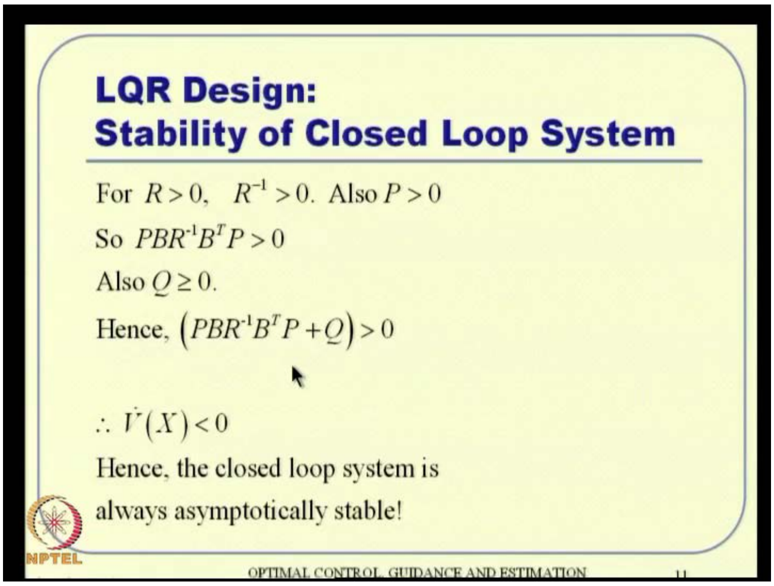
Alright, So, we have a closed loop system X dot equal to A X plus B U, which is nothing but that and Lyapunov function V of X is X transpose P X. So, V dot of X is nothing but, X transpose P X plus X transpose P times X dot, remember we are talking about infinite time, so that means P is a constant matrix actually, so that P dot is not involved for (( )). So, we have X dot equal to A X plus B U again, so substitute that X dot equal to this expression A minus B K times X. So, I can substitute it here, a minus X dot is nothing but, A times B K, A minus B K times X whole transpose P X plus X transpose P times X dot again, so X dot is nothing but, A minus B K X substitute like here.

So, that results to like if I expand this transpose and that is also X transpose A minus B K transpose times B, so that you can substitute it back here, where K you know that sorry K is nothing but, this R inverse B transpose P. So, that you can substitute back here, R inverse B transpose P nothing but the K basically, so this similar thing, this is also K. So, substitute back and then expand it fully and then do this plus Q minus Q thing here, once

I do this plus Q minus Q, you can observe that one part of the equation is nothing but, the Riccati equation actually. So, then this being the Riccati equations, obviously this X, this part of the equation will go to 0 and you will be left out with the rest of the term actually.

So, what is this? Now B dot you can is nothing but, this expression actually, this is remember, this is already in quadratic form, so all that we need to so and this is personately there is negative sign in both the expression actually. So, we take out the negative sign and all that we need to show is, this V dot needs to be negative definite function, so these are all Lyapunov theory based arguments sort of thing. So, we start with a positive definite P, that is our Lyapunov function that means, V of X is a positive definite function and we need to show that V dot is a negative definite function. So, that is another reason why we need to select positive definite free, I mean P may be multiple solution in this Riccati equation, but we need to select a P which is positive definite actually. Anyway, so this being a positive definite, so this positive definite function and V dot turns out to be this expression with if I take the minus sign out, it will turns out to be Q plus this expression, PBR inverse B transpose P actually, now you can already see that Q is a positive semi definite matrix.

(Refer Slide Time: 11:09)



So, if Q is a positive semi definite matrix, what about this expression (Refer Slide Time: 11:12)? Remember R is a positive definite matrix, so R inverse is also positive definite, remember if a matrix has Eigen values lambda, then it is inverse is 1 by lambda is Eigen values. So, if all this Eigen values are positive for R, there R inverse also will have positive Eigen values actually.

Now, this expression you can see B times B transpose, that is a quadratic expression sort of thing and this is positive P already a positive definite matrix. So, this will lead essentially lead to this entire P B R inverse B transpose P is a positive definite matrix actually. So, you have a positive definite matrix and negative that of negative that is negative definite and minus Q is also there, so that means, that is negative semi definite. So, the total expression become something like negative definite matrix actually, these all explained here, same thing what I told for R; this notation what you see here is not just greater than, they are not scalar quantities R is a matrix, so by symbolically it means R is positive definite matrix.

So, R inverse is also a positive definite matrix, P is also positive definite matrix, this happens to be a positive definite matrix and Q is a positive semi definite matrix. So, the combination of that is positive definite matrix and hence negative of that happens to be negative definite matrix. So, V dot happens to be negative definite and by the Lyapunov's theorem, direct theorem actually that if V of X is a positive definite function and in V dot of X happens to be a negative definite function, then the close loop system is always asymptotically stable. And this is the this being the linear system and this is like if you can extend further, these are all radically unbounded system, I mean radically unbounded function and things like that and so essentially it happens to be globally asymptotically stable.

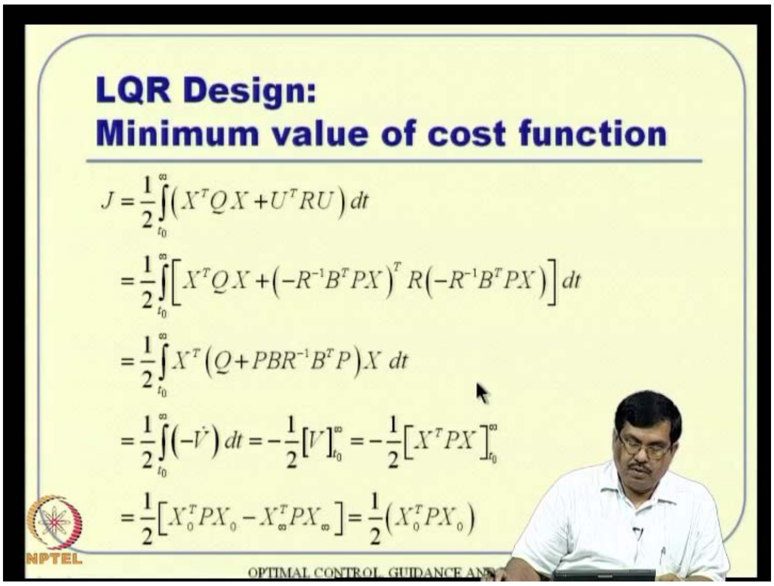
So, that kind of thing, but also remember we are all talking about linearize system, linear system or typically linearize system, so for the linearize system, the linearization is valid in a local neighborhood any way. So, what you are talking here is, it is a global asymptotically stable for the linear system, I mean not for the original non-linear system actually, so just keep that in mind it is a comment sort of thing. As far as linear system is concerned, if you apply LQR you will have global asymptotic stability guarantee basically and also it turns out that because of this expression, this nature X dot equal to this is something like a close loop matrix times X.

So, you can talk about X of T is nothing but, E to the power A minus K into T times X 0, so that is the solution. So, essentially it turns out to be exponentially stable also basically.

So, it is the strongest motion of stability in non-linear systems theory, it satisfies actually, so it is globally exponentially stable sort of thing actually. Any way, so this is I mean more on that, if someone is interested what to see, what not to see and all that actually in the Lyapunov theory, I encourage my other course which is already available in NPTEL program itself actually, which is advance control systems theory, where I have discussed about two three lectures on Lyapunov theory actually. So, you can see that for details somewhat actually, any ways the summary is we have we have one V of X which his positive definite, for which V dot of X tends to be negative definite.

Hence by lyapunov's direct theorem or sufficiency condition and all, it turns out that the close loop system is always asymptotically stable and hence it is globally asymptotically stable, because this is a radically unbounded function and all and is also exponentially stable, so there is very strong motion of stability actually.

(Refer Slide Time: 14:49)



Alright. So, there is stability there, so what about the minimum value of the cost function, is it possible to estimate that? So, now you can go back and see this value of the cost function is nothing but that, this t naught infinity we are talking about infinite cost functions here, infinite time problems here. So, J equal to this half of T naught 2 infinity all this expression, now U you substitute is minus R inverse B transpose B times X, so that is the expression for U and this is U transpose.

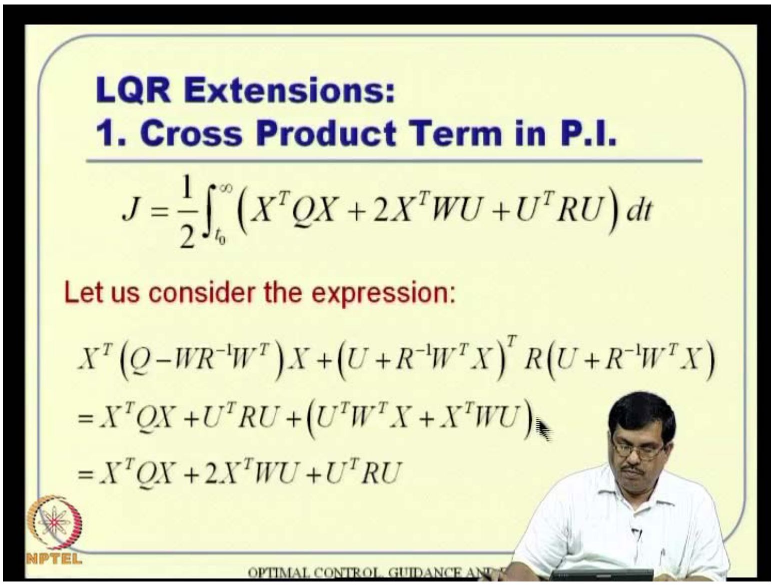
So, you expand that again X transpose Q X from this side and X transpose P is a symmetric matrix, so P transpose is P and then B, because B transpose is B then R inverse coming from here. So, essentially you have this expression, I mean if you if you expand this expression, you have this X transpose and then P transpose, which is P, then B transpose transpose which is B, then R inverse transpose, so that is R is also symmetric matrix. So, R inverse transpose is R inverse times R, times this matrix you have actually here. So, it turns out that, this is identity you can talk about taking it out and hence we left we are left out with this expression, half of X transpose Q, Q coming from here and then X transpose here and X is there, so that goes out, and then you have this P B R inverse P B R inverse B transpose P, so this is expression actually (Refer Slide Time: 16:00).

But, what is it? I mean this expression what you have here is nothing but our V dot, V dot is negative of that actually; so, minus V dot happens to be that basically. So, we put it that expression that this is minus V dot, so integration of T naught to infinity minus V dot times d t, remember this is D V by d t sort of thing actually. So, this integration and differentiation will cancel out and left out and we are left out with this V, half of V T naught to infinity and this is minus half of X transpose P because X transpose P X because V is that that one we selected that actually.

So, put it back and then it is T naught to infinity, so we evaluate it actually. So, X naught transpose times P times X naught minus X infinity transpose P times X infinity, but we just proven, just this is a asymptotically stable, it is always asymptotically stable, so that means, X of infinity is equal to 0 basically. So, with that term analysis and you are left out with that; that means, if you just know the initial condition and the Riccati matrix solution, you straight away have a value of the cost function actually, really I mean if you operate based on that, you are supposed to get this and probably this expression will help you in cross validating your results or kind of verifying results and all that actually.

So, this is minimum value of the cost function and now some extensions of LQR design, this is I mean LQR procedure, we know by now and infinite time procedure is also we know, we are guarantee to have globally a asymptotic stability for linear systems, stability theorem we proved and minimum value of the cost function also we derived actually.

(Refer Slide Time: 18:12)



Now, we see some extensions of LQR theory, LQR design; now let us see one by one actually. So, the first thing is cross product term in performance index. So, remember for originally, we had the performance index X transpose Q U plus U transpose R U, we did not have a cross product on something like this, X transpose W U. So, this W is also like a (( )) matrix sort of thing but, remember X and U are not of same dimension; that means, W need not be a square matrix actually.

So, with this expression let us analyze, how do you do that and the whole idea here is you can always go back to this Hamiltonian, try to derive the necessary condition and try from the beginning and proceed further, as we proceeded before, but that that turns out to be not necessary actually; if we manipulate this expressions little bit, then with existing solutions available, we can actually solve this problem as well.

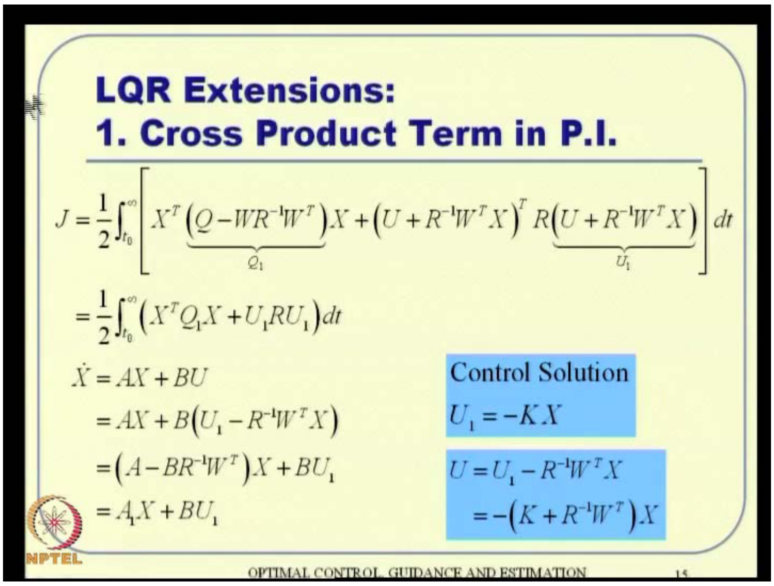
Now, how do you do that? Now, just see this expression whatever you have and then let us start with this expression, if somebody starts with this big expression and you try to simplify. So, this first term will be X transpose Q times X, the first term over here is U transpose R times U and some of the terms and all will cancel out and you will be left out with this term actually and then ultimately, it turns out that this is nothing but, X transpose Q X and minus see this one it is a scalar, ultimately all this things are individually scalar quantity.

So, the same expression times its own transpose equal to that actually, any time they were matrix multiplication is a scalar quantity, you know that scalar transpose is a scalar value itself. So, we can tell that this expression X transpose W U whole transpose is equal to X transpose W U. So, whatever we have this expression is a transpose of that, so you can tell that this is nothing but, same thing of that. So, that turns out to be 2 times this actually; that means, whatever expression we had here, we can actually substitute by this long hand expression actually.

Now, somebody can ask me, we have in how you got it and all, so this certificatelly does not happen overnight, but the idea here turns out to be in the reverse direction, you start with this expression here and this is now easy because you can talk about 2 terms like that and one term you keep it as it is, the other term you take a transpose of that. So, coming from third line to second line is rather very easy actually, but here we do little bit more manipulation, we add and add and subtract the term, try out I mean a couple of times and then and finally, you land up with this expression actually.

So, this one we have to big hand expression, let us assume that it is, it can be done actually. Now, what you see here? If you just look at it a little bit close to it, I mean closely, we have a quadratic expression here, X transpose something into X transpose and we also have a quadratic expression here; that means, some expression some vector transpose times R times the same vector, so that means we have a quadratic expression here and a quadratic expression here actually, so that turns out to be an advantage actually, how? Because we can actually define these as some sort of a new control variable and then we will have a direct expression actually.

(Refer Slide Time: 21:40)



So, now let us go back and substitute it here, instead of this expression what you have, we will substitute that actually in the cost function. So, we substitute it and J turns out to be half of T naught infinity and we substitute all that and this turns out to be a U 1 expression. So, this U 1 expression turns out to be I mean its available, we put it something like it is a small type of mistaken here is U 1 transpose, anyway this expression what you have here is nothing but, A X transpose Q X but we will substitute by X transpose this term, Q minus W R inverse W transpose X actually, so that is what it is here.

So, we define this matrix is something like A Q 1 matrix and define this vector something like U 1 vector. So, we land up with this half of X transpose Q 1 X plus U transpose R U, U 1 transpose R times U 1, this now turns out to be I mean looks very familiar to what we had actually. But, wait a second, we had do not have a cost I mean we do not have a system dynamics in terms of U 1 basically. So, we have to manipulate a little bit here and tell if my U 1 is define to be something like this expression, this my U 1 then I can actually this U, I can substitute from this, their U equal to U 1 minus this expression.

So, this what you what you substituted here and then what you have here is something like combine these two terms, we have A minus B times R inverse W transpose coming this side times X plus B U 1; that means, if you define this something like A 1 matrix what you have is X dot equal to A 1 times X plus B U 1. So, you have a system dynamics in the forms of linear system dynamics, X dot A 1 X plus B 1 and the cost function is also available to us in a modified form, this is X transpose Q 1 X plus U transpose R U 1, actually that kind of thing.

So, now this form this cost function and this system dynamics are both comfortable. So, we can use this A 1 B and Q 1 R matrices in the Riccati equation and solve with gain matrix actually, solve for a Riccati matrix, then we compute the gain matrix. Essentially this control will be available, but remember this control what you are talking is U 1 not U. So, U 1 will be minus K times X, but we know once we have U 1, you can also recover U from there, U is nothing but, U 1 times that actually, that is what you are substituted here, so once you have U 1, then you can get you U,U is nothing but, that.

So, you have a gain matrix K calculated using this A 1 matrix, B and Q 1 matrix R and then using that gain matrix and then using this little bit algebra here, we got the actual gain matrix for U basically. So, little manipulation of algebra saves a lot of I mean this necessary conditions sort of algebra and all that actually, you do not need that.

Alright. So, this is our first extension with cross product term, what about the kind of second extension? Well, weight-age I mean second extension, where we talk about weight-age on the rate of state; that means, we initially we talked about X transpose Q X plus U transpose R U, but sometimes it is necessary to minimize X dot also basically, you do not want the system dynamics to change very fast actually. There may be other verifications for that, there may be some safeties used, there may be some if it is temperature is there and you want to pump in some heat and then you can if you pump to much of heat and then probably this X dot becomes very high, then material may melt, all sort of things vary actually.

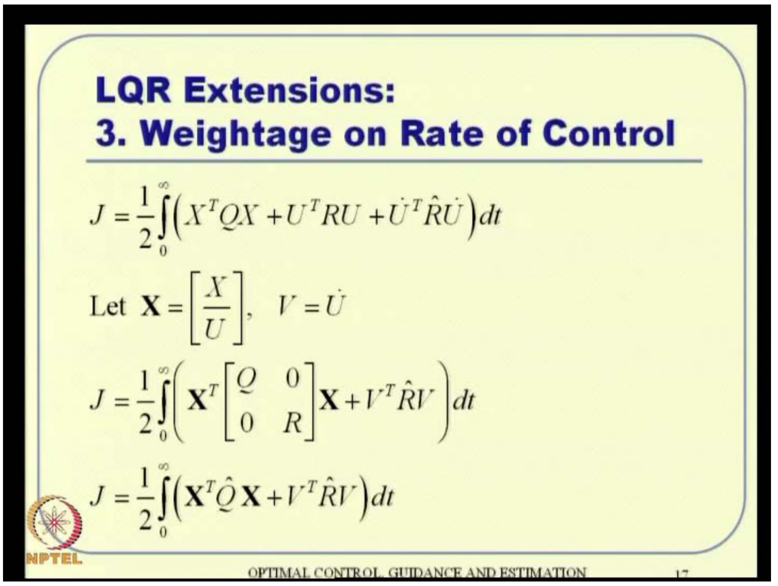
So, many applications are there, where we really do not want this X dot to develop as fast as possible or even decay as fast as possible, we have a smooth decay or smooth rise sort of thing actually. So, in those situations we like to have a minimization of that, but remember here X is a symmetric matrix because they I mean it is a square matrix as well as symmetric matrix sort of thing actually and typically when you see this kind of term, it is also assume that it at least positive semi definite matrix basically, so that the cost function remains convex actually.

Anyway, so if that is the case then, how do go how do you go about actually? The idea to go about is something like this, we have X dot naught, so X dot we have A X plus B U. So, substitute that X dot A X plus B U, so we have this J which is X transpose Q X plus U transpose R U plus all this, A X plus B U transpose S into a X plus B U alright. So, if that is the case, then you have this half of all this, X transpose Q X, U transpose R U plus all these actually. So, X transpose A transpose S times A coming from here and then all other terms this will essentially 2 plus 2, four terms will generate and all these four terms are given here actually.

So, now you collect this, let us say we collect this X transpose X sort of it terms, so we have one term here and one more term here. So, we can split here and U transpose, U sort of thing, so we can put it here, U transpose U is R coming from here and U transpose U some other thing coming from here. So, we have this X transpose this times X plus U transpose this times U basically plus 2 X transpose is this one, A transpose S B times U all this things you can put it back here.

Alright. So, essentially this leads to a cross product case because we have a term like this, two times X transpose W U, so that that term will give you something like this. So, we essentially define this matrix as Q1, this matrix as R 1 and you can define this matrix as W, essentially with this cost function J comes out to be this form, X transpose Q 1 X plus U plus U transpose R 1 U plus 2 times X transpose W U. So, this expression what you have here, essentially we can see that you have a quadratic term in X, quadratic term in U, but also a term with a cross product thing; that means, whatever we just discussed here, this term it leads to that, but now we know how to how to solve a cross product term case actually, if you have a cross product, how do we deal with that we know it actually, just know we discuss all that actually. So, this type of thing we can handle, that is not a problem at all actually, so now, what about extension number three usually?

(Refer Slide Time: 28:12)

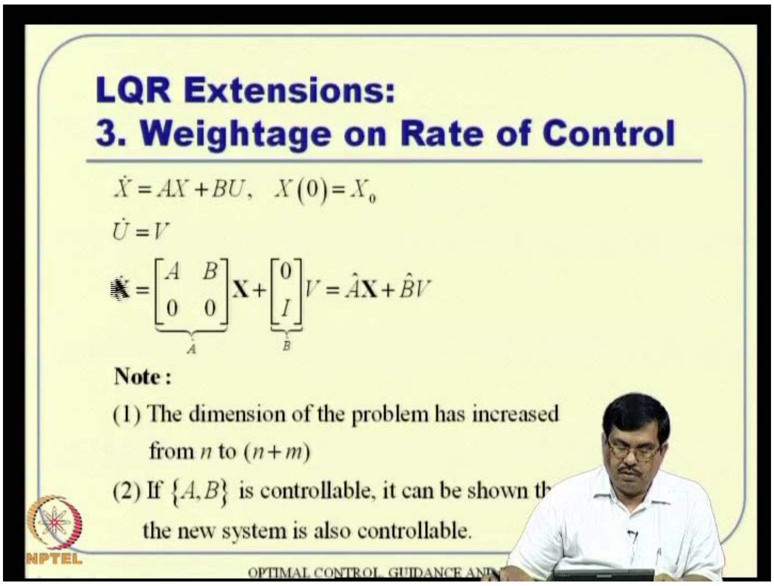


So, we now talk about, what about control rate minimization basically, we do not want to apply too much of control, I mean we do not want to see too much of build up of control rate basically and this is a very critical parameter, because if you talk about anything, any control, the real energy spent is through U dot actually. In other words, if you have a ((fin)) deflection of missiles or aircraft and thing like that, then the way to deflect control surface is through motor actually; so that means, that whenever there is A U dot term, there is a current, I mean usage of current actually if U dot is high, the current drawn is also high.

That means, the all energy stored in the battery will go, will drain out very fast and essentially this is also kind of preventing this scattering problems and things like that, we scattering is remember high fluctuate high fluctuating oscillation sort of thing; so that means, we have this U dot term very high basically. So anyway, so there are lots of practical applications, where we purposely want to minimize U dot also basically. So, how do you handle that and in this particular case, remember our R hat matrix has to be positive definite actually. So, how do you do that? Now, one way to do that is let us say you define some other variable let say capital X, X and U put together in a vector actually, so that, that is I mean that is our big vector state, big state vector capital X, X and U put together.

Then it turns out that V, I mean you can also define some auxiliary variable V, which is nothing but, U dot actually. So, now, we can see that if I do that, then this cost function J, we can write it this way, X transpose this capital X transpose this matrix times capital X plus V transpose R at times V basically. So, essentially if I define this as Q hat, then I have this J equal to X transpose Q hat plus V transpose R at V.

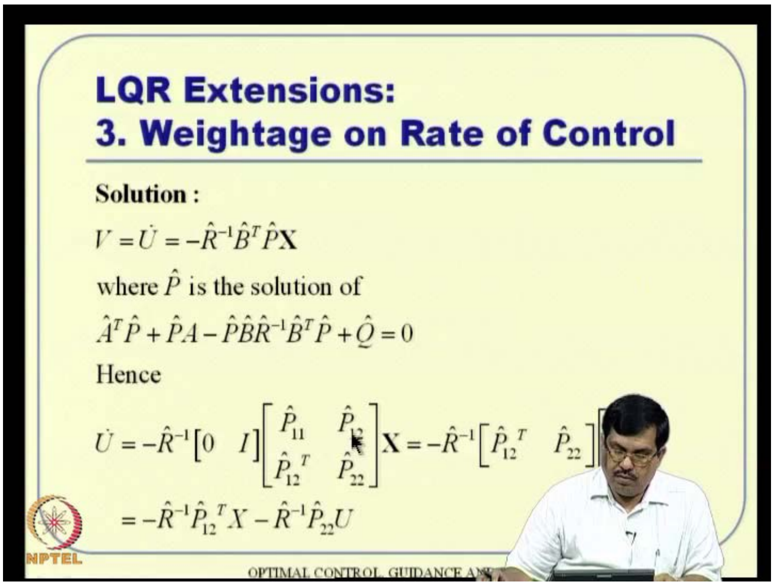
(Refer Slide Time: 30:30)



But, I mean you also remember that we have a system dynamics to get offer and system dynamics comes out to be like that, we have X dot equal A X plus B U and U dot equal to V; that means, if you see that in a capital X formulation, then this turns out to be like this; that means, this big X dot capital X dot turns out to be A hat X plus B hat V, where A hat is defined something like this and B hat is defined something like this alright.

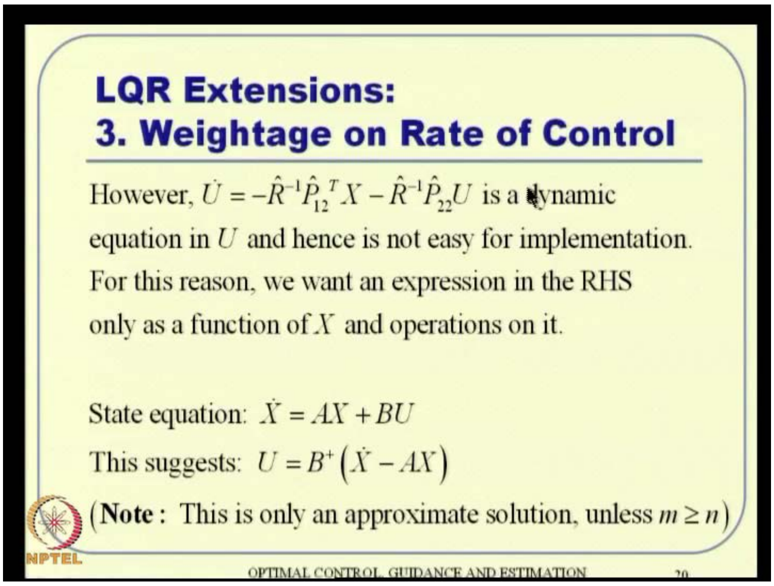
So, then first thing to note is the dimension of the problem is increased from n to n plus m, remember n is the state number of states and m is the number of control actually. So, we have put it together, so the big capital X dimension is n plus m actually and it personately it can be shown if pair A B is controllable, the original A B is controllable then, A hat and B hat is also controllable. So, that is in very important and interesting observation actually. Now, what the message here is we have a system dynamics X dot equal to A hat X plus B hat V and a cost function also in the quadratic form.

(Refer Slide Time: 31:39)



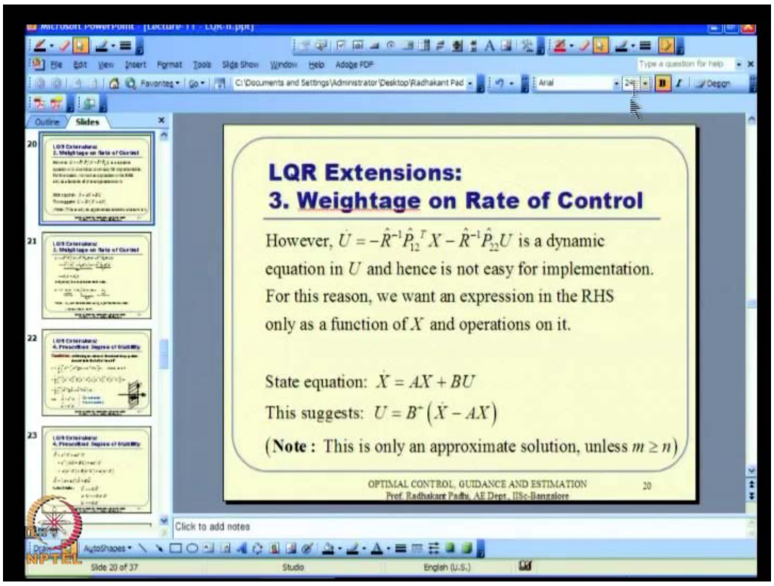
So, we can actually go back and use our LQR theory and hence we tell that, control solution, remember this control is actually V what you are talking here, that is the control variable. So, V is nothing but, minus R hat inverse B hat transpose P hat X, where P hat is the solution of this Riccati equation all that actually and remember what is the V is nothing but, U dot. So, U dot turns out to be something like this and X this point of time, I can split my X, X is nothing but, X and U and tell corresponding matrices and all that I will take it and I can write it this way. So, U dot equal to minus R R hat inverse P 1 to transpose X minus R hat inverse P 2 2 hat times U all this coming from this algebra actually.

(Refer Slide Time: 32:30)



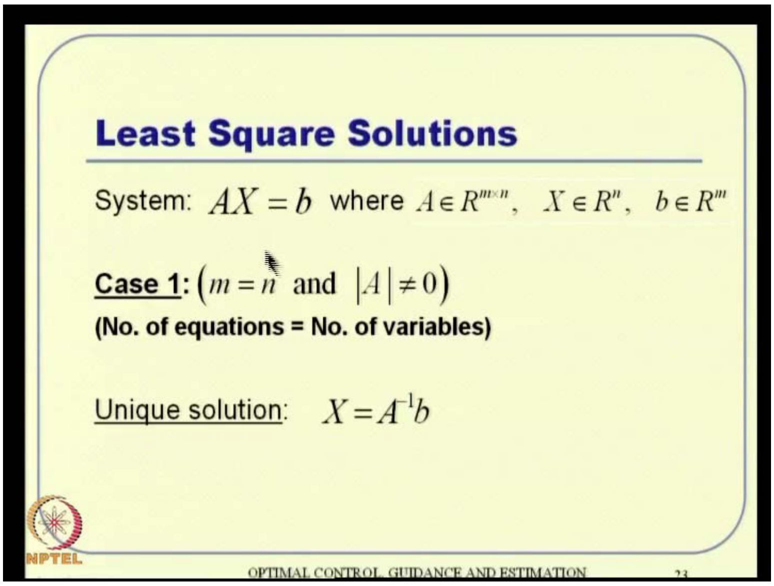
So, what is it? Now, if you see this, it is actually a differential equation, U dot equal to something times X and minus something times U. So, that is actually a differential equation that means, it is a dynamic equation; it is not easy for implementation and it prone to error also. That means any initial condition, see does not talk about initial condition on U, so any error initial condition is suppose to be propagate also basically, does not matter that much, but typically a dynamic control or is never good because there is some degree of kind of open look nature that actually. So, we do not want it, we want to operate it kind of a close loop some way basically. So, one way to do that is we will go back to this state equation and you see that if this is the case, then U can be solve something some like this, where B plus is nothing but, the pseudo inverse actually and this point of time probably it make sense to I mean kind of revise a little bit or you I have seen that, I mean we have discussed about that in lecture number two actually (( )).

(Refer Slide Time: 33:16)



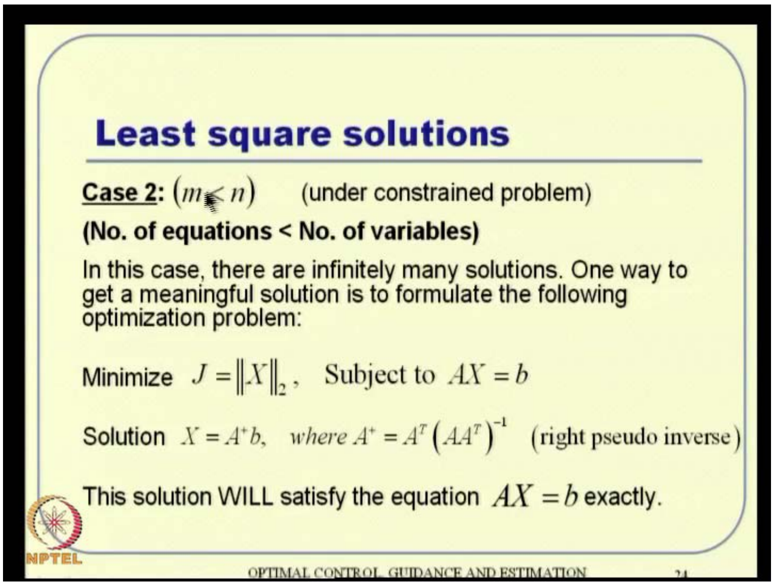
So, when you discuss this matrix theory and all that, we discuss those things and let me just quickly grave it. So, this is what we discuss that point of time, if you have A X equal to B then, m equal to n determinant A is not 0, then is a unique solution and all we know that first here and what about m is less then n; that means, what you are talking here is something like X is n dimensional and number of constraint is m dimension actually.

(Refer Slide Time: 33:35)



So, if a number of constraints is less than number of free variable; that means, its under constrained problem, then you can some do something like this, we want to minimize this norm of X and other words, you want to find minimum norm solution subject to this constraint and solution turns out to be pseudo inverse, where pseudo inverse is given like that and this is remember, this is under constraint problem.

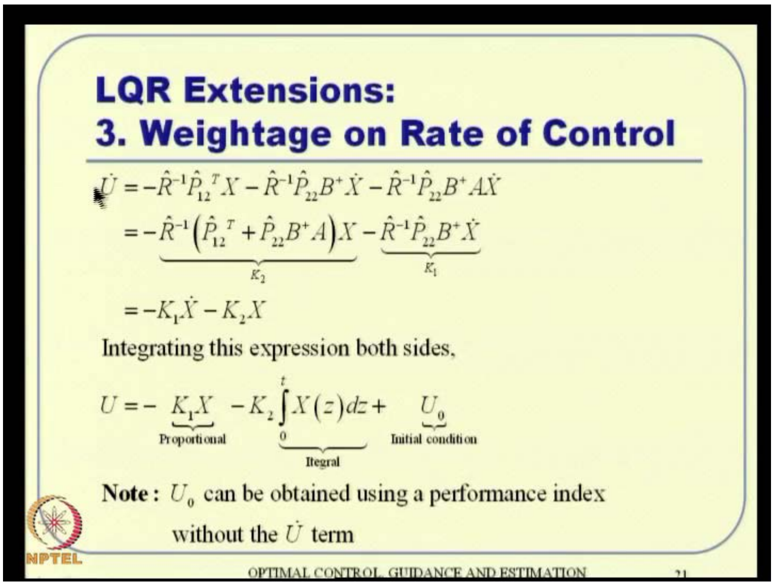
(Refer Slide Time: 33:42)



What about over constraint problem? If you are, if the constraints are something like this, then you can you can do something like this J equal to like that and other words it does not satisfy this equation exactly A X plus A X equal to b, but we want to kind of desperately attempt to kind of minimize this error A X is equal to B. So, A X minus B error minimization sort of thing, if you do that it turns out that the solution is nothing but, A transpose A pseudo inverse B again, but this time it is a left pseudo inverse, that was a right pseudo inverse actually.

So, these kinds of things we discuss, so what it turns out in our cross here, that one can always do this. And remember typically U is of lesser dimension than X; that means, this will always left turn out to be left pseudo inverse not very good, but in some sense it is it is close to that, I mean close to what it should be actually and as long as this number of control is equal to or greater than number of states, then it is very much correct, actually there is nothing wrong in that. Unfortunately, we will not have that case most of the time, that is a different case anyway. So, this is actually an approximate solution, so let us not bother too much in to that, but you can put it back here. So, we put this U whatever you see here with this expression, B pseudo inverse X dot minus A X.

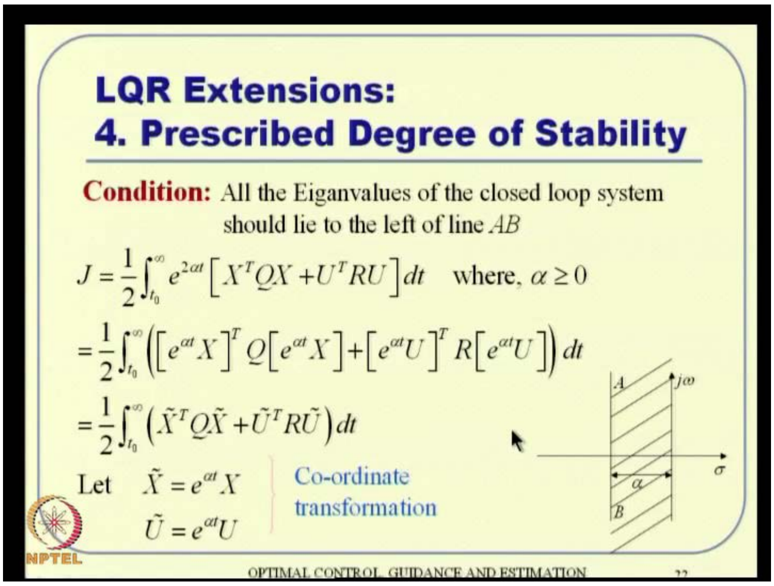
(Refer Slide Time: 35:33)



So, we will land up with U dot equal to all that, essentially tells out that U dot can be expressed in the form of minus K 1, so this is our K 1 X dot and K 2 X this is again small type of mistake here, the brackets will somewhat they will end here actually. So, this part is your K 1 and this part is actually is K 2 basically. So, we have this is K 1 times X dot minus of minus K 1 X dot, minus K 2 times X actually.

So, now, can integrate both sides in time and you can have this U equal to minus K 1 X minus K 2 times integral of X plus U naught, which an initial condition actually, now U naught can be obtain using a performance index without U dot term we know that, right. If you if you performance index does not contain any U dot term, then you have a different gain, U equal to minus K X; that means, for X naught U naught is minus K times X naught actually, so that is available. So, using this expression you can get an expression for U and what kind of this? This is a proportional term and this is an integral term, so essentially it leads to this PI control sort of ideas actually, some gain times of proportional terms and some gain times an integral term actually, so we land up with this PI control actually.

(Refer Slide Time: 37:02)



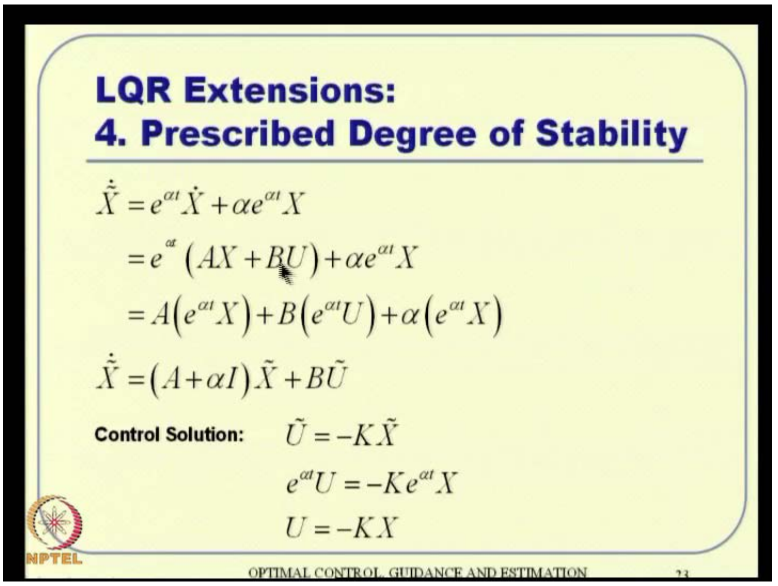
Alright. So, this is all about this rate of control and all that actually, now the last extension is this prescribed degree of stability and all that actually. Then remember, the very I mean some time back in this lecture, we talked about this control is asymptotically stabilizing lead to asymptotical stability behavior, but asymptotic stability behavior means all the Eigen values are bounded away from the J of omega axis, that imaginary axis which is ok, there is nothing wrong in that, but as far as robustness behavior is concern, that means, if the Eigen values are to close this imaginary axis, then the real Eigen values for the actual system there is a danger that it may go to the right hand side actually.

So, we want some design which will at least assure some sort of a alpha margin from the J omega axis; that means, all the Eigen values are suppose to lie left of left side of the A B line, which is separated away from the J omega axis by amount alpha actually. So, can we do that? And turns out to be, yes you can you can do that, by how do you do that? By taking a cost function in this in this form actually, e to the power 2 alpha T and I will also show why it should happen actually.

So, now imagine that this is possible, then what is what it can lead to? Just remember e to the power 2 alpha T is nothing but, e to the power alpha T into e to the power alpha T and e to the power alpha T is a scalar quantity. So, we can we can manipulate this algebra like this, so we will have e to the power alpha T times X times transpose times Q times all that plus e to the power alpha T times U, U times transpose R e all that actually. So, this J equal to all this and then e to power alpha T times X turns out to be X tilde we define it like that, X tilde e to the power alpha t times X and U tilde is e to the power alpha T times U, so this is nothing but, our coordinate transformation actually.

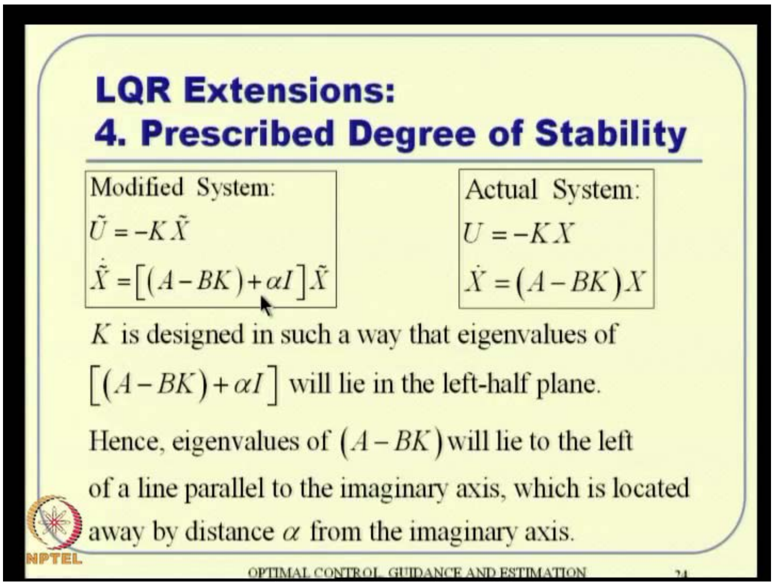
Alright. So, this coordinate using this coordinate transformation, this turns out to be a quadratic function in terms of X tilde and U tilde basically. Now, what about system dynamics? Again you can go back and think X tilde dot is what, X tilde is like that, so X tilde dot happens to be something like this, e to the alpha T into X dot plus the derivative of that, that is alpha times e to the power of alpha T into X actually and again X dot we could substitute back and then it turns out that, you can write A to the power A times e to the power alpha T into X, remember e to the power alpha T is a scalar quantity. So, you can take it right side, similarly you can write it B times this expression, e to power alpha T into U plus alpha times all these actually.

(Refer Slide Time: 39:09)



So, when you see this, these two terms you can combine now, you can combine and then remember X tilde is nothing but, your e to the power alpha T into X, so that can we combine actually. So, A plus alpha I coming from here times, this is X tilde plus B times U tilde basically. So, we have X tilde dot is nothing but, a plus alpha I into X X tilde plus B times U tilde basically, a by quadratic cost function now in terms of X tilde U tilde and you have a system dynamics in terms of X tilde U tilde is well actually. So, you can go ahead and find the control solution and control solution can be obtain using a Riccati matrix, considering this as A matrix, this as B and then Q and R, Q and R actually and then once you have a control solution in gain K and control is u tilde is nothing but, e to the power minus K into X tilde by U tilde is that and X tilde is that and these are all scalar quantity again and they will never go to 0 also, e to power alpha T never goes 0 only when T goes to kind if alpha is positive, t goes to in minus infinity happens to be 0 actually. I mean that never arises anyway, so we can cancel this out and have the same gain actually U equal to minus K X. The only difference is Riccati is matrix is completed based on this A rather then that I mean rather than the original A, all other things remains same, V remains V, Q remains Q, R remains R actually, just that the system dynamics get part of the little bit there actually. Now, somebody can argue how does it guarantee, this is what we wanted, now it turns out to be not that difficult.

(Refer Slide Time: 41:25)

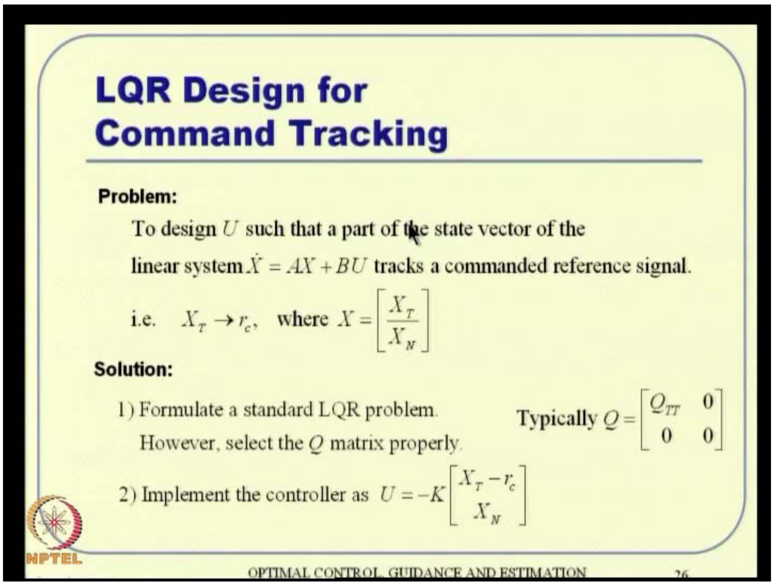


So, we this actual system and close loop system given something like this, modified system is something like this, with a modified close loop system is something like this (Refer Slide Time: 41:32).

So, LQR system it will guarantee asymptotic stability for this system, so that means, all the Eigen values of this matrix what you see are guarantee to be remain left top of the imaginary axis and if you carry out the algebra, it enough of this Eigen value and this Eigen value, that means, if all the Eigen values of this are I mean it actually have a real part, some real part whatever that is and this real part would if you compute Eigen values of the real part, you can see very well, I mean you can easily that the real part of the Eigen values will be pushed by alpha actually. So, this algebra you can do it yourself probably, the core point is the design assures that all the Eigen values of this matrix are left hand side, left top of a plane and hence, Eigen values of this only A minus B K will remain left side of this line A B, which is separated by alpha actually.

Alright. So, that is that is about it actually, now the next one which I want talk is how do you use LQR design for command tracking let us say. That means, so far we have been discussing about stability and all that actually, now suppose there is R of T which is comes as a command for a class of problem and how do you handle that.

(Refer Slide time: 42:57)



So, here you talk about a problem something like this, you have you have our aim is to design an U such that a part of the state vector of the linear system, that means, X dot equal to A X plus B U, but we have here a state vector, a part of that tracks a commanded reference signal R C. In other words if I if I split this X vector X T and X N, that means, track state and non track states and all that, then the objective here is X T should track R C basically.

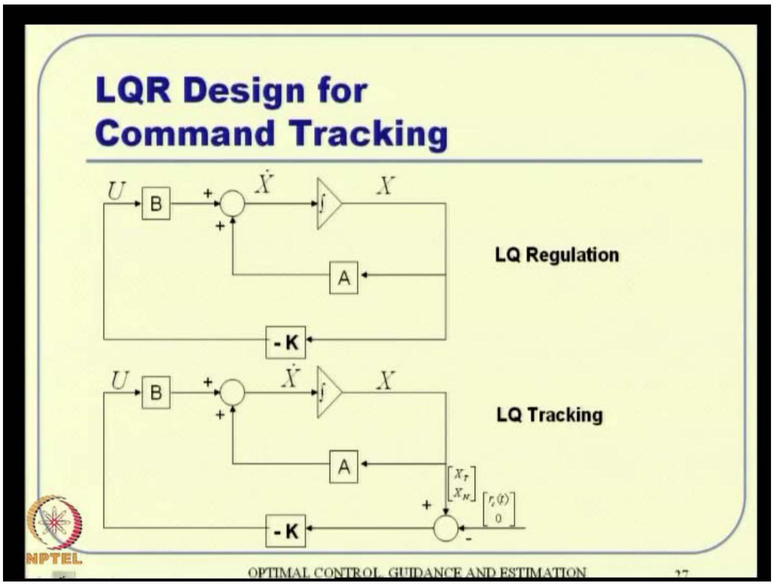
Alright. So, how do we go ahead and find a solution to this? Remember X N is non track state, we really do not worry where they go and all that, typically if X T goes to a tracking value, then X N does not cause too much of a problem with the problem formulation is good, so that kind of ideas there actually; so they remain bounded in an indirect sense, even though they are not directly (( )) and all that you get.

So, we do not worry about that actually. So, in that case what happens to this actually? So, we will we will see that, now what it turns out that just because you want this X T to track something, that means, X T minus R C should go to 0 we should have a term for Q T T, Q can be split into this partition matrix sort of thing, this is a partition matrix, first I mean this first entry Q T T involves X T. So, there is some term here and other things can be taken as 0 actually (Refer Slide Time: 44:31).

Now, it turns out that you can simply solve this problem using this Q and then implement the control like this, instead of minus K times X; that means, minus K times X T X N you will tell minus K times X minus R C, X T minus R C and X N; so that means, instead of simply feeding X T, X T we will feed X T minus R C in the first component actually.

So, this is how were you can see, how this evaporates and in a block diagram actually. So, this is a typical LQ regulation problem, you have this X dot equal to A X plus B U. So, A times X and X dot integrated over will give you X, so A times X plus B times U coming from here that is X dot, so that is how it is constructed and U is nothing but, minus K times X, so that is how it is.

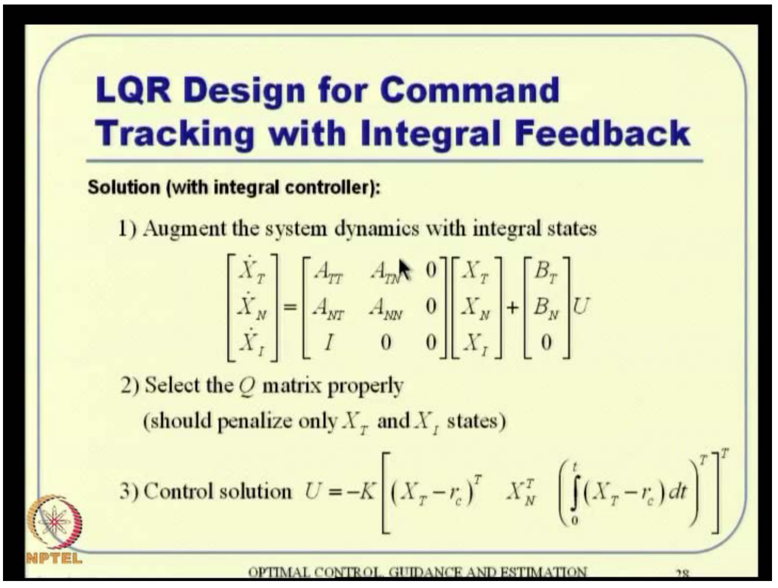
(Refer Slide Time: 44:57)



So, LQ tracking turns out to be something like that, R C of T it is spread into that and hence if you X T and X N if you put it here, then it turns out to be like in this junction what you see here, X T is not X C, I mean what you U should track R C basically. So, if it is R C there, that that happens to be this log diagram actually.

So, you have this addition, subtraction minus K is also there basically (( )) it should I means somebody can R C minus X T, but there is a minus term here, so we can talk about X T minus R C basically. Now this loop and this it is works well, I mean it works really well, it gives steady state errors and thing like that. So, we to nullify those effects it is also advisable, to substitute I mean to have some sort of integral feedback actually. If you do that then you can have this kind of a dynamics, where X I dot is nothing but, X T I times X T and nothing else actually.

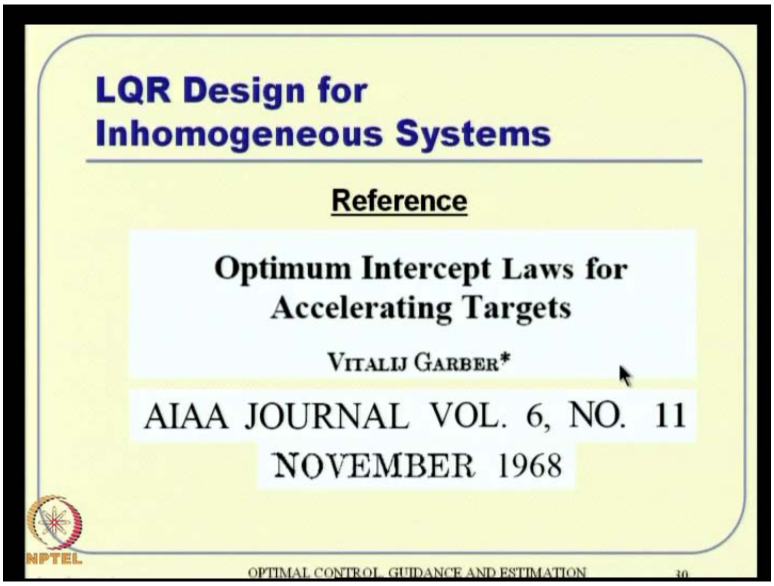
(Refer Slide Time: 46:18)



So, X I dot is X T means X I is nothing but, integral of X T basically. So, you can do that and select the Q matrix properly; that means, you have a one more terms here. So, Q T will be there and something like an integral of Q T T terms should also be their, after the 0 0 and all that actually, so you will ((do analyzation)) for both error as well as integral error actually.

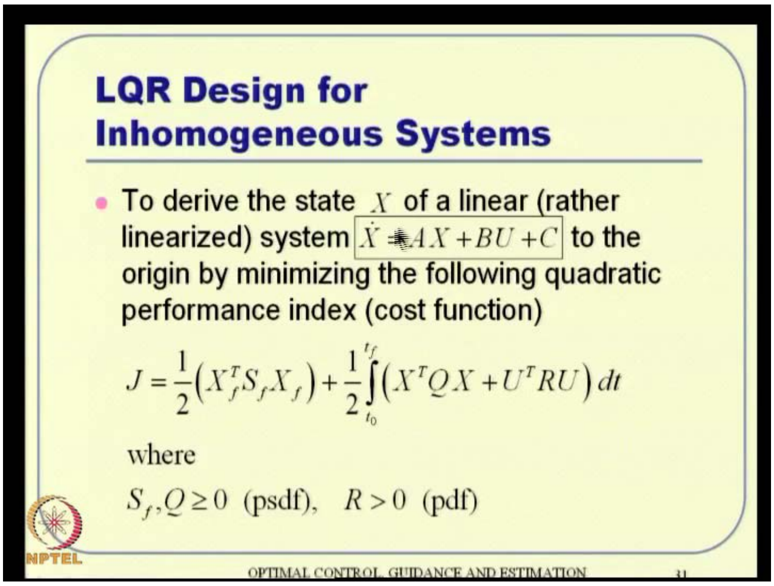
So, once you do that, you can operate the control based on this minus K times this T part X N is anyway free. So, you do not worry, but it is needed for feedback anyway, all straight feedback system, then it is U equal to this expression this is your X I, X I is nothing but, X X I (( )) I mean what it turns out is A X I is integral of X T, but you will have this X T minus R C integral basically. So, this will operate based on error integral rather than X I X T integral actually. So, error term and error integral term actually has to be there. So, this is how the control solution operates actually.

(Refer Slide Time: 48:02)



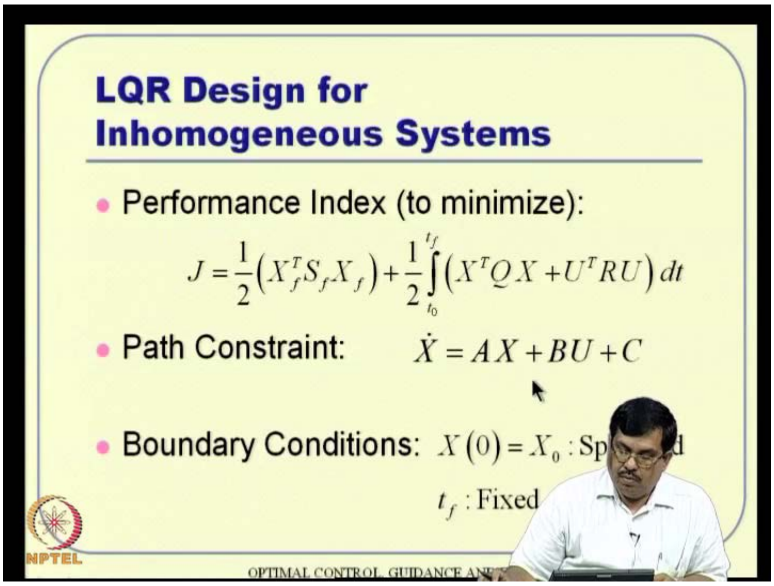
Now, lastly what, how about this design of this inhomogeneous systems? How do you handle those kinds of things actually? This is taken from very old paper by the way which is it appeared in 1968. So, AIAA is journal actually as this was an connection with a missile guidance problem also will not… will see that in a exclusive lecture probably how about I mean using LQR theory for various missile guidance and strategies and all that. So, this part of the application I will not talk about, but the formulation let us see actually generic.

(Refer Slide Time: 48:29)



So, this is whatever stuffing X dot equal to A X plus B U plus C now; that means, even though X and U are going to 0 that is still not good because X dot equal to C that there is a finite amount of rate for which the trajectory will start deviating actually. So, how do you assure that the state trajectory X of T should go to 0 in this case and state 0 also basically.

(Refer Slide Time: 49:10)



So, how do you how do you do that? The cost function is also same, I mean you have this standard cost function, where this standard conditions all good actually S f and Q has to be positive semi definite, R is to be positive definite, so those expressions remain for it actually.

Now, performance index to minimize turns out to be something like this, half of X f transpose S f plus all these and then the path constraint turns out to be like this, I mean we are going back to this summary of this formulation, we have this cost function and we have this path constraint, boundary conditions are again standard thing. So, we will go back to the necessary conditions optimality here and see what is going on actually. So, with this, with this we can we can put this you can we can see that there is our phi, this is our l and all sort of things. So, we can have a phi and we can have a Hamiltonian, this l plus lambda transpose f, now f is a X plus B U plus C that is the difference actually, state equation is also slight different A X plus B U plus C again then, costae equation remains to be same because once it take derivative all this C does not count actually, so that become 0. So, costae equation remains and similarly the optimal control equation also remain same, boundary condition also remain same, but the state equation will have a (( )) actually.

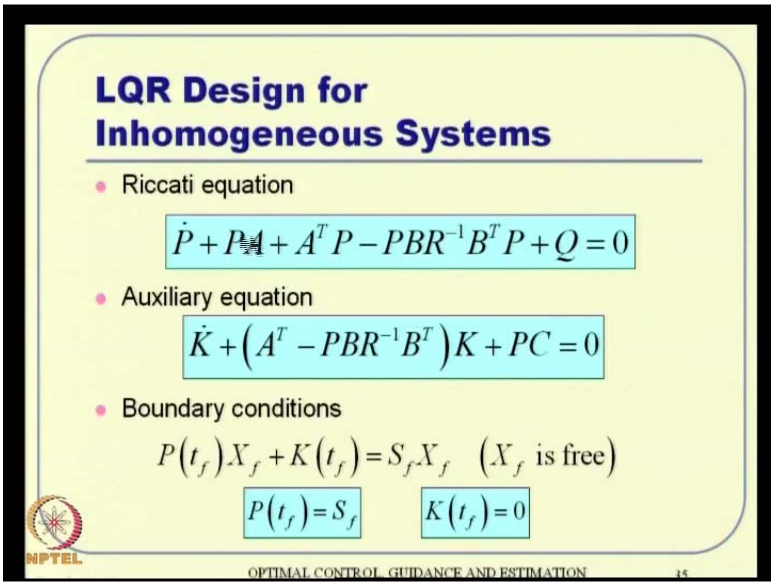
So, just to see that I mean just to see what is going on here, what we what we taken earlier, we just observe this and this happens to be homogeneous dynamics. So, we took this lambda is a linear function of x actually, we will also do very similar thing here, but with the addition of K of T, lambda of T is nothing but, P times X plus K basically. So, this has to be there, now lambda dot is nothing but, P dot times X times I means P times X dot plus k dot actually, so if it is like this and then x dot is nothing but, that X dot is A X plus B U plus C, so you put it back here, plus K dot.

And then you have this U, U is nothing but, that minus R inverse B transpose lambda, so put it back here, plus PC coming here and K dot and lambda dot is nothing but, Q X minus Q X minus A transpose lambda but, lambda is nothing but, that P times X plus K T this time, so minus Q X minus this a transpose not P X, but P X plus K because of that. So, do the similar very similar algebra, just we aware that wherever lambda is there you have to substitute P X plus K, not just P X actually.

Now, you do that and then ultimately turns out to be something like this and again this because it is two for all sort of combination of X something like that. So, let me see here little bit mistake here probably, that is a small (( )) again looks like which is probably minus of that actually, everything is taken to the right hand side actually. So, we have this K dot or this is plus I think, not minus everything is taken to the right hand side basically K dot is there. So, K dot then A transpose P coming minus A transpose P will go to positive side and thing like that, A transpose K is A, this is minus sign which will go to this right hand side will become positive and all that actually, so all this things are there.

So, now here what you what you say? You can tell that this coefficient as to be 0 and this coefficient also has to be 0; that means, we land up with some sort of a Riccati equation, which is coming from here, this is actually Riccati equation plus very close to Riccati equation some other differential equation and we call it as auxiliary equation actually.

(Refer Slide Time: 52:39)



So, these two equations as to be satisfied simultaneously, how about the boundary condition? We can again go back to the final condition like that and see that P of t f can take the same boundary condition as earlier and K of t f happens to be 0 is also commensurate with this and the because lambda t f is something like K of t f here, and lambda t f is nothing but, S of X f, so there is no bias term there. So, K of t K of t f happens to be 0 actually. So, in other words we have two sets of matrix difference, differential equations one in terms of P dot and other in terms of K dot actually. So, in the corresponding boundary condition, say one is P f t f and S f and one K f t f is 0 basically. So, using these two boundary conditions and these two differential equations you can propagate it backwards again; store the solutions then start using it actually.

So, when the control solution happens to be finally like this, minus R inverse B transpose lambda and lambda happens to be again you if you see this lambda is nothing but, P times X times P times X K times T and so this is ultimately like that and here you can actually observe some little bit crucial information that after X goes to 0, you still have a bias terms actually remember K is there actually.

Alright after X goes to 0 U equal to this term minus this term actually even if X goes to 0 because K is if K goes to 0 then, we have this U also goes to 0, but K need not go to 0. So, then you have this I mean bias control sort of thing actually and which is coming I mean, which is kind of comfortable also because you have a you have a constant forcing function X dot equal A X plus B U plus C. So, there is forcing function sort of thing is there, which is operating on the system to kind of make it deviated away from 0 sort of thing, so to nullify that we have a constant control also coming back to here. So, it turns out that these two quantities will nullify each other and make sure that X dot is 0, but that does not happened naturally remember that, we have we need a that bias control to make it happen. So, it is a force equilibrium sort of ideas actually then.

So, this is what is written here, this part of the control offsets the continuous disturbance actually, you can also think of these as a continuous disturbance acting on the system something like a cross wind or something constant cross wind and all that, if it is there in aircraft dynamics, then we can think of C, that constant bias term as acting on that actually it tries to deviate it to constant away and hence you have to apply your control system surface like (( )) reflex and thing like that to kind of not to allow the system to get influence by this constant deviation actually or constant disturbance force.

Alright. So, these are all possible to and we saw many things in this lecture, we saw stability behavior, we saw minimum cost function, then we saw extension of this LQR theory to various systems I mean various possibilities essentially and then also these I mean, inhomogeneous system design and how do you handle that and all that. So, with that I will stop this lecture, thank you.