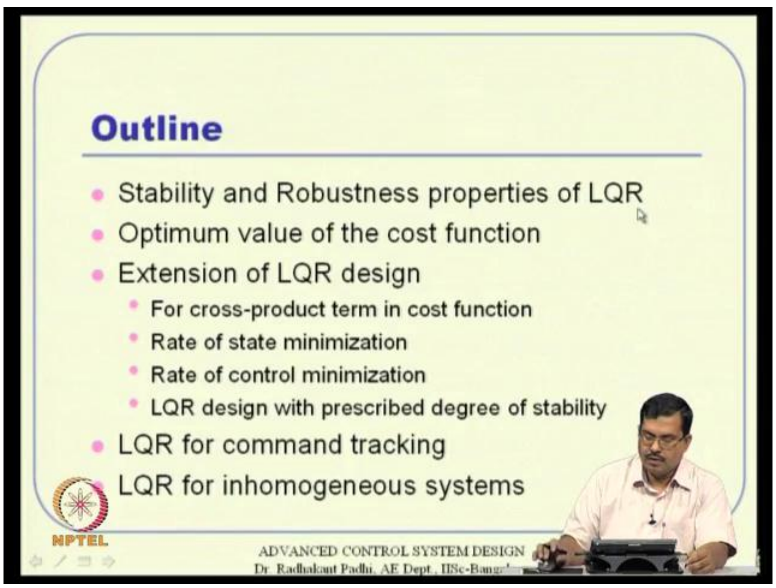
Advanced Control System Design
Assoc. Prof. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Science, Bangalore

Lecture No. # 28
Linear Quadratic Regulator (LQR) Design – II

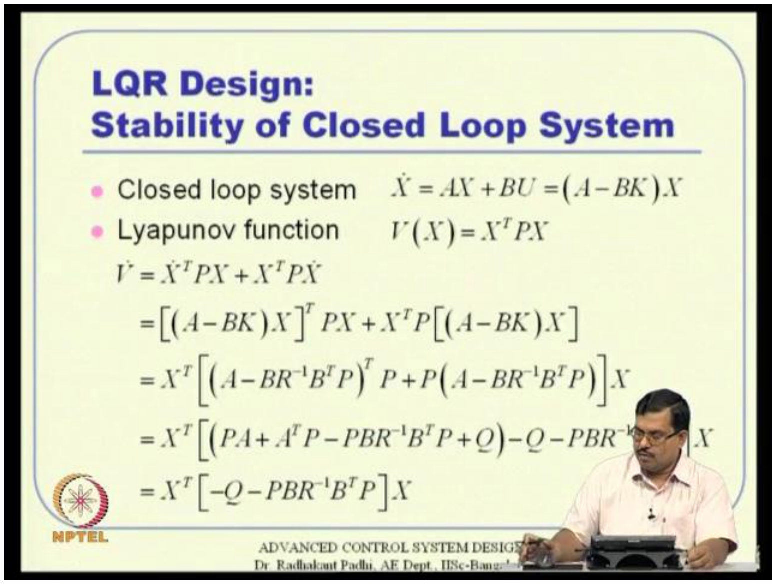
Hello everyone, I will continue our discussion on linear quadratic regulator theory. Previous lecture, we have derived I mean we have discussed the concept of LQR design as well as derived the necessary conditions. We followed up with required equation derivation as well as some examples actually. In that example inverted - inverted pendulum example especially we discussed about the stability behavior and thing like that. So, we will continue further with that generic descriptions and all that, there are many, many things that you can discuss under the LQR design. And we will cover some of the important things that comes up associated with design actually.

(Refer Slide Time: 00:54)



So, outline of this particular lecture will be for like this. So, we will first study the stability and robustness properties of LQR, and you can also see there is a optimum value of the cost function directly we can we can see in terms of LQR solution, I mean regarding equation solution. You can also see some extensions of LQR design, in case that the cross product term in the cost function, the rate of state minimization what happens? All sorts of things those are some of these things we can discuss. And we can also talk about LQR for command tracking as well as LQR for inhomogeneous systems, some of those things will be of practically usage for many different problems actually.

(Refer Slide Time: 01:43)



So, let us start discussing one by one and see how-how these things are in place. So, first stability and robustness properties and first we will deal with the stability part of it and here I assume that some. I mean your somewhat Rhopalic accustomed with this Lyapunov design with which we will going to study in a in our lecture series little while later, but all that it matters is when you have a Lyapunov homogenous system and a closed loop system turns out to be a homogenous system.

This is A X dot is A X plus B U got B equal to minus k x. So, closed loop system is X dot equal to sum at X into X which is nothing but a homogenous system and if you have a homogenous system like that the way to use Lyapunov theory is you select a Lyapunov function V of X, it needs to be positive definite function and as soon as we I mean as long as we select a positive definite P matrix this particular function is certainly a positive definite function.

So, we given a homogenous system, we select a positive definite function and somehow we make sure that the time derivative of this function is actually a negative definite function then the system is asymptotically stable actually that, that is that is what it tells and along with that if the function is actually radically unbounded. Here is everything will be satisfied that those conditions and certainly the function. I mean this result tends to be global actually; that means, it can be shown that the system is globally asymptotically stable. That is what you want to do here. So, we have this closed loop system dynamics and you have the Lyapunov function candidate like this, which is certainly radically unbounded we will see some of these concepts later in later classes actually then we can think of doing this time derivative now and v dot is the thing nothing but starting from here, it is is the this one and remember all these things we are talking about constant p matrix solution the infinite time problem actually.

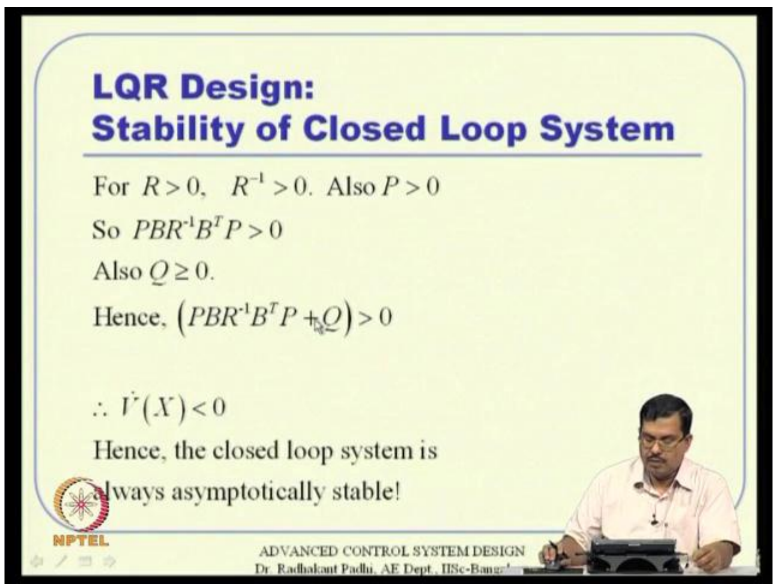
So, v dot turns out to be like this X dot X dot transposes times P X plus X Transpose times P X dot and x dot is nothing but that A minus B K into X. So, A minus B K into x here and that transpose is the times P X plus X Transpose p times X dot. So, X dot is the (( )). So, we put them together, and then tell this-this transpose can be expanded in a reverse sense. So, X Transpose turns out to be like that and A minus B K transpose turns out to be that way and k remember is nothing but r inverse B transpose P. So, that is that is the K.

So, wherever k appears we will take that way we will substitute that open this bracket. Now it will become a transpose here and the transpose will multiplication transpose will happen in a reverse sense. So, that is how, but remembers P is a symmetric matrix. So, P transpose E is equal to P. So, all sort of things we take advantage while simplifying this and then what you do is you do a plus and minus Q idents of straight Q matrix. Actually; the whole purpose of doing that is we have some ones of terms outside, but one part of the exercise runs out to be anything about algebraic Ricatti equation in the left hand side. So, this is certainly equal to 0 in the optimal path.

So, this is-this is gone as long as we apply this controller optimal controller minus k x or K is nothing but R inverse B transpose p and then such equation is satisfied so; that means, this entire equation what we see here is going to zero actually.

So, once that is entire thing is gone we are left out with only this last two terms and that is how what our V dot V dot turns out to be like that. So, if I take the negative sign out also. So, let me take the negative sign out. So, this is nothing but minus X Transpose X Transpose Q plus P B R inverse B transpose P into x. So, all that we need to show is this particular matrix turns out to be positive definite once you are there then negatives of that is certainly going to be negative definite function that you have done actually now question is this matrix what you see in bracket is it positive definite now one part is here that Q what you see is Q for Q matrix is positive semi definite by requirement actually.

(Refer Slide Time: 06:45)



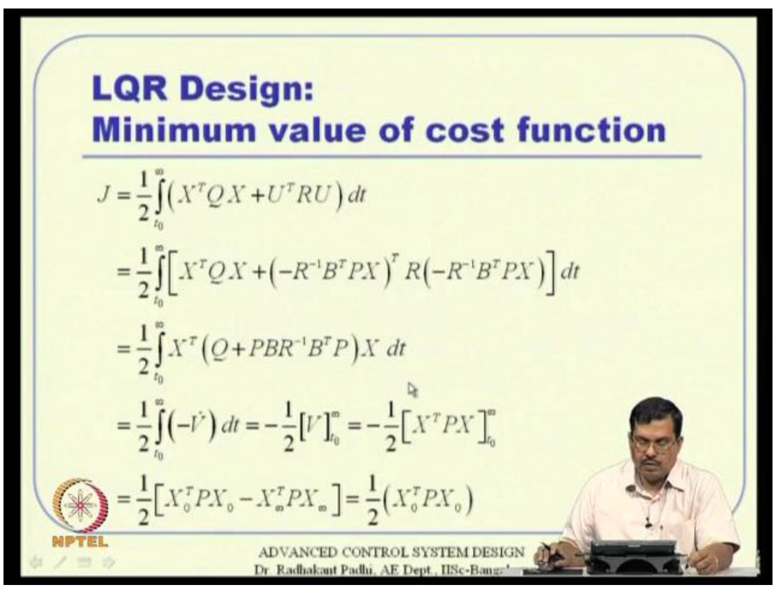
So, that part is positive semi definite anyway. So, all that we need to show is this part remains positive definite actually now because R is positive definite R inverse is also positive definite term why, because I mean Eigen values of any matrix are nothing inverse of a matrix a is one by Eigen values of the matrix actually. So, if r is positive definite its Eigen values are positive and hence one by those Eigen values are also positive and hence r inverse is positive definite matrix p matrix by selection is a positive definite matrix anyway. So, we are multiplying both sides by positive definite matrix and p times b transpose is like a quadratic term actually.

So, this entire process makes you this one positive definite matrix that was, is what is written here R is positive definite. So, R inverse is positive definite P is also positive definite and B and b transpose it is like a quadratic term out here and hence all the things that appears to be positive definite also that Q is positive semi definite and hence addition of the two is certainly a positive definite matrix. When this is a positive definite matrix V dot is that.

So, it is certainly going to be negative definite matrix, and hence this V dot is negative in the closed loop system is always asymptotically stable no matter no matter whatever values of p n i mean Q, and R you select this as long as you satisfy that Q is a positive semi definite matrix and r is a positive definite matrix then v dot is certainly going to be negative definite. And hence the closed loop system is always asymptotically stable, and as I told you before because this function is also radically unbounded this result is actually global.

So, the result does not in other words starting from any initial condition we are certainly guaranteed to go towards zero actually. So, that is the powerful result that you can rely on in LQR theory.

(Refer Slide Time: 07:54)



Now, what about the minimum value of the cost function? Can we compute depriory /a priori as a cross seeking of our result verification on a kind of it may serve as a verification tool. So, let us analyze this cross function and in. So, J is nothing but t zero to infinity this quadratic term and then u is nothing but minus R inverse B transpose P. So, we can substitute U that way and then the transpose there, and then you expand this transpose then R inverse transpose right will come here and then R times R inverse is identity in all that actually.

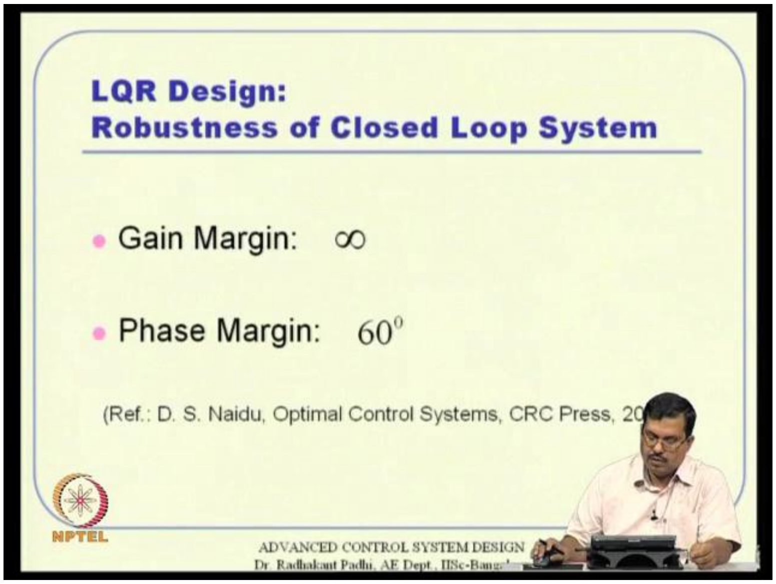
So, R is certainly a a positive definite square matrix roughly. So, I mean is positive definite square matrix. So, R inverse is also a square matrix ax symmetric matrix also. So, r inverse transpose and all in there will be nothing but r inverse itself actually. So, you do this algebra and then do not look at this particular thing X Transpose is all multiplying left and X Transpose x multiplies right. So, this turns out to be like this.

So, this particular thing remember this is what you see inside the bracket is nothing but negative of v dot that is what we just saw actually there is a negative of v dot. So, I can substitute this term as negative of V dot, and then I can I can see that it is nothing but a derivative in the derivative and integral-integral of a derivative term.

So, I can just simply take v and t zero to infinity now you can evaluate that and turns out to be like that actually. So, if I evaluate this term this turns out to be for negative sign is there remember that. So, infinity minus T 0 or T in a negative sign included it is initial condition minus final condition actually, now final condition sends X Y X X infinity is zero anyway therefore, that is what we are guaranteed to get using this asymptotic stability condition and all that. So, once that is there then the cos function of the LQR it turns out to be this way.

So, it is a half of X 0 transpose P times X 0 only. So, starting from any initial condition we can actually compute the final value of the cos function and certainly this can give us some sort of a verification tool whether our controlled design is proper or not in this simulation studies actually.

(Refer Slide Time: 10:11)



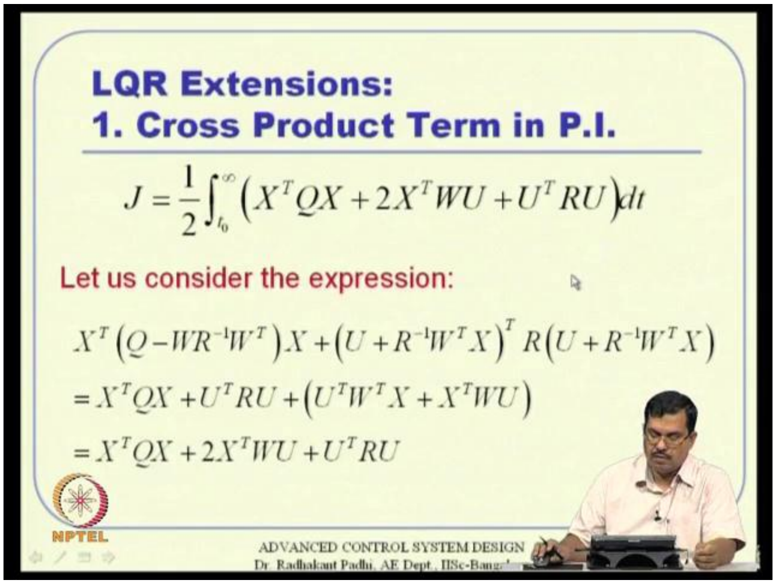
So, that is the value of a cross function actually now there are robustness concepts as well which tells us that under the ideal condition of just take feedback, and all your system will have a very good robustness margin with your gain margin turns out to be infinity and phase margin turns out to be 60 degrees actually more on that when it is available in this book we will not.

So, much into that what you can see these things that are available this guarantee is also there in addition to in addition to global asymptotic stability condition basically. So, that is how this design is a powerful design actually. Now extension of l q r, but before I proceed further also remember the moment there is this states are not feedback in exact sense; that means, you consider this.

So, called l q g design and all that where the state needs to be estimated through and observer or filter then this properties are lost actually that is the major drawback major hurdle actually and then that is why people talk about l q g slash l t r designer in that l q g design turns out to be LQR with kalman filter based state estimation and then l t r is something called loop transfer recovery and all that actually.

So, those are concepts that are available from the LQR. I mean the linear quadratic design details and all that those of you are interested can see some appropriate book actually one of that book is a certainly this book actually you can also see details of that here.

(Refer Slide Time: 11:50)



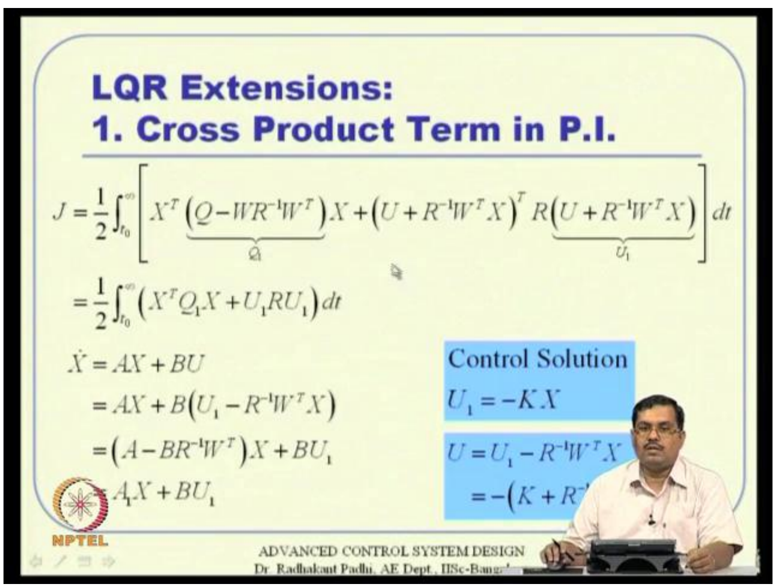
So, next study let us study about some extensions of LQR design we will study about one of three or I mean about three four extensions. So, let us see how that is proceeds first is the cross product term in the performance index. Before we did not talk about this particular term where X and u were not together either X Transpose Q X was there or u transpose R U was there now some applications you may come across where this cross product term also becomes a necessity in the performance index.

How do you do that now whole idea here is not to go back to this necessary conditions of optimality and then try to re derive the similar form of Riccati equation and all that we are not interested in doing that actually what we are interested is can we reformulate the problem in a slightly different way. So, that you can interpret the same problem as an LQR problem for a differential system and then utilizing that solution can you come back to this original solution actually I mean original problem solution that is the whole approach here.

So, to proceed in that to that without philosophy what is what we do here is we take this quadratic expression and we can see that this particular quadratic expression now this one we want to visualize as some form of quadratic expression actually. Now it turns out that if you consider this expression. It is a little complicated long end expression nevertheless. If you consider this and then try to simplify these transpose multiply and cancel out to corresponding terms things like that then this term is same as this term actually what you see here. So, instead of having this term out there I can visualize the same cos function in terms of that.

Now, what is beauty of that now this is this is my state when you guess value we decide I can visualize this term as some sort of a different Q matrix let us call that is Q one or something and I can visualize this as a different control.

(Refer Slide Time: 13:41)



So, with respect to that particular Q and that particular control vector, I now have some sort of a quadratic form and that is what I am doing here. So, one substitute that by k one. I consider that is Q 1 and I consider that is u one and hence it is something like Q one this X Transpose Q 1 x plus U 1 transpose R U actually R U 1. So, that is how I will visualize. So, this is certainly a quadratic of function and then the problem is not over because we had we had their original state equation this way it was not in terms of u one it was in terms of U;

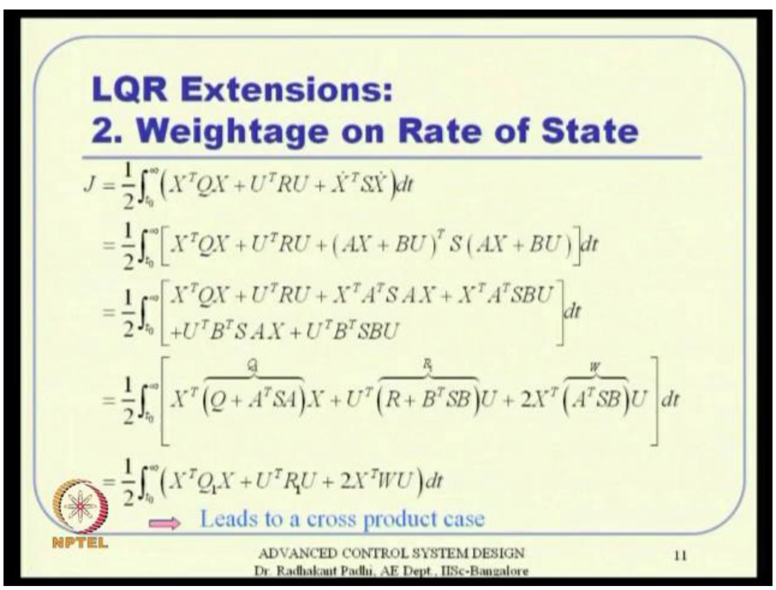
however, once U 1 is that you can you can solve for U from there and substitute from I mean substitute U as this expression U 1 minus that actually.

So, now we can combine X side of the story and then tell this particular thing that pops up is nothing but A 1 and that is my U 1. So, in other words you have a problem for X dot equal to A 1 X plus V U 1 and J is nothing but that. So, now, this linear equation in terms of X and U 1 and A 1 B matrix this linear equation with this quadratic of function you can have a regular LQR formulation actually.

So, we can get a solution from that particular problem like this u one equal to minus k x, but u one is not my ultimate objective U is our ultimate objective, but u is like this u one minus that. So, once u one is there minus K X I can substitute that and get my solution ready.

So, this is the real gain actually K plus r inverse w transpose is our is our real gain matrix that we are talking which comes by virtual gain matrix for a for an I mean for a related problem in the say setting of a in the LQR framework actually. So, we really do not have to go back to this entire idea of a necessary condition start deriving the Riccati equation again and think like that that is not a necessary here. So, that is how you handle cross product terms in the performance index.

(Refer Slide Time: 15:45)

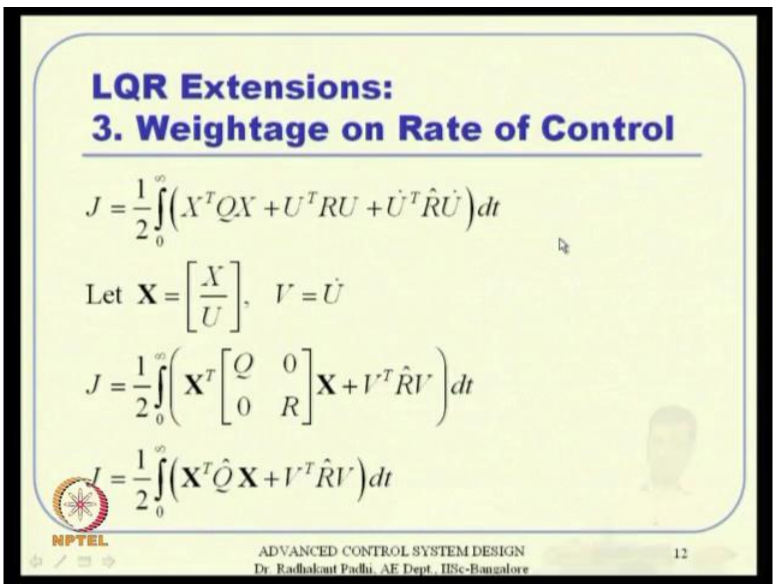


Now, let us see whether second condition weight age on rate of state sometimes. It is also essential to minimize the rate of state in addition to state itself one of that as I told in the inverted pendulum case you had to minimize theta and theta dot as well sometimes it is also advisable to minimize theta double dot as well actually.

So, for better and better guarantee of inverted (( )) ability actually in a way anyway. So, some problems it is not enough to get a minimize only state, but their derivatives as well and in those situations. I mean that is actually like separation of noise and other things are there. I mean that is that lead to that part actually. So, once you have a term something like this it is equivalent of telling that X dot is A X plus B U. So, I put X dot equal to like that way and X dot transpose is that transpose. So, expand this term try to simplify that to combine the terms and think like that we will essentially end up with this simplification actually out here which I can interpret these as a cos function that way which has a quadratic term in X A pure quadratic term in U as well, but there is a cross product case also.

So, we now with this particular formulation leads to a case of cross product terms build their now we know how to handle this cross product term already, and hence we can, we know how to solve this actually.

(Refer Slide Time: 17:14)

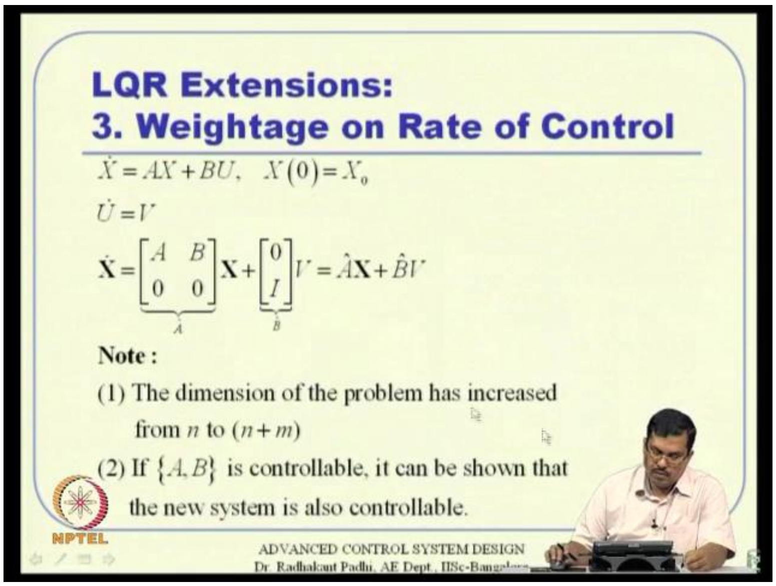


So, that is how we handle this rate of state now there is another thing which another extension which is tells what about rate of control if you have rate of control that is also another robustness-robustness formulation sort of thing where we need to minimize the rate of change of control variable as well actually. So, if you have that the solution is not very straight forward rather, but still it is doable possible in that sense what I do now let us, I will visualize this u dot as a virtual control variable v and consider u as a straight variable actually.

So, I have augmented state vector with my original state and control vector that is my augmented state vector and V is my virtual control is nothing but U dot once this definition is in place then the cross function J what we already had here can be visualized that way. So, these two terms I will combine through this matrix this term and U dot R hat times U is nothing but V transpose R hat times V basically.

So, we have a quadratic looking formulation like this in terms of augmented in terms of an augmented state vector X and U basically, now the we need to do further things because the original problem is not in terms of X dot this bold X dot, but normal X dot actually.

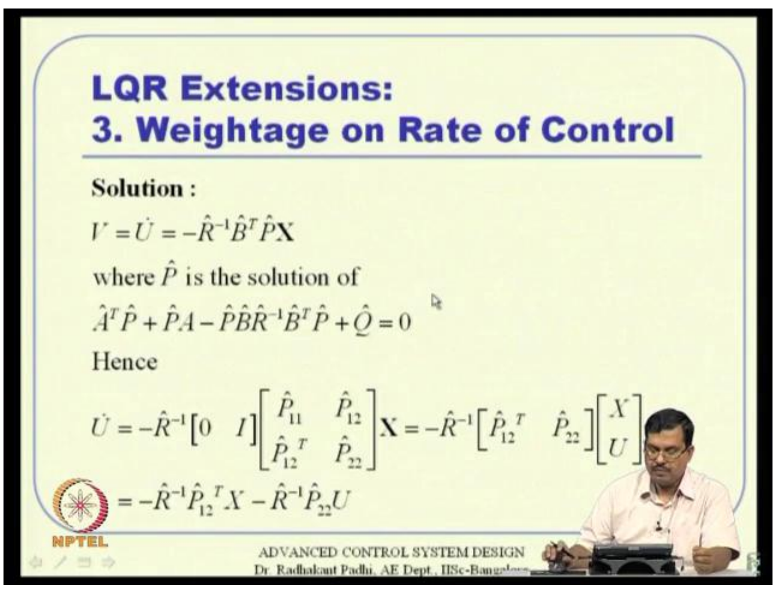
(Refer Slide Time: 18:26)



So, what about those things, so we will go back to that and if this is our system dynamics X dot is X plus B U now we have an augmented state vector where U dot is V. So, if I put that together that is my big I mean the bold X dot it turns out to be that way. So, we have got a hat and V hat matrix propping up from here.

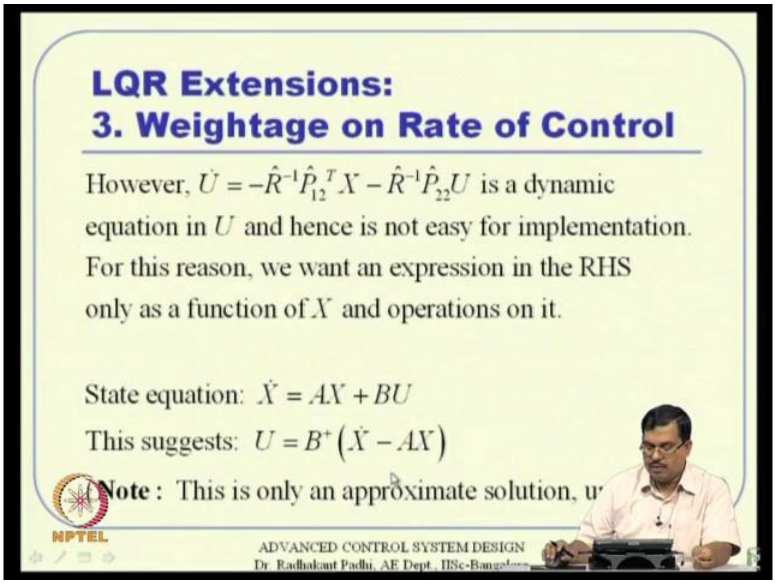
We can observe two things one thing is the dimension of the problem has increased from n plus n to n plus m the original state vector was n original control vector was m. So, now, the dimensions of the state big X actually or bold X turns out to be n plus m. So, the Riccati equation dimension or the Riccati matrix dimension is no more n by n, but it is n plus m by n plus m actually. So, that is the dimensional increase actually happens second thing it is more important to understand actually that if A B pair is controllable it turns out that a hat and b hat pair is also controllable and I encourage all of you to kind of justify that in your own matrix in your own algebra you verify that a you formulate a control ability matrix for the a hat and b hat pair and you also know that a b pair is controllable that derives a different control ability matrix for that and that is full range. So, you can show that the new control ability matrix is also full range actually. So, it turns out that if A B is controllable the new system dynamics that you see visualize here is also controllable actually.

(Refer Slide Time: 19:49)



Now, the solution turns out to be like this ultimately you get a solution of in term terms of V V as a function of bold X actually so; that means, the minus r at inverse that is B A transpose B A times X actually where P hat is a solution of that now this. So, what I am telling here once you get V is nothing about u dot u dot is that. So, expand all that and this one turns out to be like this. So, this is certainly a differential equation in terms of control variable which makes the controller dynamic actually which we do not want to operate it that way most of the times dynamic control have their own problems actually.

(Refer Slide Time: 20:40)

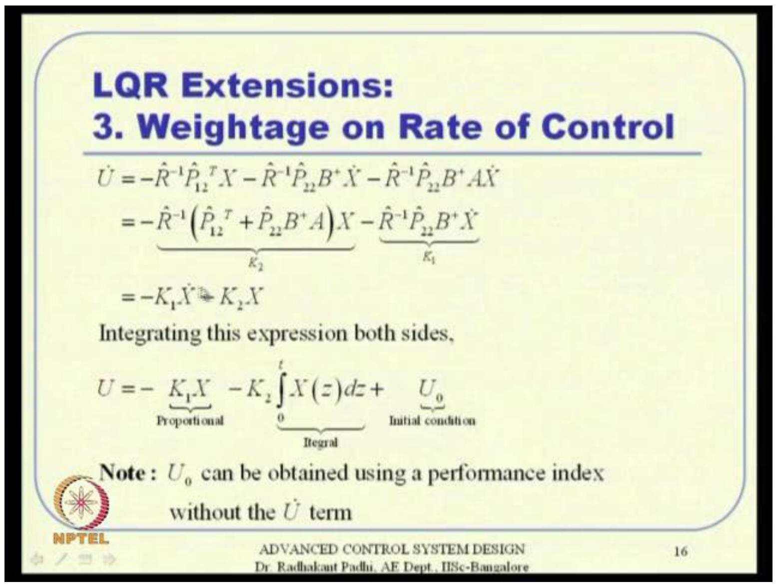


So, we want the control, we want the control as a starting function of X R its integral value that while I can make use of the past history of X actually that is what you intend to do here, but while intending do that then this U next to be kind of a eliminated from the right hand side now how do you do that we take advantage of this one the X dot is A X plus B U anyway we know.

So, U it turns out to be this solution where B plus is nothing but the pseudo inverse of e remember this B is not a square matrix in general. So, you cannot talk about b inverse here, and that is where this approximates solution and things like that comes in here actually.

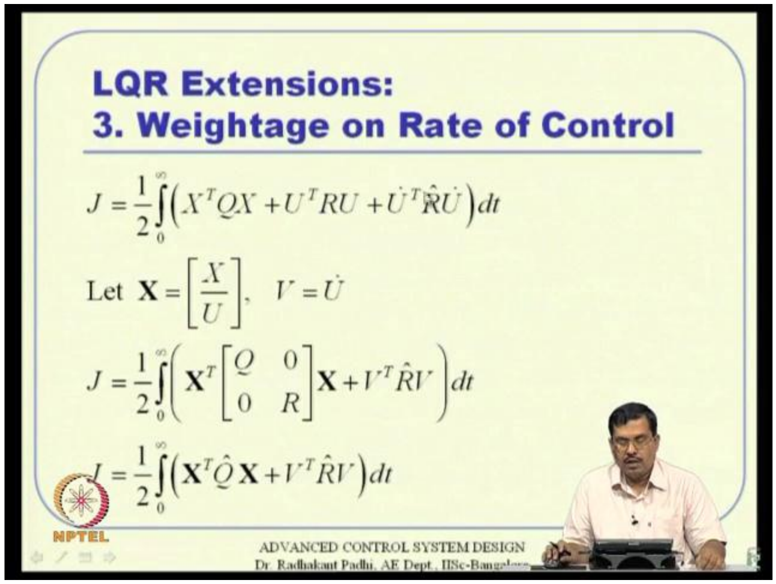
So, anyway we will not talk too much on that, but then what happens here is U I can solve it from here using this pseudo inverse solution right and this is ever constant problem the pseudo inverse solution is typically not a got solution that is knowing one personal comment out here actually.

(Refer Slide Time: 21:33)



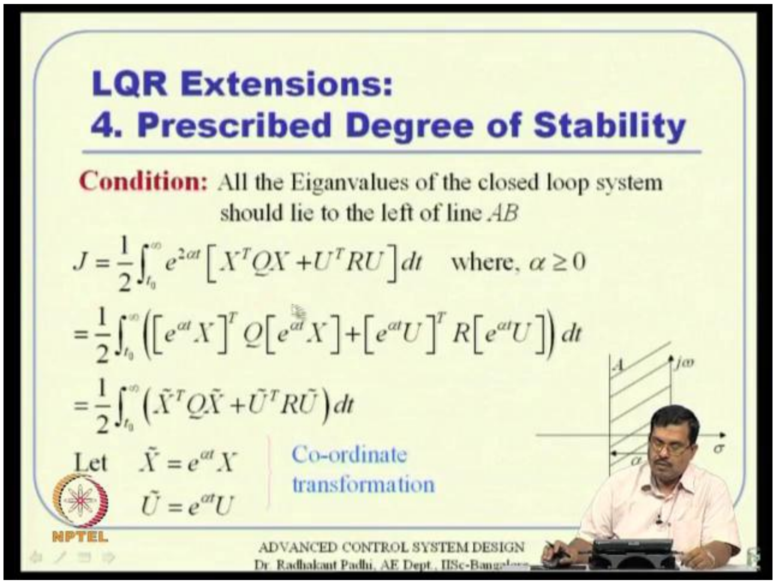
So, do not jump into this idea unless this is badly required (( )) anyway once I solve this u in that way then I can put it back U that U dot turns out to be that one and hence I can visualize it partly as a function of X and partly as a function of X dot basically all right anyway this bracket will terminate some terminates somewhat here. So, that is not there whatever you see as coefficients are actually the definitions of K 1 and K 2 all right. So, that is an obvious from the explanation as well. So, this is a u dot is nothing but minus K 1 times X dot minus K 2 times X. So, now, I can integrate the both sides and visualize that U is a proportional gain U is minus K 1 times X That is; that means, it will throw me a proportional gain and it is a proportional term actually you can say that way K 1 times X 2 get (( )) sort of thing there's a proportional feedback and there is a integral feedback and there is a initial condition for U that will also be required actually.

(Refer Slide Time: 22:42)



Now, how do you get this U 0 and my suggestion is you consider a performance index a parallel problem without these terms. So, that is a regular cross function anyway and there you get a solution for U that you consider as a U 0 actually. So, that is that is how I will recommend actually. So, that is how you get a solid control where the control is consisting of three part one is the proportional part and there is an integral part and there is an initial condition part actually anyway. So, that is, did least to the p i sort of controller and this U 0 can be obtained from a regular LQR formulation actually. So, that is our third thing.

(Refer Slide Time: 23:19)

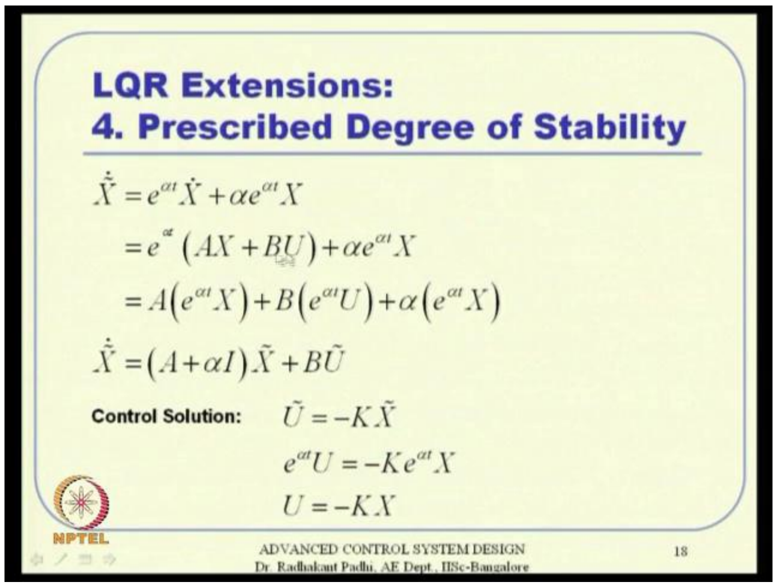


Now how do about fourth extension, now the here is a problem which discuss which talks about the asymptotic stability guarantee is all good, but it tells me that the Eigen values are only look at located at left outside actually. Now if that is the case then somehow of the Eigen values can. In fact, be located little too close to this imaginary axis actually it is too close to that then because of this parameter inaccuracy in other problems. So, other practical problem there is a chance that the some of the poles that are situated close to the imaginary axis can jump to the right hand side and then my system will go unstable actually.

So, to avoid that situation can I guarantee that all the Eigen values are situated left of an artificial line A B which is situated by a distance alpha from the imaginary axis if this is the A B line is situated by a distance alpha from the imaginary axis. So, that I have at least this much of margin and if at all there is a inaccuracy of parameters and all the poles located out here can only go to that side which is still in the left hand side of the J omega axis. So, this is actually some sort of a robustness consideration actually. Can we do that it turns out to be yes?

The answer is yes, we can do that by selecting a cross function this way let us say the performance index if you say like that way with an augmentation of e to the power two alpha t there then it turns out that we can do it how is it possible let us analyze that. So, this two alpha T is nothing but e to the power alpha t into E to the power alpha T. So, I put E 2 the power for t one term here one more term here. So, that term I visualize that way and similar thing I do it there define this X Tilde u tilde vectors that we have now. So, I can talk about this cost function is nothing about that actually that it is a quadratic term in terms of X Tilde and u tilde variable.

(Refer Slide Time: 25:22)

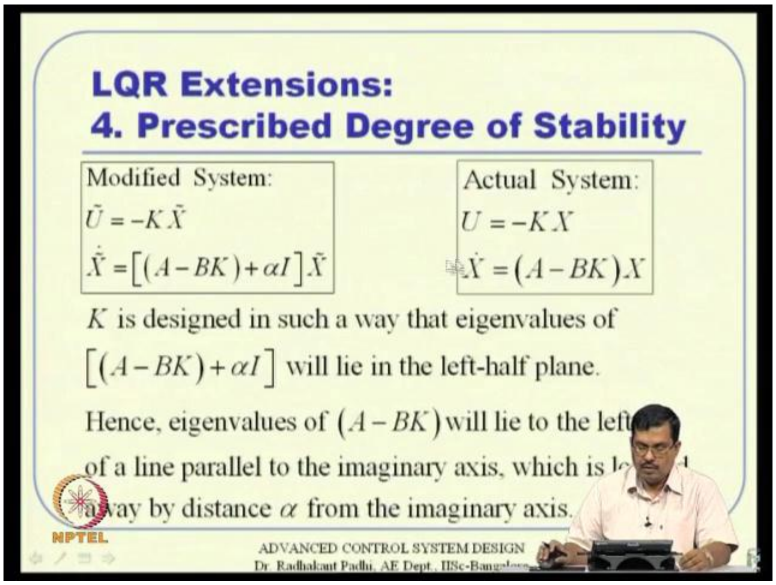


So, now can I proceed further because this cost this original equation also I need to transform there we write I need to analyze this X Tilde dot actually. So, if I go back to the definition X Tilde, and to make X Tilde dot out there that is the form and x dot is A X plus B U, so that is the form and wherever I can use of alpha T is a is a scalar quantity. So, I can take it anywhere I want. So, I can combine with X and U and then I can combine these terms associated with X Tilde this is X Tilde this is U tilde this is also X Tilde. So, I combine this first and last term. So, that is a plus alpha e times X Tilde plus B U tilde actually.

So, control solution from ultimately what you are getting from this solution control what you are getting this is a this is the linear system equation and that is the cross function with respect to that LQR formulation in can get a solution in this way u tilde is minus K times X Tilde. So, you tilde U to the power alpha T times U by definition X Tilde is also there.

So, E to the power alpha T alpha E to the power alpha t gets cancelled out and all that you are left out is this terms get cancelled out you left out with E equal to minus K X. So, the same gain that you are computing through ultimate formulation is nothing but the speed by gain that you want to operate your actual controller actually.

(Refer Slide Time: 26:36)

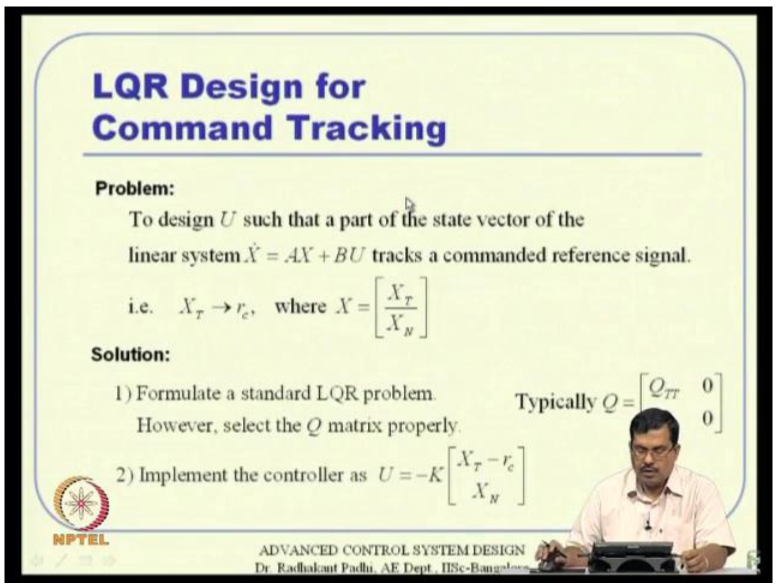


Now, how does that Eigen values R shifting I mean getting shifted from the imaginary exist to this boundary, boundary sort of things when we left of A B let us see how does that happen you can see that the modified system is like this U tilde is minus K x. So, X Tilde dot is something like this the actual system is like that.

So, what you see I mean the this entire formulation the way this gain matrix and other things are I mean the entire formulation what we are discussing here guarantee stability for this system dynamics asymptotic stability; that means, for this particular system in the closed loop Eigen values are situated left to the imaginary axis, but remember this is alpha e term which can only modify the diagonal elements of the characteristic equation if I have if I take a characteristic equation out for this I am only modifying the diagonal elements actually.

I mean diagonal elements and hence if you interpret the Eigen values of these I can take out those thing and then tell this alpha I all these Eigen values the real part of those values will be getting shifted by negative alpha actually I have to subtract to minus alpha from the solution of that really that is how we get this desirable condition that all the Eigen values were whatever all the Eigen values are getting shifted by alpha to the left actually that is what happens. So, if something is shifted something lies here that also gets shifted by alpha distance basically that is how you guarantee this prescribed degree of stability actually. So, that is how this is happening there.

(Refer Slide Time: 28:11)



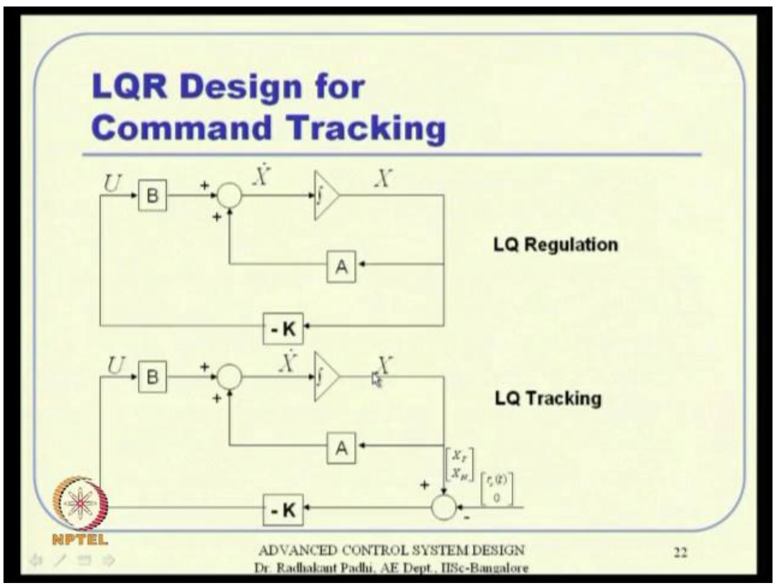
Now, let us study one more thing. So, that is LQR design for command tracking how do we do that now here is a problem which the problem is something like that we have we need to design u for command tracking issue.

Now in other words earlier we are mainly worried about X would go to zero only that is the regulatory problem and that is where the LQR system is been designed. I mean for that particular problem only actually now if you want to extend that to the tracking problem, how does it happen here we instead of it is considering a very generic problem and things like that. What you are telling here is let us say the X vector is divided into two parts where X T and x n where X T is the track to states that is what we want to track actually and X N we are typically not bothered about that.

So, much actually it can go somewhere we are not. So, much concerned about that for example, again in the if you go back to that inverted pendulum example last lecture what we discussed is theta needs to go to zero. So, let us say theta dot we did not put any condition on that actually and similar something similar you can visualize theta should go to some commanded function r c of t where x n i mean theta dot and all we do not bothered. So, much on that actually, so that is left out free that kind of problem we are talking here actually.

So, what you do here is we are not going to revisit the entire problem what you really need to do is I mean the whole idea here is you typically select a q matrix this way Q T T and q n n is nothing there here. So, q n n is left free and of diagonal you do not normally put anything that is anyway zero, but once you get a solution for this LQR problem you operate the controller that way. So, u equal to minus k x x is the X T and x n instead of making it simply X T and x n the first part of the thing you change it to X T minus r c and then make x n actually.

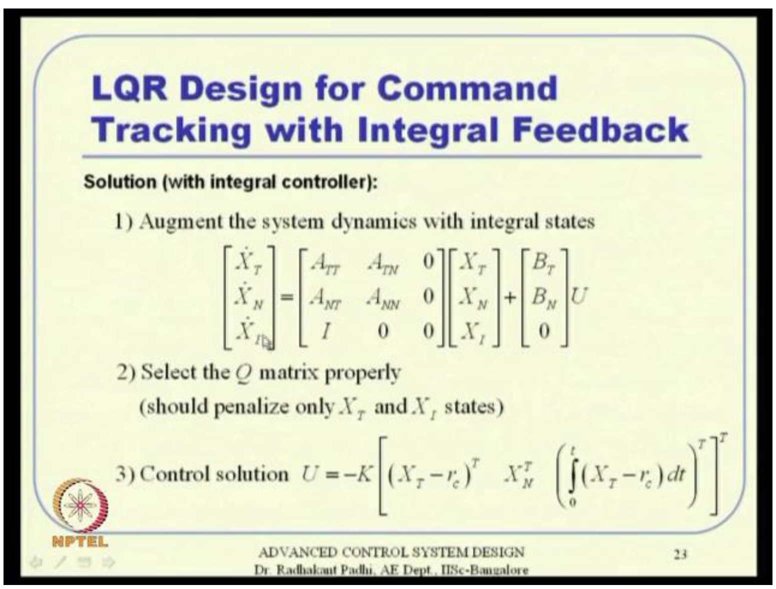
(Refer Slide Time: 30:09)



So, that is the if you operate it that way then X T will go to r c actually in pictorially visualize something like this. So, the LQR egulation problem operates this way; that means, u is going there to be in all states in the happens nicely, but here in a tracking problem what you are telling here is there is a summation block out here where X T minus r c is the something that will effect to the gain matrix actually .

So, if you do that what happens is X T minus r c will go to zero and hence X T will go to r c basically and that is the whole idea there under the assumption that r c dot is probably 0. I mean that is that is the assumption well that assumption is like quasi steady assumption at every time of every time gild you have see for every delta t duration. You assume that r c remains 0 I mean r c remain r c dot remains 0 that means, r c is a fixed quantity there in other words if r c changes to some value like let us say r c changes like that then you assume that this remains kind of this changes that way basically and that is what a normal assumption for this is r c and this is t actually there all right. So, this is a normal assumption in tracking problem. So, what you call is quasi steady assumption it is not really steady, but for every delta t it is assumed to be steady actually from now that condition it can be shown that it is X T will go to r c (( )), because X T minus r c will happen to be zero basically that will that will go to zero basically.

(Refer Slide Time: 31:46)



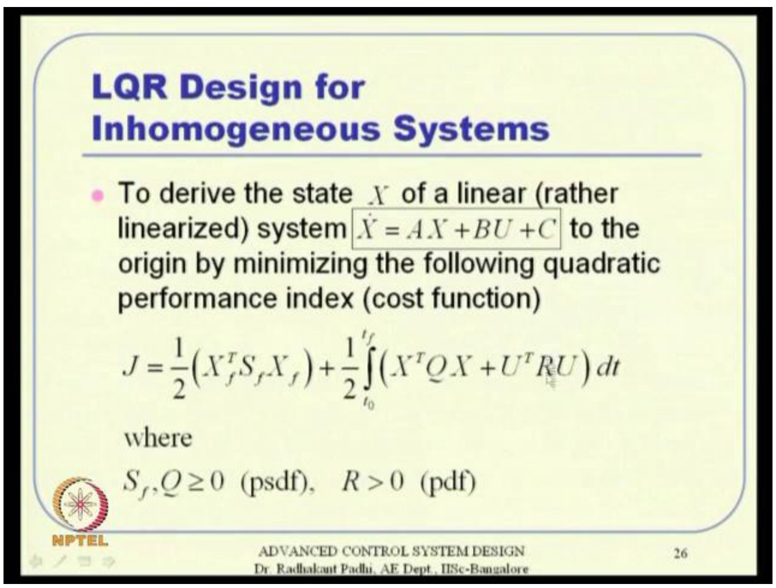
$$\begin{bmatrix} \dot{X}_T \\ \dot{X}_N \\ \dot{X}_I \end{bmatrix} = \begin{bmatrix} A_{TT} & A_{TN} & 0 \\ A_{NT} & A_{NN} & 0 \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} X_T \\ X_N \\ X_I \end{bmatrix} + \begin{bmatrix} B_T \\ B_N \\ 0 \end{bmatrix} U$$

So, now there is a problem of this particular thing when you talk about tracking thing typically people are interested in having a integral feedback in the loop and let us see how to do that. So, what you do here is we want to have an integral feedback in the loop. So, we augment the state vector in terms of an integral state actually X I dot is X T that is. So, X I dot is X T means this will become like a additional integral state basically and then the feedback solution will operate that way.

So, you have partly the proportional term the difference error and the integral, I mean the non track states and there is a integral of the tracking error that will come as a feedback to that actually, and correspondingly you have to select a gain matrix for the integral part as well you can leave the non track states to zero but this will become a like it is not really a three by three, but it is a blocked partition sort of a thing but the last block what you are talking should also contain some sort of a whitening matrix out there actually and that is what we are telling why select the appropriate q matrix actually, so if analyze both X T and X I states.

Now, the control solution should operate it that way all right. So, LQR next you see LQR design for inhomogeneous system. So, that is let us say what you what you are talking here is inhomogeneous system that is, I have taken this from here so; obviously, it is a I mean this idea is not very new all these ideas are actually not very new it is a longtime back people have studied these and all that, but still they are useful in variety of conditions actually.

(Refer Slide Time: 34:00)

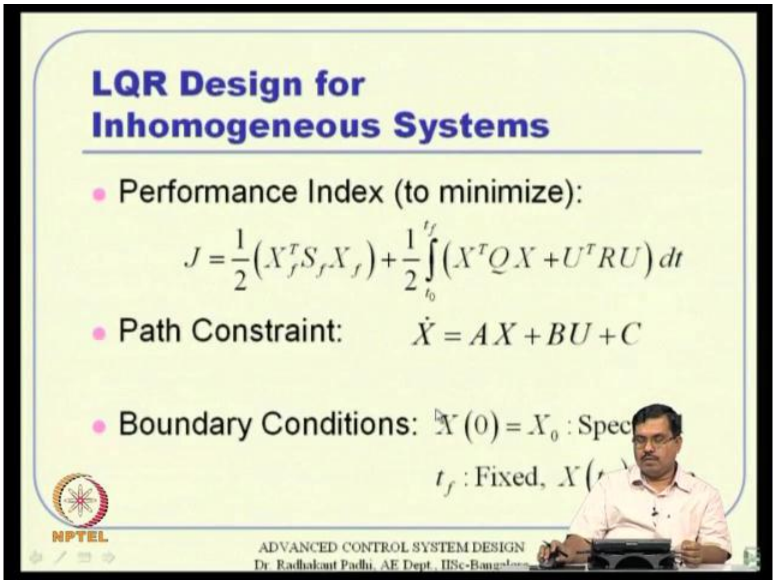


Now, this is also aerospace application those of you are interested in looking at how this particular formulation is useful for mutual guidance they can see this by (( )) actually and they can see this utility of this for mutual guidance problems now the core part of it is you have actually an inhomogeneous system.

What does it mean is, I mean what does it mean this means that instead of X plus B U only you have a plus c term also so; that means, X dot equal to A X plus B U plus c is there basically, now if you have this kind of a thing how do you handle this; that means, you can visualize this as a constant input for example, let us say there is a constant wind flowing while your aircraft is flowing. So, then how do you and that is actually a bias term which is trying push the aircraft all the time actually.

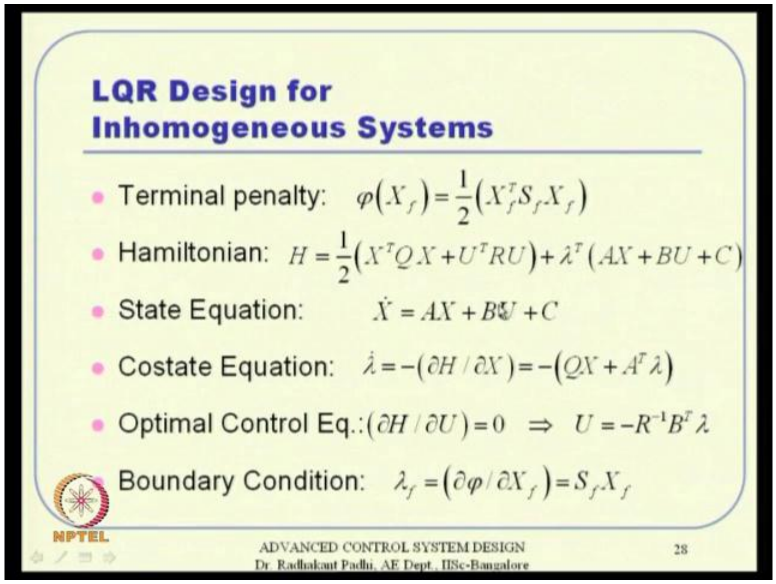
Now, let us see how do we handle that? So, whether you know direct extension for say that we will discuss here we have to go back to the Hamiltonian and necessary condition things like that actually anyway. So, let us study this is cross function is a same, I mean what you started with and similar conditions were all good actually.

(Refer Slide Time: 34:58)



So, now, we go back and this is our kind of performance index, but the state equation becomes something slightly different it is plus c terms out there Hamiltonian will also get different, because of that there will be a plus c term there and the corresponding equations will take its own form actually.

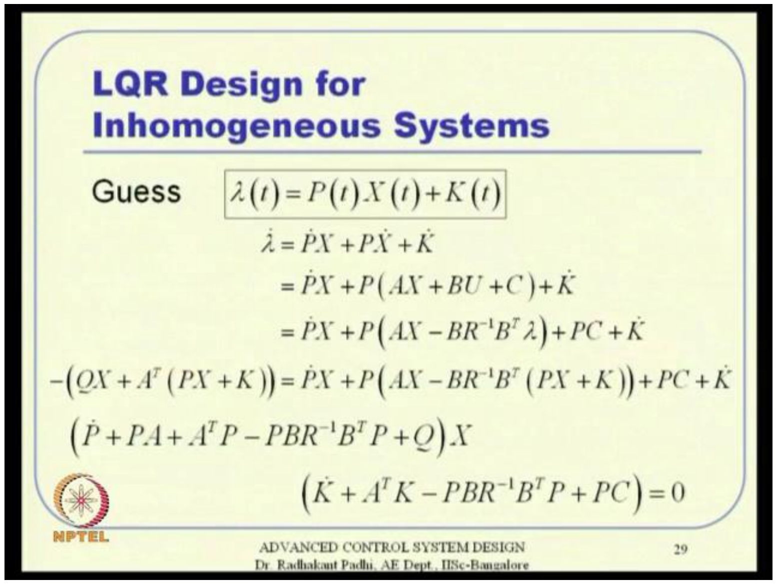
(Refer Slide Time: 35:11)



So, we can see that the any derivative what constant C is actually 0 and hence this lambda dot which is partial derivative with respect to X and then this is partial derivative with respect to U their optimal control equation will not change, because that term will not contribute to any additional term anyway.

So, cost ate equation will remain like that and optimal control equation will remain like that now how do you handle this kind of situation what the state equation; however, remains like that like this the whole idea here is because there is a constant bias here we will, let us try to kind of a visualize a constant bias control which is which in the essentially is to nullify that to bias actually. So, if you see that then lambda then ultimate control that I am looking for should have a bias term and hence lambda should also have a bias term actually.

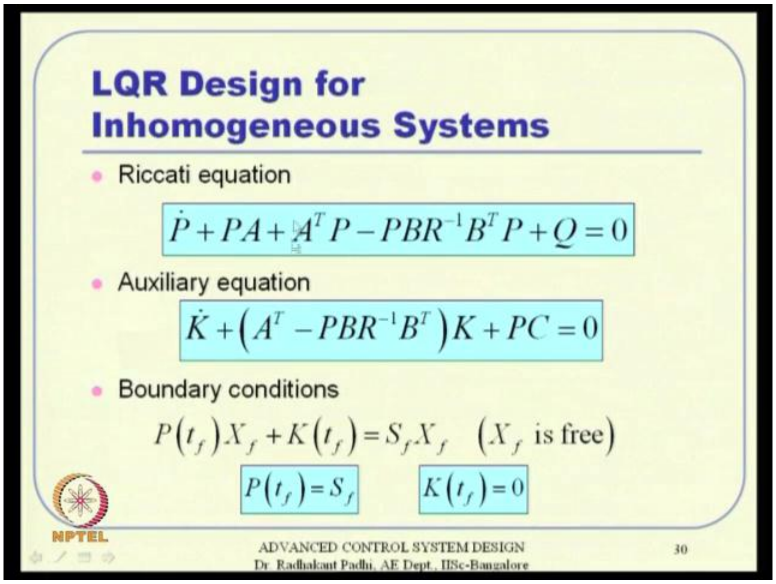
(Refer Slide Time: 36:10)



So, let us with that philosophy let us start with this thing and tell lambda is not only of a linear function now, but it is something like a state line equation sense is sort of idea where I have this k f t i addition to this term actually now let me let me see that whether I can handle this problem through this type of a lambda I mean solution or no actually.

Now, if I have this lambda form known to me then lambda dot turns out to be like that this is P dot times X plus P times X dot plus K dot which is a very obvious from there K dot part. I will not change that will remain. I will keep it, but x dot and all I will try changing remember straight in the core I mean cost ate and optimal control equation will not change too much actually only state equation will change. So, that is where x dot will become like that and then because U is R inverse b transpose lambda. I will put it that way now with a negative sign and then I will try to P C will come out from here and lambda is again P X plus k this time is not now it is not no more P times X only where ever lambda is there it is p P X plus k and wherever lambda dot is there that is Q minus Q X plus a transpose lambda basically minus Q X plus a transpose lambda where lambda is P X plus K this time one more time. So, P X plus K comes here P X k comes here and X dot equal to X plus B U plus c comes here actually. So, because of all these things together now I try to simplify attempt to simplify and then take all the terms to the left hand side and then what happens here is this similar form of ricatti equation pops off with a I mean which is multiplied with X plus there is a plus term missing out here probably plus this term has to be zero actually.

(Refer Slide Time: 38:15)

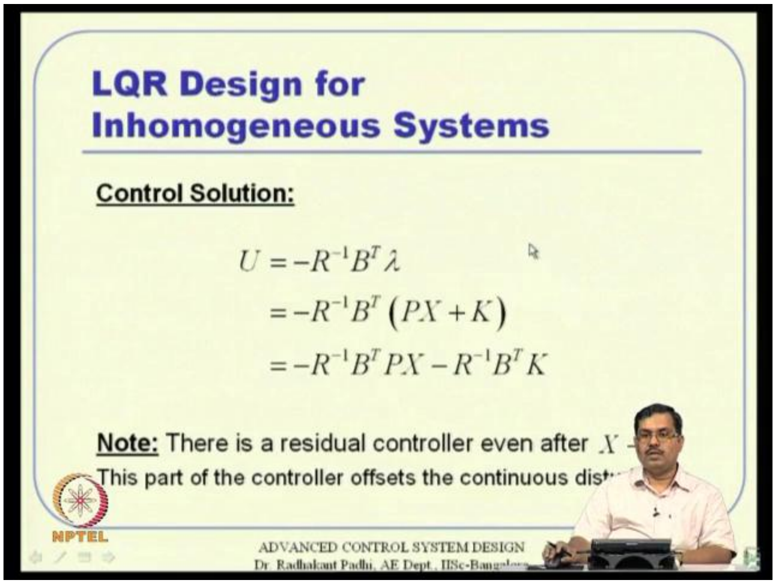


Now, what happens if I really want to make this in equal to 0 then the way the way to do that is then let us say the coefficients will go to 0 and hence I will get two equations not not one, but two equations I will get one is the Riccati equation standard Riccati equation another equation is something like auxiliary equation actually so; that means, K dot equal to a transpose minus P B R inverse K the whatever it turns out from here that is an auxiliary differential equation. So, if I really need to know P of T and K of T differential equations are ready, but I need to talk about boundary condition as well and the same boundary condition all stood here basically here the outside term was not changed outside term remains same actually that is why phi remain same and lambda f remains same. So, let me satisfy this term this equation if I put the boundary condition out here and try to equate the coefficients what it turns out is P of t f is equal to S f and K of t f is 0.

So, if with K f K of t f is 0 and this differential equation I can propagate back I mean in integrate this auxiliary equation backward from t f to t 0 and with this boundary condition and this difference equation. I can observe, I can integrate the equation backward actually again. So, I will have the solutions for both P and K ready that can with this integration process can again be done backward and make sure that you store these things this solutions and all on board. So, you have this solution ready offline and then you can use it online actually.

So, you are using this two differential equations and these two boundary conditions you can actually solve this equations and then you are ready to apply the controller because lambda of t after that is like this and U U is equal to minus R (( )) B transpose lambda.

(Refer Slide Time: 39:55)



So, lambda is all like that this is known to us that is what I have doing here, control solution turns out to be like that, and the control solution if you see carefully. What it turns out is this U has a proportional term like an LQR controller before, but it has a bias term also basically. So, what it turns out is that the residual controller even after X goes to 0, there is this is the term. So, there is a residual controller which will naturally come after X goes to 0 ,and this natural this residual controller part actually tries to upset the continuous disturbance e, but that is the meaning actually.

So, that is the way to handle this. So, what you have seen in this lecture is extensions of LQR control design in various forms as well as how do we extend it for let us say command tracking and inhomogeneous problem problems. We also solved stability and robustness behaviors, so I suggest that you can select some proper book on l q design and study more actually. So, with that I will stop this lecture them, thank you.