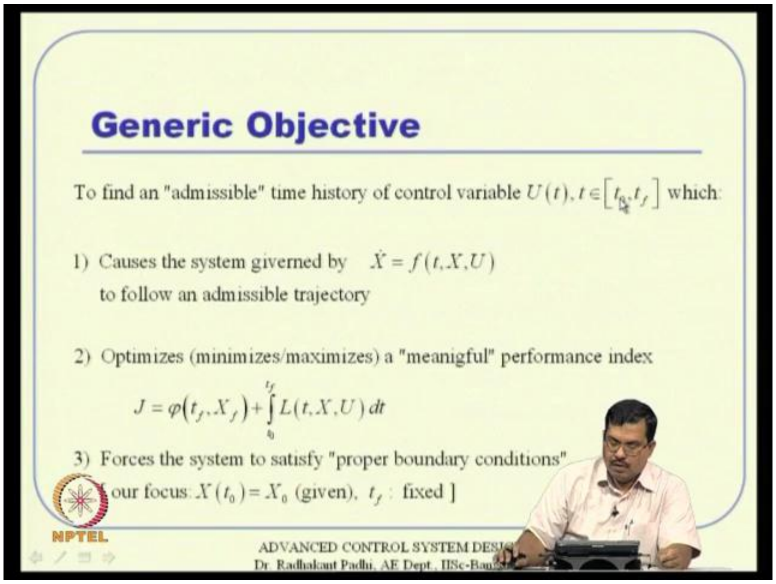
Advanced Control System Design
Assoc. Prof. Dr. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Science, Bangalore

Lecture No. # 27
Linear Quadratic Regulator (LQR) Design-I

Hello everyone, lets continue our lecture series on optimal control theory, and today this lecture 27. I will discuss about what is called as linear quadratic regulator design; however, popularly known as LQR design. We will continue this concept in next lecture as well several things to talk about. Let see the fundamental things and then associated examples, and all in this particular lecture.

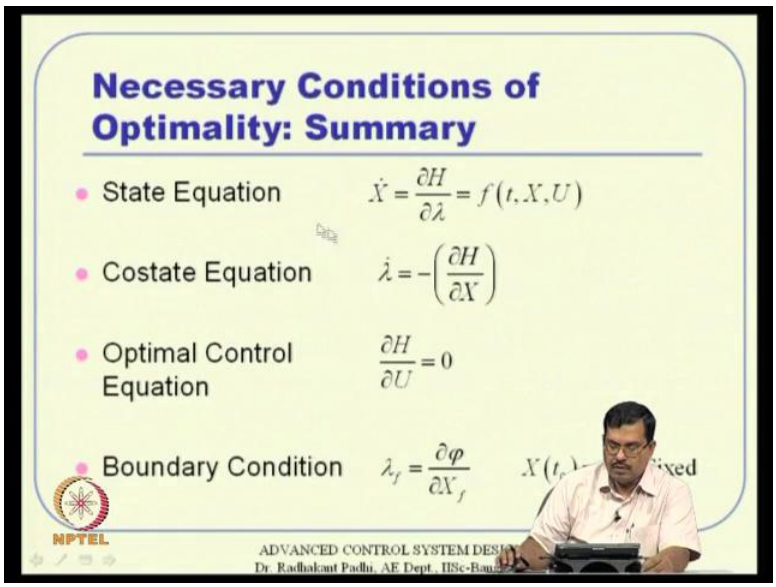
(Refer Slide Time: 00:43)



So, generic objective as far as any optimal control problem is concerned, we have studied this particular class of problems. So, very large class of problems rather which tells that the aim is to find admissible time history of the control variable U of t from t 0 to t f which does these three things, which causes the system governed by this non-linear system dynamics X is dot equal to f of t X U to follow an admissible trajectory. And it optimizes all that means, minimization maximization certain meaningful performance index that way.

And it also forces the system to satisfy proper boundary conditions, and whatever we discuss in this this course we discussed about this particular class of cost functions, which is rather fairly generic talks about many class of systems you can formulate in this kind of cost function. As well as we also assume that the initial time and initial condition of states are all fixed it is given, t f is also fixed only the X f is free. So, those are the class of problems that we are interested in this particular course actually

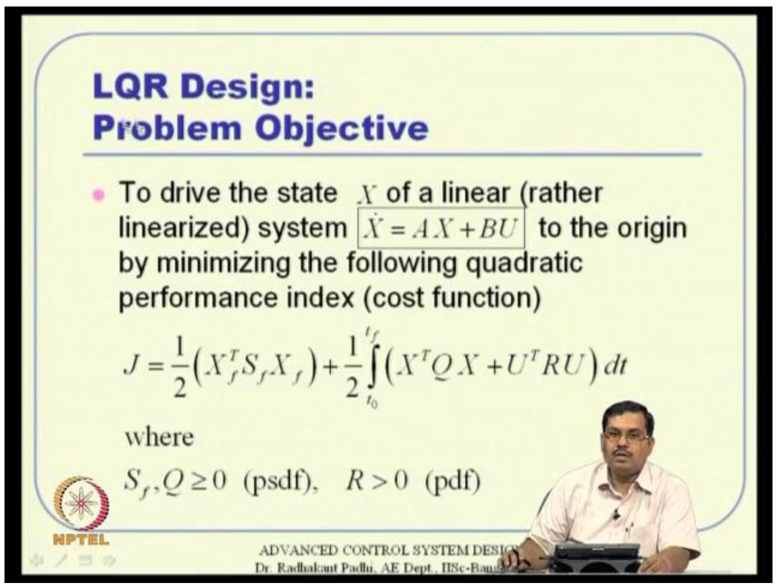
(Refer Slide Time: 01:52)



And associated with these four problems, we also went ahead and derived these necessary conditions. So, essentially it talks about three path conditions and one set of boundary conditions actually. So, path constraints one is I mean two of them are dynamic equations one is state equation, and there is cost ate equation and associated with that there is optimal control equation which needs to be satisfied at all point of time. So, that is del H by del U equal to 0.

So, these are the state equation being and cost ate equation both are dynamic equation what the problem here is we have split boundary conditions half of the conditions are known at initial time, whereas half of the conditions are known only at final time that leads to this problem of two point boundary value problems, which are difficult to solve I mean in a close as well as it is also difficult to solve typically. It is possible, but it is computational intensive to solve in a numerical procedure as well nevertheless, we have seen this two three numerical procedures last class which also gave us a feeling how do we really try to solve it. In case we want to solve these problems offline solution. Basically I mean if you are interested in offline solution you can pick off some of those algorithms and try to solve these problems actually.

(Refer Slide Time: 03:16)

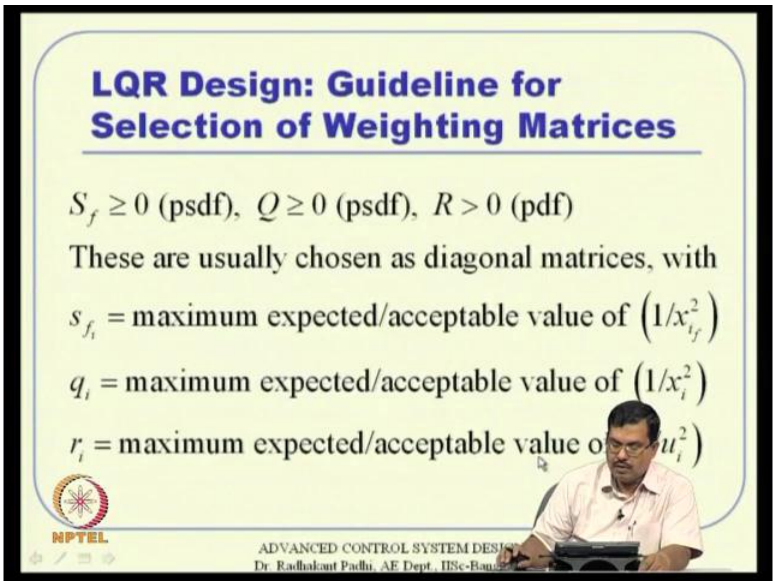


But the question comes is there a special class of systems problems for which we really do not need those computational intensive procedures iterative procedures to get a solution, and one of the class reference to be this LQR design which is linear quadratic design where you are interested in linear state equations x dot equal to AX plus BU, and quadratic performance index. So, if you have a linear system or rather linear zed system and; obviously, when you have a linear zed system this x is delta x and u is delta u, you have seen before in the linearization lecture. And when you have this deviation perturbation sort of ideas and perturbation dynamics associated with that then; obviously, the aim is to kill that deviation or the two kind of nullify perturbations actually.

So that means, you are interested to derive the drive the state x two zero whatever x is there my aim is to drive this state x to 0 actually. So, if you drive that I mean, if the aim is to drive the state x of the linear system or rather the linear zed system which is like this to the origin;

that means, x would go to zero. Then how do you do that you can do that by minimizing the following quadratic performance index or cost function actually. So, with these if you see each of the terms are quadratic in nature there is a final penalty which is X f transpose S f X f and there are path penalty which is X Transpose Q X plus U Transpose R U if you have this quadratic function and in the I mean on the path and this quadratic function at the T at T equal to T f then the aim is to minimize this state actually. So, x would develop towards zero I mean that is our objective actually provided these conditions are met actually S f and q are positive and semi definite matrices and r is typically positive definite matrices those are necessary conditions that will pop up later also basically. So, we will see that a little later.

(Refer Slide Time: 05:03)



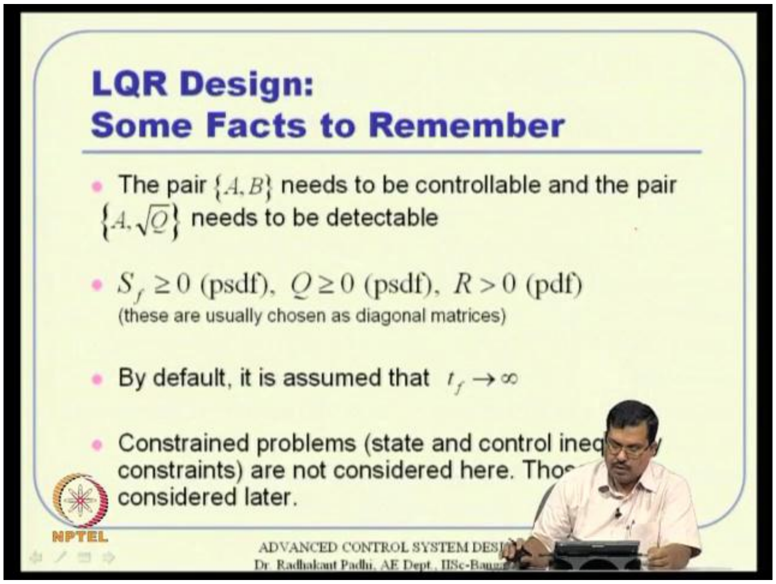
Now, as long as these conditions are met and there are several little bit other conditions that we will see later these-these kind of conditions then things will be in place actually. Now the question is how this does the form of the cost function is fixed, but how do you select these matrices S f q and r.

So, if I mean these are the only conditions there, but remember these are matrices these are not even scalars actually. So, selecting matrices which needs to be positive definite and positive semi definite is also not a trivial task actually. So, there are, but there are some guidelines. So, one of the guidelines is is like that which is popularly called as Bryson's rule actually, but anyway. So, the whole idea here is if you have three different terms they are computing against each other then somehow as an intuitive case you do not want to have kind of referential bias to anybody. So, try to kind of normalize each of the terms.

So that they have each of the term has a fair chance to contribute actually that is that is the whole idea there. So, if I select this S f q and r as diagonal matrices and then each of the diagonal element I select something like that suppose S f i is maximum expectable maximum expected or acceptable value of this one over x i f square then what happens is this particular thing gets divided by norm of x in a way basically. So, it is X Transpose X f transpose S f divided by norm of X f basically I mean that way it is roughly type of norm not really norm of x, but norm of maximum value of x basically that way.

So, it is similar terms upon this similar term upon this. So, you get some sort of normalization by selecting this way. So, S f i you select maximum expected value of the one by S f i x i f square think like that actually and I mean the whole idea here is S f I let say if you take q then what you one by say q let us say draw this probably. So, q i i put it like 1 1 by q i i mean this is q i basically like q 1 q 2 q 3 in the diagonal everywhere else it is 0 0, and then q i i want to select something like one by x i max square that way I will select. So, if I select if I if I see that this is like half q q i this particular diagonal element is something like half q i x i square. So, then if you I am multiplying this by half x i square basically. So, if I do that then it turns out that this particular term is nothing but half x i square by divided by x i max square actually. So, that way it is getting kind of normalized there.

(Refer Slide Time: 07:54)



Similar thing you do for q and r actually, now there are some facts to remember or some conditions to remember this particular design or any other design this pair a b needs to be controllable that is that is the first requirement actually for any control any control design for linear system the pair a b needs to be controllable, and there is also associated condition that a square root of q needs to be detectable as well actually that is something related to regarding matrix solution, and all that we do not discuss.

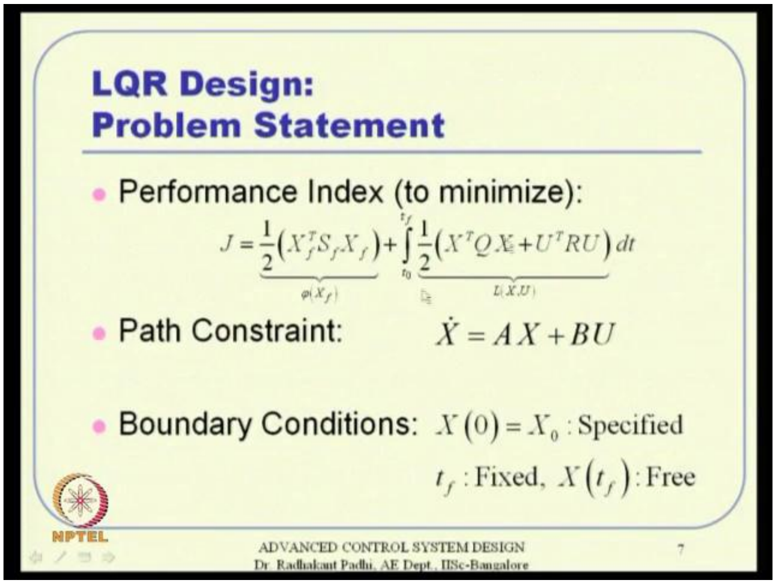
So, much on that, but a b if the pair a b needs to be controllable is very apprehend it has to be needs it needs to be satisfied for any control design and LQR is not an exception actually then this conditions we already discussed. So, these matrices are usually chosen as diagonal matrices and S f and q needs to be positive semi definite whereas, r needs to be positive definite and then by default suppose nobody tells us what is the value of this t f then the value of t f by default is assume to be infinity; that means, you are talking about a large final time problem actually. So, there is a necessity there is a necessity. I mean we will see that that assumption t f goes to infinity simplifies the for development quite a lot later and most of the time LQR is used in the infinite of anyway we talk about stabilizing problems and all where you inbuilt assumption is t f goes to infinity.

So, if nobody tells that it is really a finite time problem I mean this is also finite time. Where t f is actually fixed at infinity, but if it is not told that t f value is exactly that much then by default we need to assume that t f is infinity actually we will see that in a second why that is beautiful both it needs lot of simplification actually, and also remember that constrained problems that state and control inequality constraints problems are not considered as part of this lecture probably those will be considered later or may be in a different course. And thing like that actually not necessarily in this particular course, but there is a there is a mechanism of handling that in a soft constraint manner; that means, if you if you really do not want too much of control here in any channel then increase this corresponding r to to high value.

So, that the corresponding u will become small out here actually, so in a soft constraint when things are in place what really if you want to impose hard constraints on top of it then there are different conditions like principle we are not going to discuss those things in in this lecture actually.

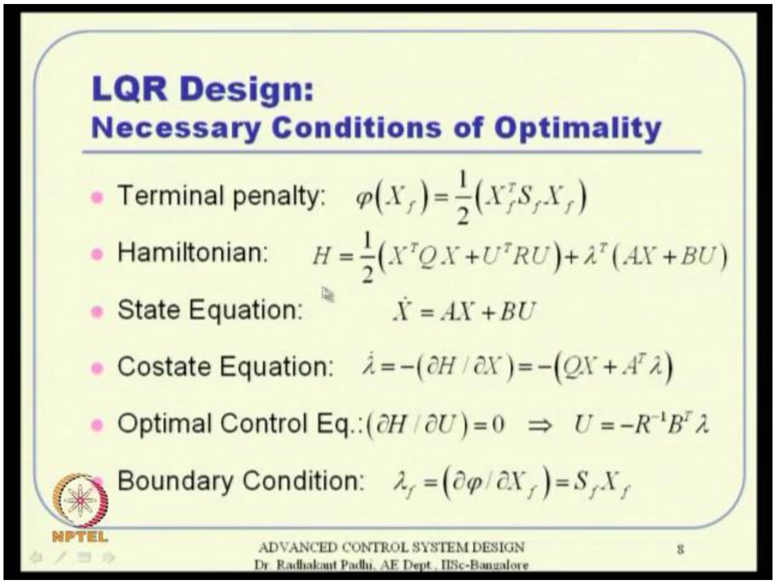
So, this anyway coming back to this, so that is the problem definition. So, we want this system dynamics in place. So, we want this state to go to zero along I mean while doing that we do not want too much of control. So, there is a control penalty as well and this is a particular cost function that we are interested in initial condition of the state is fixed, but final condition is free; however, final state is actually getting minimized through these particular component of the cost function that is the problem.

(Refer Slide Time: 10:55)



So, let us go back to this problem, and then try to visualize what is happening here. So, the path constraint the performance index to minimize what a near certainly we are considering a minimization problem, because X of f needs to go to 0 anyway. So, this particular cost function J contains this phi of X f and this L of X U, in the in the original setting what we discussed here this phi and this L are given by those those terms actually this phi and that L. So, path constraints happens to be X dot is A X plus B U boundary conditions X of 0 is specified, and t f is fixed, but X of t f is free actually. So, under those conditions how do we handle these how do we get that actually.

(Refer Slide Time: 11:39)



So, if you see that I mean lets go back to this problem and try to utilize these necessary conditions of optimality and in that setting we needs to first see that phi of X f is this one and Hamiltonian is l plus lambda transpose f. So, l is that part of it that is l plus lambda transpose f f is that part of it. So, l plus this is l plus lambda transpose f f is a x plus B U. So, once this phi and h are ready we can talk about state cost ate and optimal control equation as well as the boundary condition the state equation already we know x dot is a x plus B U cost ate equation is negative of del h by del x.

So, h is already known to us. So, if you see del h by del X This term will throw us q x, because of half of X Transpose q x del del y del x of that is actually q x plus there is nothing coming from here, but something coming from here and del y del x of this particular term is nothing but the coefficients getting altered in a reverse sequence; that means, what you really have is a transpose lambda it is not lambda transpose a, but it is a actually reverse order a transpose lambda. So, that is how we get it negative of q x going from here plus a transpose lambda coming from there. So, that is nothing but lambda dot actually.

Then the optimal control equation is del h by del u equal to zero and if you go back to that and see what is del h by del u h is like that. So, del h by del u again from this component is nothing but R U and then from this component its b transpose lambda, because lambda transpose B U Take partial derivative with respect to u it will turn out to be the coefficient transpose in a reverse order actually. So that means, it is b transpose lambda basically. So, you have these u u I am sorry, R U plus b transpose lambda equal to zero; that means, u equal to minus r inverse b transpose lambda.

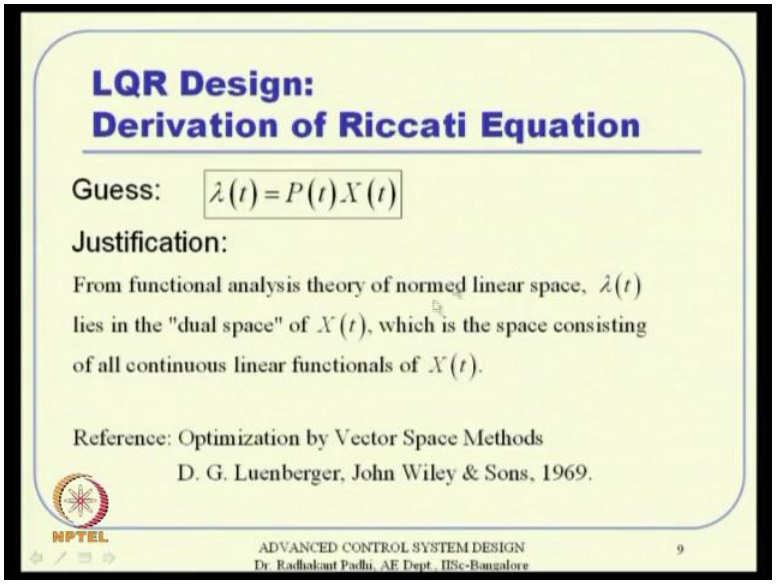
So, that it is how we get it each of this one and then the boundary condition sense lambda f is del phi by del x del phi divided by del X f and phi is that. So, del phi by del x f is nothing but that. So, these are the things that we need to account for state equation cost ate equation optimal control equation which gives you that minus r inverse b transpose lambda equal to u and then this lambda f is a boundary condition which is nothing but S f equal I mean lambda f equal to S f X f .

So, these conditions needs to be satisfied together remember if any of the condition is not satisfied then it is not really an optimal controller, now if you just just see this equation this u equal to r transpose minus r inverse b transpose lambda then once you have lambda known to us then u is known to us. So, what you really want in this case is the lambda as a function of state.

So, that you have a control will let as something like a state feedback from actually suppose lambda is a function of X, then r and b are typically fixed anyway. So, those are known to us. So, as as as long as lambda is a function of s the control becomes Rayleigh function of x and we do not want any direct function on time actually; that means, we really do not want to solve this control in an open loop sense we just write this lambda as a function of x, so that you can visualize this control as a feedback control state feedback control actually.

Now, you also let see that at t equal to t f lambda f is actually a linear function of X f S f into X f; that means, lambda f really is linear function of X f, now if you really want to extend that t equal to t f then if it is a linear function of X f then how about considering lambda at any point of time as a linear function of that particular x in other words the lambda of t is a linear function of x of t. So, that is intuitive guess actually the second thing that you should relay is you can start with a guess like that, but also remembers that LQR problems will admit unique solution ultimately. So, if you get one solution then you got almost I mean you got every solution.

(Refer Slide Time: 16:13)



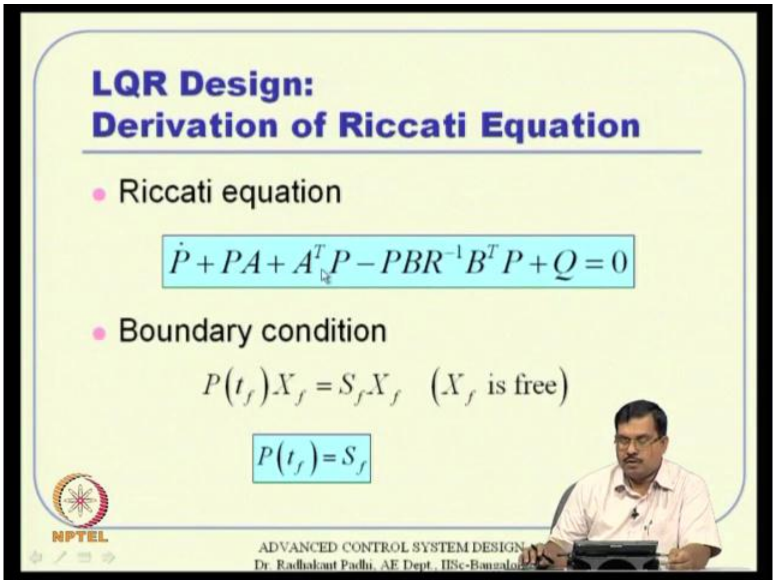
So, we are interested in let us say you guess this lambda as a function of X taking the clue from there, and let us assume I mean lets somehow try to see whether that particular solution satisfies all these conditions or not if it satisfies, then you can invoke the uniqueness of the solution. And then tell we are we have done anyway, but these are in the this intuitive argument say there are nice mathematical arguments as well which actually tells that this lambda of t is the lies on the something like dual space of x of t i will not get too much detail into that those of you are interested, you can see this reference the details are there in that but in a mathematical vector space sort of ideas you can also justify that lambda of t lies in the dual space of x of t. And dual space means it is a space consisting of all continuous linear functional of x of t, that is the meaning actually. So, if I this justification point of view also the lambda of t need to be of a linear function of x of t.

Now, the coefficient matrix p of t remember lambda is n dimensional x is n dimensional. So, p of t is needs to be n by n and we do not want to visualize this p to be a constant p we are bring in the flexibility that p of t can. In fact, p time varying P now with the with this relationship in place let us see how do we kind of derive some sort of simplified version of this three all these four equations in place embedded into one equation actually let us try to derive that. So, we note that lambda of T is equal to P of T into x of t we start from there, and then take derivative both sides lets though lambda dot is nothing but p dot times x plus P times x dot directly from this equation then x dot is nothing but a x plus B U state equation and u is nothing but minus r inverse b transpose lambda that is optimal control equation and lambda is again P times x. So, I put P times x. So, I took derivative both sides time derivative that is P dot x plus p times x dot.

So, P dot x i will not say I will just keep it as it is, but x dot I know it is a a x plus B U using state equation u is minus r inverse b transpose lambda using optimal control equation that is that is what it is here let see this one first and this one next and then we will see that lambda is nothing but P times x. So, we will just put P times x here, but lambda dot is also that if you go back to the cost ate equation lambda dot is minus Q X minus a transpose lambda and lambda is again p times x. So, minus Q X That within bracket plus a transpose b a transpose into lambda lambda is P x again.

So, I put put that here. So, then I see that all these I mean this this equation is also valid now I will take all the terms in one side of this equation. So, that x becomes common to everything it is it a post multiply everywhere that is equal to 0, and remember x need not be zero at any point of time actually the entire state vector x is 0 only when t I mean either it is or t f really goes to infinity even in that sense x is not strictly equal to zero it will only asymptotically approach to 0 basically. So, that sense this equation is satisfied for all possible x only if the coefficient becomes equal to 0.

(Refer Slide Time: 19:19)



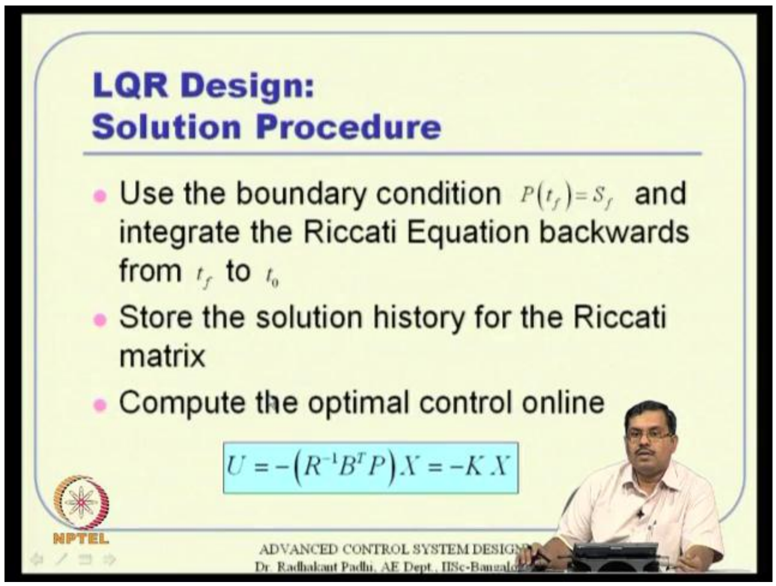
So, that is how we we tell that this coefficient matrix has to satisfy this constraint that is this P dot plus P A plus a transpose P minus PBR inverse B transpose P plus Q equal to 0 which is commonly called known as Riccati equation all the this differential term here P dot. So, it is also called differential Riccati equation actually.

So, only this is remember this particular equation is completely independent of initial condition of the state actually now that is the beauty no matter, where you are operating you can simply visualize this differential equation independent of that that particular initial condition actually now if the. So, the idea here is can we integrate this equation independently at least now for that we need a boundary condition and for which we will go back to that that boundary condition that we have lambda f equal to S f X f, and and then we try to derive that, because the lambda by definition is p times x. So, at t equal to t f it is p f t f into X f X f f is equal to S f X f from boundary condition, but X f is free. So, it is not necessarily equal to 0.

So, from there you can derive that p of t f equal to S f. So, we have a differential-differential equation matrix differential equation associated with its own boundary condition actually. So, if we take this boundary condition and this differential equation together then the problem is fairly independent and we do not need this initial condition information as far as integration of this equation is concerned actually.

So, what is the procedure now procedure is we can use the boundary condition P of t f equal S f and integrate the riccati equation backward from t f to t 0 remember this equation this boundary condition is available at t equal to t f only.

(Refer Slide Time: 20:08)



So, starting from t f we can make use of this equation and then integrate this equation back work. So, that we can get a solution of P of T at every point of time and we can store that P of T we can store the solution history for the Riccati matrix and then we can compute the optimal control online by using this relationship ultimately.

So, remember u equal to minus r inverse b transpose lambda and lambda is nothing but p times x. So, I can interpret that this particular thing r inverse b transpose P And nothing but k that is the gain matrix actually. So, remember even if a b where constraint matrices; that means, you had l t I systems even then p of t is actually solution of the differential equation and hence it is actually time varying matrix. So, even for l t I system you will end up with time varying p matrix solution from this Riccati equation actually.

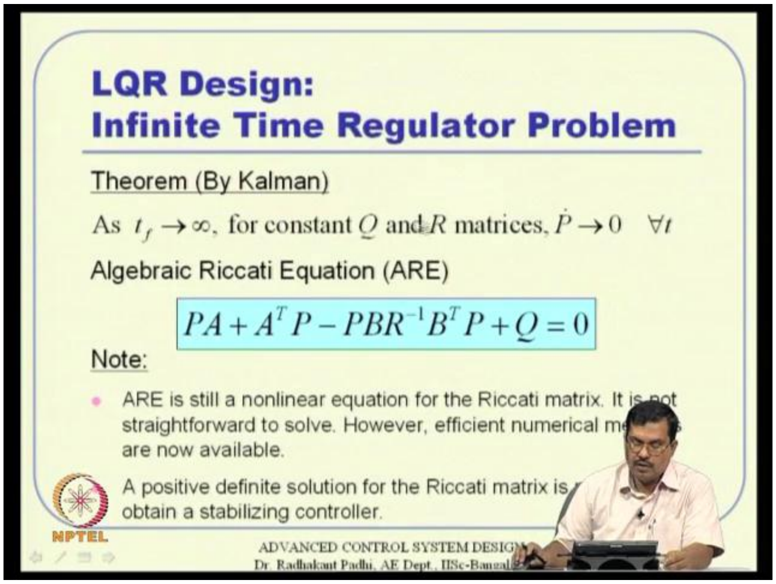
So, that is that is the way to do that, so but nevertheless-nevertheless. It is, it has become completely independent main cell condition all this P solution can be computed offline and we can store that solution and online you just need the information of x and then your control is ready, because gain matrix is already computed from there. So, select that particular P At any point of time, and then compute the corresponding gain that way and u equal control u equal to minus k times X That is the solution that we get actually.

All right, so that more simplification is there is no problem of as long as we integrate this differential equation online I mean offline and store it we really do not have to solve this two point boundary value problem online basically that is that is the message there. So, leads to some sort of simplicity-simplicity and we do not have to go back to this iterative numerical intensive procedure to solve this class of problems really and then what happens this no matter it has we have got some simplicity, but also it remember that we need to compute the solution offline integrate the solution offline store it use it online.

So, there are several problems associated with that in other words suppose t f i mean our control operation duration is greater than t f, then we do not have the solution of the for the for using it online actually that is one of the problems, and normally speaking we do not want to invite this this differential equation solution I mean if possible you want to avoid that actually.

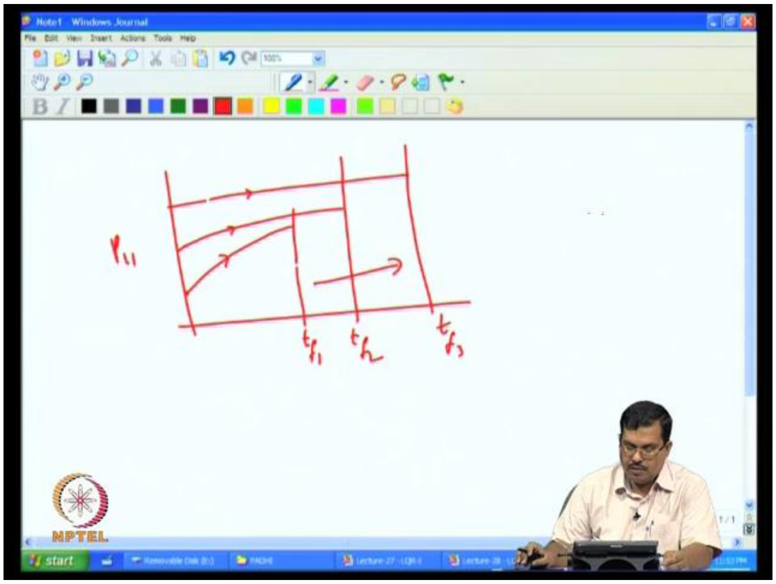
So, is it is it possible in that sense remember most of the time this LQR formulation is useful for for for infinite time problems actually, because the control operation duration I mean is is typically large compared to the problem constant time constant that you are talking actually.

(Refer Slide Time: 23:51)



So, that in that sense kalman has given a great theorem which tells that when t f is in fixed at infinity; that means, t f tends to infinity and your q and r matrices are actually non-time varying, but constant and in addition to that A and B are already constant let assume that this you are talking about l t I system. So, A and B are constant in addition to that Q and r matrix also you you fix it constant and t f goes to infinity that mean is a large final time problem under those conditions the theorem tells that P dot tends to 0 for all time; that means, this P which was actually time varying is no more time varying actually.

(Refer Slide Time: 24:45)



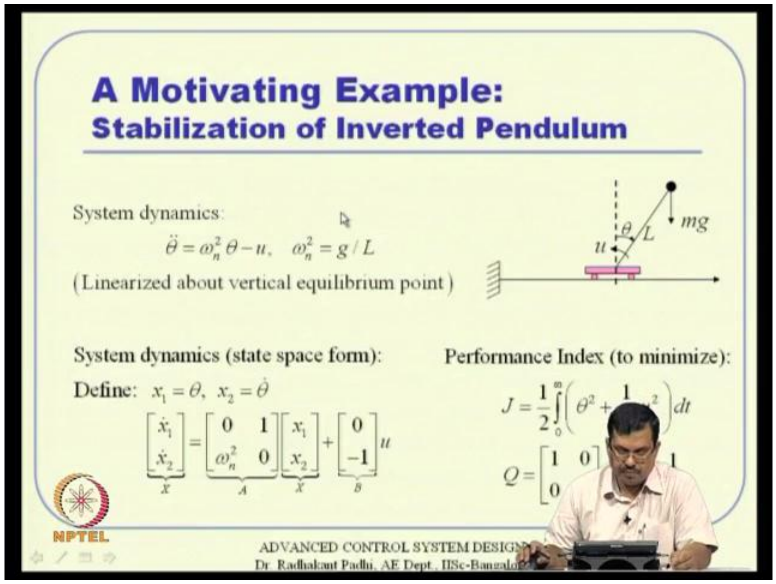
So, p remains constant let us say k if you kind of visualize this let us say if you if you are one particular element of p lets say i put p 11. Let us say initially let us say it was this like that lets lets draw this actually. Let us say for this particular time t f this is our t f one like that let me increase the final time little more. So, let us say t f 2 is like this then it turns out to be somewhat flatter and t f three is something like this then it will become even more flatter . So, this is that I mean this happens this trend happens for for every kind of element of p matrix. So, as as t f goes increasing I mean t f increases the final time t one increasing these curves are no more time varying actually they become flatter and flatter actually. So, that is the meaning of that theorem.

So, using that theorem; that means, P dot goes to infinity I mean P dot goes to zero you can suddenly see that this entire differential Riccati equation terminates to or reduces to some sort of algebraic Riccati equation, because only differential term is P dot here which is zero now. So, the remaining terms left out is like that which needs to be 0. So, that is how we get this famous algebraic R44iccati equation which is given like that. So, this is simply this P A plus a transpose P minus B P R inverse b transpose P plus q equal zero just a small comment for those of you know or we will see that in later class that if you take out this non-linear term the rest of the term also feels like what is called as lyapunov equation that is a linear equation, but this differential. I mean this algebraic riccati equation is certainly an algebraic equation; however, it is still a non-linear equation because of this quadratic term in p here, this this particular term makes the riccati equation nonlinearly.

So, that is the point here a r e is still a non-linear equation nevertheless it is not a differential equation really, but because this Riccati equation appears in many, many places lot of people have paid attention to this and efficient numerical methods are now available to solve this particular algebraic equation even though its non-linear algorithm is do exist. Now, which can solve this Riccati equation in a very computational efficient manner actually and second thing to notice this because it is a non-linear equation it can admit multiple solutions, but we are interested in the positive definite solution only and that positive definite solution is going to lead us to a stabilizing controller, we will we will prove that in the next class actually. So, whatever positive definite solution comes from here this p matrix we that will lead to a stabilizing controller actually.

So, we are interested in solving this equation using the one of the computational efficient methods that are available now and out of the multiple solution that is particular we are interested in that particular solution which will give us a positive definite solution from from this equation actually . So, I just remember that now once you get a solution of that; obviously, the the lambda is ready lambda is p times x and hence the controller is also ready because that is the same formula that we are interested actually. So, u equal to minus r inverse minus r inverse B transpose P, now the P is no no more a time varying matrix, but a constant matrix and hence the gain is also a constant matrix actually. So, that is how we get a constant gain state feedback control using LQR formulation. So, that is how it is actually.

(Refer Slide Time: 28:47)



So now to clarify our concepts, So, we will discuss two examples and then it will give us better idea of what is what is going on here actually first example is a standard example we talks about stabilization of an inverted pendulum by moving the base cart actually. So, we have a inverted pendulum attached to a cart where the cart can move horizontally. So, that I mean the the task here is to move it appropriately. So, that the torque that why moving this you will actually produce a counter torque and this counter torque is going to balance that actually.

So, we are not interested in the linear motion of the cart per see, but let us directly interpret that whatever torque that you are generating by this movement is actually the control variable and the control variable per se is not really the torque, but torque divided by movement of inertia which is nothing but the negative of angular acceleration actually.

So, that that is the that is the control variable that you have taken for solving this problem, but remember once you know this control the torque is known and once the know the torque depending on other dynamic other parameters you can actually compute this leads linear motion as well actually. So, ignoring those complications and all we will just directly take this control u as a contracting angular acceleration that can stabilize this this mass actually in in inverted position .

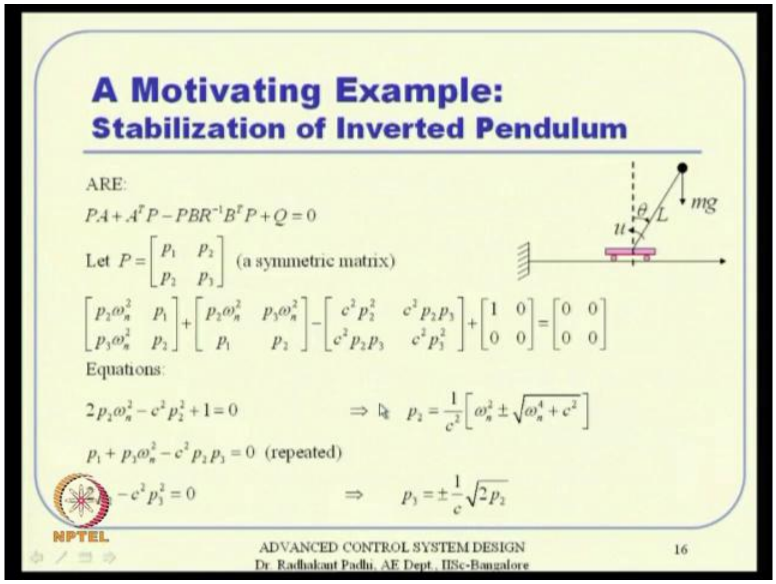
Now, we will not go through the details of dynamics how they are derived and all that the dynamics of a pendulum is fairly will non-probably, but we can linearize this equation that particular equation about the vertical equilibrium position and get this kind of a linearize system dynamics actually where omega n square is the system parameter given by g by one one is the length of this line actually. So, first thing is to write this system dynamics in state space form and once I take theta x one equal to theta and X Two equal to theta dot I can visualize this system dynamics this way which is very straightforward because x one dot is x two which is theta dot that is straightforward and x two dot is this equation this omega n square times theta which is x one that is what it is here and minus one times u basically. So, that is how it comes here. So, we have got a matrix and we have got b matrix and that is x dot equal to x plus u is known to us now.

Now, we have to select a performance index which can stabilize this thing about the vertical equilibrium point. So, let me select a performance index this way because theta needs to be minimized theta is to be zero actually ultimately. So, theta needs to be minimized then let me select cost function this way where one over c square is nothing but the waiting thing for the e square term actually. So, if I do not want too much of control then I can aim to increase this this term one by c square of one by c square term; that means, c i have to decrease actually and vice versa in other words if I want to control theta in a very tight manner then this part is to be much more than that; that means, I have to increase c.

So, that one by c square will be less actually. So, that is that is how it is now once you have this cost I mean system dynamics you have a b matrices once you have the cost function ready you have q and r matrices and we are interested in infinite times settings and whenever there is infinite time thing remember X f will go to 0 at infinity. So, there is no point in having an additional term.

So, phi lambda f is equal to p f X f which is nothing but zero actually anyway. So, the remaining is the outside term outside integral there is a zero term the phi is actually zero here as long as you talk about infinite time problems in the LQR framework. So, we do not need that term anyway. So, we have a matrix we have b matrix we have q and r. So, first thing is to check controllability and I assume that controllability check has already been done the system is attain controllable anyway.

(Refer Slide Time: 32:43)



Now, let us go to the r e algebraic Riccati equation solution. So, this is our equation P A plus a transpose p minus B P R inverse b transposes p plus q equal to zero. So, we have all the matrix elements ready now a b q r all that put it back and also remember we want a symmetric positive definite solution for P ultimately.

So, we will take the p matrix this way which is already a symmetric matrix p two p two here and p one p three there actually now once you plug in back there P A transpose P All sort of things elements matrix by matrix elements the equation-equation turns out to be that way; that means, if you if you really see the equation term by term, then first first element one by one element will give this equation and one two element will give you this which is again same thing as two two one element, and then there will be two two thing which will give us that.

So, we have three free free variables P 1, P 2, P 3 here, and three equations in place. So, certainly we can solve these three equations, but these equations are non-linear. Remember there is A p two square p two p three term here p three square here like that. So, we will get kind of multiple solutions actually.

So, from this equation it is obvious that p 2 is like this. So, we can apply this I mean quadratic term solution formula and get that get it that way and from here you can get p 3 that way. So, we do not, but we have this sign ambiguity in both places and we do not know what is right wrong actually what we see that p two we cannot tell p 2 is of diagonal term, but P 3 is a diagonal term and we want a positive definite solution. So, p 3 needs to be positive actually. So, this this minus sign is ruled out now. So, we will take P 3 as positive of one by c and square root of 2 P 2 actually, but remember 2 P 2 is also not known.

So, to get a real number P 2 has to be a positive number first actually now that helps us in deciding this this sign ambiguity which tells that sign has to be only positive quantity because remember this quantity is at square root of this quantity certainly greater than that. So, if you take negative sign here then P 2 become negative number and P 3 becomes complex quantity which we we do not want that actually. So, that way we will take only positive thing here and positive thing there.

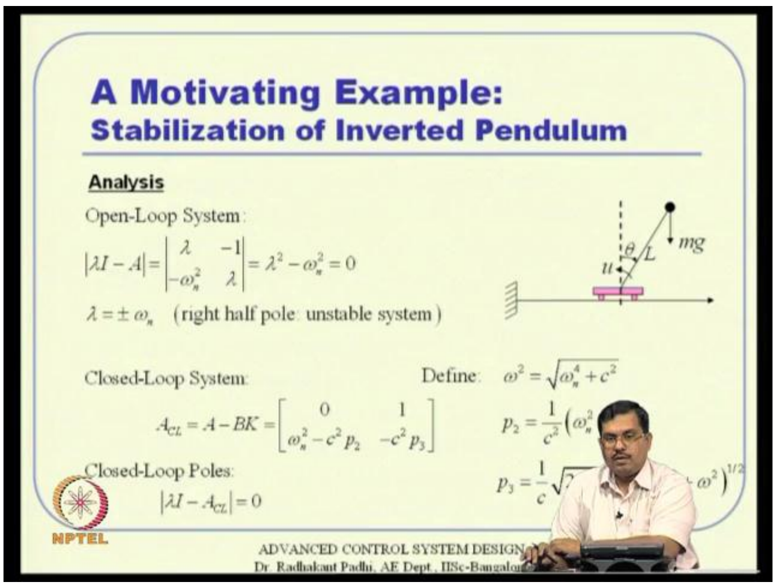
So, then we we are done with p two and P 3 and P 1 is a direct solution in terms of P 2 and p 3 we can just see from here P 1 equal to this term minus that term. So, once P 2 and p 3 are in place then p 1 solution is also ready, but in this particular problem we may not need P 1 solution, because ultimately the gain matrix is not a function of P 1 really, but certainly we can solve for that. So, that is that is what is all given here as I told p three is positive square root of that and p 2 is also positive positive sign out here actually and p 1 is can be computed this way anyway ultimately k is gain matrix k is r inverse b transpose P.

So, if you plug in that you'll you can see that this gain matrix turns out to be like this and hence the control is nothing but minus k times x where the k is like this row vector and x x is theta and theta dot. So, you can you can pull that theta theta dot and then negative sign. So, you control is actually this way c square into P 2 theta plus P 3 theta dot actually.

So, as long as you keep feeding this theta and theta dot information which is state information then your control is is actually ready through the gain matrix k. So, that is the whole idea of synthesizing the control for the inverted pendulum this way.

Now, also remember that if I change the cost function matrix and have a theta dot term which is certainly better because theta should not not only go to zero, but around zero it is should not keep on vibrating actually that that is not allowed it is not desirable actually. So, you can also nice to analyse theta dot term as well here; that means, in that situation I suggest that you can derive the rework on this example and derivable everything with respect to one more term which is q one times theta square plus q two times theta two square I mean theta dot square basically, if we have that there then what there are necessary results and all I suggest that you do it yourself actually.

(Refer Slide Time: 37:10)



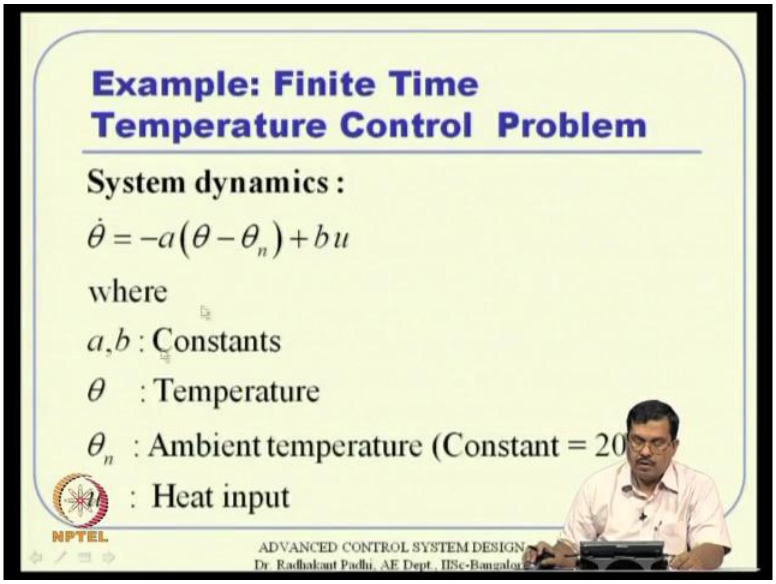
So, now the question is we got the solution is there any guarantee that. So, I mean this particular controller needs to be stabilizing controller let us examine that question. So, first we analyse the open loop solution open loop solution remember considers-considers only this a matrix whatever a matrix is there here without any control term control is zero.

So, that let me analyse the Eigen values - the Eigen values of this particular matrix and that one turns out to be lambda I minus open loop a determinant which is nothing but that is equal to zero then lambda is certainly plus or minus omega n and certainly this system is unstable because there is a right half pole in the a matrix open loop a matrix actually. So, that is it because of this one right half pole and the real axis. So, certainly it is an unstable system.

What is the closed loop system stable at least? So, let us analyse the closed-loop matrix now which is a c l is a minus b k that is what will operate right x dot equal to a x plus B U and x x dot is equal to a x plus B U and u equal to minus k x. So, if I put them together then x dot equal to a minus b k times x. So, that is that is what I have here. So, that is what my closed loop matrix b k is basically. So, let us analyse the eigenvalues of that closed loop matrix and see what is going on here actually. So, a minus b k turns out to be like that and for simplicity now will define another term omega square is nothing but square root of omega fourth plus c square term and then p two and p three we can write in terms of omega n n omega that is that will lead to simplicity in algebra actually then closed-loop poles will be by this characteristic equation now, and you can plug all those numbers that you have here in this in this a c l matrix and try to see and try to see what is going on here and then lambda one two will turn out to be something like this remember there is a negative sign here; that means, the real part of this solution is certainly negative for both the thing.

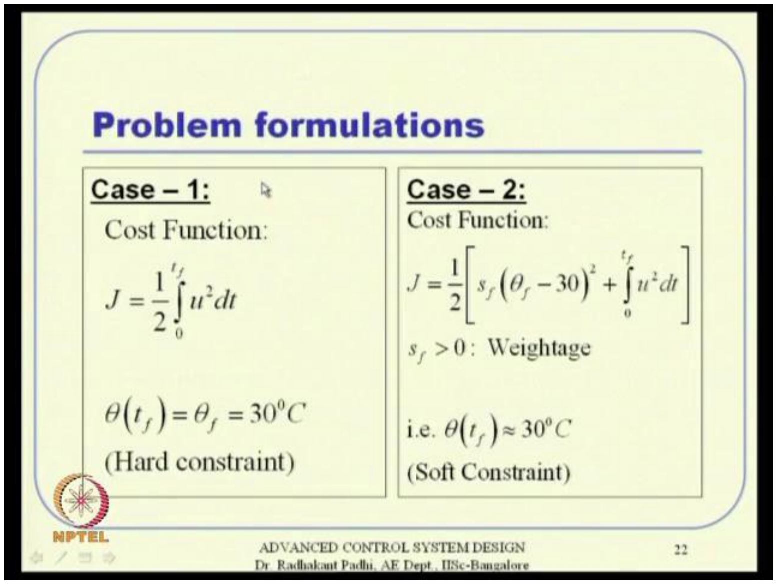
So, what you what originally you had something like two poles out there now this these two poles got shifted to these poles actually this is no more this is open loop openzymes open loop is not there this is certainly the closed loop pole actually so; obviously, the open loop thing was not good whereas, the closed-loop system is stabilizing actually. So, that the thing the closed-loop is guaranteed to be asymptotically stable, and we will also see that in next class how do we kind kind of guarantee these per all possible cases not just for inverted pendulum and all that actually that prove will be there in the next class actually.

(Refer Slide Time: 40:12)



Before We closing this lecture let us study one more example and see what all goes on here actually. So, this is the finite time temperature control problem let study and here we are talking about a system dynamics like this let say theta is nothing but the temperature and theta n is the ambient temperature let say it is fixed at twenty degree for just an example situation and u is the heat input and if u is negative its heat taken away is a cooling condition if I mean u is positive then it is a heating condition which is; that means, in the in winter season we can think about giving heat input to the room and all that actually. So, this is the system dynamics certainly its linear linear dynamics, but there is a bias terms out here actually minus a theta plus a theta n. So, that is the bias term the theta n is ambient temperature which is constants a is also a b are constant anyway.

(Refer Slide Time: 41:04)



So, how do you handle this particular situation and now is rather easy to handle by doing a change of coordinate sort of ideas we will see that little later in this this one let say we will talk about that anyway now the what is the what is the objective here objective is to take this theta to a desire temperature actually and that desire temperature let us talk about that I mean some sort of thirty degrees what we really want the desire theta basically.

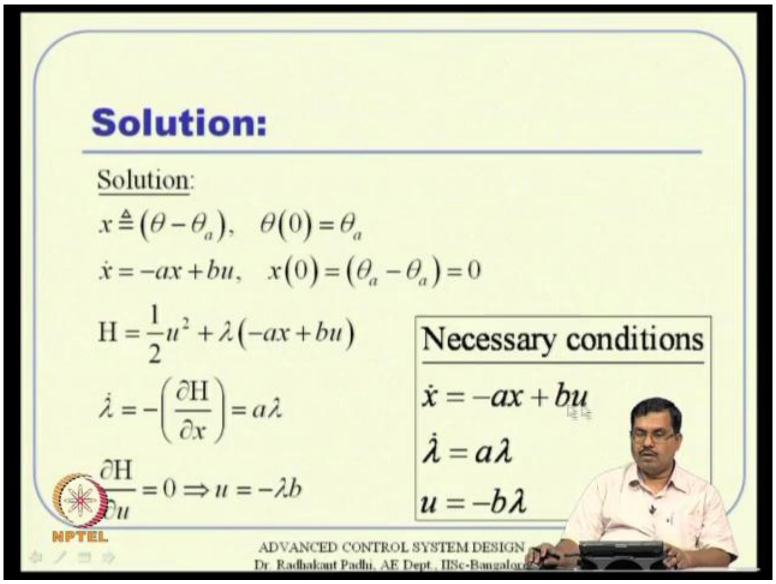
So, if you see we can formulate these two cases in two different ways one case I while I while I aim for this thirty degree taking this 20 degree 230 degree I also should aim to minimize this control effort actually that is another object anyway. So, if I can formulate this entire problem in two different ways one case I can talk about a cost function which is control minimization only and which is hard constant in place; that means, at t equal to t f we want theta up to be exactly equal to thirty degree.

As an alternative case two we we do not aim for that exactly thirty degree as long as you are closed to thirty degree you are fine. So, hence the cost function that we are talking here is some penalty function here outside integral at t i mean at t equal to t f theta f should be closed to thirty actually.

So, as long as I take S f equal to I mean greater than 0, and have a term like that then we are actually this. So, theta will be forced towards thirty degree, but how much closed to thirty we do not know that for that question remains open that depends on the selection of S f actually. So, that that we will see from example example if you can talk S f, if you S f S f become higher and higher ultimately it it leads to this hard constraint sort of problem actually.

So, here we talk about hard constraint problem here we talk about soft constraint problem actually. So, let see how we handle this actually anyway anyway. So, that is let see how we handle this.

(Refer Slide Time: 43:10)

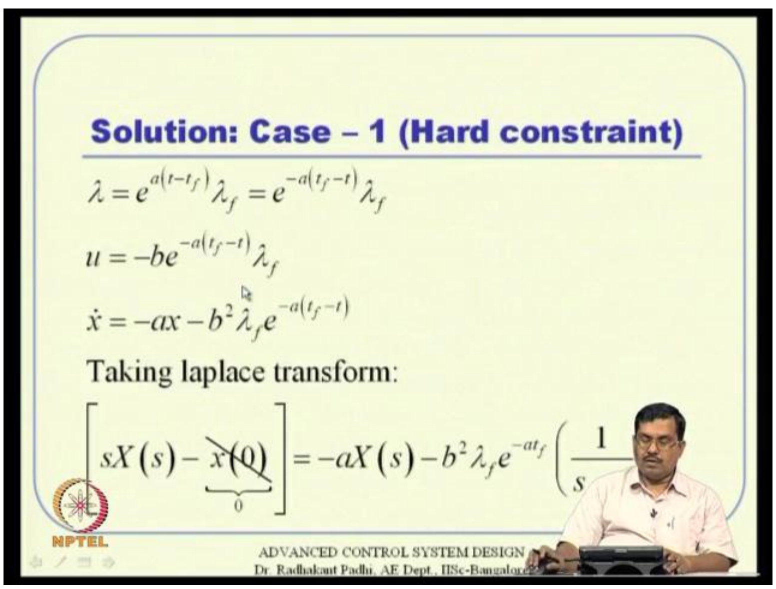


Now the solution part of it first does a change of variables. So, we take theta minus theta the ambient temperature that is what our x is and also we go back to this this this dynamics then this dynamics remember theta is is fixed. So, maybe this is theta a yeah this is theta a actually sorry this is theta a well alright. So, if theta a then we can talk about change of variable here. So, of course, change of coordination essentially then theta x is define as theta minus theta a. So, x dot is something like this and correspondingly you have to change the boundary condition values. So, x of zero turns out to be theta minus theta which is zero and X f we will see in the later in the hard constraint part of it.

Now, what we what we are doing is no matter whether you have hard constraint and soft constraint Hamiltonian remain same because this is nothing to do with the this particular type I mean whether is outside term Hamiltonian is only l plus lambda transpose f which is either inside the integral or system dynamics. So far both of the current problems that part remain the same sp Hamiltonian is like that. So, lambda dot is that way minus del h by del x equal to x and you have this optimal control equation del h by del u equal to 0 from which we can derive equal to minus lambda b.

Now, necessary condition hence if we we can summarize that the steady equation that is why it is cost ate equation that is what it is and control equation that is what it is. So, these three equations needs to be satisfy together basically now nice part of the equation. I mean observation here is this lambda dot equation is is independent of itself I mean independent of any other variable x and u. So, because lambda dot is a function of only lambda this equation can be integrated directly basically and then once lambda is there u is there and once u is there you can put it back and then think about solving the state equation also actually and all this will happen in a closed form manner which is the nice part of it actually.

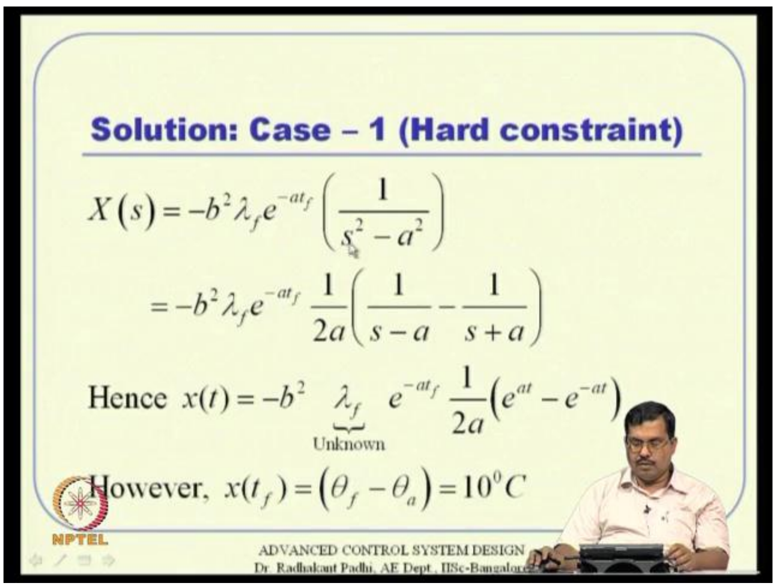
(Refer Slide Time: 45:38)



So, let us solve this equation lambda dot is a lambda first and then we can visualize the solution that way with respective to lambda f the solution takes this form remember typically you take lambda 0, but in this particular case we take lambda f is the reference value. So, e is the exponential term become naught t minus t 0, but it is t minus t f actually that is that is the difference. So, lambda of t is given something like that which is nothing but that and t f minus t is is typically written because that is a positive quantity and we shall guidance especially it is called as time to go t go. So, this is this is this form actually.

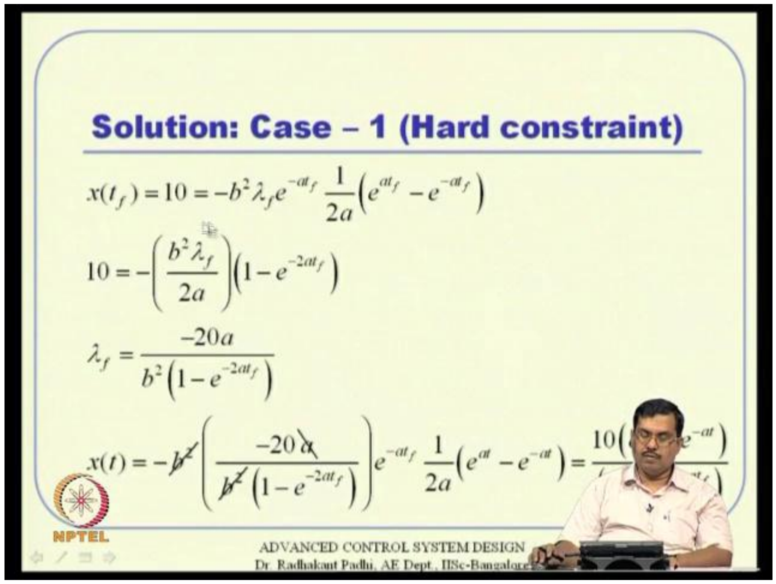
So, lambda a takes to this form and hence once lambda is that form then we can substitute by, because remember this completely time depend I mean it explicit function of time now. So, we can solve this equation also equal to minus b time's lambda or lambda comes from there actually. So, that is what u and once u in place you can go back to the state equation and then substitute there, and then then look that is my state equation actually.

(Refer Slide Time: 47:09)



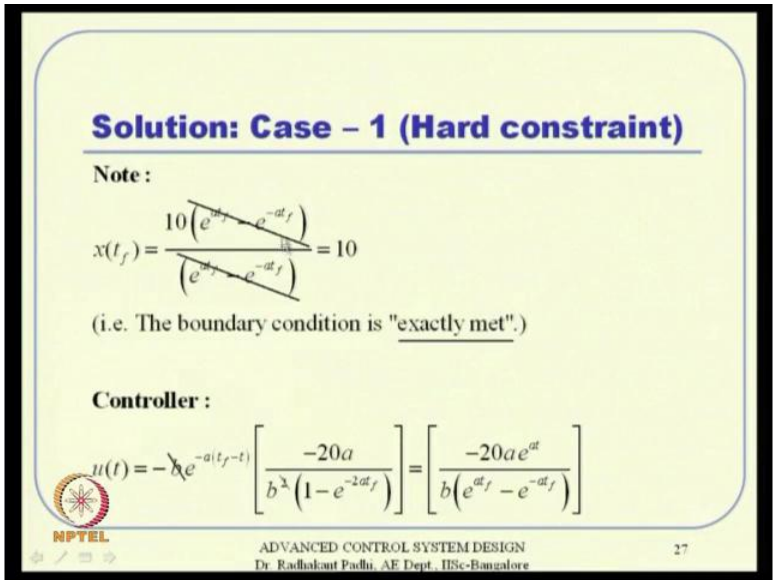
So, let me try to solve this state equation which consist of partly the homogeneous system partly the forcing function actually there are several ways of solving this we can take the help of Laplace transform and then tell this is if I apply Laplace transform both sides then that is what it is and then x of zero is fixed at zero remember that is the boundary condition x of 0 0. So, we plug that x of 0 0, and then I will try to solve this x of s and then x of s turns out to be like that take the parcel fractions the composition of these standard procedure and then we take the inverse Laplace transform of that and hence you obtain the solution actually we are still not done, because lambda f is actually unknown that is what that is what needs to be found out actually.

(Refer Slide Time: 47:44)



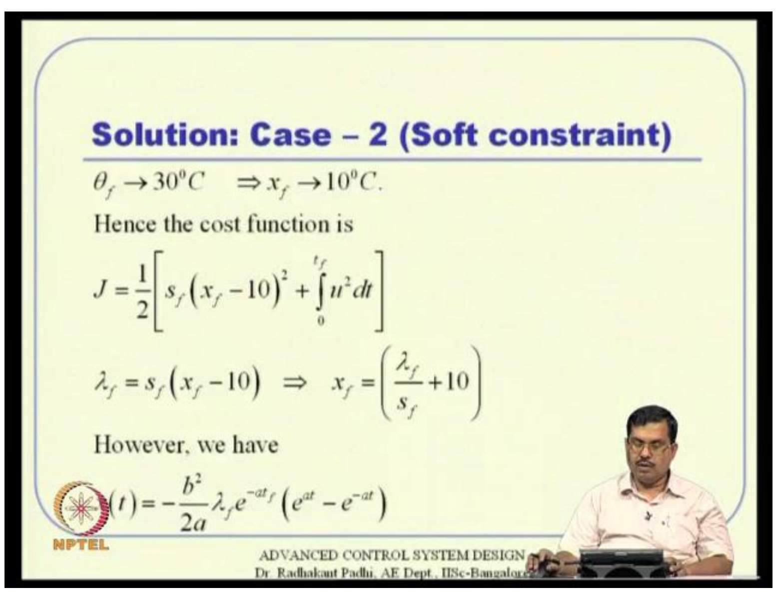
Now, for that we will take make use of the hard constraint let say case one and then x of t is equal to ten that is what we are interested in. So, if I put x of f equal to t a i mean equal to tend sort of thing in this equation and take t equal to t f then what is whatever you get here we will give you the solution for lambda f actually. So, equating this two we will get simplify and then get a solution for lambda f now once you get a solution for lambda f we are we can go back to your x x of t solution, because that is of the only unknown here now lambda f is unknown to you. So, let us talk about solving that actually here. So, x of t takes out turns to be like that all right.

(Refer Slide Time: 48:18)



So, what is what is beauty here now if you put t equal to t f here then obliviously these two terms will cancelled out and x of t f will be equal to ten. So, in other words the boundary condition is exactly met actually using this particular controller whatever now control is controller is minus b term lambda where lambda solution is given like that and lambda f we have found out using this particular thing.

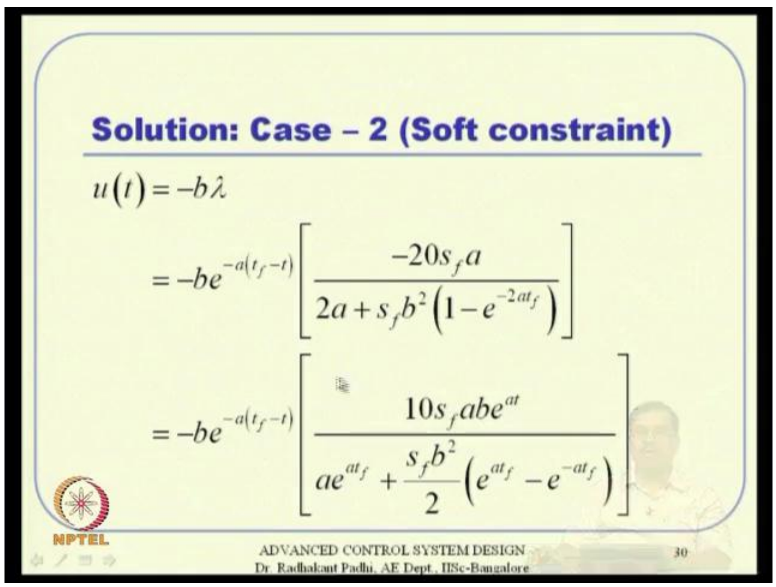
(Refer Slide Time: 48:44)



So, that is case one now suppose I mean the the problem here in in case one is a kind of control variable actually when t approaches t f the control variable is typically not nice actually it may tends to infinity that is the typically hurdle in any finite time optimal control problems first actually, but we can see that this x of f is actually met in a very nice way basically

So, now let see that how we handle this problem in a soft constraint manner, because that is the hard constraint way now case two is soft constraint. So, we go back to this cost function and then look lambda f is this way now from where you can solve X f in terms of lambda that way.

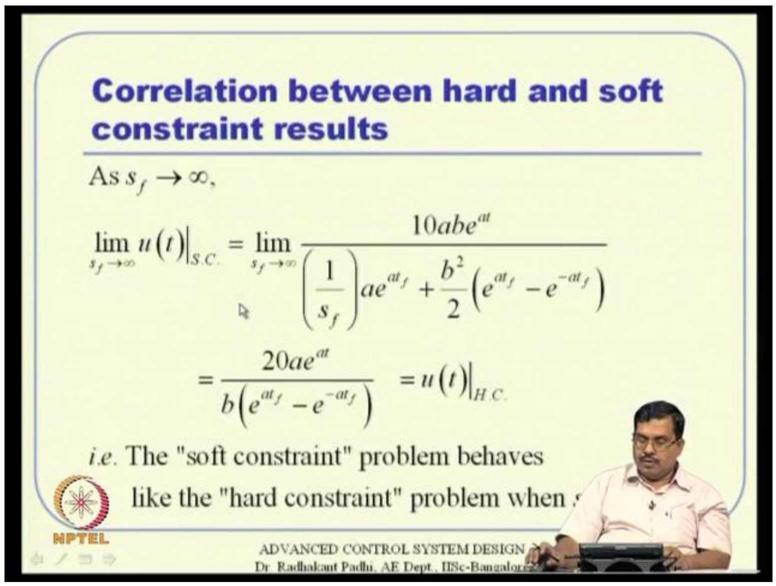
(Refer Slide Time: 50:00)



So, we have the solution in place already whatever that part remain same we have to solve the lambda f in a different way though actually. So, we put this only solution we put equal to t f and try to find a solution for lambda f and that is what it is done this way. So, lambda turns out to be like that and hence lambda turns out to be that way which is given that way that that expression actually. So, u of t is equal to beta minus b times lambda which given like this basically. So, that is a soft constraint case.

Now, the question is if I solve the same same problem using soft constraint and hard constraint the only thing I can actually approximate the hard constraint situation through the soft constraint formulation by taking S f going to infinity basically this cost function if I take S f infinity then this term is predominant.

(Refer Slide Time: 50:36)



So, that is what the hard constraint I mean the soft constraint will try to approach the hard constraint problem does the solution also approach the hard constraint problem I mean that is that is the question actually. So, we can do this analysis-analysis and then look when S f goes to infinity that u of soft control soft constraint formulation is given something like this and if S f tends to infinity one way S f goes to zero. So, I can cancel out that term and then further simplify this term and then look that is what the what the formula should be because the soft constraint under the limit S f tends to infinity and this formula is nothing but the same control will turns out to for the hard constraint formulation actually.

So, the soft constraint problem we have slight the hard constraint problem when S f goes to infinity that is the message actually which is very comfortable with what we thought in terms of cost function selection basically when you select the cost function that is the meaning. So, as long as you put not S f tends to infinity, but some finite number here then there is a nice formulation where you talks of relative weight age between the how much close you want to go to the boundary condition and how much compromise on the control f for that you want to make actually these will be a nice interpretation that basically.

So, that is how the soft constraint and hard constraint problem can be handling. So, probably I will stop here in this lecture and we will continue further LQR theory in the in the next lecture. Thank you.