Optimal Control, Guidance and Estimation
Prof. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Science Bangalore

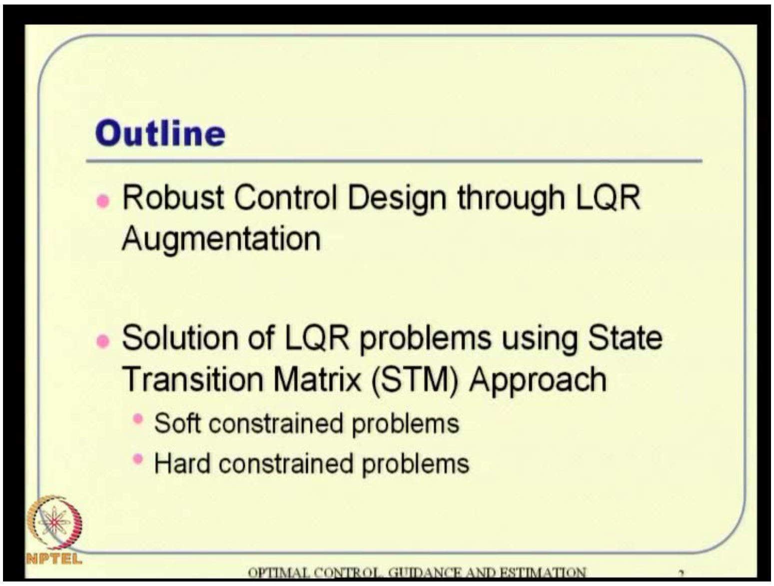
Module No. # 05

Lecture No. # 12

Linear Quadratic Regulator (LQR) – III

Hello everybody, we will continue our lecture series on LQR design. Last two lectures we have seen this foundations of LQR followed by several (( )) and things like that and we will follow up with further ideas on LQR design in this lecture as well.

(Refer Slide Time: 00:33)



So, primarily two things I want to discuss in this lecture, one is robust control design through LQR augmentation ok. So, it is possible to have some sort of a robust control design and we will see what the kinds are. When we talk about robust control, the robust control can be from several factors actually, robustness with respect to let us say external noise, robustness with respect to modeling and certainty, robustness is part of that is robustness with respect to parameter variation something like that actually. Because no matter how much information we can have or how much modeling you can do, they will be some sort of on model dynamics of the time, which includes parameter uncertainties as well actually.

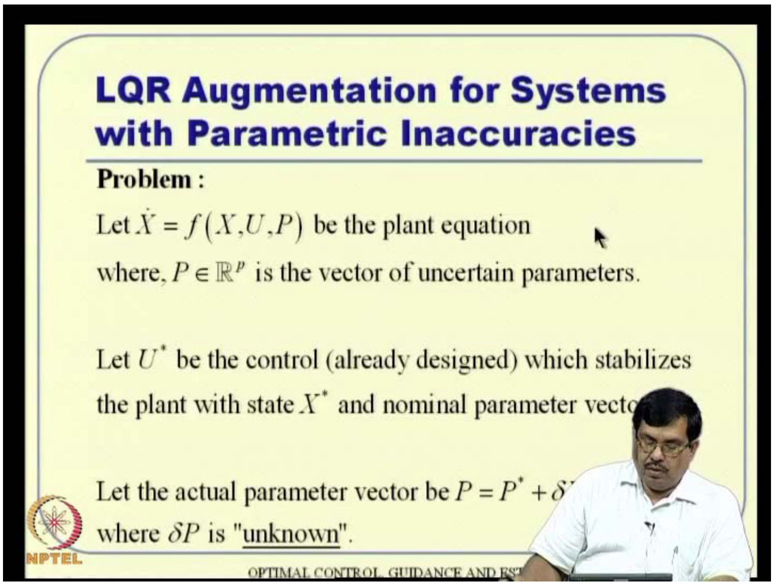
So, that kind of things can be handled through LQR and when we talk about external noise into the system, we will see later that it is still possible to use LQR under the assumption of kalman, I mean under the augmentation of kalman filtering basically. We know that means, that essentially leads to this LQG design short of ideas actually, Linear Quadratic Gaussian I think, you we did not talk about that here yet; we will talk about robustness with respect to parameter of variation or some sort of a forcing information of sort of thing actually, we will see that ok.

And the second thing what you want discuss here is solution of LQR problems using this state transition matrix approach sort of ideas. So, what happens here is if you follow this record equation approach, you essentially land up with N by N non-linear, either differential equation or algebraic equation in terms of the recovery matrix actually, ok. So, solution becomes kind of and involve and many times we will end up with numerical solution. So, that part of the computational basically, so we do not have to do that one, you follow this state transition matrix approach, the drawback here is that at the problem dimensionality increases from N to 2 N actually.

So, we consider this X dot and lambda dot something like together sort of ideas actually, remember X is n dimension means lambda is also N dimensional lambda actually. So, you increase the dimensionality, but kind of reduce the complexity sort of thing, so that is the whole idea there. So, it is a different approach and I thought it is good to know that actually.

Under this umbrella, we will talk about soft constrained problems first, what we have discussing so far and it is also possible to discuss hard constrained problem. That means, you really want some part of the states to go to a desired value, that is in equality sense and all that, it is possible to do that, there are certain drawbacks also doing that way, but we will talk about that as we go along actually. So, let us get going first is robust control design for systems with parametric inaccuracies through LQR augmentation that is what we are discussing (( )).

(Refer Slide Time: 03:17)



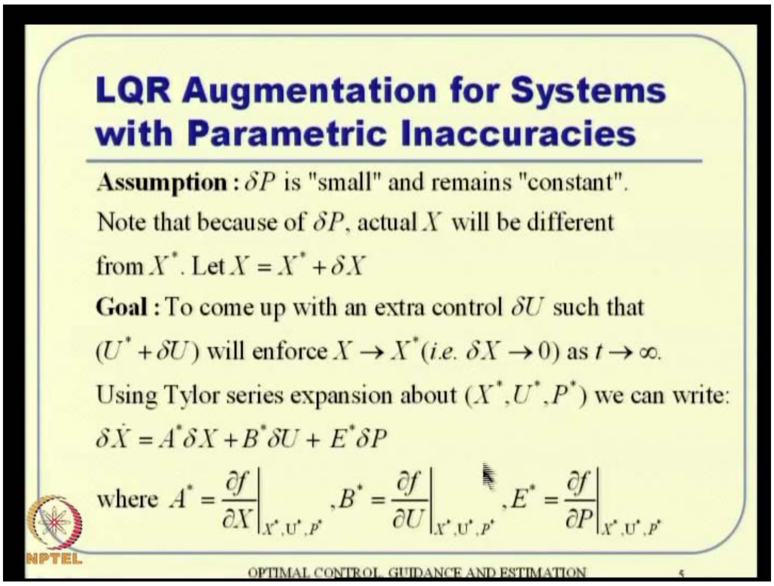
So, what we are discussing, normally write that X dot equal to f of X U for non-linear system, assuming that, that set of parameter values are kind of constrained actually, they do not vary or they are known basically. So, they do not play any significant role sort of thing, but in general we can also write this X dot equal to f of X U P, where P contains the parameter, I mean parameter vector actually. And for example, if you have a moving mass, then X the dynamic equation is m X double dot equal to f; that means m is a parameter of the system. Similarly, if a rotating body, moment of inertia becomes a parameter, if it is aero aerodynamic controlled aircraft, then aerodynamic forces and moments there are several coefficients, which can be thought about as parameter of the system dynamics, think like that actually ok.

So, those are the values that we are talking about as part of the system dynamics actually and we have a non-linear system like this, X dot equal to f of X U P and this what you are talking here is we already know a U star, some method is there, I do not know we do not want to do alone on that, we assume that U star is some control which is already designed, assuming there is a parameter vector P star. That means, that is that is the knowledge which is known to us actually, in other words if the mass of the vehicle something like 102 kg and we assume that it is 100 kg, that means, that P star turns out be 100, the 2 we really do not know whether it is 100 102 105 or 98 whatever actually.

So, that part is uncertain actually, but p star is largely known, I mean exactly known the other, if you if you take p star is a parameter vector which is completely known, then you can design a new star based on and that will result in a next star trajectory. So, that means for a known set of parameter vectors, we already have the controller ready basically, it can come from optimal control, it can come from any other controller as well actually, we are not bothered about that at this point of time actually.

And one possibilities certainly optimal control, non what nonlinearly optimal control that trajectory optimal ideas that we discussed before actually alright, so this is what it is actually. So, now the, but the reality is the mass is not 100 kg and it can be 100 200 598 whatever it is. So, we consider that part is something like delta P, so P is actually a combination of P star, which is already known to us and an unknown factor which is delta P actually ok. So, remember we are not having any sensor of anything to measure that online, we just have to leave that error actually. So, that is the concept, can we design something like some control or which will which will (( )) actually.

(Refer Slide time: 06:06)



So, what is the objective here? Objective can be described something like this, what you assume first is delta P is small we do not have to talk about large uncertainty and all that, we talk about small uncertainty, that too that is what I just told about actually, if mass is 100kg because consider 100 to 105 like that, it is kind of plus or minus 5 percent in accuracy or may be 110 kg or 10 percent accuracy like that. We do not talk about something like 50 percent, 100 percent, 200 percent accuracy like that accuracy; that means, that information is a different class of problem altogether and we do not want to talk about that here, ok.

But, largely it is like that, in a in a normal situation well there is a small comment, large uncertainties are not unrealities, they are also realities when they talk about now people talk about something call reconfigurable control. If you have large uncertainties the system plant dynamics is largely different from what we discuss actually, if for example, if your aircraft is flying and you have a vital damage, that means half of the wing is gone or something lost and things like that, that kinds of uncertainties are large uncertainties, we do not talk about that. What we are talking here is delta p, that means, some parameter inaccurate information; that means, wind tunnel data some coefficients have been computed through polynomial think like that, that polynomial numbers can be inaccurate, but that inaccuracy is not suppose to go beyond 5, 10, 20 percent actually ok.

Similarly, mass moment of inertia cannot go much more than 5 10 percent actually. So, that kind of inaccuracy we are talking about, anyway coming back what we are talking is parameter vector P is can be described something like this, P star plus delta P, where P star is known, delta P is unknown actually. So, delta P by assumptions remains small and it will also remains constant, it does not vary with time as well, we assume that it remains constant, we just do not know that part, but it does not vary with time, if mass is instead of 100 it is 105, it is 105 it does not keep on varying 105 to 107 to 110 and think like that on line actually, so that assumption is this.

So, what happens now if you have P instead of P star, your state will develop as X naught X star, so we consider that our real state, I mean state variable X of t is something like X star plus delta X actually. What is objective? The goal is to come up with an extra control delta U such that the controller also needs to be a part of U should be equal to U star plus delta U basically.

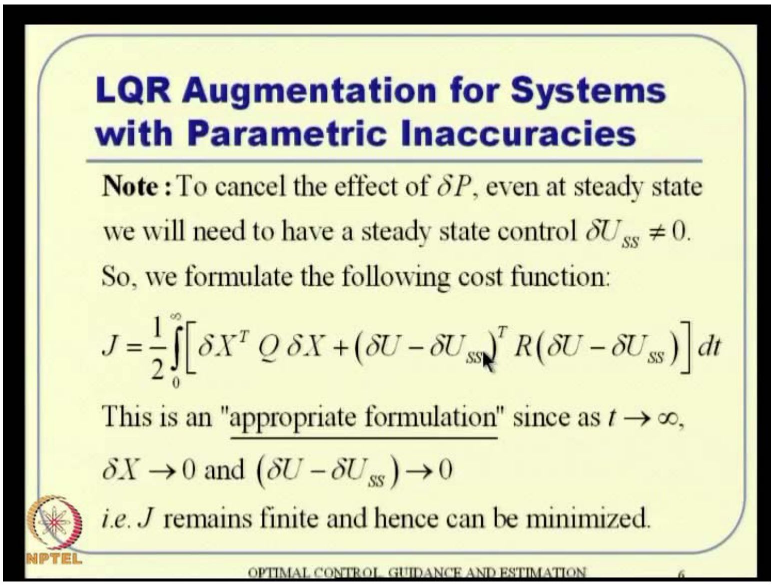
So, remember what we have done here is we have a plant, here we have a U star already designed for parameter vector P star, which will result in X star. Now, P is not P star and which will result X is not X star and hence we will need some other U not U star to compensate for all that actually. That means, what we really require is to compute a delta U such that, U star plus delta U will make sure that X (( )) if it is X star, so X goes to X star goes as soon as possible actually. And why because X star is already available to us, we know that everything good about that, because that is a nominal control design already done actually. So, in essentially if we can design some delta U, so that u star plus delta will enforce this, the actual X 2 to follow X star then our job is done actually.

In other words, we want this delta X 2 to go to 0 actually that is the problem, so how do you do that? Remember delta p is not known actually, now using the Taylor series expansion about X star u star p star and all that, you can you can also write this actually right, if starting from this non-linear system dynamics following the same principle of linearization of systems and again that is a very standard procedure and some of you do not know how to linearise system of non-linear system, why actually have a small discussion about that in the very beginning class, several lecture in my other course as well as actually.

Anyway so this is what it is, we can go back to this thing and we have a nominal X star, nominal U star and nominal P star available, so using those values we can talk about a linearization about those values actually. So, we will end up with this kind of thing, where A star, B star and E star can be computed this way, these are matrices can be computed that way ok. Just see the Jacobin matrices and all evaluated at this nominal value actually. So, we have some sort of a linearism expression all there alright.

So, what is the idea here? Idea is to cancel the effect of delta P basically, so if you have to cancel the effect of delta P, also you need some sort of delta U steady state because remember P is a parameter, that delta P is a constant value. So, delta X dot is there, so to cancel that effect, we need a constant bias control sort of thing, that delta let us the talk that is some steady state control actually, delta steady state bias control, the delta U ss.

(Refer Slide Time: 10:30)

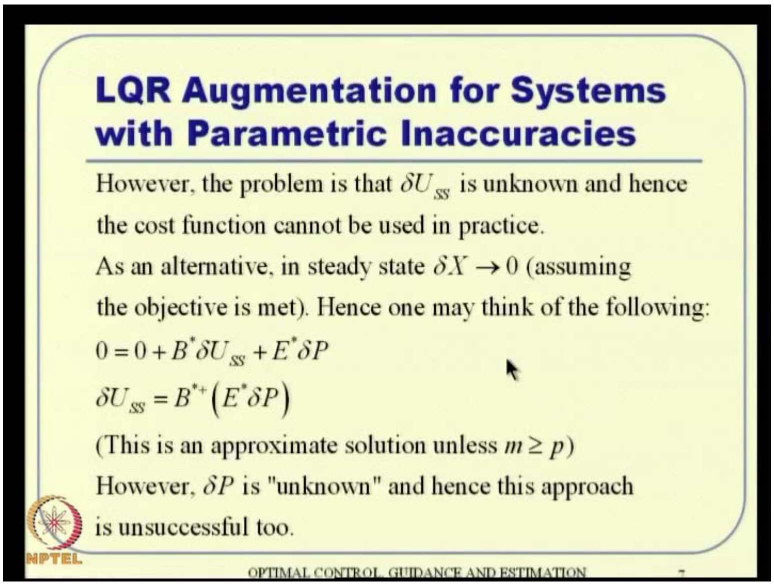


So, assuming that is available which is not 0 anyway, so what we really need to have a cost function is something like this, delta X should go to 0, so that means that part is a quadratic function and remember delta u should not go to 0, but delta u minus delta U ss should go to 0, because that this steady state control we need to nullify that we are aware actually.

So, this difference should go to 0 that means, you can construct a cost function with this difference being a quadratic function actually and ultimately remember as long as you talk about 0 to infinities sort of thing, the integrant values should ultimately go to 0, otherwise the cost function will remain unbounded, it will go to infinity and all that and which cannot be minimize actually ok.

So, that is a incompatible formulation a very quick check is whatever you are putting it as a quadratic function here, that at some point of time it should go to 0 and it should remain 0 actually. So, you put it quick and then tell this cost function can be minimize actually; however, this turns out to be an I mean it is it is ok because this has to go, but the, but the problem here is we do not know this value, delta U ss we do not know actually. So, even if you compute a difference like the delta U minus delta U ss equal to some express and think like that, it is virtually of no use because we cannot recover delta U, because we do not know delta U ss actually; so that is the problem there, so solve that is explained here actually ok.

(Refer Slide Time: 12:17)

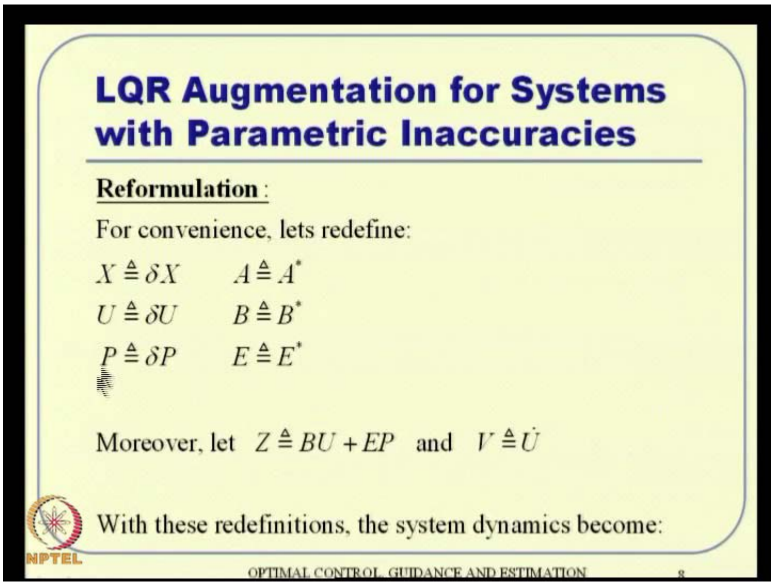


Since delta U ss is unknown and hence the cost I mean cost function cannot be used in practice, so let us find out some alternative ideas actually. So, as an alternative U what will happen in steady state? Steady state delta X is going to 0 that is our objective actually and delta X dot should also go to 0, this expression what you have here this should go to 0 and this should also go to 0. So, let us put it back 0 equal to 0 plus all that and hence delta U S can you can think about I can compute it basically, because this is there and it is not a very good solution is a pseudo inverse involving and all that, but still it in approximate sense I can compute it and as long as my control number of dimension of the control variable are more than the dimensional parameter, which is sometimes true sometimes not true whatever; if that is true this pseudo inverse is not good actually, I mean pseudo inverse is not bad. So, then you can think about I can compute it and hence I can use this cost function, but again wait a second delta P is fundamental nature of the problem, the delta P is not known, ok.

So, even though this expression is true in general, we cannot compute it because the value for delta P is not known actually, so this idea also fails basically. So, now it will look out for some other approach actually, so let us see what this is. Now, for to further algebra will define some of these variables which is typically non-linear systems also, we redefine this state variable, control variable and all that, X is redefine as delta X and U is redefine as delta U like that. So, do not have to keep on talking delta X delta U delta P all that actually.

So, we define it and we remember that when we talk about X, U, P and we are actually talking about delta X, delta Y, delta P here, it typically done in linearization I mean, linearization procedure actually, ok.

(Refer Slide time: 13:39)

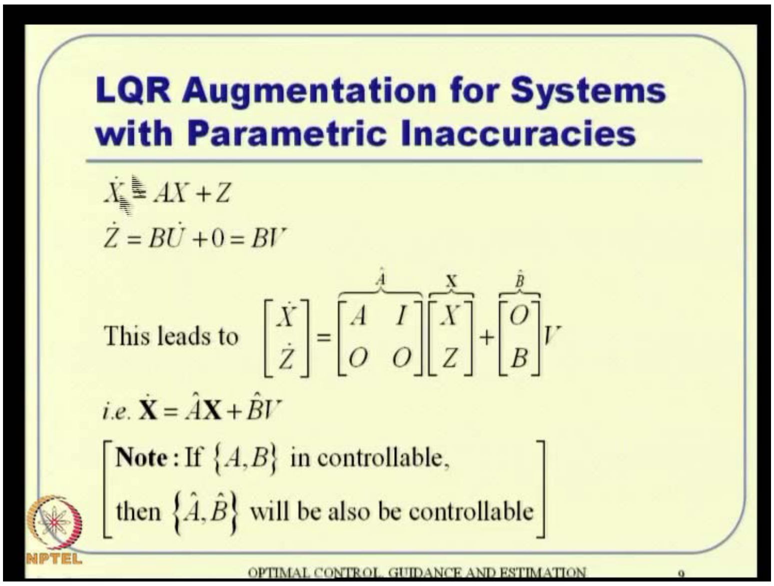


And similarly we define A, B, E as something like A star, B star, E star so that we do not have to keep on talking about this star values also, we do not have to keep on writing star unnecessarily, alright. And after this redefinition, we also define Z as a new variable which is B U plus E P part, remember this is now X dot equal to A X plus B U plus E P in our new definition and all.

So, this part what we have here we define it is Z actually, Z is B U plus E P total actually and we also define some auxiliary control variable V, which is U dot actually and with these redefinitions, the system dynamics turns out to be something like this, X dot equal to A X plus Z plus Z and Z dot is nothing but Z is defined like that; remember P is a constant thing parameter and then E is constant, this derivative will not be there that is 0, Z dot is B is also a constant matrix, so Z dot equal to B times U dot actually, ok.

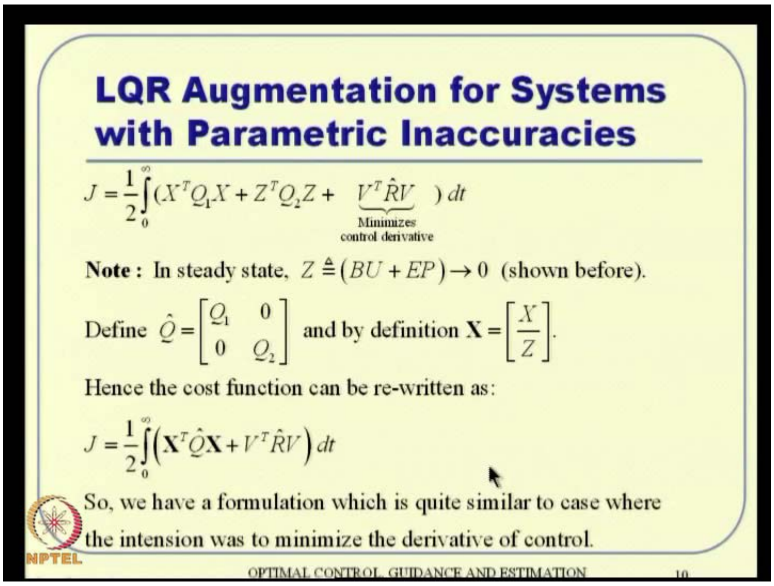
So, but U dot by definition is V, so you can put it has B V. So, X dot equal to A X plus Z and Z dot equal to B V actually; so, this essentially leads to this, you have this X dot and Z dot is nothing but, this kind of ideas. You can put this system dynamics together and tell A is there and then this part is I, Z and X and Z are there and then you have the Z dot, Z dot is B also 0 0 and you have a B actually.

(Refer Slide Time: 14:50)



So, X dot and Z dot is nothing but, A dot X plus P dot B, so that is this. So, that is once I define this capital X now, this thing is forgotten to put that and this is obvious actually, X dot is defined here anyway, so this capital X is this vector X and Z. So, this capital X dot is something this one, this X dot is nothing but, A head times X plus B head times V, again the similar to what we have discussed in the last class, it turns out that if the pair A B is controllable, then this pair A head X also controllable actually, ok.

(Refer Slide time: 16:23)



So, that gives us a hope to proceed further with the LQR design. So, we now formulate this kind of a thing, we have this cost function X transpose Q 1 X plus Z transpose Q 2 Z plus V transpose R at V, because V is a control variable now, but remember actually V is nothing but, U dot. So, we are forced to write because, this formulation new formulation V is the control variable and R has to be there because R cannot be 0, it is an initial positive definite. So, this R term is there, but physically it means that it take actually minimize we attempt to minimize the control derivative also and this is also true in a physically meaning full sense, because whenever we have a disturbance storm system and all, we do not want to kind of make the control variables since of that too much actually.

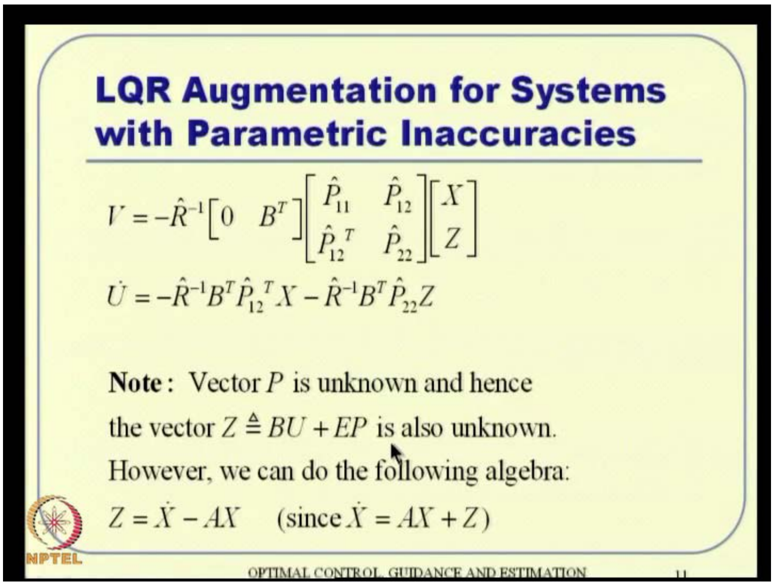
So, we want to have this control rate minimization as well also, that is a philosophical argument, but mathematical argument is because V is a control variable, we need to have some term for V transpose V also and that is whatever it results in the physical interpretation actually. So, we have X transpose Q 1 X and Z transpose Q 2 X, this is just because we have to we can split this and interpret that way, this part is Q matrix largely and this was a R matrix actually ok.

Now, what happens in a steady state situation, now is remember you go back to this, in a steady state situation, this entire term which is now in a in a new notation, it is B times U plus E times B, B U plus E P is has go to 0 there is no doubt about that, that has to happen; just that here we are not able to solve for delta U ss because delta P is not known, over in steady state this condition should happen. So, this condition this is what is written here, in steady state Z has to go to 0 ok.

So, if we define this Q head something like this, partition matrix Q 1 0 and 0 Q 2 like that and by definition this capital X and Z, the cost function turns out to be like this. So, you have a cost function which is quadratic in terms of capital X and quadratic in terms of V. So, we have this standard quadratic cost functions and we need to see this standard linear (( )) dynamics also basically. So, with respect to this standard the equation and this cost function, now it is problem is comfortable actually.

So, we can go ahead and solve it actually ok, but also just remember this is a formulation which is quite similarly to the case, where the intension was to minimize the derivative of control. We have discussed all that in the in the previous lecture actually, you can some of you forget that, you can probably revise that as well as actually, very similar to that the development happens to be somewhat very similar to that. Anyway this is the cost function and this are the steady state linear and this is a I mean cost function, which is quadratic. So, we can go head and use our LQR theory so to get this one.

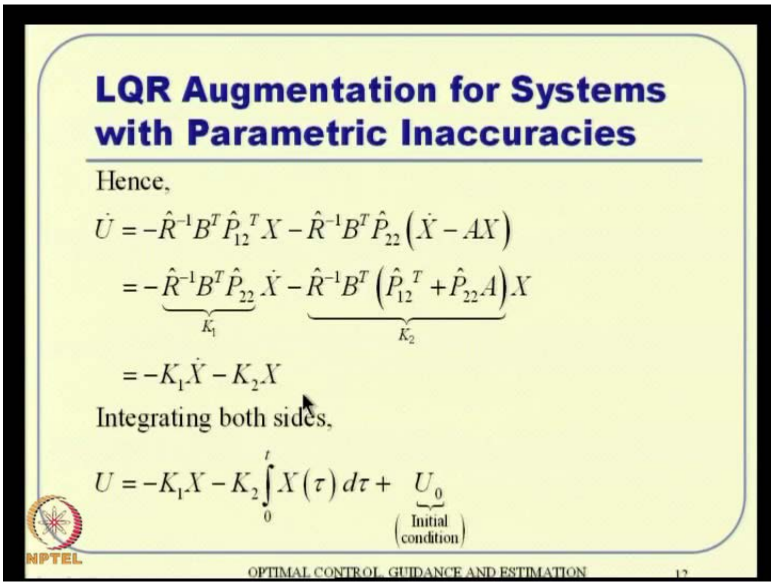
(Refer Slide Time: 19:13)



So, what ultimately my control variable is V here, so it is V equal to minus R inverse R is a R head here, R inverse B transpose B is nothing but this vector, this vector 0 B transpose, R inverse B transverse this one, very knows B transpose record matrix solution P head times capital X, capital X is AX and Z. So, my V is nothing but U dot know, so I can substitute that is U dot something like this and remember Z is not computable again, because we do not know this delta P, the P is nothing but delta P, remember that actually ok.

So, this is P is not known, so Z is also not compatible actually. So, we cannot implement this and also remember this is a dynamic controller U dot and think like that, so we do not want that to implement the actually. So, how do you go about that? Now, the idea here is we know Z, Z is something like this, so what is X dot? X dot equal to let me let me write it here probably. So, we have this X dot equal to A X plus B U plus E P right, so that all over our formulation, so this is nothing but, A X plus B U plus E P that part is nothing but, Z actually. So, if I if I want Z, then Z is equal to X dot minus A X, this is what is what we can do here to compute Z indirectly, we cannot compute Z directly like this. But Z can be computed something like this actually, but remember X dot is probably not available in sense or not advisable to use it also, so what we do it is probably algebra like this.

(Refer Slide Time: 20:56)



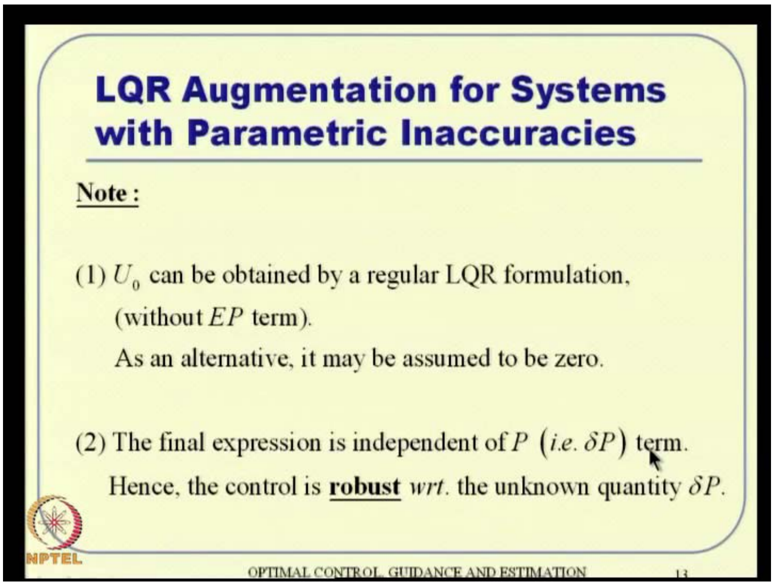
Now, remember U dot is something like this and Z is something like that. So, we can substitute Z is a something like this and then get it as Z is nothing but that, so substitute that and recover the X dot part here and recover the X part here and here, X dot part coming from here and this part and that part can be combined here actually, you can define this K 1 and define this K 2 and U dot is nothing but, minus K times X dot, minus K 2 time minus K 1 times X dot, minus K 2 times X actually, but that is U dot, what you really need is U.

So, you can now integrate both sides and tell this results in a very similar situation, what we discussed in U dot minimization all that and the previously lecture, essentially results in a PI controller actually. So, will K 1 times X minus K 2 time integral of X and all that ok. So, this I mean ultimately we will end of with something like that here K 1 K 2 can very well be computed, that is not a problem and this terms can very well be computed now, because it all involves that deviation delta, remember X is delta X actually and also what we talking here is delta U ok.

So, delta X is available, so delta X integral is also available, I can compute the integral as a numerical computational sort of thing and it is all their actually, the initial condition as we discussed before, we can think that there is no parameter variation to begin with and hence, whatever controller you get that can be initial controller, sometimes people use that as 0 also, but I will suggest that you use typically this, I mean assume that there is no parameter variation and then you I mean you land up with this u naught computation as a regular LQR formulation sort of thing actually.

Anyway these are subject to implementation, once you start implement see which is better which less all that actually, but the whole idea is this control is now available in the form of a PI controller it some sort of in initial condition also, alright. So, this is how we can do that this computation, this is what I have already told, that u naught can be obtained as regular LQR formulation and also remember the final expression what we are getting here is completely independent of P basically ok.

(Refer Slide Time: 23:00)



So, this you can think of that my control does not require P and at there is A all of the good objectives is already there here, because X goes to 0 and Z goes to 0 Z is nothing but, B U times B U plus E P, that is what ultimately should happen, right because you have this expression right, X dot equal to A X plus B U plus E P. So, once X goes to 0, X dot goes to 0 and this term also goes to 0, then we have done B U plus E P. So, that is Z will also go to 0 by this formulation actually because quadratic term for Z also basically. So, essentially this P I controller guarantees that Z goes to 0 and hence everything in same, hence delta X goes to 0, hence the original objective that we started with that delta

X will go to 0 is matrix actually. So, this is how we land up with some sort of a robust controller with respect to the unknown quantity, we do not need the we do not need to know the value of delta P; however, as a compromise we need to know the value of delta X, which is also I mean which is justifiable actually, we most have sensors which should tell us how much delta X is appearing actually.

So, using that delta X in an integral feedback and professional feedback, since we will be able to compute our control, which will be able to do the actually, so that is how we will land up with a robust controller actually ok. So, to summarize a little bit, because it is too much of A argument here, so we started with something like our objective was to I mean we have a non-linear system like this and we already design U star X star assuming a P star value, but P is something different than P star and hence we want to compute a delta U, which will which will augment with U star U star plus delta U, so assure that delta X goes to 0 ok.

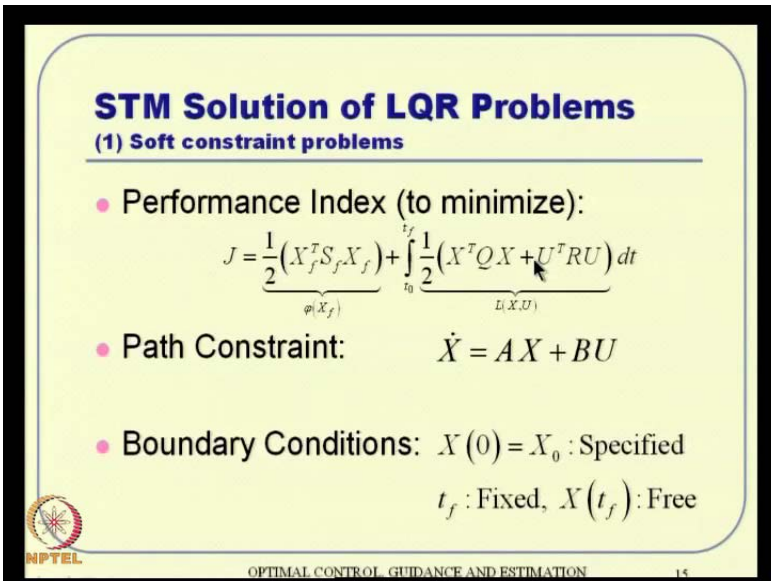
Now, we landed I mean we did this linearization and have some several argument is into work out something like this is not visible like that and ultimately we redefine all these variable for sack of clarity and here will landed up after defining all that, this z is something like this, V is something like the and all, we will landed up some system dynamics like that.

So, using this system dynamics which can be possess like that and then using this cost function, we can solve a regular LQR problem in this form and then we can talk about my control variable is available is that way, but my control is a that V is nothing but U dot. So, U dot is available that way and Z is not available, but Z I can compute it that way, X dot minus A X sort of thing, remember the Z is of same dimension all this which is also nice actually, ok.

So, we there is no approximation involved here, kind of directly compute that and hence A once I put it here, it takes this form and then I land up with this U dot is equal to some gain times X dot minus some gain times A X actually. Then integrate both sides, I got U equal to a proportional term and in an integral term, so I will land up with an integral term PI thought controller and using only that deviation value of X, we can compute the deviation values of U basically.

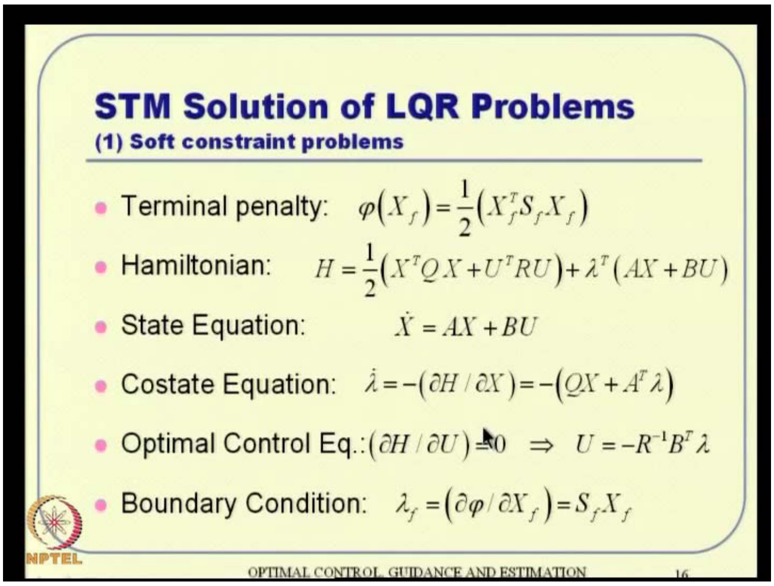
So, this is essentially results in a robust controller, alright. So, this is one way of handling this problem actually, alright. The next thing that I wanted to talk in this class is something very different and this turns out, we go back to the solution approach altogether instead of Riccati matrix approach, we will try to see some of the alternate approach which is typically called as state transition matrix approach alright.

(Refer Slide Time: 26:54)



So, we started we start that idea and the there are two things that as I told, there is one formulation is soft con I mean soft constraint formulation, the other one will be hard constraint formulation. In soft constraint problem formulation, goal or objective is exactly same as what we have been discussing actually, this it should minimize this cost function subject to this path constraint or system dynamic constraint and boundary equations are exactly same. And remember, X of t f is free variable, free means not really very free, I mean it what we talking here is it should remain as most close to 0 as possible depending on what value of X of I choose, what value of P I choose all that, it will derive towards that actually. So, this is the time of formulation that we are talking about here, so how do you have a solution in an alternate setting actually, that is that is the objective here and let us see how you go about that, ok.

(Refer Slide Time: 27:48)

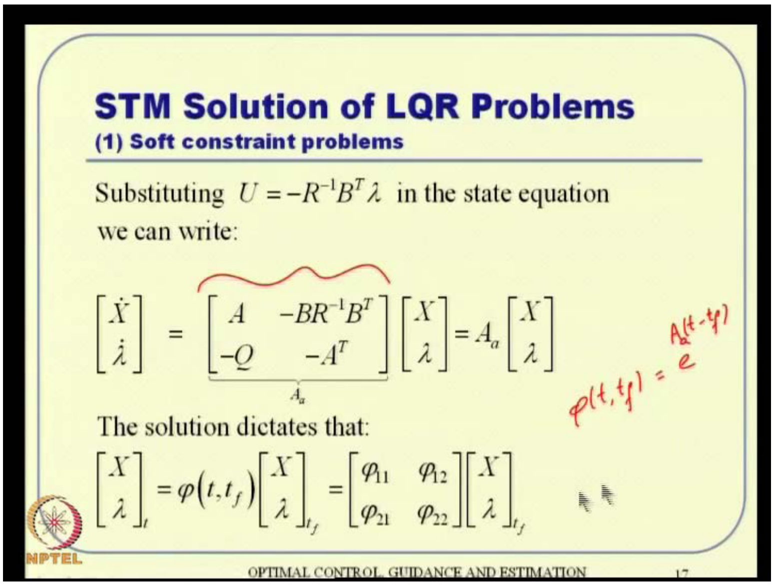


So, we go back to these necessary conditions of optimality and all, so phi is we discussed many times now by now; so phi is like this and l is like that. So, phi is available and hence Hamiltonian H is nothing but, l plus lambda transpose and l is available from here and f is available from here, ok.

So, this is l plus lambda transpose f is available actually. So, the state equation turns out be A X plus B U which is there with us, costae equation is same and optimal control equation is same, all things we have discussed before actually. And after doing all this necessary condition, the very next logical step was to assume that lambda of t is function of X of T linear function of X of T. That means, lambda of T is nothing but P of T into X of T that is what we proceeded, I mean we assume then proceeded with all this lambda dot and then substituted X dot U lambda dot all that and then carried out further algebra to land up with Riccati matrix equations, remember that.

But here we are not going to do that, what we are going to do is something different and also I mean, let us see what is going on here. And remember this U is a close form of solution is available in the as a function of lambda basically. So, this function of lambda whatever you are doing here can be substituted right here, once you substitute here this two function this two equation, X dot lambda dot can be kind of coupled equation to each other basically, X dot is a function of both X and lambda, and lambda dot is a function of both X and lambda ok.

(Refer Slide Time: 29:37)



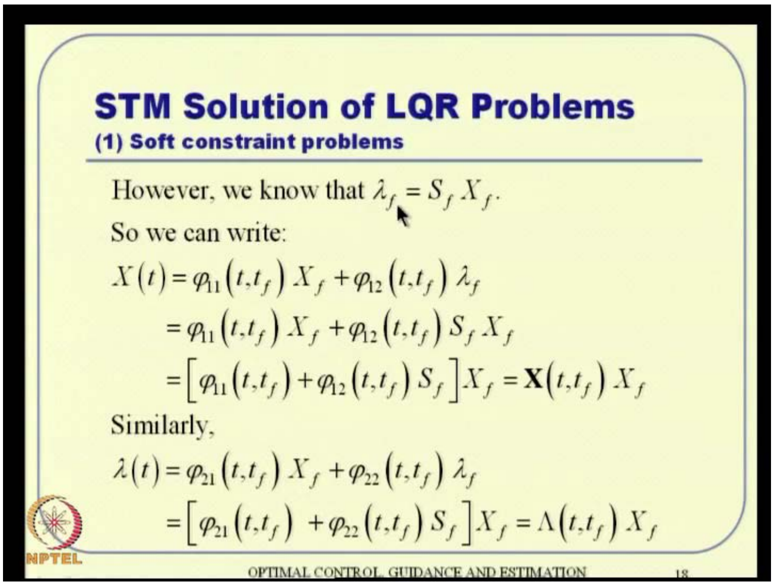
So, that is the type of analysis that you want to carry out for that, so we substitute this control expression and the state equation here and hence you write X dot equal to A minus B R inverse B transpose lambda, that is that part of it and lambda dot is already available minus q X minus A transpose lambda, so minus q X minus A transpose lambda ok. So, X dot lambda dot is something like that, A times this one X and lambda, so it is variable this way. So, this solution dictates, now remember this is actually a linear system equations X dot equal to kind of A X sort of form actually. So, the solution can be written in the as a function of this state transition matrix and all that actually ok.

And if it happens to be a constant matrix, that means, A B R Q all these are constant things, then this phi of t t f happens to be some like A to the power sorry e to the power A t minus t f, that kind of formula. Let me probably write it also, phi of t t f equal to equal to e to the power A t minus t e f, if A means, this A a whatever you are talking this thing A a times t minus t f provided all the entries of the A a matrix are constant actually, we does not we does not very it time, otherwise you cannot write it and you to leave with this state transition matrix actually.

As the special case, this structure goes form that way; anyway solution dictates that X and lambda of any time t is nothing but phi of t t f in all that actually, then what actually. So, remember this is also many times, you might have seen that this is phi of t 0 and X and lambda at 0, but phi is a general thing, that means, state transition matrix can start from any time to any time, so we can take the final time as reference time and then write the solution in the form of t f, not necessarily t 0, we can, so that is the small observation here actually.

And primarily because lambda of t f is available, that is the that is the reason for that actually and this expanded form I can partition matrix, remember this matrix is now 2 n by 2 n matrix, because X is of n dimension and lambda is n dimension. So, we have 2 n dimension in the left hand side and 2 n dimensional right hand side, this has be 2 n by 2 n, that 2 n by 2 n I can partition it to 4 n y matrix and write it something like this actually, alright.

(Refer Slide Time: 32:12)

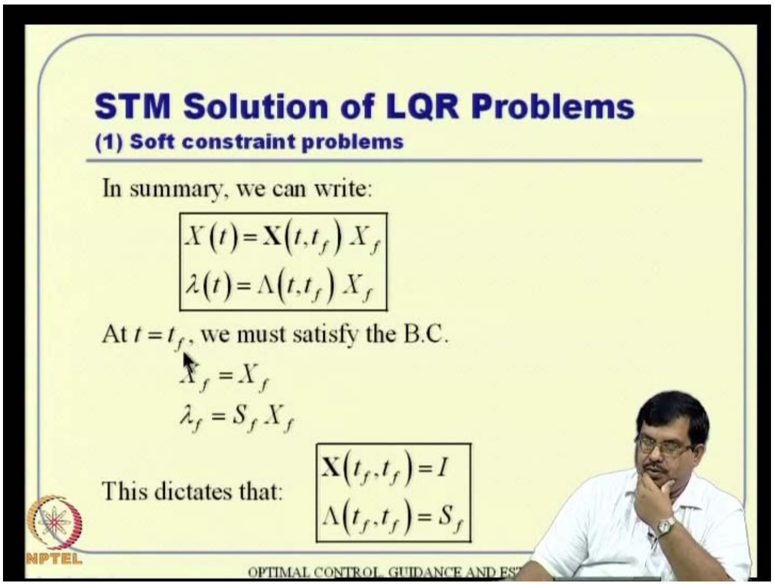


So, now we would to be know that s f lambda f equal to s f X f is a boundary condition, right, this final boundary condition. So, using that final boundary condition, I can I can write X of t from this equation, X of t is nothing but phi 1 1 X plus phi 1 2 lambda. So, phi 1 1 X plus phi 1 2 lambda, but remember this is X f and this lambda f and lambda f is something like this, so I can substitute lambda f is s f X f basically ok.

Once I substitute that, I know this is X f s f multiplying to the right and put the expressions. So, I can take out this common and then this turns out to be phi 1 1 t f plus phi 1 2 t t f into s f X f actually. So, this entire matrix what you are getting here, I can define it as some capital X t f, remember this is not argument state vector know, this is actually state on this matrix with transitions this X f to any other time X t basically ok.

So, if you know X f value, then you can calculate X of t value using this thing actually, if you know all this entries. So, for this is still a symbolic expression, we do not we are not talking about how to compute all that, in second I will talk about actually. Similarly, you can you can go back to this lambda of t expression here and lambda t turns out to be phi 2 1 times X plus phi 2 2 times lambda actually, so phi 2 1 times X f plus phi 2 2 times lambda f actually. Again you use this lambda expression, s f X f and then define this entire thing is something like capital lambda t f, we have lambda of t is nothing but this capital lambda times X actually, ok.

(Refer Slide Time: 34:01)



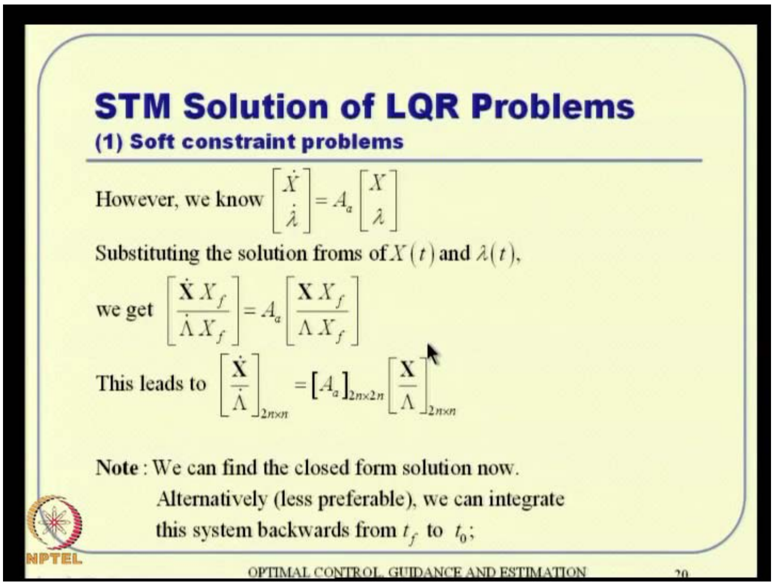
So, this X of t is this way and this lambda of t that way actually. So, this what you summarized here X of t we got it something like capital like X of t t f times X f and lambda of t we got something like capital lambda t t f times X f actually ok. So, at t equal to t f, we must satisfy the boundary condition as well, remember that and what is the boundary condition we can think of something like this, X f of is equal to X f well, we can think about the real expression, but it is true anyway, but lambda f of has equal to s f X f that is also true, these two must be true actually.

So, using this two, what we getting putting it (( )) here, what we getting here is X of t f or X f is equal to this time t f t f into X f, X f of t X f is equal to X of t f t f into X f and similarly lambda of t f and lambda of f is equal to this capital lambda of t f t f into X f alright.

So, and this lambda of f nothing but that basically, s f X f has to be if you consider that this is s f X f is equal to lambda of t f t f times X f. So, if you use this expression, so you can easily see that from this expression what you are having, this capital X of t f t f is nothing but identity, because this expression has to be identity and this from this expression you can see that lambda of t f t f has be equal to X f. So, we have the boundary condition for these two state transition matrix, you can this is state transition matrix for X, starting form X f and this the state on matrix for lambda I mean lambda starting from again same X f only, ok.

So, the boundary conditions are now available, what about the differential equation, is that the still available? It turns of out to be yes, because this is if you put it wake you are I mean expression and all it turn not be like that, let us see that. We know this is true, the X dot lambda dot are given something like this, A times X lambda we just derived that, if you substitute by this is how it is actually.

(Refer Slide Time: 35:53)



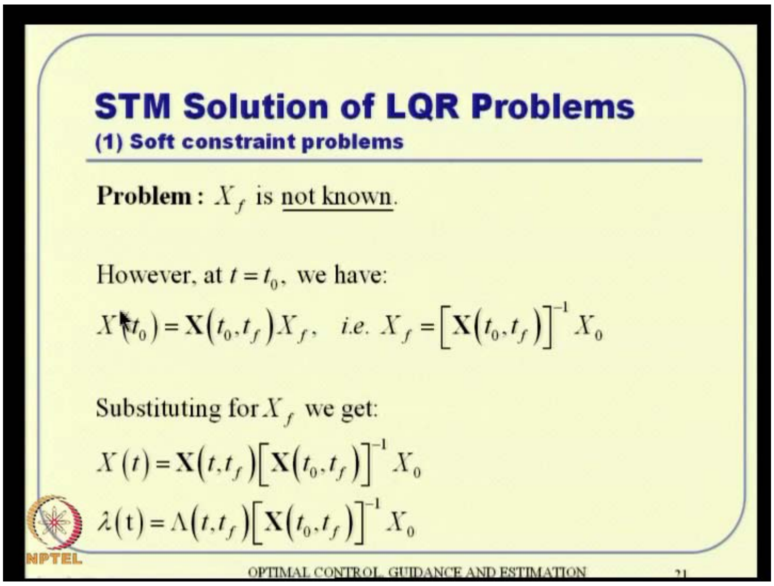
So, we have this X dot lambda dot is A times X lambda and we know this A this expression of X t and lambda t now, this is X t if we know this, lambda t which is know that actually. So, if we substitute it with whatever you know and remember X of f is fix quantity, this is not time varying quantity X f is a number final actually, fix vector. So, if substitute this X dot nothing but, this capital X dot times X f and the lambda dot capital lambda dot times X f and right hand side is same thing actually, A times this X is this capital X times X f and lambda is capital lambda times lambda X f.

So, this leads to like conclusion that, this dot that X dot and lambda dot has to be A times this matrix actually, so that means, the state transition matrix is a dynamic variable, but the differential equation for that is known to the actually and remember this actually matrix, this is a matrix, so this also matrix. So, what you having here is matrix differential equation actually, but this matrix differential equation is a not non-linear differential equation, it is actually differential equation. I mean linear differential equation, it exactly satisfy the same differential equation that state and course satisfy that is the property of state transition matrix as well, if you if you remember that actually.

So, this state transition matrix is a is a linear differential equation and so, you avoided this non-linear sort of record matrix all that and this differential equation also we know the corresponding boundary condition actually, so we can solve it. So, we can solve the you can find the close form the expression solution now, because once you have this numbers known to you, this differential equation known to you, as long this are not time varying it should be exponential solution. If there are time varying you have to see, if you if you have a expression explicitly available has a function of time and then you can probably put that and try to still get some close form solution typically and if it is not explicitly available, but most of the time it is implicitly available anywhere. So, for example, if you are mass is burning out and something is there, which will give you some sort of a numbers only, but it does not give you X explicit formula for phi, then you cannot talk about close form solution for say basically.

But you can still integrate the system backward from t f to t naught, that is actually that is always possible; numerical integration from t naught to t f is always I mean t naught to t f that always possible, but that is naught a measure motivation here, we wanted to have a linear form, so that most of the problem we can actually do a close form solution and many times if the if the everything is time invariant, that is state is state matrix sorry I mean system matrix A B are time invariant and Q and R by choice are also time invariant, then A A matrix has to happen to be time invariant and then you can solve a close form solution in the function of matrix exponential actually, ok.

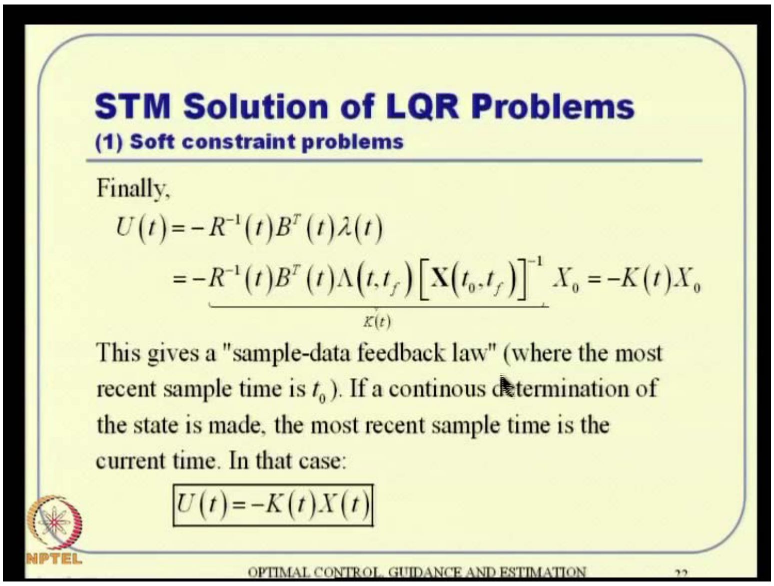
(Refer Slide Time: 39:16)



So, that is the motivation with corresponding boundary conditions known to us, so that can be solved actually. Only small issue here is we have this everything in the function of X f, but X f is not known remember that, X f is free actually what we know is rather X 0, initial condition for the state is known, final condition for state is not known. So, how do how do kind of solve about that problem. So, we know that X of t is nothing but phi I mean this capital X of t f X f, so X of t naught has to be capital X of t naught t f X f and also remember the state transition matrix is typically notable, their they do not there are not singular matrix never actually. So, we can talk about inverting this actually and get it this one ok.

So, essentially the idea here is you have this boundary condition and you have this equation, you propagate it or have solution whatever it is and you ultimately land up with this expression which is available to you, t naught phi this capital X of t naught t f will be available to you and because you know this actually, so you can compute that also because this is always invert. So, you substitute for X f now, because X f where ever you have this X f term and all you can substitute that.

(Refer Slide Time: 40:29)

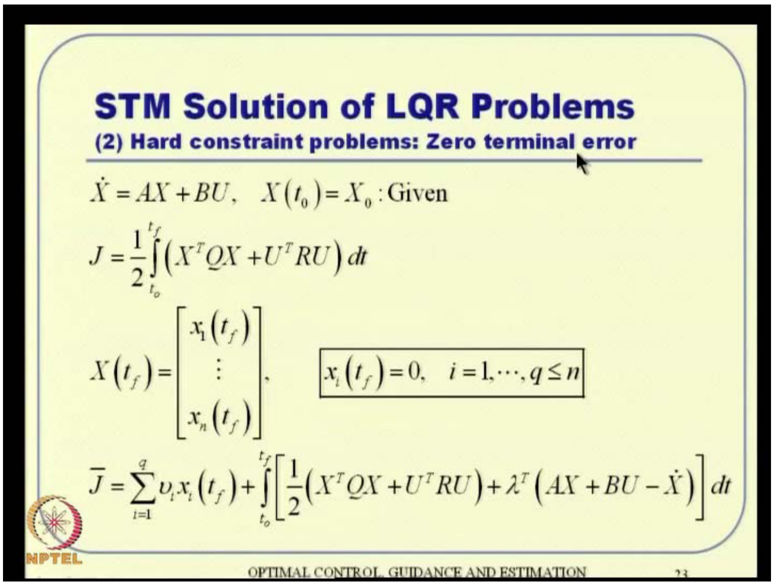


So, you can substitute that and then finally get a solution for this control. So, what you will see that substituting for X f we will get this expression, this was X of t remember was X of t t f times X f. So, X f is now given a something like this, so I can substitute that and lambda of t is lambda of t t f times X f and X f is this one, so we can substitute this one, because now lambda of t is available, I can talk about a control available. So, control is minus r inverse B transpose lambda and lambda is available now here. So, I can substitute it back and tell this is my expression now; remember that lambda are coming from here and I kind of combined all the thing and that is my gain matrix k of t actually ok.

As a small comment, this k of t will happen to be same value numerically, whether you come this way or you come from Riccati approach, Riccati equation approach, you can take any example and verify yourself also. So, this LQR problem typically admits and unique solution anyway, so it will you will essentially land up with some sort of a solution, but coming from a different expression actually.

Anyway, so this is available now, U of t is starting from X naught it is all available, but X naught can be your value at any point of time. So, this are something called that sample data feedback law and all that. So, where the most recent sample time is t naught and hence X naught can be can be replaced to X of t, at any point of time that you are there you can consider that is initial time, for rest of your time actually that which is which is typically true basically. So, instead of X naught, you can substitute that as X of t and then you have got k of t, so you can write it that way. So, U of t is nothing but minus K t times X of t actually, where K of t is computed that way.

(Refer Slide Time: 42:26)



Alright. So, this is all about this in state transition matrix approach, but these are all for soft constraint problems actually, now what about hard constraint problems? Now, this hard constraint problem, what we are looking for is 0 terminal error and sometimes these are very appealing, it is some difficulty also, but these are sometime the formulations since appealing. Especially, suppose let us say you talk about emission guidance problem, and then you want to land up with zero terminal error, zero terminal resistance actually.

So, the hard constraint turns out to be little more appealing than soft constraint, we cannot talk about falling somewhere close to the target, but we want to fall on the target basically. So, that kind of idea is there, can we do that. So, let us talk about some sort of similar formulations here, let us talk about X dot equal to A X plus B U, again the same linear system dynamics with same quadratic cos function actually and purposefully we avoid this additional term here, this terminal penalty because what we are talking here is a hard constraint penalty; that means, X i of t f has to be 0 where i is 1 to q, where q is less than equal to n.

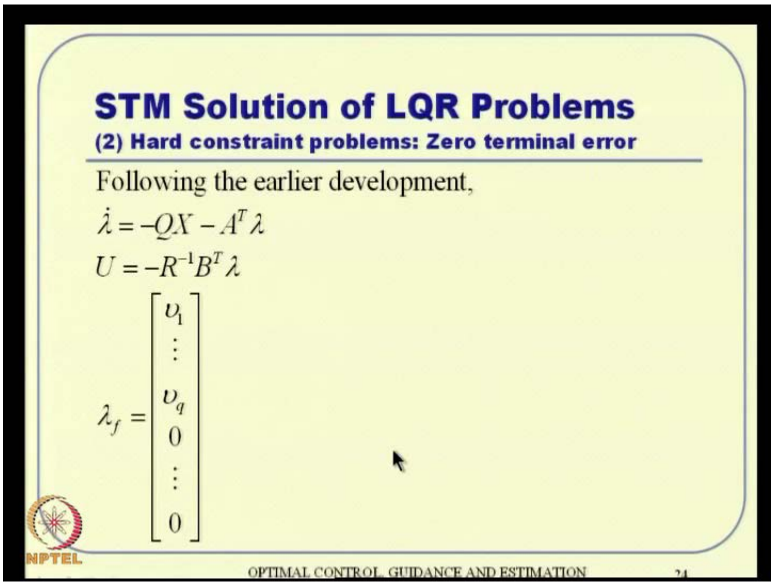
So, in other words, you can have constraint on all straight variables, if you want to or you can have part of the state variable that you are interested in, other part you forget it actually for example, if you have again talk about initial guidance. So, you talk about position error only, then velocity errors and all you can forget about it or sometimes this angle constraint guidance are there, then it will be part of the velocity vector it contains the velocity magnitude as well as two angles actually. So, magnitude you can forget, what can we say angle constraints actually, that way. So, those kinds of problems can be discussed and can be put framed in this actually.

Alright, go back to that and then talk about X of t f is nothing but, X 1 of t f to X n of t f, where this constraint is given something like that X i of t f is equal to 0 for i running from 1 to q and q can be less than or equal to n, it can be equal to n also basically. So, how do you handle that, when you have a hard constraint like that, remember that should be equal to 0. So, the j bar are augmented cos function should see that constraint and that constraint is seen through this expression, summation of this variable is equal to 0 anyway, this new are like Lagrange multiplies, but these are constant numbers, they are not time varying things actually.

So, this additional expression is coming to picture 1 to q obviously, and the rest of the things are free anyway, plus this augmented cos function t naught to t f with this one and that one. And also remember this formulation makes more sense when you have t f as a finite time formulation, because we are talking about some state variable going to 0, at some point of time, we do not talk about it will go to 0 at infinite time and all that, and that does not have too much of physical meaning actually, but typically these formulations are finite time formulations actually.

Anyway j bar is something this one and then this l plus lambda transpose f minus X dot we are talking about j bar, remember it is not Hamiltonian j bar; now this is this function plus all that plus lambda transpose, lambda is a Lagrange multiplier, new is also a Lagrange multiplier, lambda is a time varying function, where new is not time varying function.

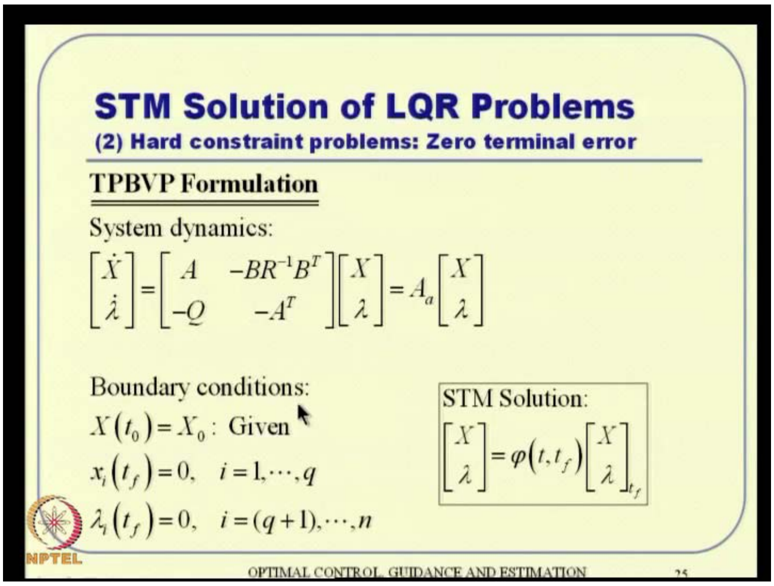
(Refer Slide Time: 45:58)



So, now we have some formulae I mean some augmented cos function available to you, so we can implement the follow the earlier development of co state equation, optimal equation like that. So this co state equation because at the for the expressions are not changed only the boundary condition is changed. So, the dynamic equation what you have state equation course and optimal control they will not since, so you can a variety that way, lambda dot is minus q X minus a transpose lambda and U equal to minus R inverse B transpose lambda and lambda f; obviously, you can calculate from here, this is lambda f actually, it will come from here del phi y del X of sort of thing.

So, if you talk about this, take this expression and in operate the del phi by del X of sort of thing, then you land up with this expression lambda f is nothing but, nu 1 to nu Q coming from here and rest of the terms are not there, X i X i when it is beyond I is beyond Q is not there; that means, for those variables it is 0 basically partial derivative is 0. So, lambda f del phi by del X f, phi is something like this, we carry out the algebra, we will land up with this expression, where nu 1 to nu Q will be available, rest of the things will be 0 actually. And when Q equal to n, then it will run through the entire vector that is also there, so but if Q is not n, so it will stop somewhere actually. So, following the earlier development what you can what you can see is, like this lambda dot is Q minus Q X minus A transpose lambda, U is all that lambda f is nu 1 to nu Q.

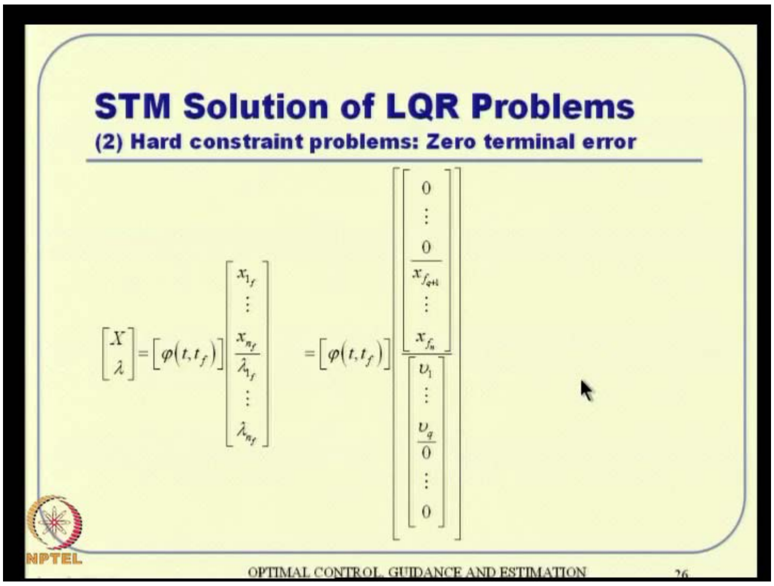
(Refer Slide Time: 47:32)



So, the two point boundary value formulation talks something like this, it I mean it has this standard expression, which I have already derived basically before, just put it your control variable U and then consider X dot a lambda dot together, we have done that just before actually. So, this is your system dynamics and what are our boundary conditions now? The boundary conditions X of t 0 X 0 is always available and that is given, but at time t f this is constraint set and that X of t f has to be 0, for i running from 1 to q and the rest of the value for that lambda of t f has to be 0; remember this is coming 0 0 and this guys are not known this nu 1 to nu q are not known, so it is foolish to kind of have a formulation which uses numbers for that, this actually helps us in finding a solution, but we cannot actually talk about a solution, where you need this number information for this, we do not want that actually. So, what is available to us is a boundary condition in terms of X from i running from 1 to q and lambda of t f equal to 0 nicely coming from here. So, you have n boundary condition 1 to q in that way and q plus 1 in this way ok.

So, this has to be accounted for actually, but anyway coming back this is the system dynamics what we had earlier. so we are following the state transition matrix approach we can always write it that way, X and lambda of any point of t is state transition matrix t t f times the final time value X and lambda t.

(Refer Slide Time: 49:05)



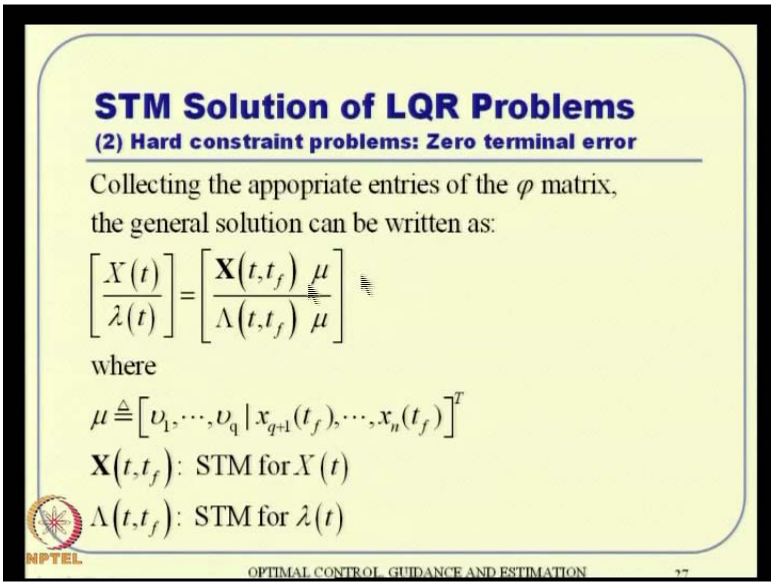
Exactly same as what we have done for the soft constraint problem actually, if we go back and then find out all these things are I mean compatible because software, this soft constraint and hard constraint the dynamic equation remains same, the solution form remains same, but the boundary condition is different and hence the solution will be different actually. So, we have to concentrate more on the boundary condition, how do you account for and all that. So, as far as the differential equation is concerned, this differential this state transition matrix will satisfy the same differential equation that we have studied earlier.

Let me recall that, this is the state transition matrix differential equation that we discussed we will follow that, but the boundary condition will be different, let us see that (Refer Slide Time: 49:30). This has to be accounted for, so how do you account for the boundary condition now? Because lambda X and lambda t t f is given like this, we can we can write it that way this expression, right. So, we can write it X 1 to X n and lambda 1 to lambda n, but it turns out nicely that, if you consider this straight transition matrix I mean this is just expanded this one, but this expanded form, you remember from for first 1 to q, this are zeros and there is something for q plus 1 to n and I mean we know that actually this are few variables. So, there some value for that, but on the other hand, for lambda's is we know this first to1 to q is to be mu, because this one and rest of the things as to be 0 basically.

So, we have we can think about X of X of lambda is something like this, which can be expanded all way like that and nicely is you see this is 1 to q and then q plus 1 to n. So, essentially 1 to q and q plus 1 to n means, 1 to n is available, some values rest of the things are 0 anyway and this is a 2 n dimensional matrix, I mean vector, this is also 2 n dimension thing and this is 2 n by some sort of thing; lot of this 2 n dimension I mean 2 n elements the first give was 0 here and last this q plus 1 to n those n minus q variable those are the there actually.

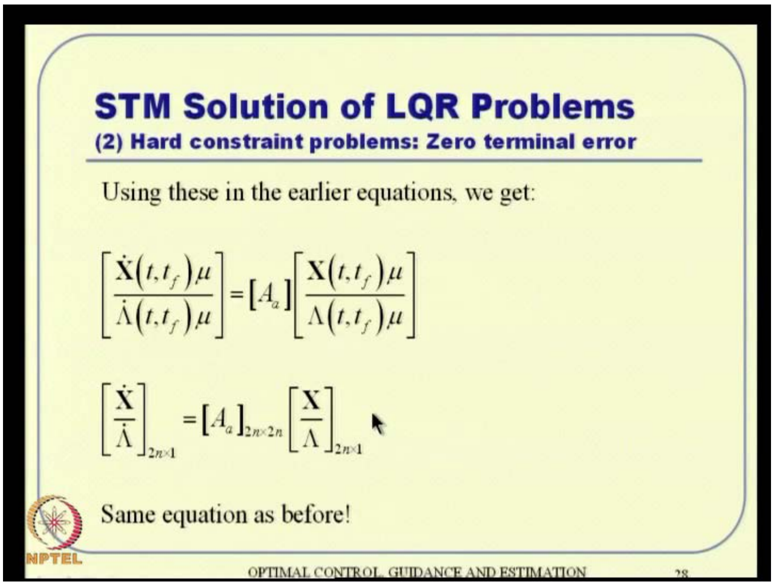
So, whatever is non zero we assigned that, we define that something like a mu, mu will define 1 to q coming here and then q plus 1 to n we put it next to each other and try to rearrange the terms actually, it is always possible, we can write in expanded form, right. Phi 1 1, phi 1 2, phi 1 3 also some sort of phi 2 n by 1 and then you can put a big matrix for that and then take the corresponding elements, that is relevant to this one and then you take the corresponding element that is element to that one and try to put it together sort of thing and then will have something like this, X of X t lambda t is something like this actually ok.

(Refer Slide Time: 51:37)



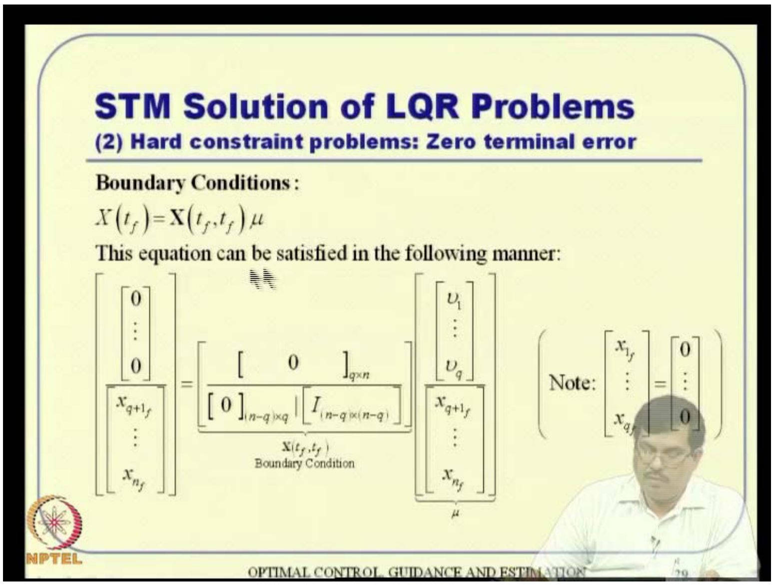
So, collecting the appropriate entries of the phi matrix, the general solution can be written something like that actually, where this is true. So, may want to make you and all sort of things are true and then we have this system dynamics which talk about this X of t and lambda t can be written something like this, which means state transition matrix, state transition matrix here not in terms of X f, but in terms of mu, where mu is something which is non zero basically, that we put it there.

(Refer Slide Time: 52:19)



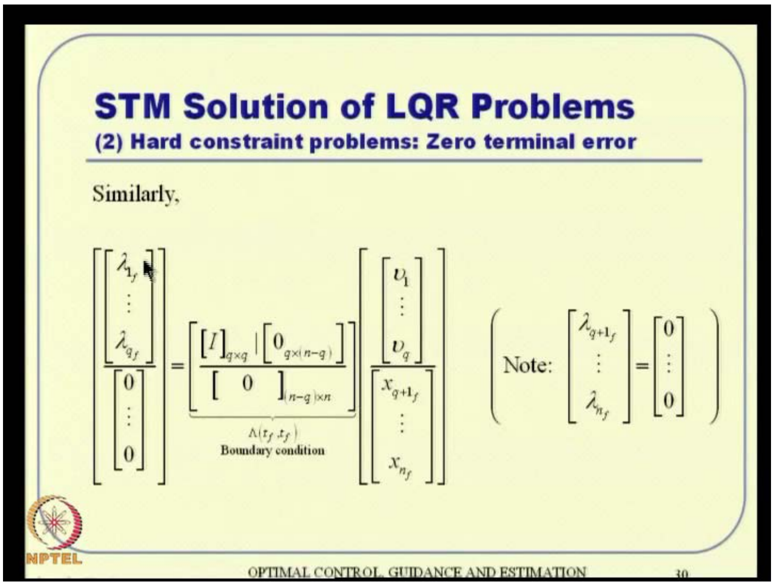
Now, how do you compute mu, that is the thing and then before that we have to differential equation I mean we need some differential equation for that and we need some boundary condition for that. As per as differential equation is a concerned, we can put this solution, whatever solution is here talking here, wake in to the differential equation original system dynamics and put it there and tell mu is non zero and hence the coefficient should be true and that is how we can get differential equation here. So, this is exactly same equation as before, the differential equation part of remains same, what is different is the boundary condition and boundary condition can be thought of putting like this, remember the boundary condition, I mean the expression is like this, X of t f is something like this X of t f times mu.

(Refer Slide Time: 52:52)



And mu is something like this, we define it nu on to nu and then X of q plus 1 to X of n. These are free variable these are also free variable, but this is non zero values, so put it there and then you see that X of t f, what is my boundary condition? My boundary condition tells that, first the first q element has to be 0, then q plus 1 to n we do not know, this is free actually. So, put 0 0 and then this, and then the right hand sides like that, if you really want match it what is happening here, then the this you can partition the matrix like this way and that way. So, this first q by n entries as to be 0 because the ultimate result has to be 0, no matter these are non 0, but result has to be 0. So, they have to multiply by 0 and then we have this part, which is equal to that. So, this is not contribute, that means, that means first n minus q time 2 that element has to be 0 and this should be equal to that, this one should be equal to that, so that is identity matrix.

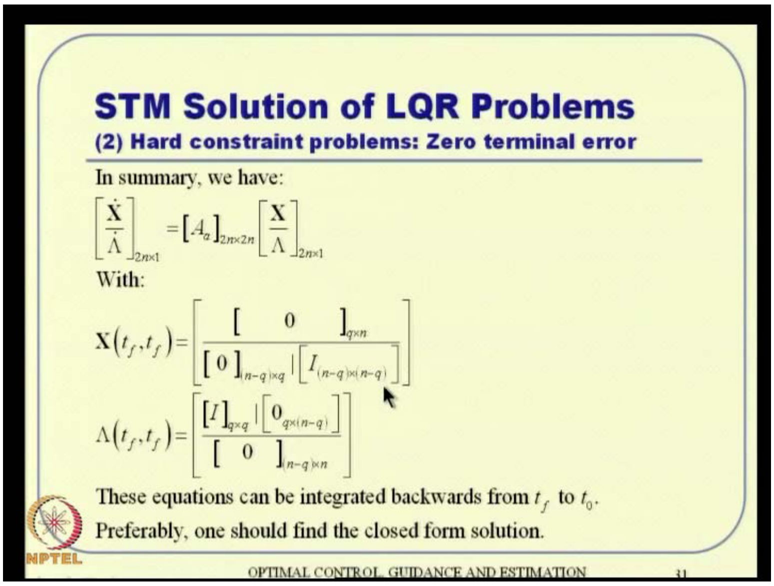
(Refer Slide Time: 54:13)



So, that is how we get X of t f t f, this is what the matrix that you have to take in to account, but you also need lambda of t t f I mean t f t f. So, similarly exercise you can do lambda as well, you put lambda 1 to lambda q and then 0 is their actually X of t f. So, lambda of t f and remember, lambda of t f we discuss we derive somewhere here, lambda f of lambda as 1 I mean 1 to q and then 1s and 0 actually, so you put their ok.

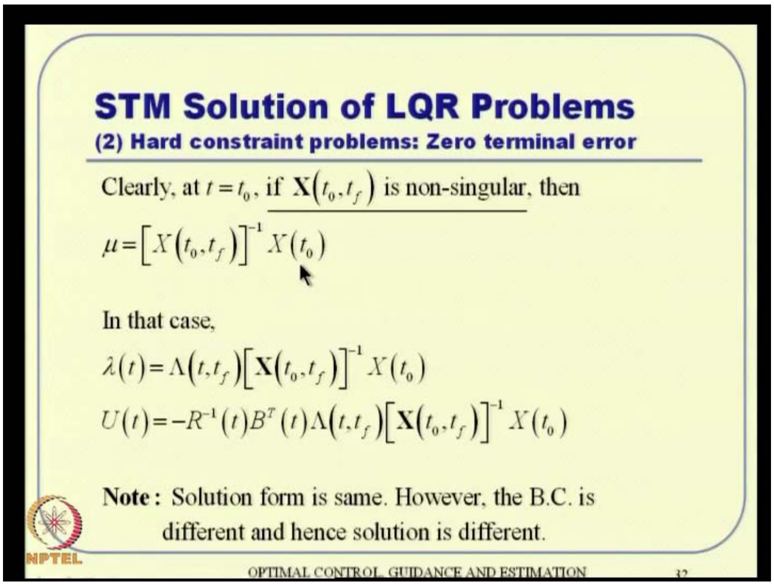
So, that is our expression lambda 1 to q at final time. So, lambda 1 f to lambda q f then once of 0 actually and right hand side is our mu vector and mu vector is defined that way that is our definition. So, we put it that, also the same expression similar analyze that this part has to be 0, the bottom half of the this matrix has to be 0, bottom not really half, but this dimension n minus q by n that kind of thing actually, these are n minus q dimensions actually, so this has to be 0, but this has to be equal to that actually, we know that that is the boundary condition ok. Because this boundary condition is to be satisfy, so we have this one equal to that and hence we have this identity matrix actually here and this will not be contribute use so that means, that is 0. So, we have this lambda of t of f is equal to that and X of t f has to be equal to that and the differential equation is also available to us actually, now we can have close form solution actually, alright.

(Refer Slide Time: 55:45)



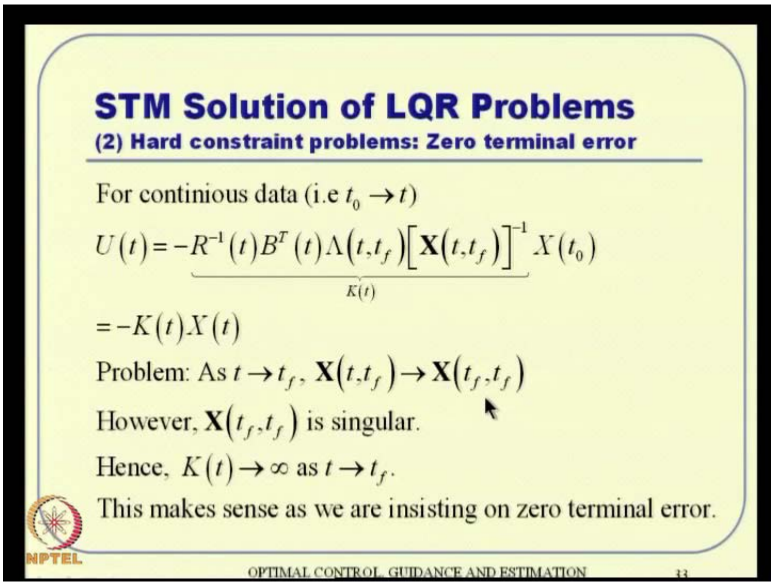
This is close the form solution, this is what we have here and this is what move, I mean this is boundary condition for that, so we put it. Essentially we talk about this is differential equation, but the boundary condition are not same as what you had it in the soft constraint formulation; boundary condition has to be something different actually. X of t f lambda t f t f, using this boundary condition and this differential equation, the procedures exactly remains same as before. And now we cannot guaranty that, X of t is t f is non singular and that is another observation here, it is part of matrix the final condition the bunch of rows are 0 and both the both the boundary condition actually.

(Refer Slide Time: 56:02)



So, may the work late, ultimately even if you integrated backwards and all that, it may or may not then be the case, this matrix is non singular, but if it non singular and then mu can be computed, remember this times mu is equal to X naught actually, right. So, mu can be computed and hence we have this solution there, so then our control is given like this, exactly same as before and again the sample data system, that wherever you are initial condition is that, so t naught can be current t, then it can be argued that this is nothing but your gain matrix actually ok.

(Refer Slide Time: 56:56)



So, just a small little more observation, for continuous data and all the that, that is what I told you, it is has to be t naught, t naught goes to t and then we have this gain matrix all that actually, but only problem here is, when t goes to t f this X, this state transition matrix X go to t X of t f right, here is t goes to t f. So, this is as by definition is go to that, but if you see, if you observe X of t f t f, what is a of X of t f t f this matrix and this matrix is guarantee to be singular, unless q is equal to n, that we do not know and unless we put a very I mean constraint in every state and all that, that can be assuming for the difficult problem anyway, but we put that is the different case, but it otherwise it is guarantee to be non singular, it guarantee to be singular basically.

So, we have this difficulty coming of here basically, X t goes to t f you are guaranty to get some sort of a infinite gain basically; that means, your control is guaranty to flow of actually. So, this is not always advisable to you have a very high ambition has 0 terminal and think like that, if you have then be careful, it may lead to control solution which may not be implementable at the final time work or close to the final time actually, which typically happens in p n guidance missile dynamics also, that we see later also actually. Anyway, so this makes sense as we are insisting on 0 terminal error, you are insisting on hard problem, so we will end of with typical to their actually, so alright.

So, that is what I wanted to discuss in this lecture, in other words, we saw two formulae's two things here, one is the robustness thing, how do you tackle some parameter inaccuracies and all, the second thing was how do you come up with alternate solution of through this state transition matrix ideas. And there we discuss about two ways, again one for hard constraint problem and one for the soft constraint problem; soft constraint problem is not much of a problem, we can go and do that, hard constraint problem we land up with singularity, I mean not initially towards end actually when t goes to t f is guaranty to singular actually.

So, K of t has to be I mean, typically flows up to infinity actually. So, keep that mind and application all that using this ideas and all, we will we will see that in other class actually, another lecture, with that I want to stop here, thank you.