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

Module No. # 14

Lecture No. # 33

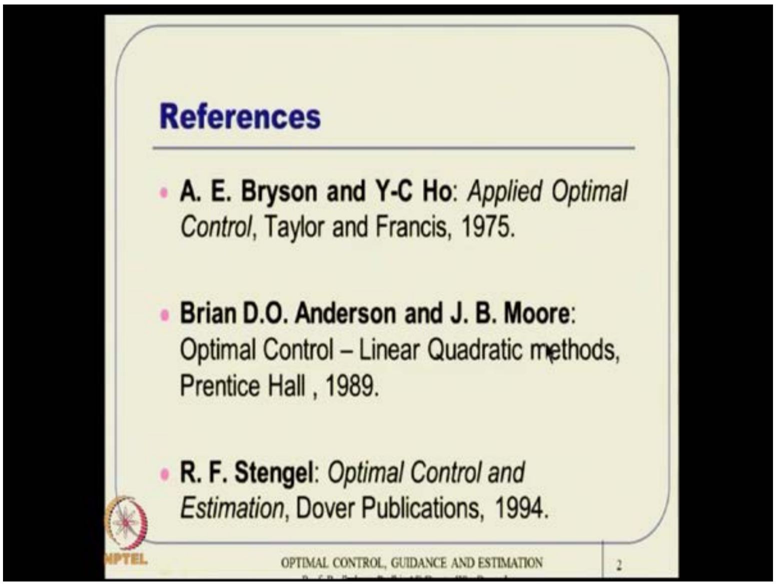
Optimal Control Guidance and Estimation

LQG Design; Neighboring Optimal Controls & Sufficiency Condition

Hello, everyone, we will continue with our lecture series on this course, "Optimal Control Guidance and Estimation". Up to last lecture, we have talked estimation and then integrated design approaches and things like that and on lot of advanced topics like what is happening in the in the research community and at the moment , but then again in this next couple of lectures, we will go back to the classical legacy material thing like that. Here, we talk about various conceptual things in the text book like that. One thing that comes to mind is L Q G design, which is Linear Quadratic Guassian design. I have talked little bit on that before, but we will more systematically talk along with examples from a text book also very basically.

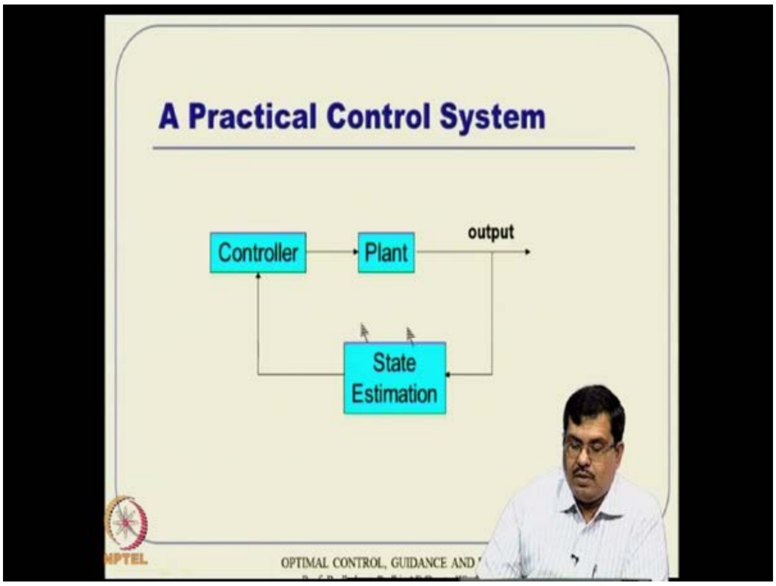
And then we will also study something like neighboring optimal control, which if given an optimal path already, then how do you find out another optimal path, which is close to that . So, that is neighboring optimal control, and then along with that sufficiency condition as well. Sufficiency conditions are classified in various categories like weak sense and strong sense and things like that. We will just summarize the results in the weak sense. In strong sense we will need this weak…, Even in weak sense, if you really did not understand the details of this design I mean sorry details of the analysis there it talks a lot of mathematical tools and all that, we will try to avoid, but rather try to understand the summary of whatever the condition what the condition says .

(Refer Slide Time: 01:48)



These materials are taken from these books. First thing is a classical book as I told in my first lecture. This is one book, which is heavily referred in lot of papers. Probably, the most referred book ever in control theory; entire control theory, basically that is my guess. Some topics are also derived from Anderson and Moore, especially for linear quadratic methods; very rigorous books, it talks about lot of this time domain as well as the frequency domain analysis and all that. So, somebody is interested, you can read that. Also, a little bit concepts can be seen in optimal control estimation from Stengel and also one small concept, some example problem that I have taken from Francis book, I did not include that, but you can see that same thing in other books as well. Anyway, this first thing is this concept, robust control design through L Q G concept, linear quadratic Gaussian concept.

(Refer Slide Time: 02:52)



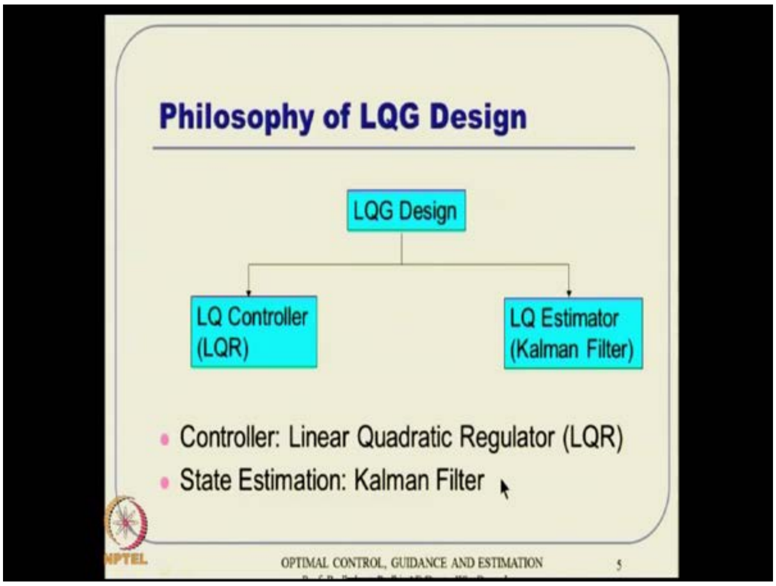
So, this is very simple like what we know in plant operation is we have a controller and we have an output, various sensors and all and so far we have been assuming that everything is available, so we directly feed it. So, we sort of kind of bypass this state estimation thing and then directly feed it to the controller and operate. Now, the question is outputs are noisy and also plant can have some process noise and all that. Now, the whole idea is how do we tolerate it or how do we design more and more robust control and all in that way.

So, one idea that comes to mind is why not you put estimation in the loop. we have a state feedback controller that control requires a information anyway, but in case of instead of directly measuring all the states or whatever doing some algebraic manipulation from the output and feeding it back all the states, which is noise anyway, why do not you put an estimation in the loop, which will not only filter out the output noises or sensor noises and also try to kind filter out the plant noises. So, then I will feed that information to controller and get much better performance.

So, that is the whole idea of how a controller is typically synthesized in practice. And even if you think that I do not need to kind to design and estimation and things like that estimator, then it turns out that the sensors that you use, typically they have their built in estimator. The sensor output what you are getting in let say in inertial navigation system and things like that they themselves have filters inside this instrumentation package basically.

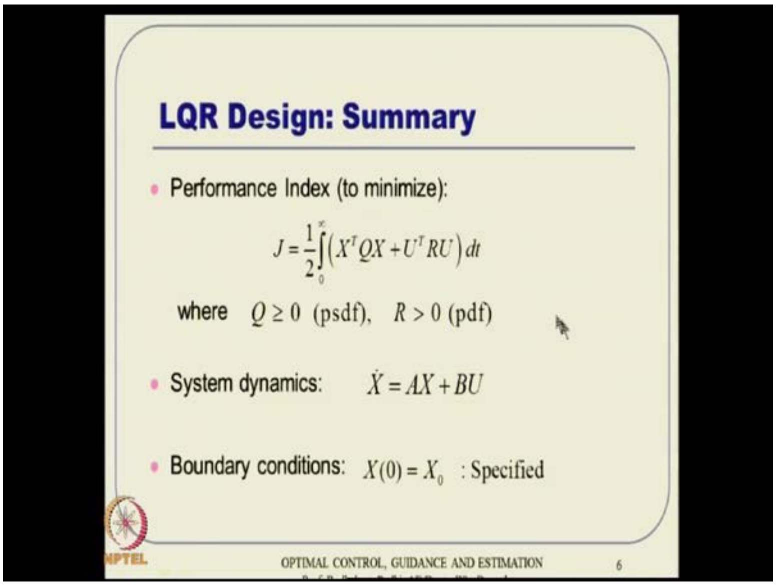
So, we do not see that explicitly and we do not to design it as long as you are concentrating on control part of it, but they are also part of the problem, part of the plant in general, part of the overall system basically. So, without state estimation in the loop probably the entire thing remains the open loop anyway. In other words, if you really do not have any state estimation, you do not want to get over with that, and then it may not work also. So, this would be as important as that what I mean. If you really want to have a complete system either you have a built in estimator, which is typically do not see or explicitly you have to design an estimator and then put it back into the controller . So, that is how it will operate.

(Refer Slide Time: 05:15)



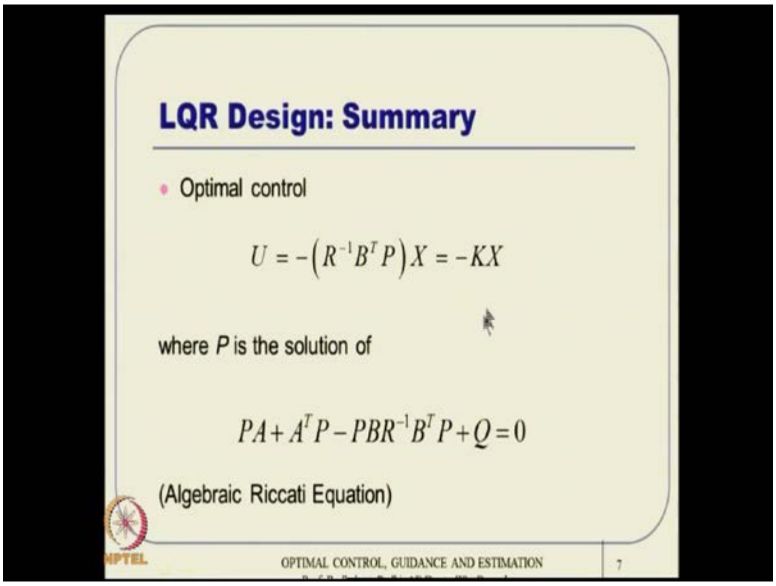
So, the philosophy of this L Q G design is simple, in fact. It has two components, one is L Q controller; typically synthesize through L Q R design and then there is L Q estimator typical designed through Kalman filter and here we talk on linear systems only. So, L Q controller and L Q estimator are everything. And essentially when you I mean there are various extensions of L Q R, as we know before are including some tracking of problem and all that, so this gives us some sort of a good platform to have some I mean fairly good robust control design in the loop sort of thing. So, the bottom line is the controller is designed using L Q R synthesis; the estimator is used I mean design using Kalman filter synthesis. So, these two together and then you have got this design (( )) sensor essentially. Lot of other various things and all, you will find in the book, in many books, in the different themes, but essentially this the key component.

(Refer Slide Time: 06:25)



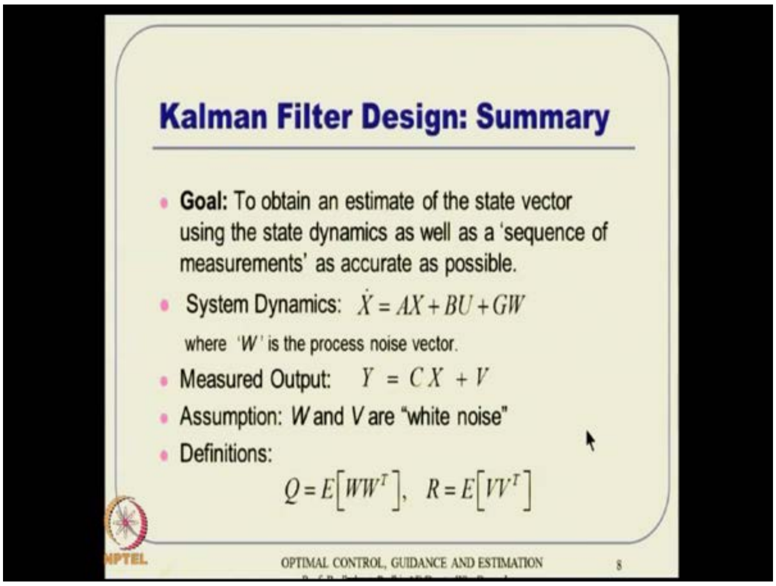
So, those two summarize L Q R design, we have studied that extensible before in the few lectures. I mean, you can see all those things if you have forgotten or you want to revise. the performance index is like this and if nobody tells us anything, we assume that T f is infinity; that means, the cost function is quadratic in state and quadratic in control and final time is infinity, and Q is possible to semi definite where r is positive definite. the system dynamics is linear; X dot is A X plus B U. The boundary conditions; its initial conditions and it is known, final conditions lambda of x equal to 0 and also if it is infinity time because of quadratic function, all these condition will guarantee that X f will go to 0 as t go to infinity.

(Refer Slide Time: 07:16)



This is the problem and what is the solution? we have done this solution already that control can be completed this way, essentially some sort of a gain matrix K times X with a negative sign, U equal to minus K X, where K is computed as R inverse B transpose T. R is already, I mean R and Q are region parameters (( )) parameters and already we have it. So, R inverse already we have, B is the system dynamic matrix, so we have already from there. What you do not have is P? P is computed as a solution of the Riccati equation, which is something like this; P A plus A transpose P minus P B R inverse B transpose P plus Q equal to 0. So, essentially you compute, you solve this Riccati equation, compute the P then compute the gain matrix that R inverse B transpose P and we have a controller structure ready. The whole problem here is this X is not ready, your K is ready but X is not ready or even if it is ready, it is noisy. So, what you do for that.

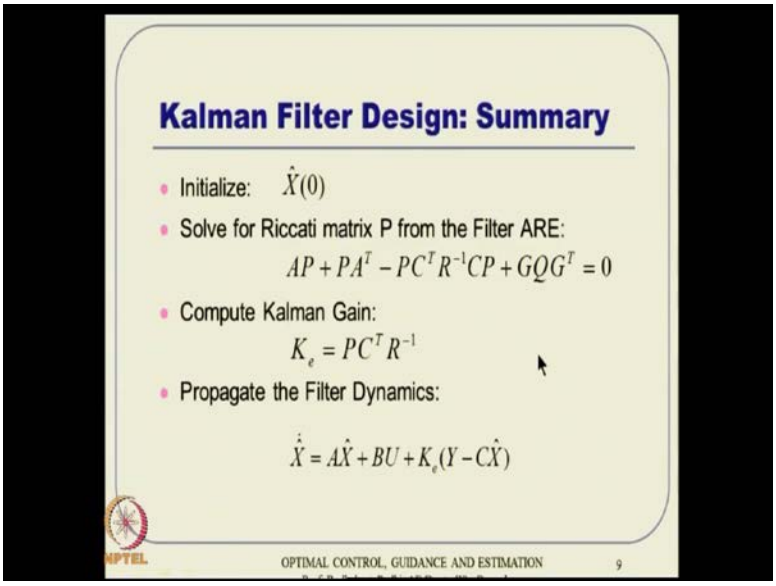
(Refer Slide Time: 08:09)



So, then this idea comes to mind (( )). This technique comes to us very handy that we also know well we have a way to kind of reject noise. Now, we know Kalman filter design and which can tolerate both process noise as well as sensor noise. Why not using that? So, the Kalman filter design summary in continuous time domain, other things are all also there as a choice basically. Then in continuous time domain what you show is X dot is not only A X plus B U, but it also as a G W component and the output is not only C X, but there is V component also.

So, X dot is that way and Y is that way. So, these are assuming to be white noise and Q is expected value of W, W transpose process noise covariance and R is expected value of V, V transpose which is sense or noise covariance. So, Q and R are typically process and sensor noise covariances, R is known from experiments and all that with respect to particular sensor side that you are using. Q is typically a tuning parameter where the whole idea is to assume some sort of a larger Q in the sense it also terminates the modeling in accuracy on that. Whatever is inaccurate in the modeling part it will come under the noise basically. So, they are all I mean we discussed before in detail.

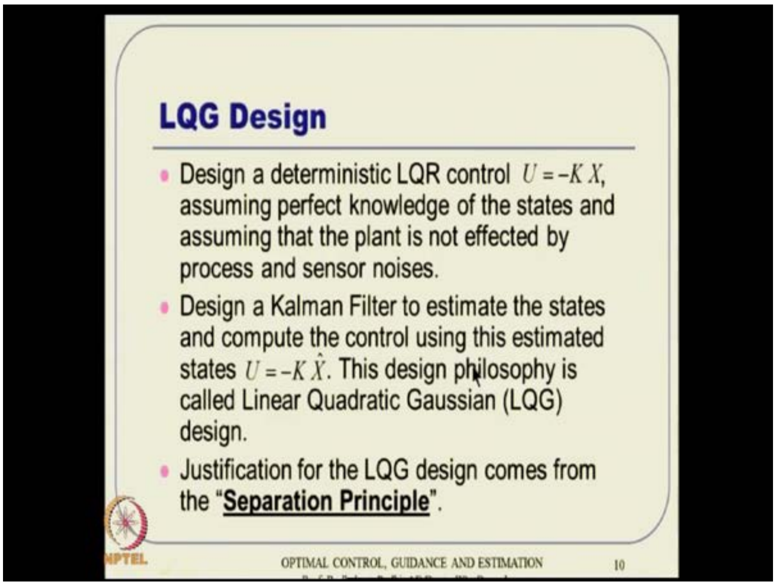
(Refer Slide Time: 09:32)



So, we have this and then you talk about estimating this state using this sensor and output information. How do you do that? You initialize some state to some values and then solve the Riccati, I mean the filter Riccati equation or filter algebraic Riccati equation, which is slightly different from what we see in the control Riccati equation, and this is the one. Once, you solve that your P matrix will be ready and then gain K e can be computed as P C transpose R inverse. Detailed derivation and all, we have all already done much in some one of the previous lecture.

So, we compute the gain that way and once you compute the gain, we have an observer dynamics, and this observer dynamics talks about a little bit innovation components, which is actual output to minus predicted output. Then you multiply that K e and then operate this filter with this initial condition, I mean, have to operate this observer dynamics with this initial condition. The whole idea is as time goes that means, as time involves your X i, it will converge to real X. That means, estimation is guaranteed to work as far it is as linear equation like that.

(Refer Slide Time: 10:47)



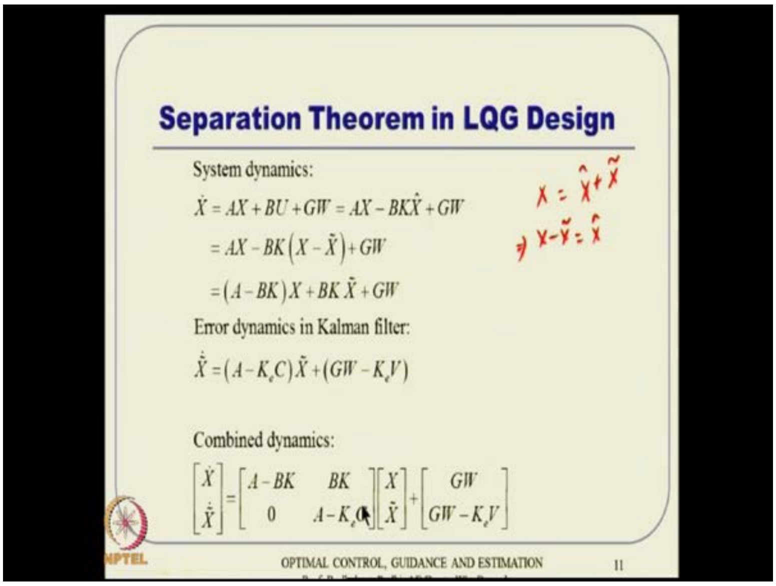
So, L Q G design talks about fusing these two; it talks about the design and deterministic L Q R control and forget everything about estimation design at this point of time. Do not worry about it all. So, it assumes perfect knowledge of the state and it assumes that the plant is not affected by process, I mean process and sensor noises. You do not worry about that at all. Then you design a Kalman filter in parallel and do not worry how this control is designed. This control and gain, how it has come and all that, do not worry about that all; in estimation design problem, you design that and find out X hat from this part of it and K from that part of it and then fuse them together as U equal to minus K X hat; now, not K X, but K X hat.

So, this design principle is called L Q G and the justification of the L Q G design comes from the peak part. There is a big good theorem separation principle. So, these two things can be done separately. It does not affect the overall system. All these things by the way remember that these are all valid only for the linear system. This separation principle which is so nice, it does not be good in non-linear system. In general, nobody has proved that in a way, in other words, it is still in a full problem.

If some of you really want to do some research on this probably pick up a non-linear control design, there are various designs of course, and then also because something like filtering design, either (( )) cap, U cap, I mean (( )) filter or particle filter. Whatever it is, if you pick a particular non-linear filter and pick up a non-linear controller and then try to see whether this separation principle is there or not, if you can prove that, I mean the comment is it works in general. We have seen that in a previous lectures also estimation operation E cap and guidance and control operation dynamic inversion things like that.

But, now days it has come up with a proof for that and a rigorous confidence for that. So, if you can take some of these problems and then come up with some ideas like that then that will be a kind of path breaking sort of results. Anyway, by coming back to this, this separation principle was an idea of Kalman and his co-workers and he tell this two things can be done separately while doing one other one you can ignore, but finally, we can operate the control based on this and this operates on separation principle.

(Refer Slide Time: 13:24)



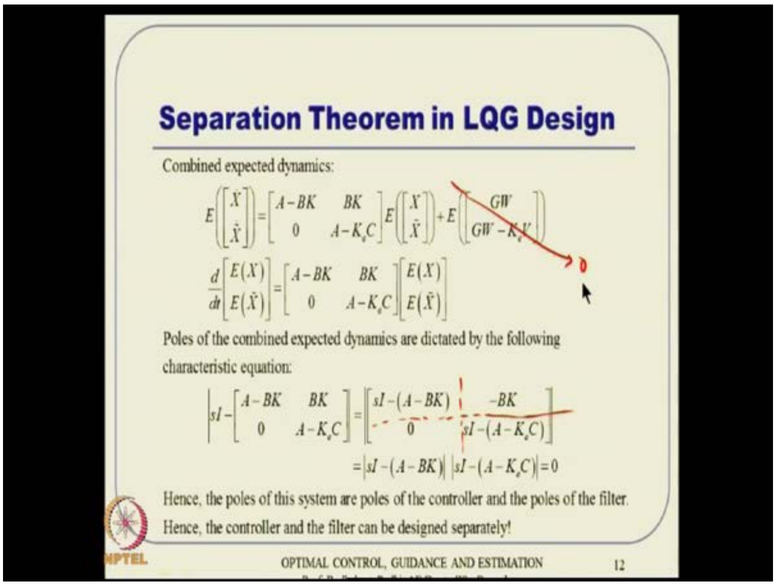
So, what is the separation principle? This has various proofs. Separation theorem can be derived in various ways, but a very quick way is something like this. We have the system dynamics, X dot is A X plus B U plus G W, but remember U is now minus K X hat. I mean U is minus K X hat. So, this is what it is. If you put it back in this U is something like A X equal to minus B K into X hat, minus K X hat comes from here. So, this minus this term is because of U. Now, if you see these two quantities, these quantities, then this quantity when A X can be retained here and then this is something like X hat, I can represents as something like X minus X tilde.

Because X minus X tilde is X hat, means our original state, remember it is estimated state plus error in the state. So, that is why this gives out that. So, this one, you put here, B K into X minus X tilde. And then if this part A minus, now you can see that these two can be combined; X is happened here and here also. So, these two can be combined here like that the rest one is B K times X tilde here and G W here. So, the error dynamics in Kalman filter and I mean in a close loop, in error dynamics, well not in the Kalman filter, but in the close loop operation really. No, sorry, the error dynamics in the Kalman filter, we have derived it like that and if you see that derivation of Kalman filter is something like X tilde dot is A minus K e C X tilde plus this thing.

Then remember we have got the solution of these and then considered these as something like a time bearing input and then this Kalman filter integration problem all these big derivations we done before. So, essentially this is the error dynamics in Kalman filter. So, this is the system dynamics and this is the error dynamics. Now, the question is can we not see them together. Remember, the error dynamics contains the information of filter dynamics also basically. So, as long as we prove that this system dynamics and error dynamics are separate then we are also done, because one is dependent on other.

So, whether we see that whether the actual system dynamics or the estimated state dynamics, those things are separate or this is equivalently telling that system dynamics and the error dynamics whatever I am telling X tilde dot, if it is separate then X hat is also separate because of these relationship anyway. To see that thing, we put them together now, where X dot and X tilde dot, so X dot and X tilde dot. Now, these one, these A minus B K times X, A minus B K times X here plus B K times X tilde, so B K times X tilde plus G W plus, external input, something like these. X tilde happens to be a function of X tilde only and plus some external input. When you talk about stability of the system, we typically do not worry about the exogenous input. So, this is this time varying input to the system is ignored, we want to worry about the system matrix only. So, this matrix, so X dot and X tilde dot is here and this part appears like that.

(Refer Slide Time: 17:37)



So, if you see the expected value of the error dynamics now, because everything happens, these are all random white noise and all that remember that. So, we cannot talk about this dynamics per se, but remember these are 0 mean white noise and all. So, if you take expected value of that, this is nothing but expected value and will come here, because this is a constant matrix. Remember, these are time invariant linear system, L T R system. So, this is this is constant matrix. So, expected value will come directly here, because it is a linear operator.

Take the expected value of that, this matrix time expected value of that, plus this expected value of this quantity. But, this expected value of this quantity, now you take, I mean kind of and if you see that this is essentially 0 mean white noise both W and V, so if your expected will go here and here and all these things will be 0. So, this is as if it is not there. This quantity happens to be 0. Now, if you put d by d t of this one, I mean this one remember, these are dots, so it talks about d by d t, because this is written in the same thing, X dot and X tilde dot. So, d by d t of that essentially happens to be like that. This component goes to 0.

Now, what happens to these dynamics especially, at least in the expected value sense? So, to analyze that you can have these Eigen value analysis and Eigen value of this matrix is determined by the characteristic determined. So, the characteristics equation is given by this matrix determinant; s I minus this matrix whole determinant and that determinant, I mean, now s I is a diagonal matrix, remember that. So, in a smaller sense, this I is different from this I in dimension sense. So, remember that this I is larger dimension, this I is of smaller dimension and this is also smaller dimension, but I did not say that this small discrepancy that this one can be written as something like this.

Now, this is typically a diagonal matrix. So, part of that is put here and part of that is put here. Then this determinant is, if you take a partition matrix sort of thing and try to evaluate this partition matrix everything you know and try to evaluate. Then this is nothing but the determinant of this, it times the determinant of that minus 0. May be this entire (( )) becomes 0. So, that is essentially talks about determinant of this and it is nothing but, determinant of that. So, that is equal to 0. Remember, that we are constructing a characteristic equation to find out Eigen values, so that has to go to 0 and essentially you can see now this is a scalar quantity and this is scalar quantity.

So, both are determinant; A times B, sort of thing has to be equal to 0; that means, either this is 0 or that is 0; that means, if this is 0 this are Eigen values of the controller close loop control plan, if that is 0 this is the Eigen value of the error dynamic. Now, the thing is both controller and observers are designed in a good way. We have proven that also. So, the control is design, the gain is designed such a way that it results in a stable close loop system dynamics. So, that means these Eigen values are all in the left top line and also same thing for the filter dynamic; filter error goes to 0 ultimately; that means, I mean expected value sense. So, that means, these Eigen values are good. The combined eigen values, it all that the this particular results shows us that the poles of the system, the combined Eigen values of the entire dynamics are nothing but, the poles of the controller and the poles of the filter. This one only result in the poles of the controller and this one result in poles of the filter.

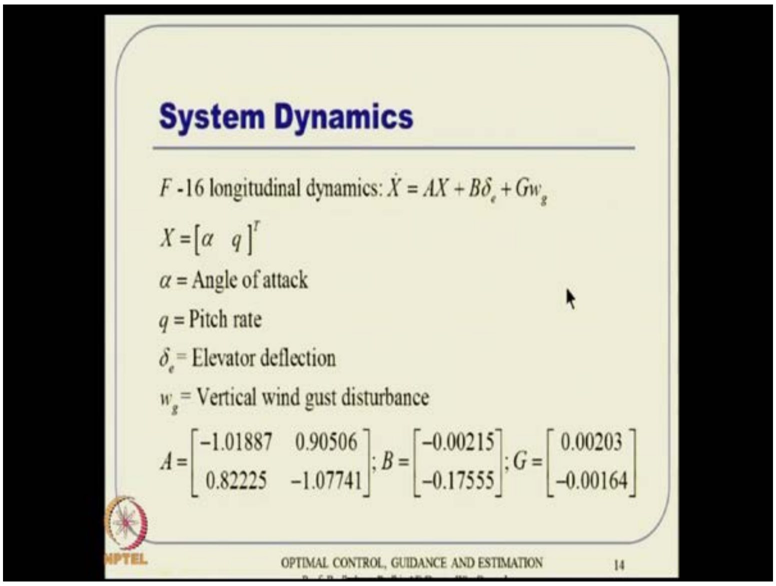
So, essentially the whole poles of this entire system consist of nothing but, the poles of the controller and the poles of the filter. Hence, the controller and the filter can be designed separately. Even if you consider this and that separately, the combine system does not go bad easily. So, this is the separation theorem in L Q G design. It is a big achievement and that gave lot of confidence to the people working on this domain that things can never wrong if I implement that way. Of course, provided both happen in the frame work of linear time in variant systems.

The small comment you have also seen and many people have implemented on Kalman filter in non-linear frame work and that is the way to implement in (( )) essentially. So, even if the controller happens to be linear frame work then the estimator is always happens to be in the non-linear frame work typically. Even the controller, even if you design in a linear setting, linear system setting, typically the implementation will talk about at least some sort of gains gradually; gains will be interpolated that time varying. So, in that sense is also is a non-linear control design and in that set of many success results have been reported. Do not have to doubt or I mean, you do not have to get afraid about that. The fact of the matter is something like separation theorem is not there, just keep up in the mind.

Next, we will see a small example to have a confidence. So, here the short period control using L Q G design and this is one of the design especially has been used extensively in aerospace industries. Remember, these things can be computationally not taxing. All that you are talking is just solve two Riccati equation; one filter Riccati equation, one control Riccati equation, and if it can be done offline or compute the gains offline, and this control gain offline and filter gain offline also and then try to interpolate on that. So, these things you find heavy use in industrials basically including the aerospace industry.

So, one of the uses is how to kind of do gust elevation and this is also about short period control using L Q G design. Remember, this short period is typically excited because of the gust and those of you flown in aircraft might have repeatedly encountered this experience. They are called turbulence and all that. As turbulence is there, so tighten of seat belt, they will tell. Those are typically the thing when there is some wind gust, which will affect the dynamics little bit. The aircraft will vibrate for a little small time and all that. These are short period dynamic. So, pilot does not have to do anything. The control phenomenon, this autopilot design or the way it is implemented in aircrafts will automatically take care of that.

(Refer Slide Time: 24:23)

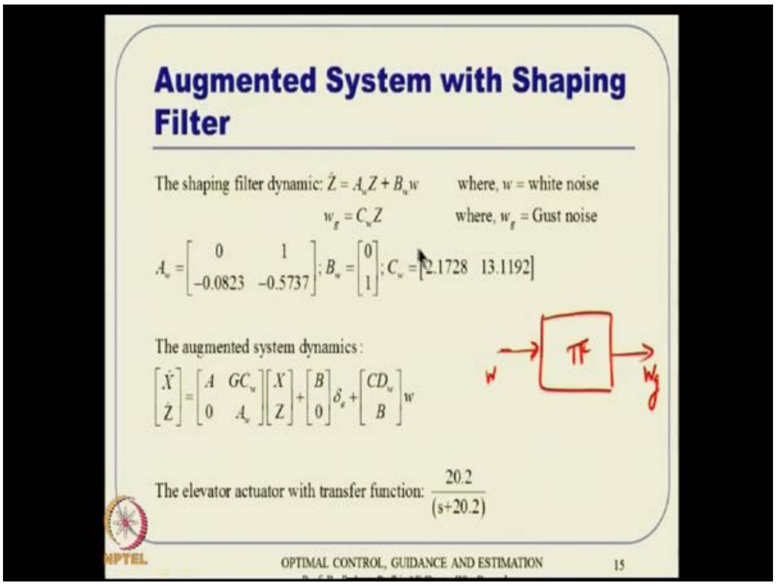


How does it operate? Let us see a small example problem here. This is F 16 longitudinal dynamics and F 16 is longitudinal dynamics can be replace is about some operating point can be designed that way, I mean can be written that way. X dot is A of X plus B time delta e. the way to control this longitudinal motion phenomenon is by deflecting appropriate delta e to suppress that I mean that oscillatory behavior. So, the dynamics that you consider is alpha dot and q dot; let it be angle of attack rate and pitch rate. This angle of attack rate and pitch rate is strongly coupled with delta e, elevator deflection, and the dynamics happens to be about some operating point and happens to be like this where A and B matrix are give like that.

Now, G matrix is also kind of some modulator, I mean this kind is identified that way. So, anytime if there is a white noise, sort of gust here coming, need not necessarily white noise, any noise that comes here, that is gust influenced in the system dynamics get influenced by this noise through this G matrix. At this point of time, it does not matter whether it is white or not, it is essentially it is a wind gust phenomenon which is non white also. So, the moment there is a noise, there is a noise influenced matrix and that X dot will be like that. The movement there is a control deflection; there is control influence matrix that will also alter extra. So, if you compute this delta e properly then this effect can be canceled out, I mean.

But, also remember exact noise number and all will not be available. So, knowing one idea is ok such as why do not we directly cancel it algebraically. It is simply not possible because this number is not available. So, we have to estimate it and estimation takes time, a small amount of time at least, then based on that it can have a control design to filter out this one. But, before we do that, here alpha is angle attack, typically the angle between body axis and the velocity vector in case plane, q is pitch rate, delta is elevator deflection, and w g is vertical component of the wind gust. Those of you do not know, I still suggest you to go back to flight dynamics lecture as part of this course itself, as some of the early lectures of flight dynamics will give you some definition like this and all or you can see a flight dynamics books also.

(Refer Slide Time: 26:55)



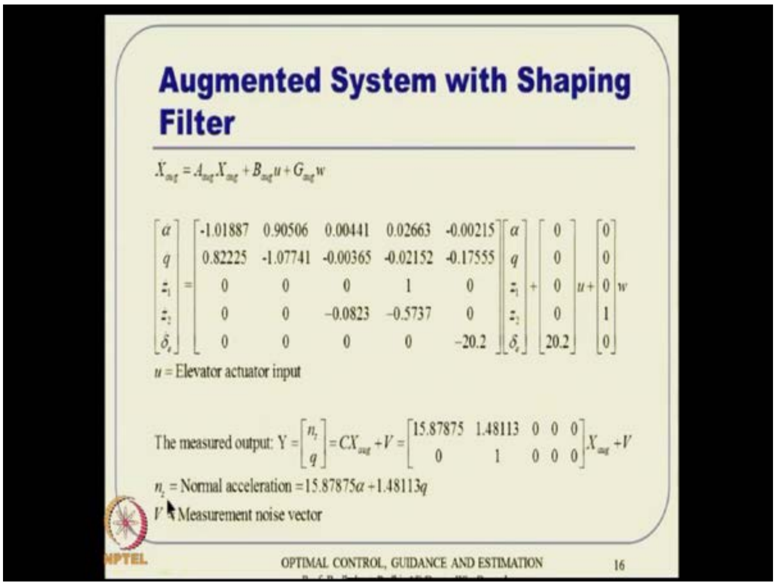
Now, what happens is this is as I told; this wind gust is not white. So, how do you handle that? Kalman filter only talks about white noise. So, this is the way to handle something like a subsystems sort of thing, which will be modeled as this system dynamics and this output equation together and essentially it comes through a transfer function realization. Here, we have some sort of something like a transfer function and with this transfer function, we will take white noise, (No audio between: 27:29-27:39) what happen here? This transfer function will take white noise and it will give non white noise. So, this means this will give w whereas w g is wind gust really. This w g is the one which will go and affect the system dynamics. Remember, this is w g, the gust component.

This transfer function if you realize the system, I means state space realization, it turns out to be like that and because it is a filter dynamics filtering problem or estimation problem, this observable canonical form is the one which is recommended here. Remember, the various forms of realizing a transfer function is not unique. In other words, if you know a state equation realizing into transfer function is unique, but whereas, if you know transfer function realizing that in state space is not unique. That various phase exits and because it is a estimation problem the recommendation is that we realize that in observable canonical form.

So, you do that and then A w, B w and all that will come into picture. Now, how to tell is I have got a system dynamics, but there is a wind gust dynamics which is represented as this system dynamics. So, I will put them together now. X dot comes from here and Z dot comes from this part of the story and then I will put them together here. Now, what happens is this w of what we are talking here or here is essentially a white noise; that means, we can apply Kalman filter to this system dynamics, but not to this system dynamics because this is not a white noise.

This way of doing things is something called setting filter and that is what this transfer function will be identified for this particular purpose; to model that wind gust phenomenon. This modeling is something like this; if you take a white noise and then pass it through this model then it will give that particular noise, which is physically happening. That is what is called as shaping filter (( )). And also after getting this delta e, it is also passed through some sort of actuator dynamics; actuator dynamics, for control I mean this elevator deflection is first order system with this transfer function. So, this is what it is. Then ultimately this realization of that can be done separately or together, they are both ways of doing that.

(Refer Slide Time: 29:56)

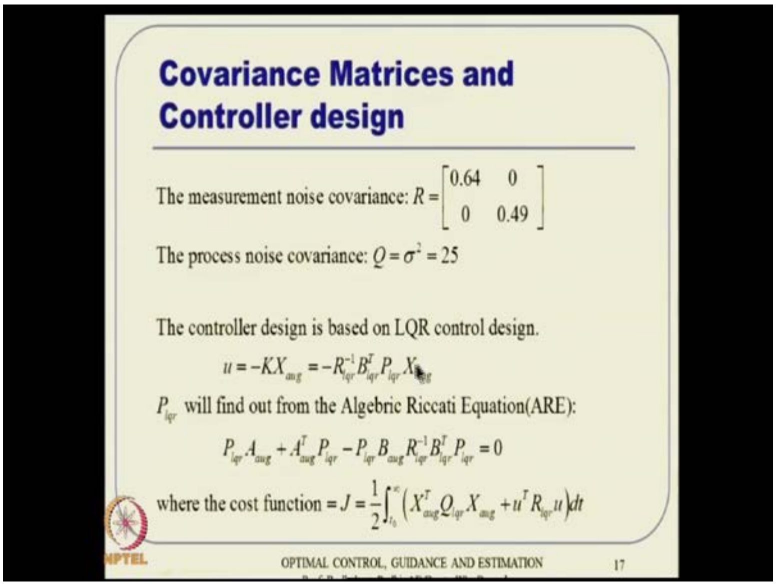


If you do hard coupling sort of thing and that also will come into picture, This can be realize as a first order of state (( )) equation and then it directly put into here and tell that I have got two system dynamics; alpha dot and q dot, then these are the wind gust dynamics realization, this part will come here and delta e dot is actuator dynamics, I will put it there also. So, I will directly compute the input to the actuator through this 5 state formulation. Even though we started with two states and eventually we landed off some sort of 6 states. Sometimes, again I repeat, sometimes this is not preferable. You just put it as 4 states then this will also talk about realization of actuator state now; this delta e, if you put it this way.

Now, for actuator state realization, we also need a sensor for that and some actuators are equipped. Especially, aircrafts if you talk about, it is there, I mean, the delta information is available. Sometimes, when it is not available, you cannot recover that from other states; obviously. So, it is not a good idea to put it that way. But, in this particular case, because it is F 16 and heavily equipped and expensive also and things like that, the actuators are capable of giving us the information about the actual delta e, so that is available. So, you can put it that way. Now, what are the measured outputs? In this particular case, we are assuming that we are not really assuming this delta e sort of thing, but what we are telling is we are aircraft is equipped with sufficient sensors in measuring two quantities only. These two quantities are nothing, but normal acceleration and pitch rate.

So, the normal acceleration and pitch rate, this is assumed this as a (( )) as this exclamatory will give us normal acceleration and the pitch rate will be given from (( )). These two values, these two sensor outputs are available with us and this can be represented like this. Normal acceleration, remember has to be represented in the form of alpha and q. Remember, that the output equations should be a function of states only. So, n z, this is the model part of it is a function of alpha and q this way. So, using this numbers now and I also strongly suggest that you repeat this exercise yourself. You do your own matlab coding or according through simulink or whatever it is and to generate the results of this, will give you lot of confidence. So, n z is given like this.

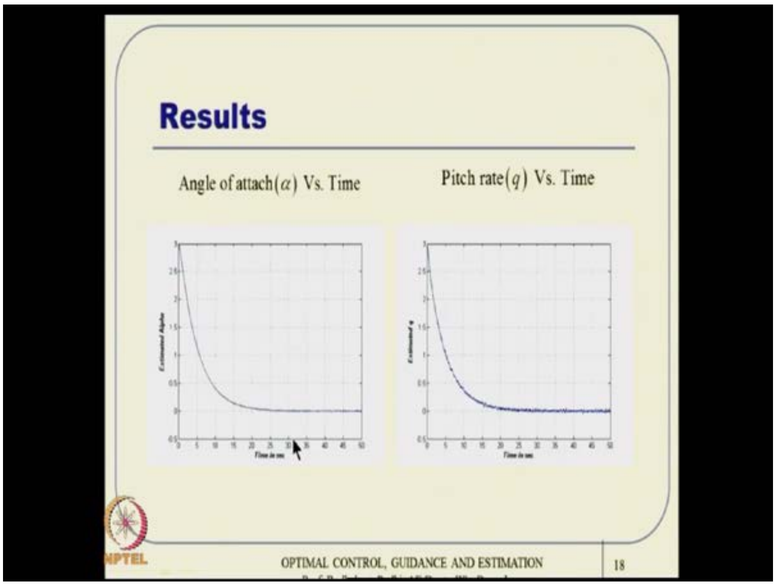
(Refer Slide Time: 32:36)



Then the measurement noise covariance is something like this. R is designed that way and Q is sigma square and it came to 25. So, the controller design is based on L Q R control design; obviously, this talks about that. So, u, what is u here? It is the input to the actuator, remember that u, this u comes from since kind of realizing this transfer function in a state space. So, the input to the actuator is directly computed to this augmented state information and for this you need a control, this Riccati matrix P l q r; P l q r is computed through a controller Riccati matrix sort of thing solution basically.

We solve the controller Riccati matrix; put it back for the gain design and you are ready with the L Q R control. What about the filter? The filter operated through this dynamics, this state equation, this output equation and then you have this as the white noise, because of the gust modeling and this is also a white noise sensor output. Now, you can use filter, I mean Kalman filter and then controller can be computed this way.

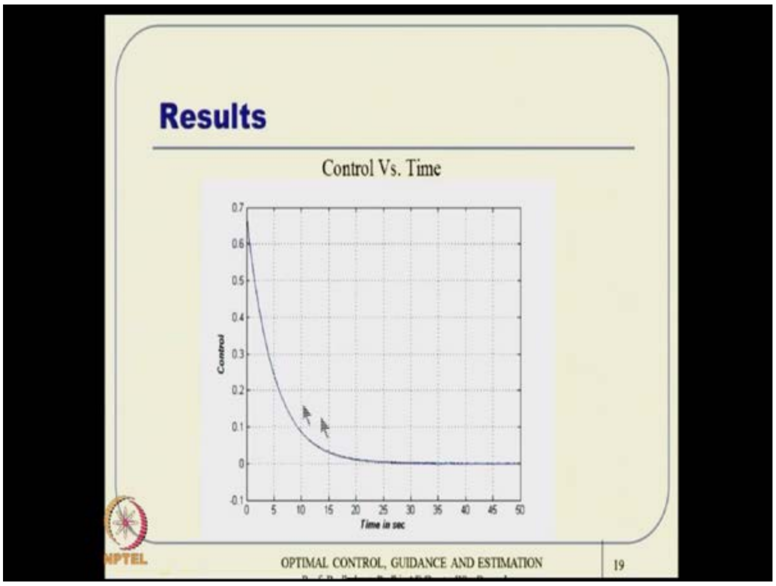
(Refer Slide Time: 33:51)



Now, what are the results? The results are very good; the alpha is getting suppressed. Remember, 20 seconds is not a very bad time or 15 second really, not a very bad time as far as aircraft equations are concerned. Especially, if you are really concerned about commercial aircraft and all that it does not matter that much. Fighter aircraft performance is more, so even if there is, I mean import performance is more. So, even if it takes a little longer time to suppress this gust phenomenon is still tolerable.

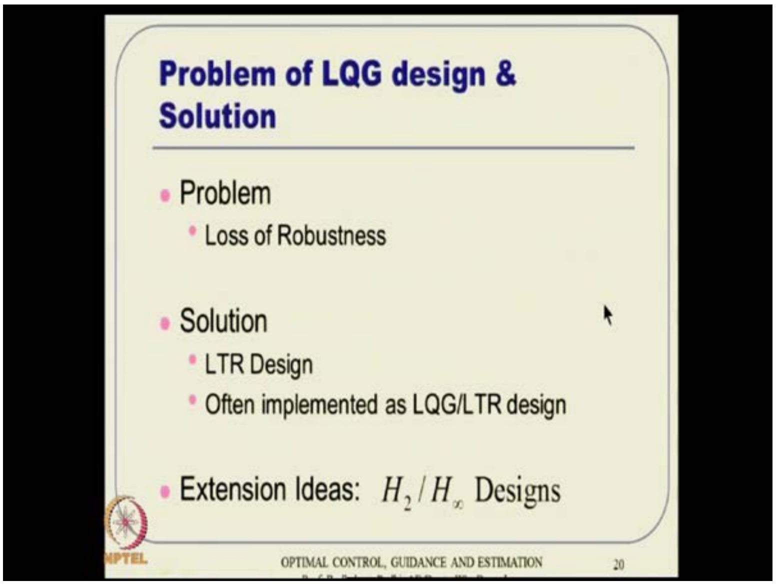
this is alpha and this is q and remember that alpha is one other smoother than q. If you things about the dynamics of alpha dot and q dot and if you see this equation alpha dot is a angle that talks about angle between two velocity vector components and the q is directly the pitch rate. So, pitch rate is more sensitive to the gust, alpha is not smoother than that. That effect is also visible here, q is also going to 0, but there is a small residual error, very negligible error; structural kind of denting and all will take care of that. The structural aerodynamic denting will not let you have this. So, do not worry about this small (( )) here. Usually, it will not happen, but even if you see on the way there are small residual errors which are smoothened out in the alpha level.

(Refer Slide Time: 35:16)



What about control in time? We are not catering at all. Control will nicely operate in the estimator state and estimator states are these two and control will be operating like this. So, with these control, application of these control will be able to take alpha and q both to zero (( )). So, this L Q G design is kind of a popular technique to do this kind of control design.

(Refer Slide Time: 35:42)



So, there are problems of L Q G design also. The main problem is loss of robustness of L Q R. Remember, if you have full state information in the L Q R design at that point of time, in those lecture, I also told that typically, if full set of information is available is error free then L Q R has good robustness; that means, you have as good as something like infinity gain margin and up to 60 degree phase margin also basically. But, all these things are gone the moment you put an L Q G design; L Q G design does not guarantee that. Now, the whole idea is can you do some modification of this gain computation K and also the control say the filter I mean Kalman gain K e, somehow the core point are well down there.

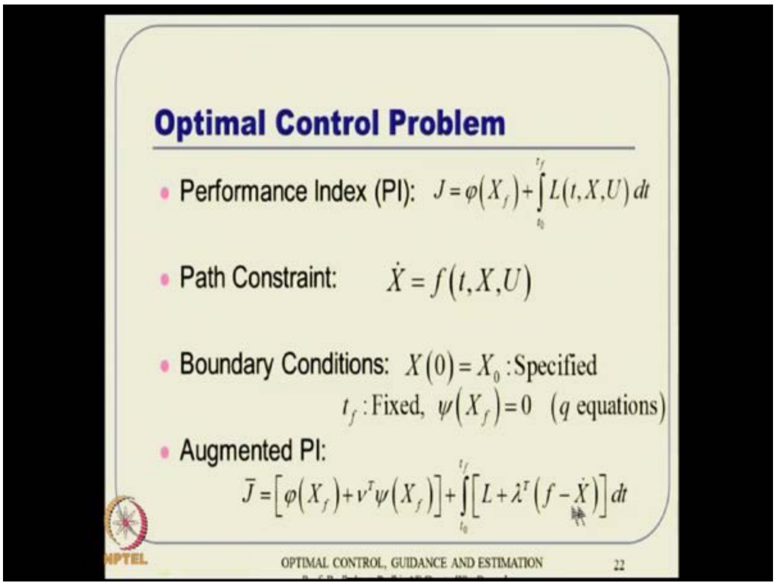
So, if you blindly select this numbers like diagonal and then work with that some time domain phenomenon like that then this robustness considerations are ignored. So, we have to go to the frequency responsive characteristics and from there you have to propose a design kind of tuning baskets sort of thing, cannot tune where ever you want to, it will allow you some sort of a basket from which you can pick up your tuning values. So, that is called loop transfer recovery. So, that idea is called loop transfer recovery and L Q G when somebody designs L Q G, L T R cannot be ignored. you have to kind of either incorporate in the control design gain selection process directly or at least you have to test your selection, whatever matrix given are selected whether it passes through the L Q R test conditions, L T R test conditions or not.

So, it passes through and we are happy with that and all that and then L Q G and L T R is kind of a good design. Then there are other ideas also and then just remember the L Q G, L T R all everything operates best on two phenomena; one is white noise, the other one is expected value. Everything happens in the sense of expected value that is average value. Average value does not give us good confidence because at momentarily, some noise can be very high and then system can be unstable and things like that. For example, surge protection, some sort of ideas when you talk about. Electrical circuit is when there are surges and that point of time the circuit breaks down. So, that kind of consideration, those kinds of problems motivated the idea of something like h infinity design; that means you expect the maximum noise and then try to have your design kind of say for the maximum noise as the input.

So, I will not talk too much on that. Those are the subjects of robust control course and all that. But, sometimes (( )) as infinity and all that are also called as optimal control extension ideas, but those are in different ball game altogether, so typically thought in a robust control course. So, let me not talk too much on that. Anybody interested in that you can also read some of these robust control books and one of the very readable book is probably book from Messijowasky who was professor in Oxford University, U K.

So, that is not part of our story basically. So, L Q R and then Kalman filters leads to L Q G and the L Q G as a little robustness problem, you kind of bring in this concept of L T R, which gives us a way of selecting Q and R matrices and all that, which will give us some sort of robustness back. So, L Q G, L T R turns out to be a good practical design approach. So, this part of the story is here. Let us move on to next topic of this lecture. The next topic is a neighboring optimal control. It is a kind of related to L Q R, but not, I mean we are hunting out for a neighboring optimal solution in true sense. So, we are not interested in (( )), but we are talking about some finding out an neighboring of a optimal control solution.

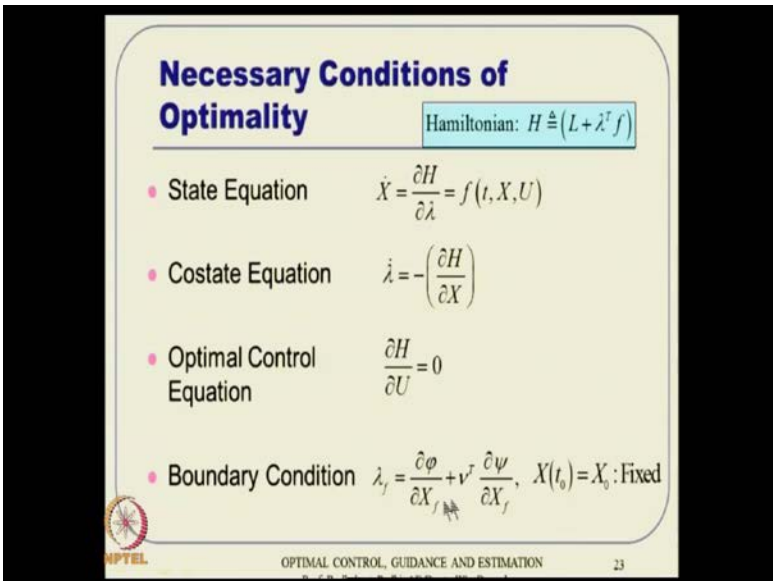
(Refer Slide Time: 39:44)



Just to summarize again it will be to have a performance index in general. Remember, these are (( )) to the non-linear domain. So, these are all happens in a non-linear frame work now. So, the optimal, I mean the performance index to optimize is some terminal penalty, some path penalty sort of thing. Then there is a path constraint, then there are boundary conditions and then we have this augmented performance index, which talks about this kind of thing.

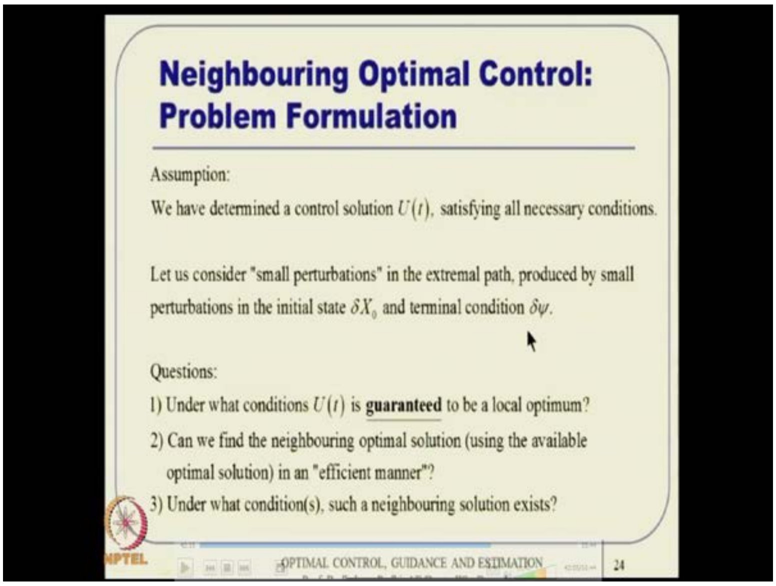
So, remember there are at final time, not only there is a soft constraint, there is a hard constraint set of equations also. Not only, has this to minimize, this function whatever typically quadratic and things like that, but these constraint also be met and that too in a partial set. It cannot be full state or all the states and all that. The dimension of the equation can be different from the number of states. This can be q equation sort of thing. So, one idea is you have this augmented performance index phi of X and then these nu transpose, this (( )) variable nu, this one is bought in here, nu transpose psi, this operates in the penalty part of it here. Then this third equation, which operates throughout the trajectory, happens and happens to loss on to that and this turns out to be like that.

(Refer Slide Time: 41:01)



Now the Hamiltonian is a function something like that and you derive all these necessary conditions; X dot is del H by del lambda, again f of t X U. Then costate equation, then optimal control equation, boundary condition, then the boundary condition; remember, will come from this full term, so it is not only this term, but lambda f is this term plus that term and forget that.

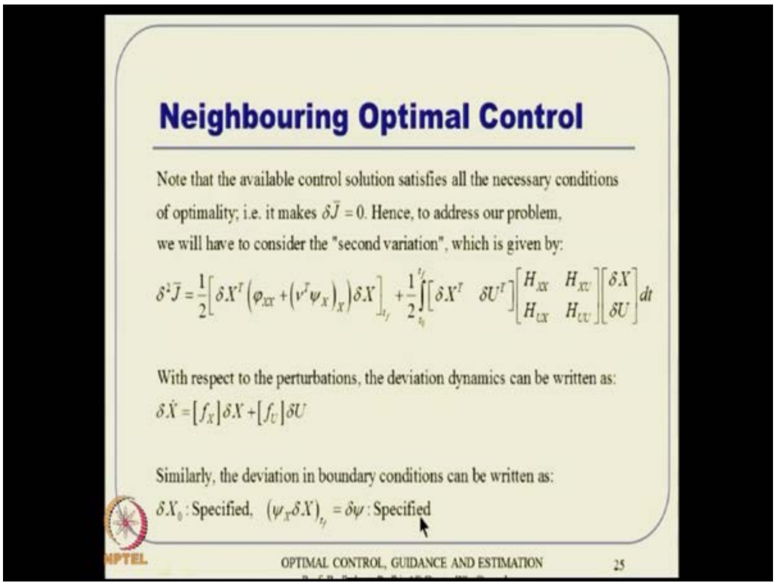
(Refer Slide Time: 41:26)



Now here is a problem definition like this. What you are interested in? First, you are assuming that we have determined a control solution U t satisfying all necessary conditions. Then we have these, like now let us consider something like small perturbations in the external path produced by small perturbations in the initial state, somehow, either because of the gust or something that you did not understand the process. Somehow, it resulted in some sort of delta X naught and from there onwards we want to find out a different path, which is close, which is optimal. So, this was it. So, what happened is there was a small perturbation that produced a small perturbation in the initial state and a terminal condition d psi also, whatever we talking here.

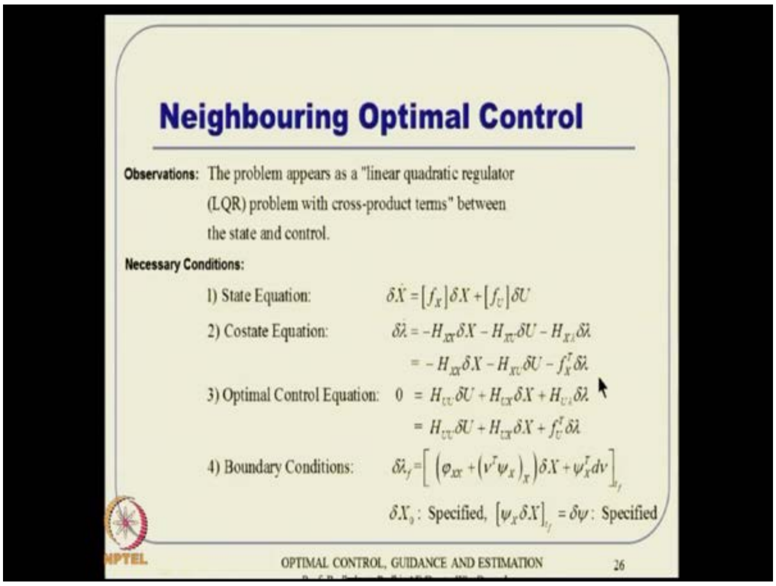
Now, the question is something like this. Under what condition U t is guaranteed to be a local optimum, first of all, and then if it is there then can we find the neighboring optimal solution in an efficient manner. While you can always think that as a new problem, after these perturbations happens and try to go back and resolve your 2.1 relative problems and then solve a new path. That is not the option here. Can you really do a little more efficient way, because we already have a closely, already have an optimal path close to it. So, can we do a better job? Can't we find out a little efficient manner? Then the third question is under what conditions such a neighboring solution (( )). So, another issue is there.

(Refer Slide Time: 42:55)



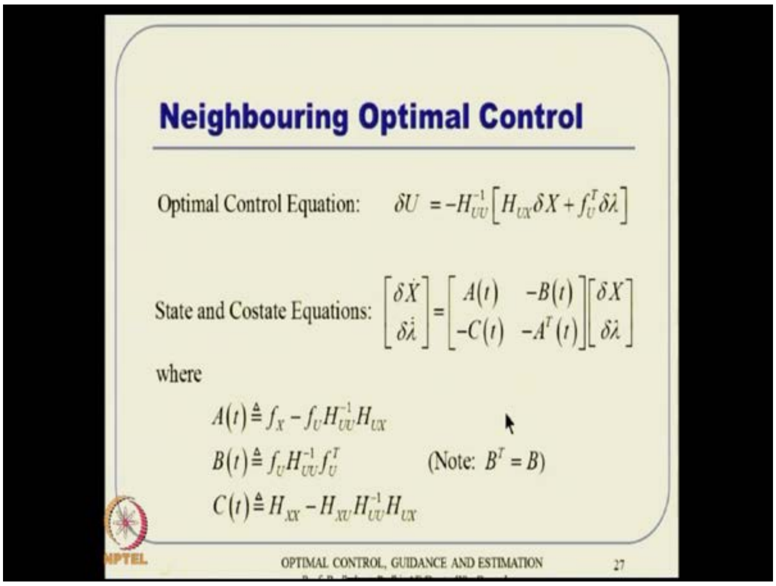
So, the detailed derivation can be found in Bryson and Ho text book. Here, we are interested in delta square J bar, the second order variation actually. So, first of all note that the available control solution satisfies all the necessary conditions of optimality that makes that means, the first variation is 0. That is already there. What you are interested in making sure that the second variation is also minimized and all and that. So, you are interested in del square J bar and del square J bar can be derived something like this. Remember, this all are Bryson and Ho notation sort of thing and this phi x x stands for del square phi by del x square. Similarly, this del H U X del square H by del U to del X like that. So, this is the performance index. So, with respect to the perturbed equation dynamics, system dynamics this can be given like this. So, this is the system, I mean the cost function and this is the perturbed third equation. Similarly, the derivation in the boundary conditions, sorry the deviation in the boundary condition can also be written something like this. This delta x naught is specified and delta psi is also specified. You know that. The perturbation that happened we know, so that is available.

(Refer Slide Time: 44:10)



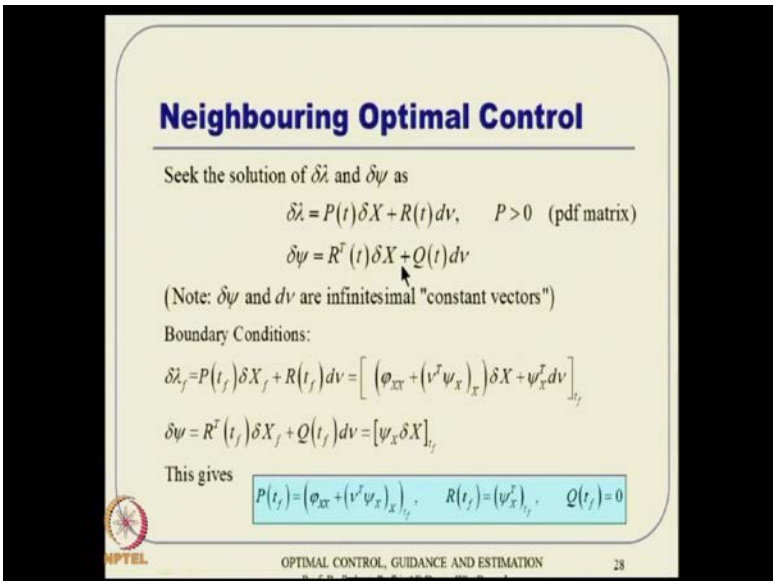
So, essentially the problem happens to be a linear quadratic problem regulator problem, but with cross product terms also basically. This term, this cross product term will appear. So, the state equation is like this and the costate equation can be something like this with respect to these cross function, remember that. You have to derive all that. So, delta lambda dot turns out to be like that, optimal control equation turns out be like that and boundary condition like this. Again, I suggest that you derive this yourself and (( )) you will have lot more understanding.

(Refer Slide Time: 44:46)



Then optimal control equation is given like this and the costate equation is given like this. State and Costate, together can be given like this, where A, B, C can be derived from this, these conditions to be like this. Because, now you remember B is a square matrix and B transpose happens to be B, if you talk reverse transpose and all that. So, this is a symmetric square matrix sort of thing.

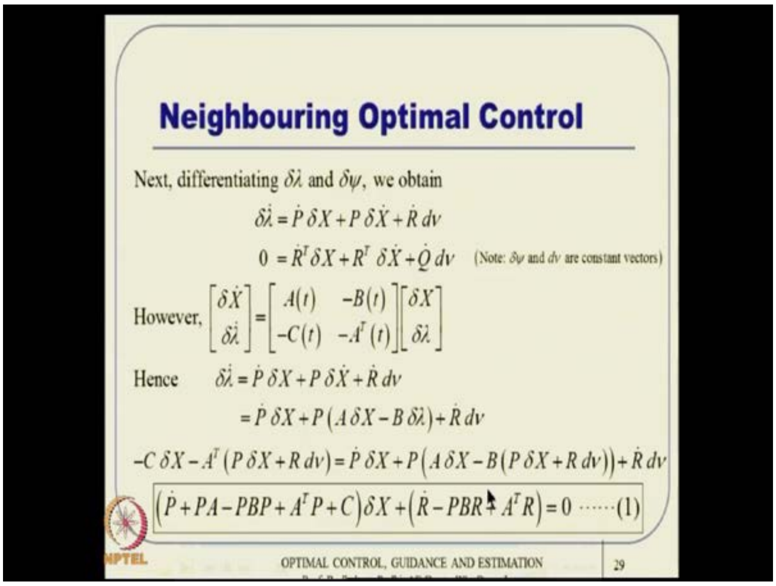
(Refer Slide Time: 45:13)



So, what you are doing is seek the solution of delta lambda and delta psi, because that is what is unknown here, so delta lambda, and delta psi, in the form like this. Now, remember the whole idea of L Q R is we seek a solution lambda is a function of X, where lambda is P times X. Similarly, like that we have to seek a solution as delta lambda is a function of delta X and delta nu also. So, this one we seek a solution like this and like that. Then you can talk about this boundary condition imposition and all, delta lambda f and delta psi f can be represented like this; directly can we derive from that and you can put it that way.

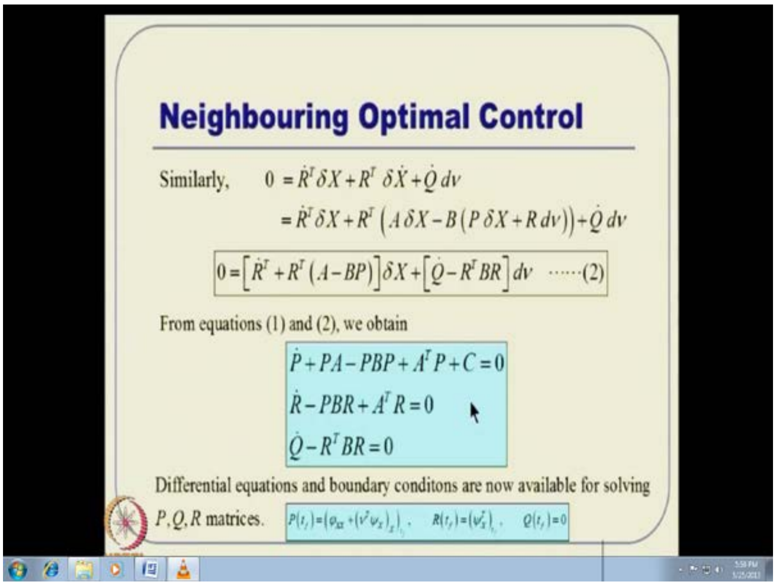
So, this essentially gives us the final, suppose you kind of compare these two guys, because this is from assumptions and this is from what you derived and all that. You put them together; this tells us that there is a delta X f component, there is delta of X f component, so P of t f has to be this component at time t f, because all variation cannot be 0 and all that. Similarly, R T f has to be like this and this is same, you do not worry about that where Q t f happen to be 0. There is no delta nu here and that happens to be 0.

(Refer Slide Time: 46:31)



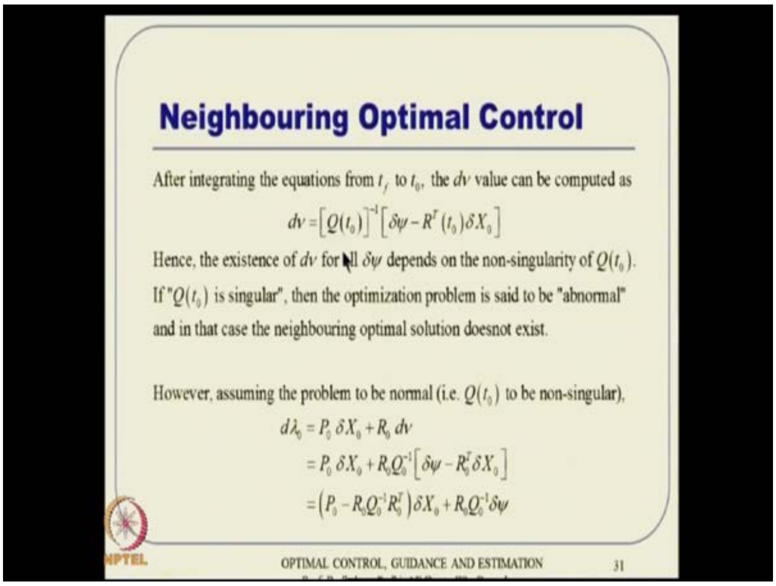
So, there are three boundary conditions known to us. Now, what about this differential equation? So, you go back and see delta lambda dot and things like that and exactly similar to what we have done in L Q R setting derivation sort of thing. If you follow the similar steps it turns out is that this equation will result and because the variation cannot go to 0 and all, we will have two equations one the (( )) or very similar to the Riccati equation that you already knew before and then there is a additional differential equation like this. So, this will all result in, if you put it back all the thing that we have know, this will result in, no sorry this is two equations, then the other equation if you analyze I mean this equation delta dot and all.

(Refer Slide Time: 47:12)



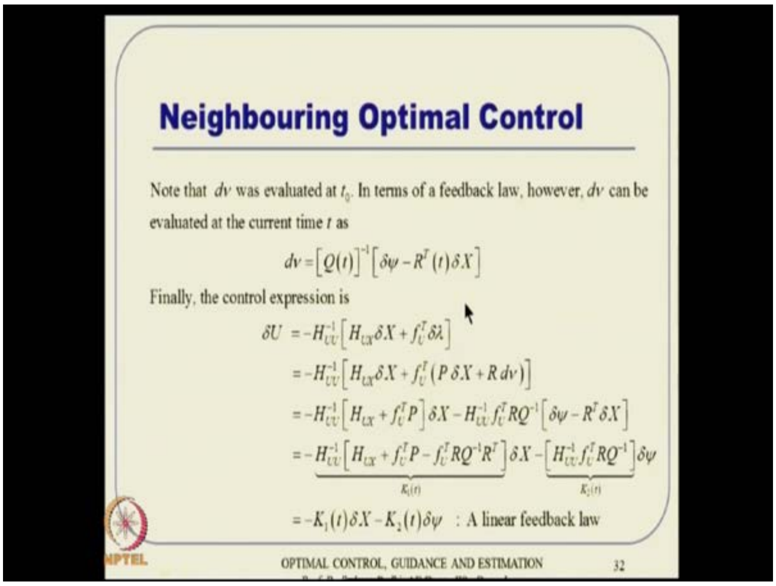
And also result in sort of equation like this and these two equations if you analyze together then this will result in this equation. Remember, this equation I mean R dot coming out of here and R dot coming out of here will be same. So, you have this P dot R dot and Q dot and P t f, R t f and Q t f are given to us. So, these three differential equations and final boundary conditions, again the idea is to propagate it backwards and then store it and use it in something like that.

(Refer Slide Time: 47:43)



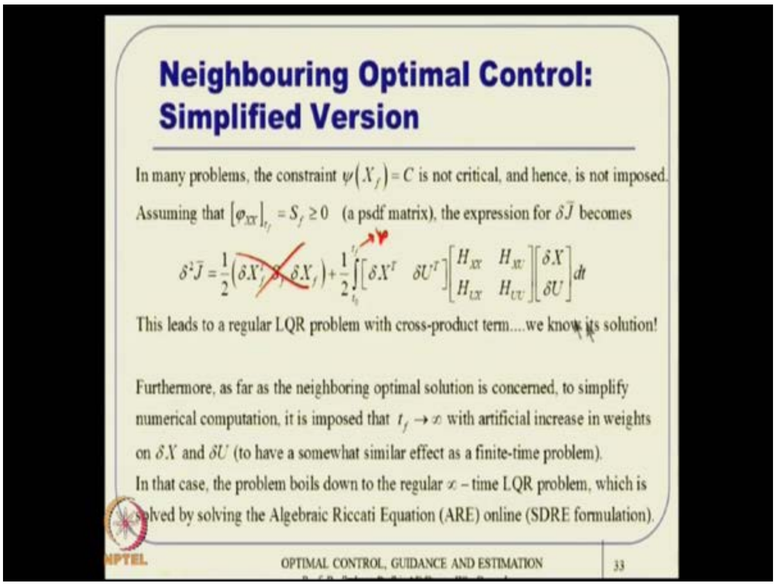
Then we are ready now with the solution and finally, d nu can be computed as initial condition value inverse sort of thing and then d lambda 0 can be computed that way. Also, there is a small point to note here that Q t naught should not be singular and if it happens to be singular then this optimization problem is set to be abnormal; something wrong here. The problem if formulated well, then Q t naught be will not be singular. So, with that assumption then you can formulate on delta lambda 0.

(Refer Slide Time: 48:15)



And then d nu also we can calculate and finally, d U will be represented as all these like that. Ultimately, it will result in k 1 times delta X and k 2 times delta t. Sorry, k 2 of t times delta psi, so that means, delta U is a function of delta X and delta psi and that is what our objective is. We wanted to compute the delta U, so that we can add it to the nominal U that is available and we can go ahead.

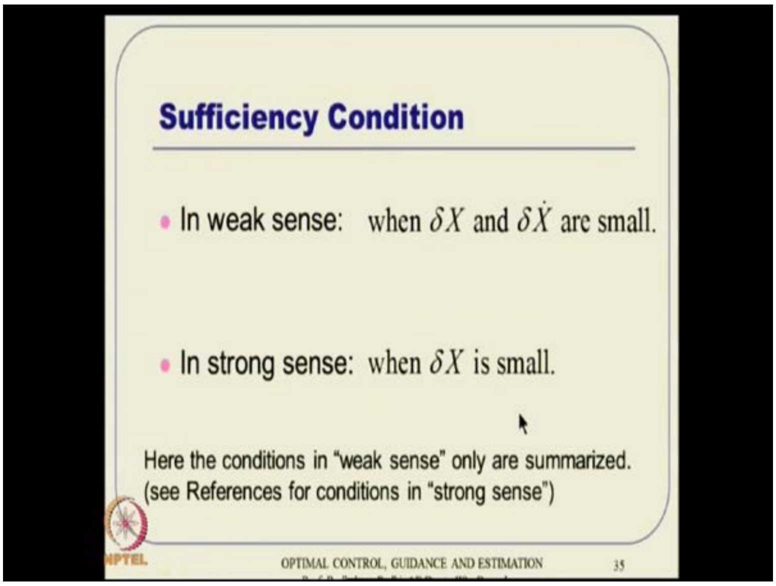
(Refer Slide Time: 48:46)



So, in many problems it turns out that this is not critical; this has to comment, a simplified way of implementing and all that. So, it turns out that this is not critical and hence is not imposed and you assume that this also goes to a psdf matrix and things like that. So, delta square J becomes like that. Essentially, it happens to be L Q R problem in cross product term of which we know how to get the solution. So, if this constraint is relaxed then all these exercise what you did is not really required in a way because we know how to solve this cross product of cost function and then you go and solve it.

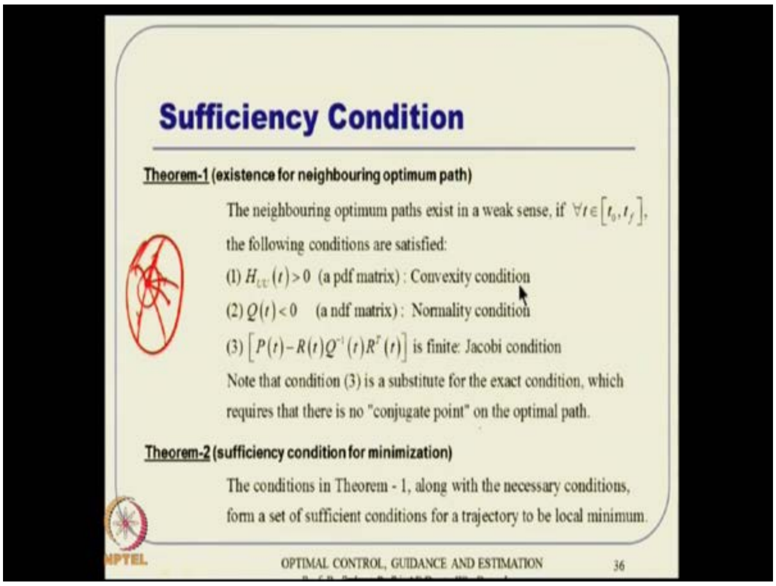
Remember, that t f has to go to infinity, I mean when t f goes to infinity, this will not be there. That becomes even more simplified. Anyway, furthermore this is what when t f goes to infinity and things like that, we can artificially increase the weights and do some engineering solution instead of doing through all these mathematical things and all that and you can also think about solving the Algebraic Riccati Equation online. Essentially, it goes to this S D R E formulation, which we have discussed before as well. Now, before ending this lecture, I will just touch up on this idea of sufficiency condition and we have been kind of or must have been kind of curious and all. Now, all the way we have been talking on necessary condition and things like that, but what about sufficiency condition. Also remember this sufficiency condition is for local optimum only. We are not talking about global rate at all here, but still what are the necessary I mean what are the sufficiency conditions after the necessary condition are all satisfied?

(Refer Slide Time: 50:22)



So, there are two, just a very quick summary, there are two ways of doing things. one is in weak sense other is in strong sense. When somebody talks about weak sense, we talk about various signs of delta X and delta X dot. Both are small and in strong sense only delta X is small, whereas delta X dot can be large. Even, if that can be guaranteed then that is called strong sense and if both happen to be small and still you guarantee the optimality that is called weak sense. So, we talk a little summary of the weak sense; strong sense, I will live it to self reading.

(Refer Slide Time: 50:57)



So, the theorem one tell something like that; existence of neighboring optimum path, this is the conditions, the neighboring optimum paths exists in weak sense from t naught to t f, if the following conditions are satisfied, that the second derivative Hamiltonian with respect to U, del square H by del U square. That is guaranteed to be positive definite matrix all the time and that is called Convexity condition. Then Q as to be a negative definite matrix and that is called Normality condition. At this Q, remember this is not the design Q and all, actually we have come up with the Q inverse and this is that particular Q.

That Q dot, the solution of that, this final condition and this differential equation whatever turns out that is this Q of t. that Q of t has to be negative matrix and that is the normality condition. Then this one, this matrix what you see P of t minus R of t Q inverse R transpose t is finite; it cannot be infinity and that is called a Jacobi condition. So, if the convexity condition, normality condition and Jacobi condition are satisfied then you call the… then it is guarantees that the neighboring optimum path exists.

Essentially, the condition three, the Jacobi condition also talks about there is no conjugate point on the optimal path. Now, conjugate point is something, in very simplistic sense if somebody wants to understand is like this problem. You have something like this, let us say, you talk about a sphere, something like these kind of a sphere and let us see you start a point a to point v and then probably you go to kind of north, minimum path from point a to point v you have north pole and all.

Now, at North Pole, something like I mean wherever you go in this direction, it turns out to be the equal distance; if you talk about a minimal length path of that thing, so that means, that there is no clarity of which direction to go after north pole. There are several ideas, there are several paths, infinity number path really where you can go to the same distance and land off with the same value of the distance travelled. So, that is the kind of a conjugate point and this Jacobi condition tells that there cannot be a conjugate point on the way. So, that is what it is and this theorem two tells us a very simple way of extension of that really, which tells us that a sufficiency condition is nothing but, conditions of theorem one. It is the condition in theorem one, along with the necessary conditions also as sufficiency condition do not mean too much without necessary condition.

So, you have necessary conditions as well as a sufficiency conditions and that is a kind of guarantee, some sort of sufficiency condition for that problem; that means, the condition of theorem one are these one, two and three conditions as well as necessary conditions form a set of sufficiency conditions for a trajectory to be local minimum. More and more details and more extension you can see some in reference book and follow up with some mathematical optimal control books and all that to get more ideas on that. We will not talk on these as this is typically kind of a engineering flavor, even though we talk to sometimes the theoretical details and all, will not go to the this mathematical control analysis to too much.

So, here I will not venture out to this further ideas and things like that and also will like in strong sense and all if somebody is very curious to see things like that, we can always refer to Bryson and Ho and all other books. So, that is what I thought, on the way it is good to have some idea about a neighboring optimal control and sufficiency condition and things like that. So, kind of summarizing it about, again these are not extensive at all. Anybody, interested can see it especially Bryson and Ho to start with and then we will learn on that.

So, this is what I wanted to cover in this lecture. Just before that again let me summarize that this particular book in a very seminal book. You can see many of the things from the Bryson and Ho. other things are also there; especially the Stengel book is very readable. You can read many things and understand what is going on. These frequency domain concepts, something like L Q G and L T R concepts, if you are more curious, you can see in this book. So, with that comment let me stop here for this lecture, thank you.