NPTEL

**NATIONAL PROGRAMME ON
TECHNOLOGY ENHANCED LEARNING**

IIT BOMBAY

**CDEEP
IIT BOMBAY**

ADVANCE PROCESS CONTROL

Prof. Sachin Patwardhan

**Department of Chemical Engineering,
IIT Bombay**

Lecture No. 23

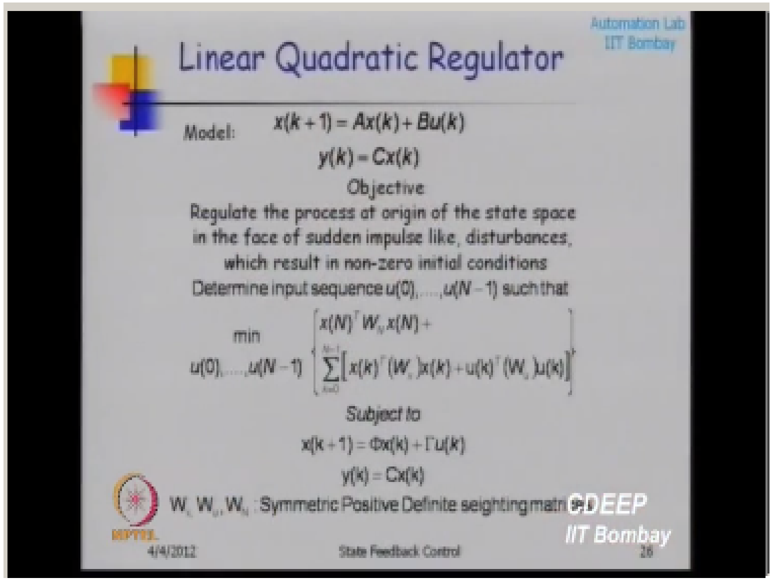
LQG and MPC

Sub-Topics

**Linear Quadratic Gausian
(LQG)Regulator Design**

Okay so in the last lecture we are looking at --

(Refer Slide Time: 00:25)



Linear quadratic regulator problem so you have taken the most basic model in the beginning which is the model which is 3 of measurement noise any state disturbance somehow the initial state is not equal to the origin and the problem is to move the question from a non zero initial state to zero initial state that is your simplest of the problem that we are able to look at that approach and slowly relax all the assumptions and then move towards the problem which is completely inversely consider disturbances not only just simple disturbances.

But different disturbances and talk about a regulation at any obituary set point so all that will do okay but let begin with the simplest problem so this is a right now determines the module a, b and c are known to me okay let us assume that system is reachable which means I can move this thing from any initial condition to any final condition and now I want to find out a control law okay that is moved at distance from initial point to origin that is the problem right now we say that is problem is now can be holder in optimization problem.

The reason for coursing at a optimization problem is to hold one is that optimization prime work particularly this one which we are going to develop will ensure two things it will also ensure the stability performances the optimization objective function allow this to specify performance so here this term here is square of the distance from the origin this is better nom of x vector with waiting matrix w is typically a diagonal matrix and it is used to do scaling of the variables that is one thing second thing is also used to tell which error should go to zero part there are two purposes that is curve one is you know difference state can have different numerical value.

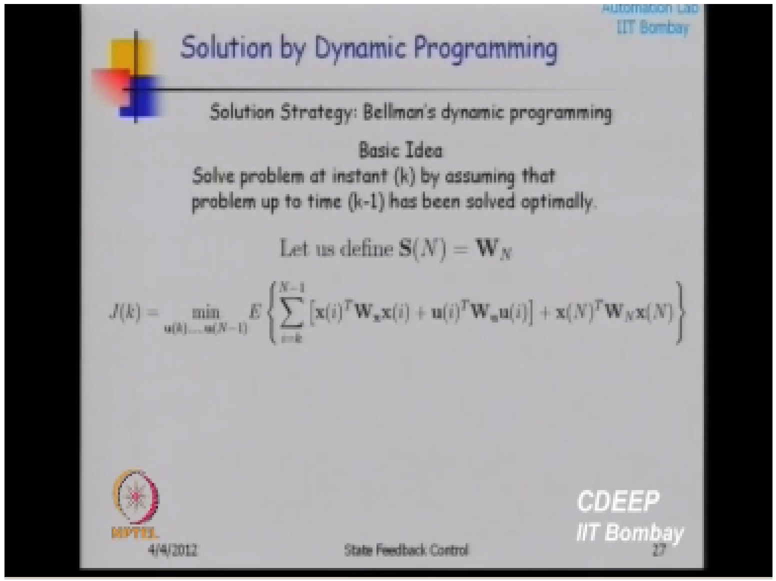
So you can make them same scooting by multiplying by a scaling process second thing is you can specify relation is important to give higher rate okay that means that error should go to zero part okay so in systems where state has physically you can specify that this particular you know this concentration should go to should be controlled faster some level we do not care if it is oscillating relation base.

So you can actually do trade off by specifying this waiting mater same is to about this second term here second term is trying to weigh the input effects okay you do not want very large input effects also you have trade out input you know some inputs are costly to manipulate some inputs are cheap to manipulate for example it is quite likely that if I am manipulating in certain scheme and cooling water and a coolant water it might be people to manipulate to cooling than manipulating scheme because scheme you know energy cost cooling water just pumping cost you know.

So you might use cooling water liberally but not in scheme liberally so all those things can be actually incorporated by using this matrix w, u this is again typically a diagonal matrix this is not going to be a so diagonal matrix diagonal elements are all positive these are all positively different matter okay and they will give you relative importance of each or you know if you hear it other way round if you put large weight on a particular input that input will not change too much okay that input will not change too much.

Because your diffusion variable are u suppose u1 first component of u I will large base associated with it in matrix wu then the first component will not change too much because if you change first component the cost function will change by larger amount okay so you specify relative importance of each manipulate variable all these things are very difficult to do when you do poles likes matrix you cannot say place the poles such that input one is not changing too fast but input two is changing fast all these things becomes very difficult coming to poles of design only this is possible to do this using optima ion framework. Because optimal framework is directly relate to time domain you can very easily give you know translate your control requirement into optimal problem that is not terrify okay so the third term is off course waiting in the terminal on the final state.

(Refer Slide Time: 05:26)



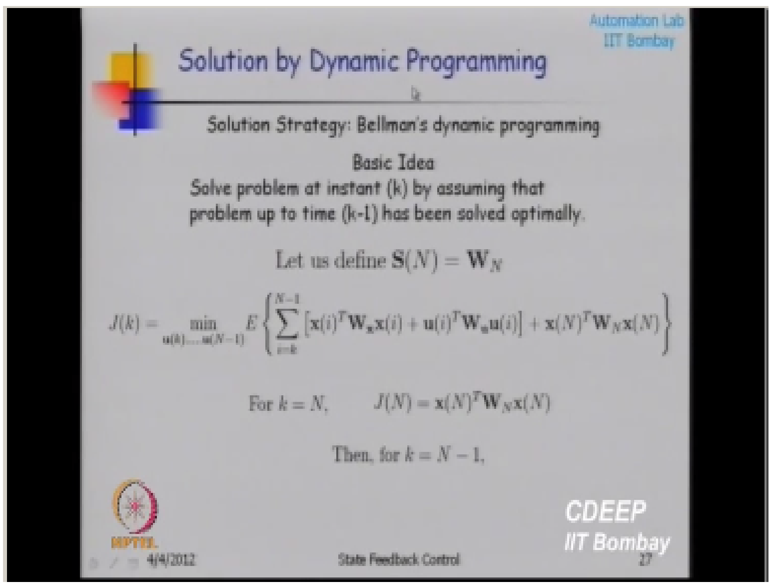
Now is this problem solved okay so solution is found by simple dynamic programming okay so this is a famous method dynamic programming because develop by bellman it has been used vary widely in optimization of optimization based p the idea here is very simple that you work here backward in time so you see here the problem if you go back here so look here this problem is forced over a finite time, time 0 to time n, n is the final time so you want to reach goes to the origin in N factor N is not equal to state dimension the N is some number okay.

So this problem is goes for a finite time okay we will then relax this also will then because the real systems you know that they come from a chemical plant or a power plant in does not work I mean in reality it works on finite time but you cannot develop a controller it will be for your one year and solve it so here right now even though n has been used you will first develop a method for n and then will say n goes to infinity and derive asymptotic device okay that is what is going to be done.

So what is done here in dynamic programming you start from the end sample okay the last sample you choose an optimal input for the last sampling having chosen in optimal input for the last circle you move back in time to n-1 okay then choosing optimal input for n-1 having chosen for n and n-1 you will back in time to n-2 okay in optimal state estimation in bellman filtering we are moving forward into okay now here you are specified you know the final time and then you are moving backward in time.

So you have to have some patients before you finish okay I am sure I can see a question on a face how can you specify a final time well displayed look at the development and then you will come back to okay now let us define this quantity this matrix s okay this is a matrix x which is defined as wn what is wn? Wn here is this weight here on the terminal okay so wn is the positive definite matrix it is a weighting on the final state okay that is after n times that is the

(Refer Slide Time: 08:50)



So I am starting with this algorithm with wn and then look at what I have to solve I have to solve I want to minimize these jk okay starting from time k to time n-1 okay I want to choose time k to time n-1 summation see this problem has been defined from k=0 to k=n-1 and now I am starting at some point k okay so at k=n only this last term remains which is it see I have this summation see I have this summation here go back here I think there is one index is around it should be k=0 so j it should j=0

So this last term is what remains see if I want to optimize this backward in time okay first I should consider the last term so last term is this term right yeah so the bellman principle is this you start with the last term and you need index to optimism the last term having optimize last term then you optimize last but 12 then having optimize those two you go back ward in time and you need to go to zero that is bellman's that is the method okay how it will lead to a controller design you will see that this tool okay no you actually solve you actually formulate you break it into multiple optimal problem okay.

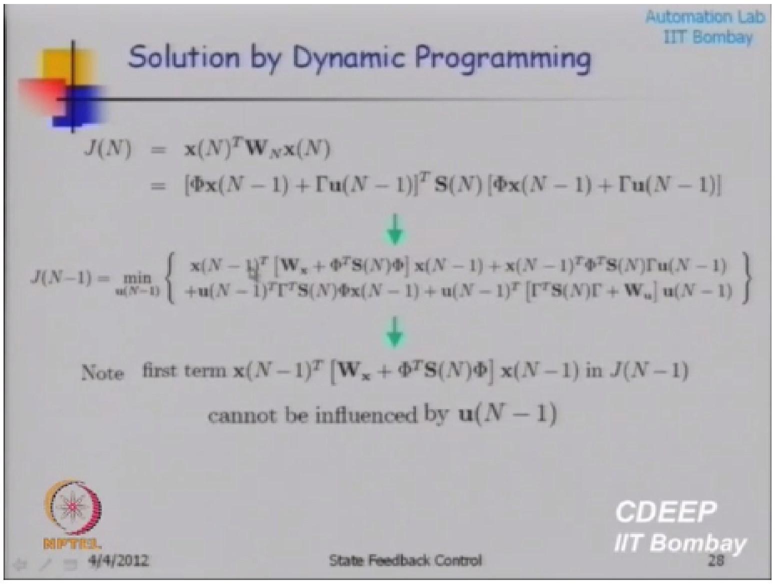
This is one joint optimization problem okay see this problem here is a joint optimization problem this is from u0 to un-1 you have to decide everything okay now un-1 see look at this particular system there is a time evolution thing here why we are going to use this particular concept of elements having difficulties see ui-1 you just tell me if I am optimization with this particular problem un-1 is it going to affect say x0 or x1, x2 it is not the un-1 will only effect xn okay.

So actually choosing an optimum un-1 can be solved independently of u0 to un-2 okay so you fix un-1 optimally I have been un-1 you move to un-2, will not going to again effect thinking the past it is going to affect only future okay so you move backward so you take advantage of this particular fact okay to break this problem one sequence of molar problem okay and that will lead to you to a very legend solution to this particular problem okay.

So what I am going to say here is a so this is staring from any obituary k it can be 0 if k is 0 it excess to the original problem I have written it clear from any orbit arty k why this is written from orbiter case and become clear as we go on okay I am defining this term jk I am defining two thing s and jk, is the problem starting from time k time, time n-1 so this is at so we can say this is at kth sub problem in the kth sub problem.

So I am going to solve actually instead one big optimization problem I am going to solve a connected series of a optimization problem and that will lead you to the solution so I am defining two things sn and jk okay so accordingly to this definition this here what you have here this which is j0 because j0 to n-1 and what is jn, it is the last term okay jn will be only this term okay sorry for jn will be the last term will be the last term yeah okay so what is jn-1 is xn-1 transpose wx+u term +jn right okay

(Refer Slide Time: 14:01)

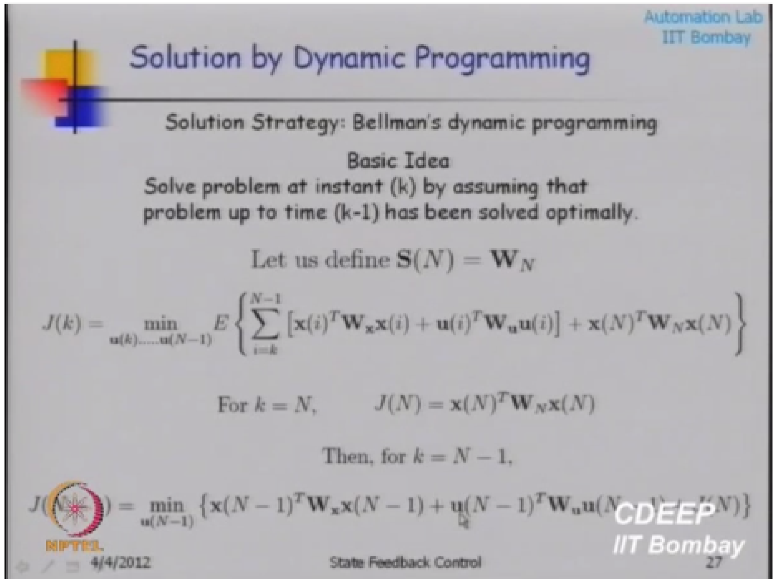


So what I am going to do next so with last term I substituting in terms of un-1 what is the diffusion variable here see what is the diffusion variable for the last term it is only un-1 okay I only have to choose un-1 because un-1 will have effect on xl okay so my jn is now written in terms of un-1, sn is wn you have defined this okay and this is the term that I want to minimize with respect to un-1 okay so if I minimize this now you have to expand this right and what you have done here is here the 84 terms right.

You just matrix multiplication you just multiple and then rearrange and then you will get this you know in between thing if you are not a very difficult to get so this is just a multiplication and rearrangement and the first term if you see here this first term here okay, $x(N-1)^T$ this term this is $x(N-1)^T$ x(N -1), this cannot be influenced by can be in influenced by u(N-1) this cannot be okay.

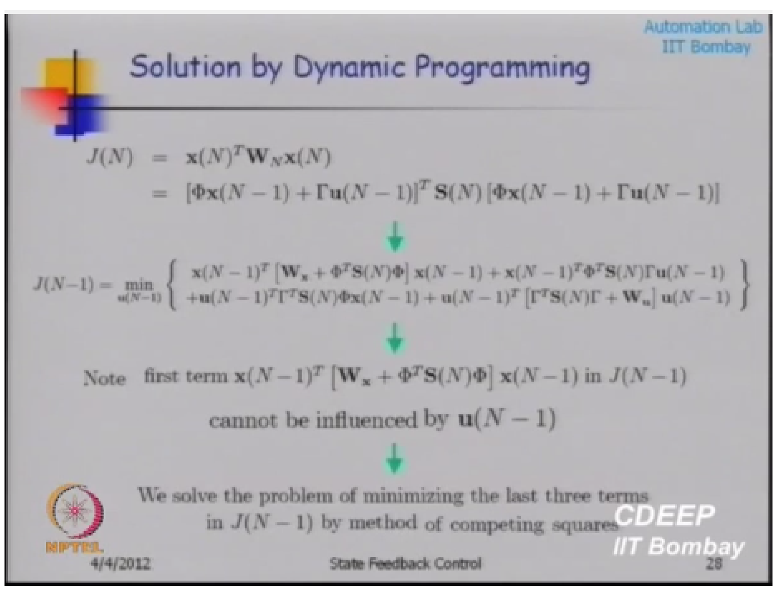
So this term is rules out this cannot be influenced by N-1 only this term, this term and this term this last three terms can be influenced by u(N-1) so actually when you minimize you only have to worry about 1,2, and 3 okay, is that okay. There are four terms x^Tx okay x^Tu, u^Tx and x^Tu so last three terms are relevant first term is not relevant okay, because first term cannot be influenced by u(N-1) it can be influenced by u(N-2), u(N) minus elected different storing it cannot be influenced by u(N-1) okay. Which one, yeah so J(N) I have defined.

(Refer Slide Time: 16:33)



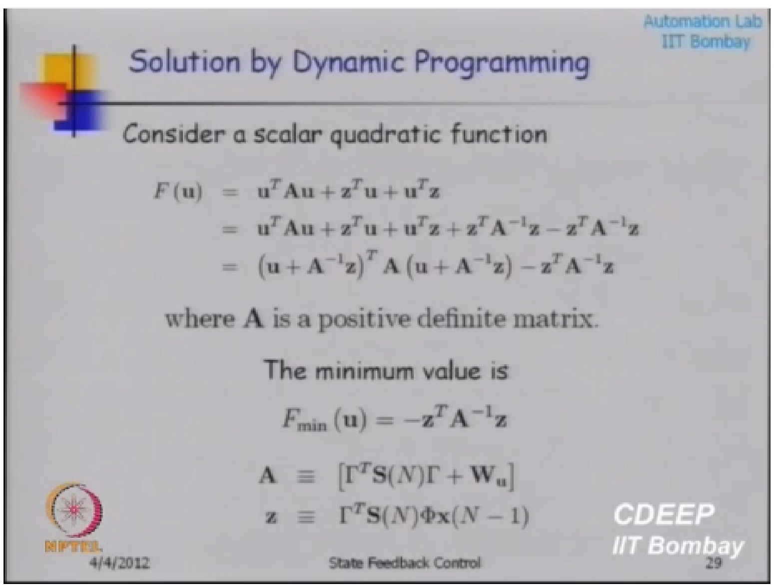
So I am going to define this J(k) loop so the first you define the last term that is J(n) then what is J(N) and -1, J(N-1) is J(N)+one mole term okay, what is J(N-2), J(N-1) two I mean there will be three terms okay, and when you collect from J(0) will be collection of all terms from N to 0. No, no, no right now so this is an intermediate set in the derivation this do not try to write now interpret okay, so right now N even though I am saying it is a final time the system may not have reached the final stage 0 at time and so you may not have become 0 so just see the derivation, okay.

(Refer Slide Time: 17:29)



So this with this J(N) and did J(N-1) is everyone with me on this I have just combined added terms together and club the terms of u(N-1) and x(N-1) they have separated them and then written it like this okay, so this last term cannot be influenced and then how I going to solve this problem okay, I have to solve this problem of minimizing this function with respect to u(N-1) okay, so this is done by a method of completing squares okay.

(Refer Slide Time: 18:13)



So thus keep that J(N-1) be this aside look at this now, consider a scalar quadratic function okay, just look at this steps are you convinced here or has been done, just see the just try to derive this can you derive this, see this form what I have done is I have written this abstract form here see what is the abstract form x^T something into u, u^T something into x, u^T a positive definite matrix into u, so there are three terms u^Tu, u^Tx, x^Tu okay, so I can now think of.

See this is I get multi dimensional quadratic equation okay, I want to converted into a complete square okay, if I can write it as sum of 2^2 okay, then I can use that to solve the problem that is the idea okay, so just see whether you can derive this, just see if this result can be derived just check, let us do the algebra. So if you have a scalar function in abstract form see that A and what is A matrix and all is very, very compression okay leave it aside right now.

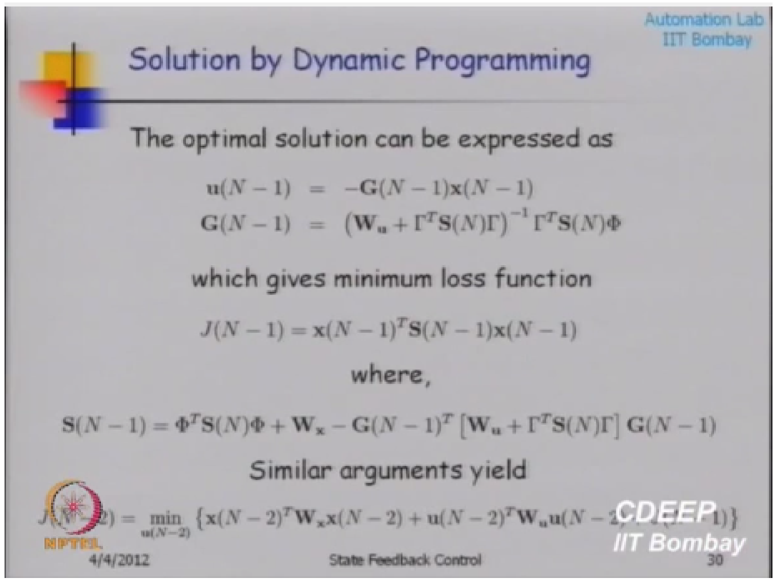
Just see whether you can derive this results okay, u^TAu+z^Tu, u^Tz so second step is very obvious the third step would require expansion, the third step requires expansion, okay. how will you find a minimum of this function, got me no, no, no see this is, is this see what is A, A is a positive definite matrix that is very, very important so what about A^{-1} A^{-1} is also a positive definite matrix, okay so what is the property of the positive definite matrix, x^T any vectors transpose A into that vector will also be a positive continent, okay.

So this is the positive quantity, this is also a positive quantity for any x any u this is always a positive quantity if I virtue of the fact that A is positive definite, what about this guy this is also positive continent okay, what is the lowest value this function can take you know, what is the definite variable my definite variable is u, v is some vector see actually here map what will you map, you map u to u(N-1) and if entire thing $x^T\phi^TS(N)\Gamma$ that you can map to G okay, that you can map to G, so we will see what is that mapping, okay.

So what is the minimum value it can take, the minimum value it can take if I am, I have to do minimize this function with respect to u it is a function of u, z is some vector which is constant yeah, so if you can put this equal to 0 see if you put this term equal to 0 this is this A is a positive definite matrix so the smallest value this term can take is 0, when will you take it 0 only when this vector is 0 okay, so the minimum value is this okay, that is when you choose u= yeah, $-A^{-1}z$ okay, so now this do this mapping choose A to be this matrix $\Gamma^TS(N)\Gamma+Wu$ okay.

This is the matrix which appeared in u^T part okay, and see z $\Gamma^TS(N)\phi x(N-1)$ okay, if you do this what is the solution A^{-1} so inverse of this matrix okay, into z, z is do you get this A^{-1} A is a I mean A always be invertible this A will it be always invertible why, S(N) is Wn okay, Wn is a positive definite matrix so is A always a positive definite matrix, S(N) is a positive definite matrix $\Gamma^TS(N)\Gamma$ is a positive definite matrix, Wu is a positive definite matrix okay, so addition of two positive definite matrices will give you a positive definite matrix, okay. So this is, this A is a positive definite matrix so A is invertible okay, and you get this.

(Refer Slide Time: 23:46)



So my controller can be written like this, this is A^{-1} see look at this here, look at here A^{-1} okay times this whole thing into x actually what I am getting is a state feedback control law, what I what you to derive as a state feedback controller okay, I got state feedback control law for one particular instant so what was this particular matrix, this $A\text{-}A^{-1}$ into this matrix is the gain okay, is the gain and x(N) is the state feedback controller okay, is everyone clear about this.

So which gives minimum loss function to be equal to this, if you actually substitute and find out the minimum of J(N-1), J(N-) will turn out to be this where S(N-1) is actually I have just done the algebra okay, you can go back and substitute and see how it works. Yeah, so what I have done here just go back and see here, see I had this I wanted to minimize this function I want to minimize this function okay, so I am just doing it by a completing the squares, okay.

I am not worried about this part because this first term is not going to be influenced by Wu okay, so to find out a minimum for the remaining three part what I have done is there I have expressed it as sum of two squares this is one square this is another square, see this is always a positive number this is always positive number okay, the smallest value this can take is 0 okay, and then I just choose A to be this and z to be this okay, this gives me a minimum of the original objective function I go back and substitute the optimum value of u and then so this is my state feedback, this is my optimum u(N-1) okay, which is this matrix times x(N-1) okay.

Now this is the result which you would have got given by minimization, if you have done τ/ τu you would got the same result okay. Why do we need Bellman principle will come to that, so this is a local result for one you can get by minimization Bellman principle talks about the series of connected problems okay, so that where so if you actually take the solution what is the minimum value of J(N-1) substitute the solution into the you know that J(N-1) expression.

You will get this, but J(N) and J(N-1) I will combined into one because J(N) and J(N-1) both will be influenced by u(N-1). See u(N-1) appears in two terms okay, it appears in J(N-1) by virtue of the fact over definition of the objective function it appears in J(N) because u(N-1) influences x(N) okay, so actually if you go back here, see J(N) is only this but u(N-1) appears in J(N-1) because the way optimization problem is defined.

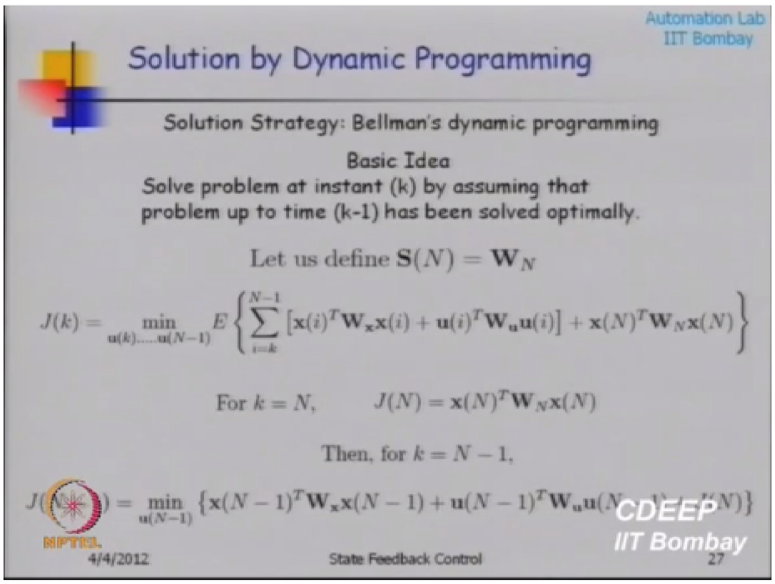
See you just, see the optimization problem is defined with respect to, see what is J(N) only x(N) okay, what is J(N-1) this also contains this term so I have grouped all the terms that contain u(N-1) together okay, so that is the reason I looked at J(N-1) okay and I optimize J(N-1) with respect to u(N-1) okay, and that optimum is optimum u(N) is formed to be this okay, I am going to call this particular time varying matrix as G(N-1) okay, this is my notation.

And when I substitute this particular solution into J(N-1) I get this quantity okay, this is pure algebra I just took the solution substituted there whatever term I get okay, that I am calling as x(N-1) as I am calling as x(N-1) my intension here is to derive or recurrent relationship between S(N-1), G(N-1) do you remember what we did in Kalman filtering we had covariance and we had Kalman gain and there were coupled with each other okay.

Kalman gain okay, is used is based on the covariance updates okay, then you again update the you have create that covariance Kalman gain calculation updated covariance again you do Kalman gain calculations created covariance as it goes to the cycle okay, so likewise here I want to derive a recurrent relationship in time okay, which are again called as Riccati equations except now it is done in the context of control other than the context of you know state estimation, so the okay, so now having optimize J(N-1) you move to J(N-1) you are moving backward in time okay.

So now I want to minimize this term here J(N-2) with reference to u(N-2) because I have see this term here is influenced by u(N-2) okay, and here this is influenced by u(N-1) okay and this has already been optimized except there is one thing here which is influenced by u(N-2) what it is x(N-1). See look at this, look at this, look at this relationship here I want to point out that they are actually analogous see here this is x(N-1), x(N-1), x(N) so this term will appear here earlier we had.

(Refer Slide Time: 31:16)

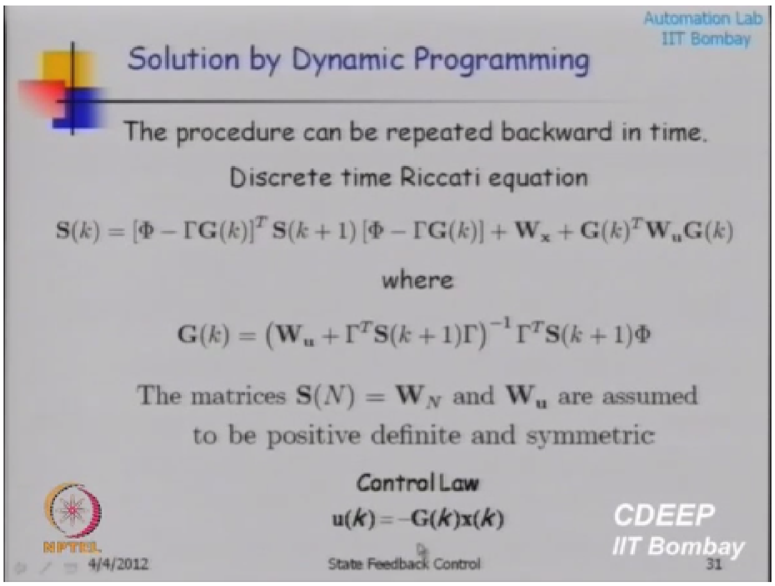


See earlier you had x(N-1) and J(N) okay, you just shift and time okay, and you have already chosen u(N-1) so now only you have to choose optimally u(N-2) okay, so algebra is multi, you know and if you sit down and do it, it is not difficult to do this algebra but just understand the concept if you are getting stuck anywhere please stop me, okay. So again okay, again see this expression looks very, very similar except this matrix here has become little complex, this S(N-1) is now a very complex expression you see this okay.

Apart from that see if you accept that somehow this S(N-1) is coming from the one side bad in time okay, then this expression is loss of equally same, I mean it is the same expression okay, so you have this term and then you have N-1 what is in the N-1 that you can influence using u(N-2) x(N-1) okay. Now see earlier what was the case even here then it was J(N) it was x(N) into $x(N)^T$ some matrix $x(N)^T$ okay, so u term was not there in the last term.

Here also now after you implement this control law u term is not there you just have this optimal you just have this optimal J(N-1) which is where this S(N-1) is given by this big matrix here okay, so now how do I minimize this again same thing complete the squares okay. Now using this term okay, if you complete this squares.

(Refer Slide Time: 33:12)



You go back in time okay, like this by completing the squares you get what are called as Riccati equation okay, well Riccati equations are moving backward in time okay, so where G is given by this and S(N) so your controller use found like this you start on S(N) okay, S(N)=W_N okay, then for that you find out G(k) that is your control that is your controller J(N) at time instant N, N is the last time in your optimization problem, okay.

So you find this G(N) okay, for the time being be aside the fact that how will you implement this just looked it as a you know optimization procedure okay, we first fix u(N-1) having fix u(N-1) we go and fix u(N-2) having fixed u(N-1), u(N-2) we fix u(N-3) so we go backward in time okay, so actually we are what we are getting here is a series of time varying gain matrices, you will do not get a control law with a fixed fit gain you get see just like in Kalman filter we got gain matrices that are changing as a function of time, okay.

So here to you get a sequence of gain matrices whether changing with time they are coupled with each other through these Riccati equation okay, through this Riccati equation except that you start from N that is your last time and you go backward in time and you will get a series of gain matrices okay. Now a practical problem you might ask is that, what do I do I mean suppose I am running some plant okay, let us say I want to take drive a motor automatically on some road and every one second I will take a decision and I have to go from here to the other end and if takes the you know one hour or two hours you go to a one hour to go to the other.

So do I actually compute 3600 gain matrices do I save them and then at each time point I just retrieve one gain matrix multiply and find the controller that is impractical okay. So we will sort this problem out do not worry about this right now yeah, I finally want to read $x(N)=0$ no, no, so now aim in this particular case is not $x(N)=0$ in this particular case is to make $x(N)$ as small as possible very close to 0, asymptotically I want to go to $x(N)=0$ this is called any arbitrary reaction.

Now what we will do they will say that under certain conditions you can find a, see this is a coupled equation okay, this is a difference equation see why it is a difference equation k depends up on k+1, see normally you are used to difference equations which move forward in time this difference equation moves backward in time okay, this difference equation move backward in time.

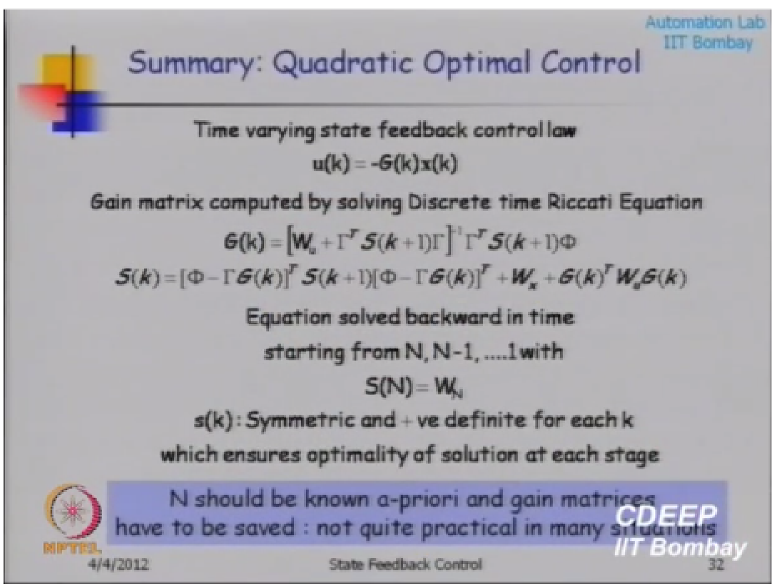
What is elegant about the solution is that we have a close form solution okay, you get sequence of optimal gain matrices okay, and you get a close form solution. Now this is not practical because many times you do not know how many, suppose you are going from here to the other are you want to you know automate your car and leave it on the road you cannot predict what time it will take, whether it will take one hour 20 minutes or two hours depends up on the road condition.

So you this is not unless in some cases if you are you know hitting some target and you will know you are going to hit a particular target in so many second then probably you know you can be calculate all the gain matrices keep them and then it will you will know, but this is not practical but this gives you a recurrent formula and how will I use this next to we come to your next.

So let me just summarize this first what is the, how do I calculate this so my feedback gain is $G(k)$ and then $G(k)$ is calculated by this recurrent relationship these are appearing here in the this difference equation back in time and then what is guaranteed in this $S(k)$ is always a symmetrical positive definite matrix, because you start with S(N)=Wn this is a symmetric positive definite matrix and the way the whole thing is constructed you are guaranteed that this S(N) matrix S(N) will always be symmetric and positive definite.

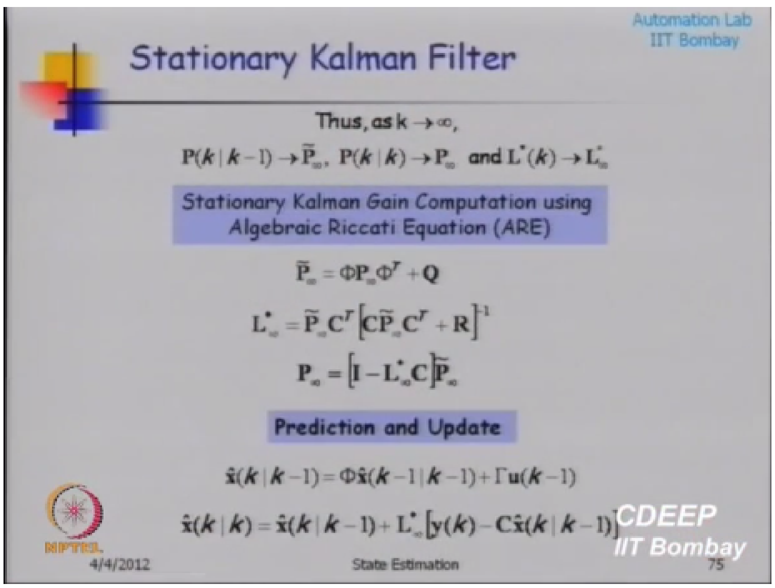
But the problem here is that N should be known a priori how many N, how many such gain matrices you find out okay, so but now for the time being you get a ϕ, you got a way of calculating the optimal solution okay, we will find the asymptotic solution of this particular problem by leading N go to ∞ okay, we will find that solution and we will use that solution to I just want to draw parallel to EKF do you remember what we did in EKF, in EKF you know we had this solution which was last time solution or infinite solution.

(Refer Slide Time: 40:11)



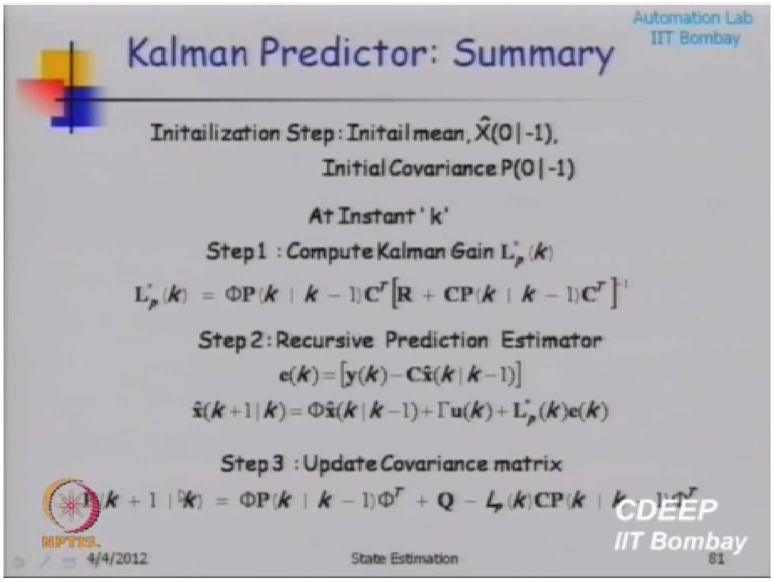
Okay, so in this case you just said that this goes to ∞ and then time goes to ∞ as k goes to ∞ the solution converges to a steady state problem okay, and then you can get here you can get L^*_∞ okay, you can get this infinite time gain which is only one matrix okay, so this is that you do not have to compute L(k) matrix and every time instant so after sometime when you program this you will realize that after sometimes these L(k) goes to a steady state you do not have to actually keep computing it every time all these matrix equation.

(Refer Slide Time: 40:48)



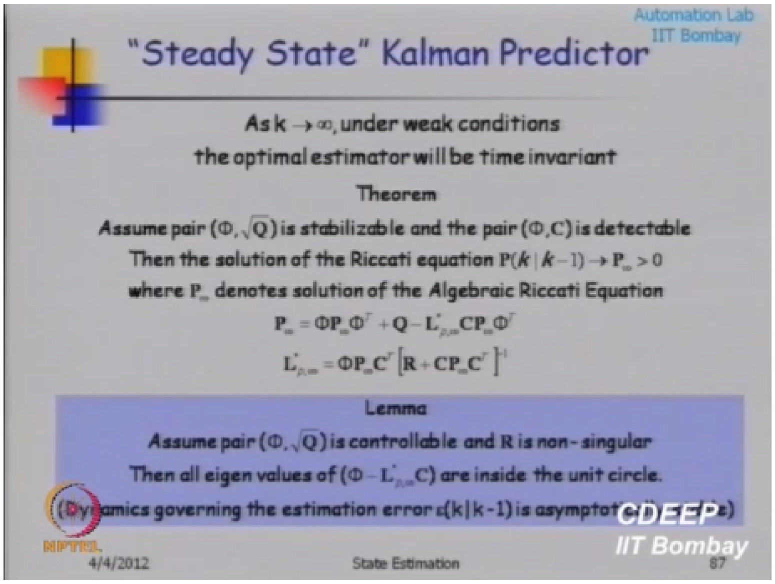
So they are under certain conditions of course on observability and so on.

(Refer Slide Time: 41:01)



So I had talked about Kalman predictor, remember Kalman predictor in Kalmam predictor we had two equations, we had gain update covariance update right, we had gain update and covariance update and then this Lp*, Lp* was the optimal gain and then we computed this gain for moving forward in time and then we had of course.

(Refer Slide Time: 41:31)

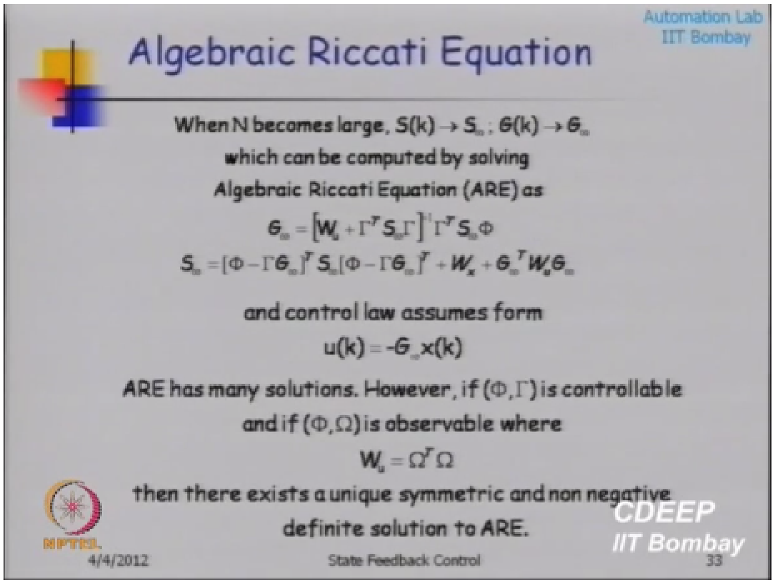


Conditions under which you know you have this coupled equations and you can solve this algebraic Riccati equation you can solve them you know the solution exits and I talks about this lemma you know which tells you under what conditions the solutions exists, when this $\phi\sqrt{Q}$ is controllable and R is non singular lemma unique solution exists solution exists such that Eigen values of this matrix are inside unit circle.

So we had solve this problem earlier and I just want to draw analogy here see there are two things here this covariance update or covariance equation and a gain equation and they are coupled okay, same thing is happening here if you just look carefully at least this LQG notes as in upload today okay, see here there are two coupled equation this is something like covariance update which is something like covariance update this is I mean qualitatively this is similar to covariance update this is similar to gain Kalman gain update.

So there is a gain calculation covariance calculation similarly here this S(N) calculation and gain calculation okay.

(Refer Slide Time: 43:05)



So now what I am going to do, if you lead S(k) go to S∞ which what happens is that G(k) will go to G∞ this will happen only under certain conditions of controllability and you know like we had conditions for algebraic Riccati equation for the Kalman filter same thing should be equal to here, so you can show that under certain conditions these couples are the equations have a solution and this equation is actually G∞ because let call it G∞ this gain, so now if you solve this coupled equations you have one gain matrix which is optimal when N goes to ∞, okay.

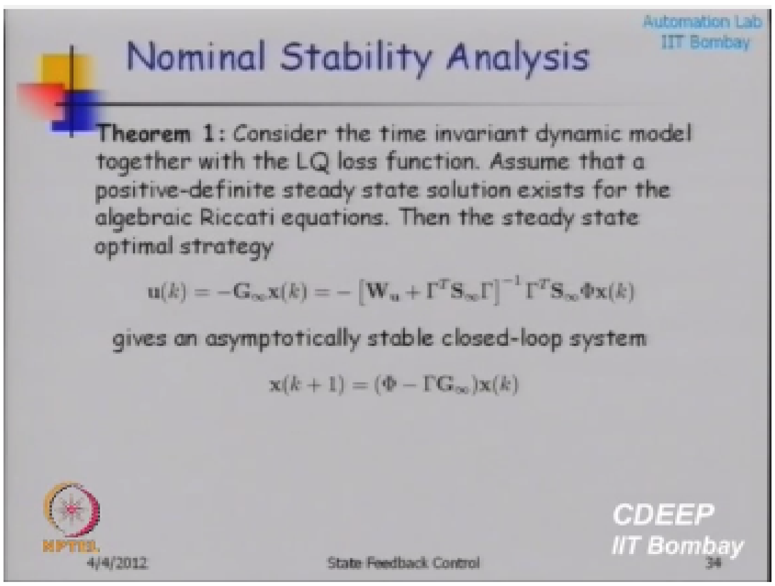
And for practical implementation all that derivation which I did I am not going to use, I am going to use this algebraic Riccati equation find out G∞ and use it for my control okay,so that starting with last point going backward in timer all that was in between step, all that where in between step okay. Now that I have derived it I am interested in this infinite solution okay, and this infinite solution is what I am going to use for designing my, so this is the k feedback controller which I have now, okay.

How do you solve this, this coupled equation can be solved under just like you has conditions there, you have conditions here okay, if you consider if ϕ and γ is controllable there we had a condition of observability ϕ and C where observable here you have ϕ and γ is controllable okay, and this matrix Wu matrix okay if you take this square root of Wu matrix and if this is observable yes, so you take square root of that okay this is called square root this ω is called the square you know about LUD composition, a positive definite matrix can be written as $L^T L$.

So this it is a diagonal matrix so it is biggest square root Wu will typically as it be a diagonal matrix so this should be just a diagonal matrix that, okay. So if this pair is observable and if this pair is controllable okay, then you can guarantee that there exist a unique solution to this particular problem so this problem have a unique solution and you will get a unique gain matrix G∞ which means stabilize the close loop, okay.

Which will bring the system from non zero initial condition to zero initial condition okay, which will ensure that the Ω of.

(Refer Slide Time: 45:56)



See if I substitute this control law what is the closed loop solution, what is the close loop behavior is this okay, is everyone with me on this. correct so now I will prove stability okay, I will show that asymptotic step this using this controller is equivalent to I will prove asymptotic stability with this controller okay.

If it is asymptotically stable it should go to 0 yeah, so well while doing the derivation whether I started by showing x=0 or not is a different story all that I say when I started a derivation that extension should be as small as possible and minimizing your objective function with respect to use okay, so my objective there was to choose inputs such that x(N) is a smaller problem okay, I did not said it equal to 0 okay, I did not said equal to 0 I just said as should be very small.

No, no but I do not have to choose now no, see I have just now got treat of that x(N) business see now I just going to say that N goes to ∞ let N go to ∞ okay, then I actually have to solve this problem this couples equation okay. So that x(N) and all that business program intermediate step to derive this algebraic Riccatic equation okay, solution of the algebraic Riccatic equation is going to give me $G\infty$ which I am going to use as, as my control law is everyone clear on this.

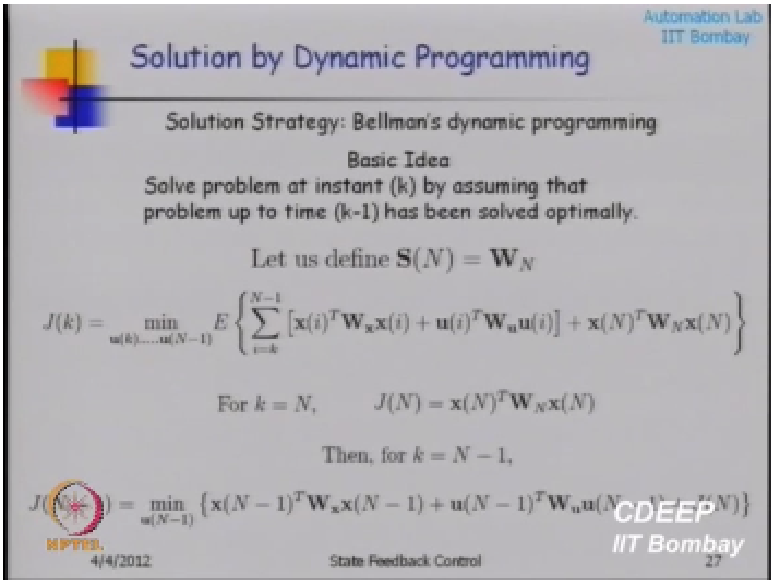
All that dynamic programming and you know all that I did just had an intermediate step to derive this particular control law okay, but this is the state feedback control law your state feedback control law okay, yeah no that is you know for existence of solution of that particular equation you have to have two condition matrix you can go and check this proof in the one of the standard text which has been listing at the end of the, so the proof is very elaborate I do not want to get into the flow.

So this is a I am just coding the, I am not written as a theorem but in any book this will be a theorem which talks about conditions under which the solution exist for the algebraic Riccatic equation. See this is not matrix coupled matrix equation it is not an easy equation to solve okay, so because these are this G is a matrix S is a matrix and these are coupled equation so solving this matrix equation is a tuff problem.

And then there is been lot of borer call how do you solve such Riccati equation, so you can just I refer you to the books. Yeah, so let us see the composition any W you can write so these things used in MATLAP and you have function called it OSP, that is standard. See the nice thing about all these things if I just use linear algebra so ultimately just algebra, linear algebra simple linear algebra that you applied you can view this entire control theory as you know in state space τ domain as applied in linear algebra. So is everyone is clear with this now you know u is $G\infty$ times this so my close loop become this okay, is this clear.

I do not see lot of convince people I can go do it again you tell me where you are stuck which part you did not understand. See let us look at like this I started doing this derivation let us go over it again.

(Refer Slide Time: 50:36)



So then what I do is once I do this I ignore the fact that gain is time varying I just look at J(N) at as N goes to ∞, I am just finally interceding deriving the algebraic Riccati equations for that I am doing these are the intermediate shape, how do arrive at the algebraic Riccati equation so I am starting with unsteady state case okay, where N is from intermediate point you are not reached the originate okay, N is some intermediate point okay.

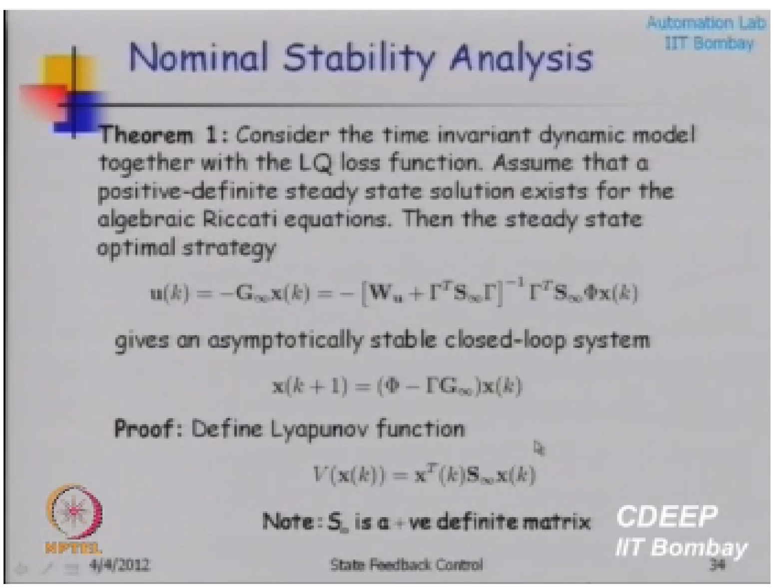
I am not specified N okay, I use this only to derive this recurrent relationship so what is the recurrent relationship, the recurrent relationship is between this I mean finally you know you got, let us go to the yeah, so you I just wanted you to derive this recurrent relationship between the gain and this positive definite matrix S(k) this is not covariance here but you know in the Kalman filter derivation this is somewhat similar to the covariance update, so this matrix update and gain update that is how you got this equation.

And then finally what I am interested is not in this time varying solution I am interested in the solution which is as N goes to ∞ okay, so I am going if it is in the steady state solution and then finally I am never going to implement a control law like this where G(k) is time varying I am interested in this control law where G I am interested in this control law okay, so all that dynamic programming I have just used as an intermediate step to arrive at this algebraic Riccatic equation, algebraic riccatic equation give me okay.

See now here you know going back in time everything has vanished because here when the solution is converged S(k+1) and S(k) are same right, when the solution has converged they are going to send between S(k+1) and S(k) so that is why any state solution is nothing like moving back in time or moving forward in time is the steady state solution.

So I am just interested in finding out the steady state gain I have used that final end as a vehicle to arrive at this steady state solution okay, and then I am stating you the, giving you the conditions under which the steady state solution exists okay, yeah. So many first Q sample it is not but as if you take N to be very large then it is optimal okay, so this G∞ is optimal G∞ okay, so this is my close loop solution.

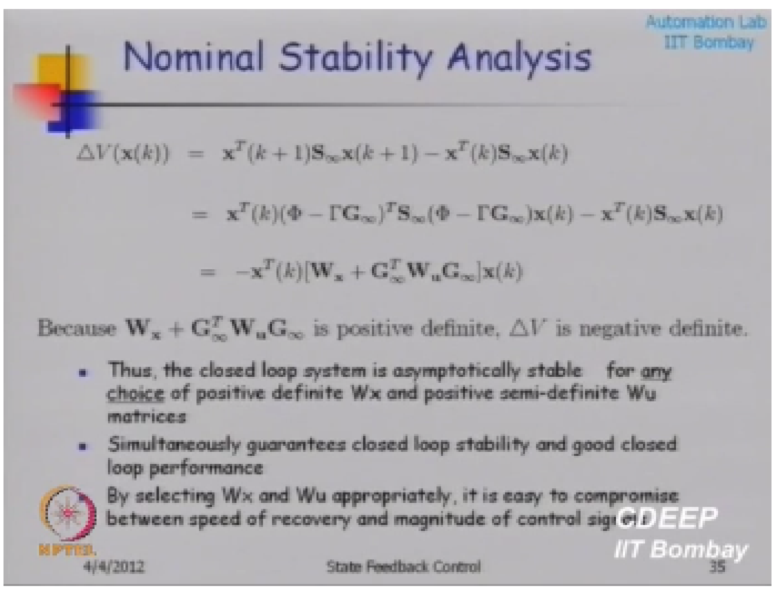
(Refer Slide Time: 53:38)



Okay, now I want to show that this is asymptotically stable okay, I want to show that if I drew asymptotic stability okay, then you know yeah then see we have seen one thing for Lyapunov functions if Eigen values of this matrix are inside the unit circle okay, then you can always construct and Lyapunov function okay, whose you know derivative is negative and strictly negative and we will you know if the system will be asymptotically stable and vice versa, so close inside the unit circle and ability to construct Lyapunov function for a linear system time in variance system or phenomenon.

So if I construct a Lyapunov function for this it implies that the poles are inside the unit circle for linear systems these two things are equivalent asymptotic stability and being able to construct a Lyapunov function okay, for the particular system. So instead of trying to find out the poles of the system and show that there is head the unit circle I am going to construct a Lyapunov function and show that you know the Lyapunov function you know the rate the change of Lyapunov function is negative definite so this is asymptotically stable system which means the poles of $\phi\text{-}\Gamma G\omega$ have to be inside unit circle so this is stable asymptotically stable, okay that is the argument I am going to use, so I am constructing this function v(x) which is $x^T(k)$ S∞ S∞ is the positive definite matrix by the way okay, S∞ is always a positive definite matrix.

(Refer Slide Time: 55:32)



So what do I have to do to find out to the Lyapunovb stability find out the difference between two successive values of the Lyapunov function so this is v(k+1) and this is v(k) okay, so is everyone with me on this I am just substituting for the close loop okay, so this becomes yeah, this will becomes negative definite because look at this term $S^T(k)$ ϕS∞ S(k) will it disappears or I have looked out some algebra in between, your very in algebra let me write.

Because ϕ will come ϕ^T and you know find out the infinite algebraic you can work out this algebra the expressions are correct okay, you have to see how one step I have so this $Wx+G^T$ W is a positive definite matrix so you can show that definite in this Lyapunov function okay,you have this ΔVx which is positive definite no I am just worried about one term you have to go and check I will just check this and confirm okay.

So the argument and I want to make here is that the close loop system asymptotically stable for any size of Wx and Wu that is very, very important I choose these weighting matrices be positive definite but arbitrary okay. Now in reality you will not choose them arbitrary but you are guaranteed to get a stable controller, stable close loop behavior.

For arbitrary choice of Wx and Wu okay, for any x of any Wx, Wu you will always get this matrix be positive definite and then I think you have simplify this using G∞, here to use that expression for G∞, G∞ expression is pretty complex, G∞ expression is this right. G∞ expression is this and you have to substitute and then expand to get so from here to here you have to do lot of algebra okay, it is not obvious from this so from here to here I have missed out lot of in between steps you can sit and work out by substituting for G G∞ that expression on the previous slide and they will do a huge expansion and then cancel all the terms and appears and so you will have to do lot of work to arrive from here to here okay.

But that is not important for us if there algebra is not so important do not worry about the algebra this is the standard textbook material in all first worked out they back and this is you know since it is positive definite you just for the time being what is important to know is that this expression can be reduce to this simple expression here, and since this is possible definite x^T this positive definite matrix so x^T this into x this whole term will always positive and negative of that is always negative, okay.

So the so you can construct Lyapunov function for this particular system and if this is negative definite then for any choice you know you will get a stable close loop behavior, okay. The nice thing about this particular controller is that it not only guarantees stability it also guarantee performance because how is the performance define, should the objective function you have given an objective function okay.

So choosing those Wx, Wu matrices make sure okay, make sure that you have desired performance transited into an objective function okay, and then that get you know accounted for when you do the gain calculation, so this is like as I said this is just leasing the poles inside the unit circle is just I saying that you know we have passed the course, you are about four. But here you know you are ensuring both performance and stability you are above four and then you are the topper, okay.

So by choosing this Wx, Wu you have to have some experiences as that is how to choose Wx, Wu you can actually shape the speed of recovery from non zero initial condition to 0 initial condition you can choose priority you can priorities different inputs you can actually give more importance to one input and let us see importance on the input and all types of sequencing, so you can weight a costly input more and not allow it to move too much if steam is costly okay, do not change it too much all that can be done by choosing this weighting matrices that is yeah.

So G∞ can see now actually when you solve it when you calculate you do not do all the optimization that is just a derivation when I am asking you to now I will now these problem is that implement quadratic optimal controller on the system that you are okay, so what you do to do is not do all these optimization what you have to do is take this particular problem you just define Wu you define Wx okay, define Wu define Wx and then give it to MATLAP there is a subroutine call ARE algebraic raccatic equations, okay.

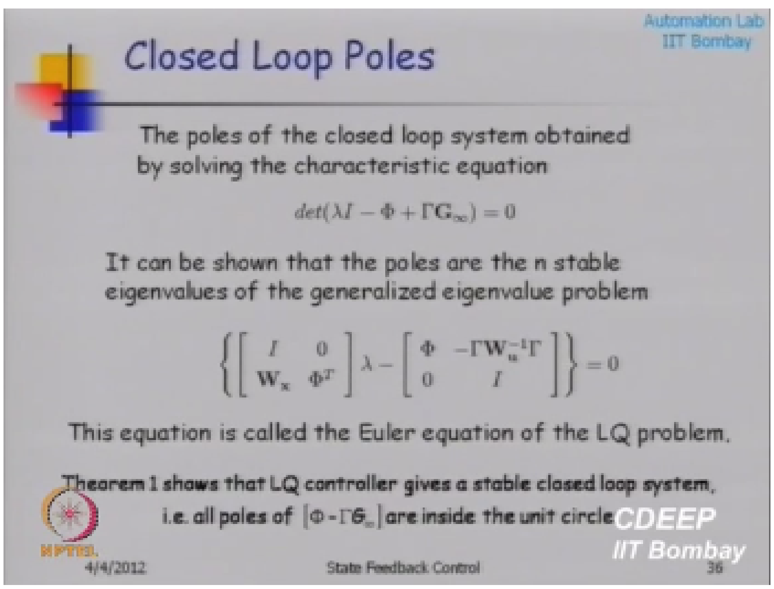
You just give ϕγ Wx Wu okay, see what all thing we are ready to specify ϕ, you have to give ϕ matrix γ matrix, Wx and Wu if this four things are given okay, MATLAP subroutine ARE will solve or SILAP will be, there will be some other subroutine equivalent to this it will solve this problem algebraic Riccatic equations and it will give you these two things what is the, what it will give you back G∞ and S∞ both will give you, okay.

Of course you should make sure that these conditions are made ϕ and γ should be controllable okay, and chose Wu in that this pair is observable, what is observable, what is controllable both you do now okay, so now you make sure these conditions are made so ultimately the working recipe is that is one slide all that was the derivation. Yeah, but typically this asymptotic so it start program if you take Kalman filter okay,I am going to run this plant for you know let us say I am doing it every one second interval I am doing calculation.

I am going to run plan for 20 days okay, so actually it is like saying k goes to ∞ because you know within some 100 samples that gain will saturated and you will get a steady state solution. See whether if you start here if you start falling the dynamic Riccatic equation okay, you will find that in some N=100 you will hit G, G∞ S∞, so will get the steady state solution when no time you know so those first see suppose I am going to run a plan for you know 20,000 time steps and if the gain saturate to G∞ within half of 100 okay, I ignore first 100 error I am not so much worried about 1 to 100 being non optimal.

I am if 100 to 20,000 is optimal I am willing to you know scarifies on 1 to 100 not be in optimal okay, so long time solution which actually mean within some you know 100 or 200 samples you will read this G∞ okay, so very short time which solutions will converged if the conditions are met, okay. So how do you find out the close loop tools.

(Refer Slide Time: 01:04:34)

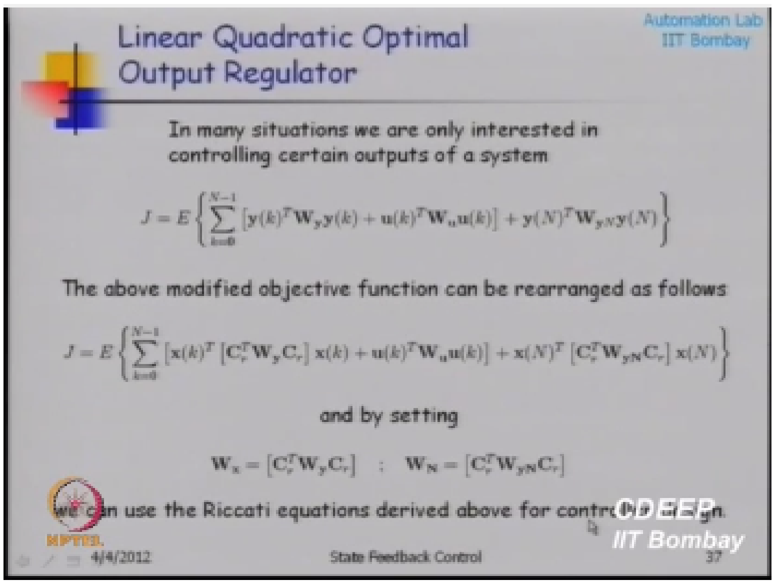


So this can be done using of course you can solve for this determinant of this it can be shown that it is also a solution of these generalized Eigen value problem and this particular solution, if this particular equation is called as Euler equation on linear quadratic problem and you can, you are guaranteed that for any choice of Wu and Wx you know you will get a stable solution okay, so stability and performance both are guaranteed by this particular approach that is the teeth in here, okay.

Now see these things you read the derivations and then of course once you start implementing it on the reason I am given the third problem is because I wanted to go back and implement it in other type problem immediately okay, that is the way you will understand in this course if you do lot of algebra and you know you cannot understand things even by solving some tiny problems of 2x2 matrices or 3x3 matrices.

Actually implemented on a real system at least in a stimulator and see how it works okay, you will now have ϕγ matrices you start about non zero initial condition and see whether you go to the 0,okay.

(Refer Slide Time: 01:05:51)

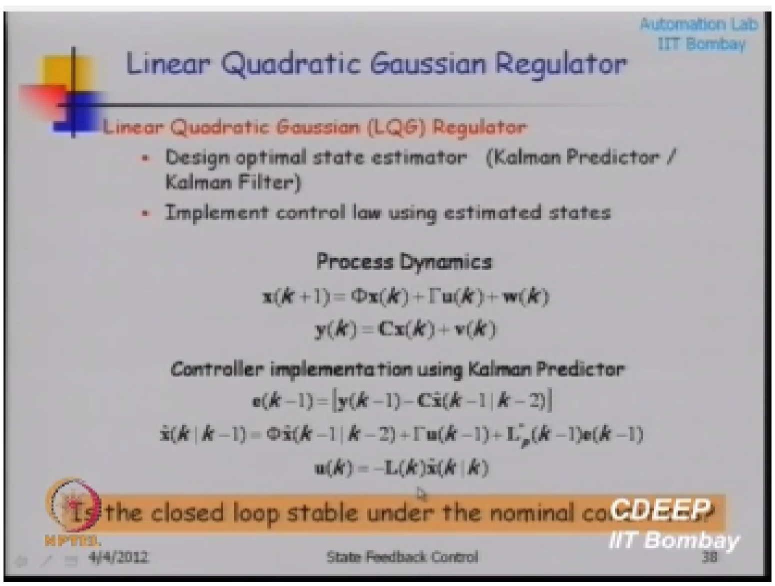


So of course you can do what about we started this problem by saying x^T Wn x x right, this objective function was sum is equal to x, if you have you know a model which is develop from data then x may not have any physical meaning only y has a physical meaning so you can actually solve this problem y^T Wy.u^TWu like this all that I have change is instead of x^Tx I have changed to y^Ty but this problem can be easily transformed into x^Tx problem.

Yeah, this can be very easily transformed into x^Tx problem so that is not an issue, so problem which is originally y^Ty can be converted into x^Tx problems okay, what is this Cr here is you know I have set that those output which you have to control, you may have mdeled many outputs not all of them you want to control so Cr will give you that part of the outputs from the states which you want to control it is okay, so this is so you can convert a problem which is output regulation to see many times you know that model is developed not from physics is developed from data, in that case x does not have physical meaning so it is difficult to choose weighting matrices and all that only y has a physical meaning, y is a real measurement.

So you can define a problem with respect to y and converted into a problem of x, because x and y are related okay, you can convert that problem and then you can solve this problem in by you choose Wx to do this Wu to be this and then you can solve the derived problem so that is.

(Refer Slide Time: 01:07:44)



Well, now let us relax one by one we said that they must know, we said that there is no state noise there is no measurement noise okay, so what do I do is there is a state noise and measurement noise, what we do is of course we have a state observer and then we have observer controller pair okay, so my process then actually now this okay, so I am not going to ignore Wk and Vk okay, I am not going to ignore them.

I will implement the Kalman predictor for example, I can do this with Kalman filter also I am just showing you this is Kalman predictor that is because Kalman predictor I have to deal only one with one equation if you here to do algebra for me for the fisrt I will put Kalman predictor here, so this is my Kalman predictor I will get estimate of $\hat{x}$ k given k-1 okay, my controller will be okay, it should be here not Lk it should be Gk here sorry, this Lk is the observer gain this is the Lk here is the I made a mistake.

It should be u(k)= this should be J∞ okay, and even here this can be okay, is this okay, so how I am going to take care of measurement noise and you know state noise I have a optimal state observer combined with a optimal controller okay, this is called as linear quadratic Gaussian controller, why it is Gaussian business comes in because I am using Kalman filter to estimate the states to construct optimal estimation of the state okay, I am using optimal way of computing the gain matrix using Riccatic equations so this is computed using another set of Riccatic equation this is computed using some of the set of Riccatic equation.

So both cases where we use the same theory okay, optimal estimation theory optimal control theory they are just you know mirror images in some sense if you try to draw parallel if you see there is a close parallel except one deals with something in future other deals with something in the past, okay. So you have this so I can have this pair optimal observer controller pair and of course I have to worry about whether this is you know jointly stable or not so I will be talking about what is called as a separation principle in my next lecture, where we will talk about stability conditions joint stability of Kalman filter and optimal controller pair, okay.

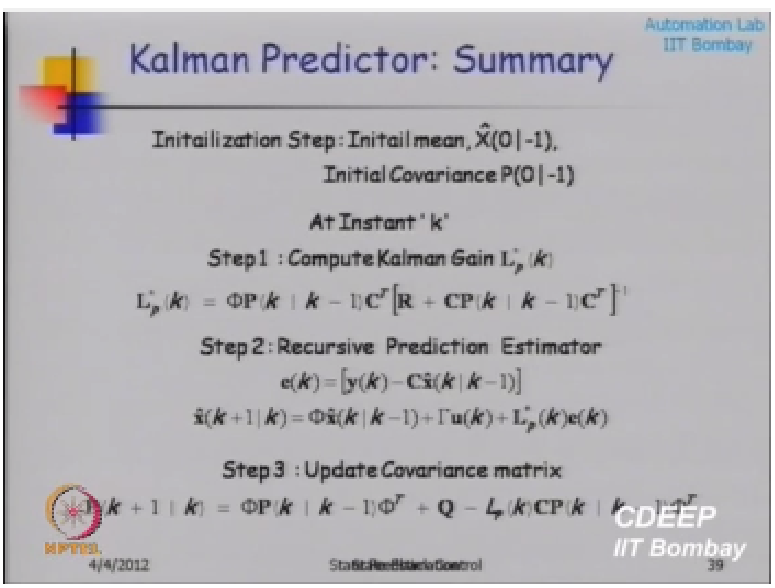
So they can be showed by jointly stable so if I design my observer to be optimal observer which is Kalman filter and then I design my controller to be optimal controller give a quadratic optimal controller and then I implement the control law to reconstruction of the states then I am guaranteed under nominal conditions they are jointly stable okay, I can separately design observer I can separately design the controller I can vary the two okay, they are together they form a stable closed loop pair, okay.

Effectively if you see here this is an output feedback controller why it is an output feedback controller, see because how is extract competence using the model and measurement okay, so the measurement if feedback to this observer which reconstructs the state that state is used to compute u(k) and that is injected here okay, see the state is not measurable so the idea of soft sensing is implicit here I do not have direct measurement of it I have only measurement of y, y is used u and observer to estimate x, estimated x is used for close loop control, okay.

I have a way of computing optimal gain for the observer given the noise model okay, I have a way of computing optimal gain for the controller given and performance objective okay, but I want to reduce it close to 0 as fast as possible by considering the input weighting and all that, there is a problem here. First of all you have to look at stability so you can think you have to see you know we said that non zero initial condition to zero initial condition very, very restrictive I want to go from any output to any other output I want to specify a set point and move from one set point to other point tracking okay.

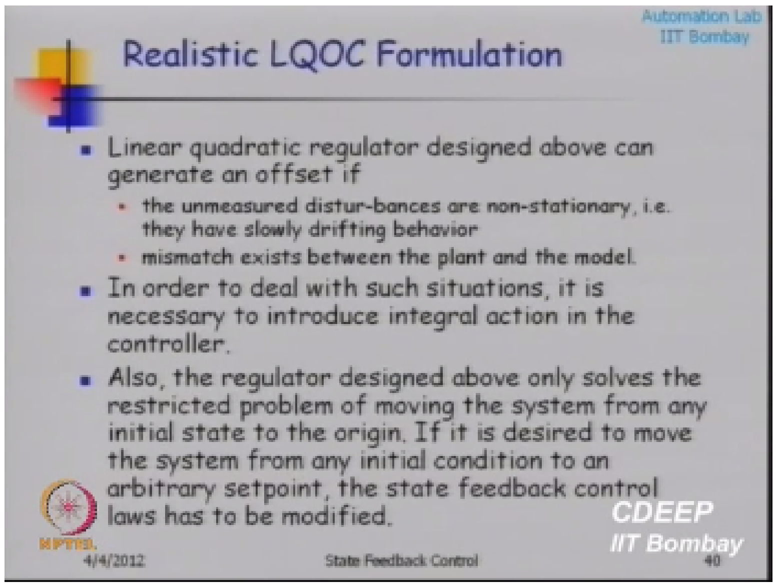
So second problem is there are drifting disturbances they could be plan model mismatch okay, and then this controller will not give you objective behavior, how do you get objective behavior with this pair okay, now you do some modifications to.

(Refer Slide Time: 01:14:16)



Do that your Kalman predictor of course I will just summarized here.

(Refer Slide Time: 01:14:24)



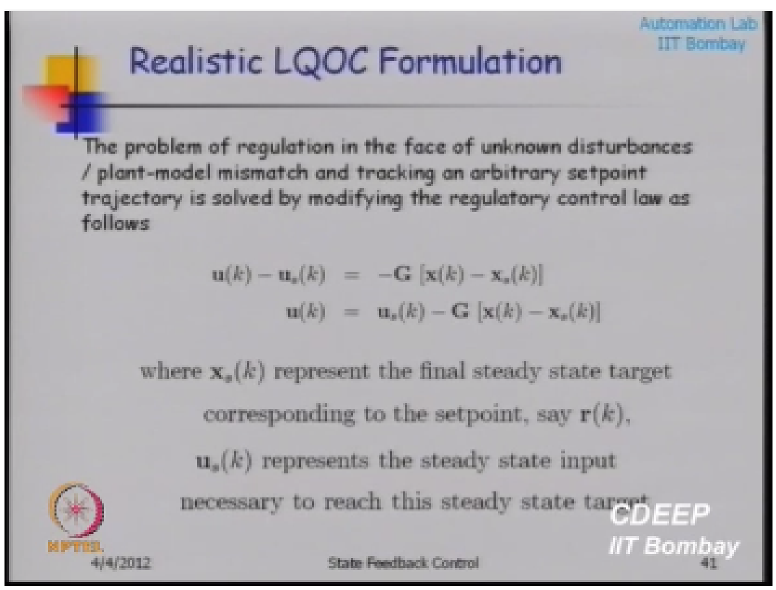
Okay, so I need to do couple of things I need to now so now I know how to compute gain matrices is that okay, I need I know how to compute observer gain matrix I know how to compute optimal controller gain matrix. Now I will done that, I have to move on and talk about two things okay, what is the model and the plan are not matching is it mismatch within the plan and the model. What if the disturbances see here when we derived this Kalman filter we said that is Wk is a white noise okay, in reality there could be a disturbances which is not of zero mean white noise.

It could be a drifting disturbance okay, which is most slightly the case in the real system there is never see one example of let us say this disturbance modeling is when you are driving a car so you know you have wind or air resistance okay, so air resistance if there is no wind but there is general motion of the air it could be probably approximated as some white noise affecting the you know some local motions of air affecting your car, but if there is a strong wind in certain directions then you cannot approximate it is a white noise you know that particular disturbance might be something like a colored noise and then you have to account for it systematically in your controller calculation, so how do you do that, so that is going to be our next task.

So now I want to move yeah, so it is a problem of regulation but for disturbances that are drifting okay, so how do I now modify my controller in such a way that I can account for drifting disturbances okay, I also account for the fact that the model and the plan may not be identical they might be a mismatch so how do I, so ϕ and γ which I am assuming might be different from what is actually you are in the plan I identified them now today, okay after a period over a period of time the plan characteristics has change, okay.

I am still using my same old model okay, see now that you are doing perturbation in the plan and collecting data. How many times you can keep doing this you will do it once and then you know you will do it after sometimes so in the meantime the plan can change okay, so now what, now will you be able to do the control with this so I need to do some tricks to account for the fact that the model plan mismatch can develop over time I also want to do something to make sure that I reject drifting unknown disturbances, okay.

(Refer Slide Time: 01:17:22)



So now what I am going to do is I am going to modify my control law like this okay, I am going to modify my control law what was my originally control law my control law was u(k)= - okay, and I wanted to go to origin okay, now what I am going to say is that origin itself is not fixed okay, I am modifying this control law by using a steady state target here this x is called as a target state, okay. So instead of saying that take x to the origin I am saying that take x2 a target set excess so difference between this should go to 0 okay, what is this target state how will I compute this target state.

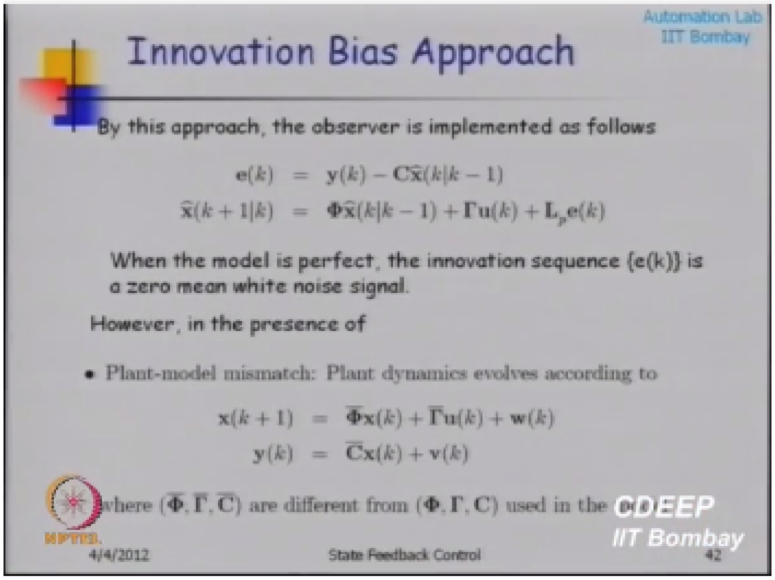
So I am going to compute this us and xs these two quantities this is target input this is the target state okay, this target input and target state are computed to account for two things one is unmeasured disturbances second it is computed to account for set point changes. See and I want to move the distance from one set point to other set point okay, using the same controller how do I do this okay.

So that is going to be done by managing, so the control now looks very, very similar if you see here are there my control law was uk=-Gx(k) I am just putting some compensation terms here okay, this compensation terms will allow me to take care off moving target you will take allow me to take care of unknown disturbances, okay so excess represents the final steady state target okay, and at excess changing as a function of time where I should reach because it depends up on the drifting disturbances and so this is the more realistic formulation of LQOC in which the disturbances are not white okay, the model is not perfect okay.

I will ever do now is now, I will now solve a more realistic problem okay, using now what I have sorted out till now is how to find out G∞ okay, all that I have sorted out till now is how to find out G∞ okay, I did not find out G∞ now I am going to modify my controller to take care of drifting disturbances to take care of any set point okay, or because I was only worried about moving to origin 00 very restricted invention.

Now I want to change it to so how now do this okay, that we will see in the next lecture you have to do lot of.

(Refer Slide Time: 01:20:23)



So when I will do now is I will upload this hopefully within a day I will upload this well and the thing that I am expecting what to do is as a part of your project is implement these controllers okay, implement state feedback, implement the state observer okay, to open the first and then implement linear quadric optimal control and observer together and control your plan okay, that is what and actually when you do it that is the time and you will learn it more because this algebra here on the you know, on the board becomes very, very messy and then you do not get deal for what is happening unless you do it, you put your hands into it you actually control one particular stimulator at least the stimulate system then you will get deal for what you really happen it, okay. Load this.

NPTEL
NATIONAL PROGRAMME ON
TECHNOLOGY ENHANCED LEARNING

NPTEL
Principal Investigator
IIT Bombay
Prof. R. K. Shevgaonkar
Prof. A. N. Chandorkar

Producer
Arun Kalwankar

Project Manager
M. Sangeeta Shrivastava

Sr. Cameraman
Tarun Negi

Sr. Online Cameraman
Sandeep Jadhav

Digital Video Editor
Tushar Deshpande

Technical Assistants
Vijay Kedare
Ravi Paswan