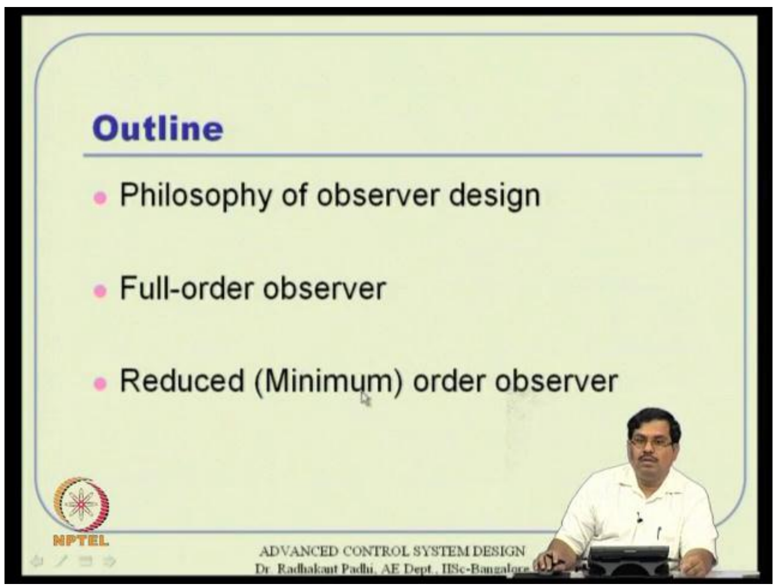
Advanced Control System Design
Associate. Prof. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Science, Bangalore

Lecture No. # 22
Pole Placement Observer Design

Hello everyone, we will continue with our lecture, this particular lecture we will talk about Pole Placement Observer Design. As I told in the previous class, we do not have to redo the entire exercise, we will rely on the pole placement control philosophy that we studied in the last class to design observer gains and all that. So, this will let us talk about in detail about pole placement observer design.

(Refer Slide Time: 00:41)

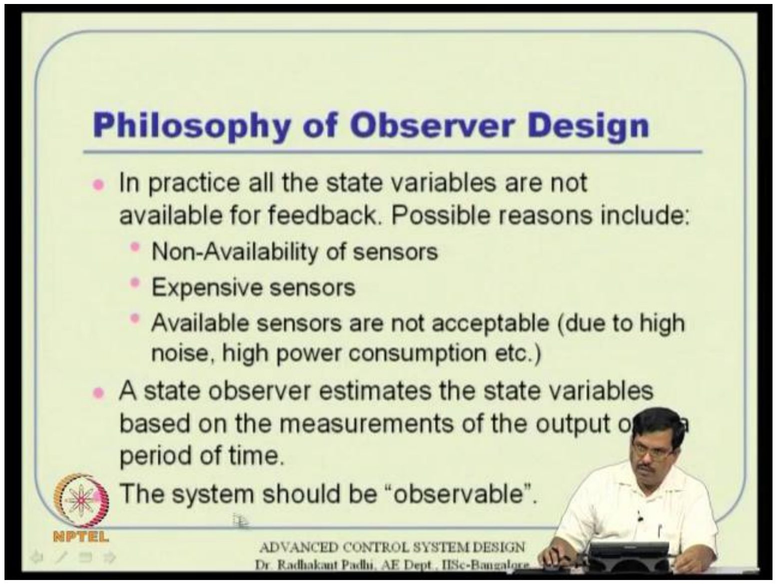


Outline of this lecture will be something like will motivate, why observer designs are necessary, and some of the associated concepts. Then primarily for first about the lecture, we will talk about full order observer design. And then we will continue with what is called is a reduced order observer design.

So, in this particular lecture, we will talk about minimum order observer, where only one output is observed actually in that is the mean, that is the minimum thing that you can have.

So, that is have the general concept is valid for reduced order, but to make our life simpler, we will talk about minimum order observers here.

(Refer Slide Time: 01:21)



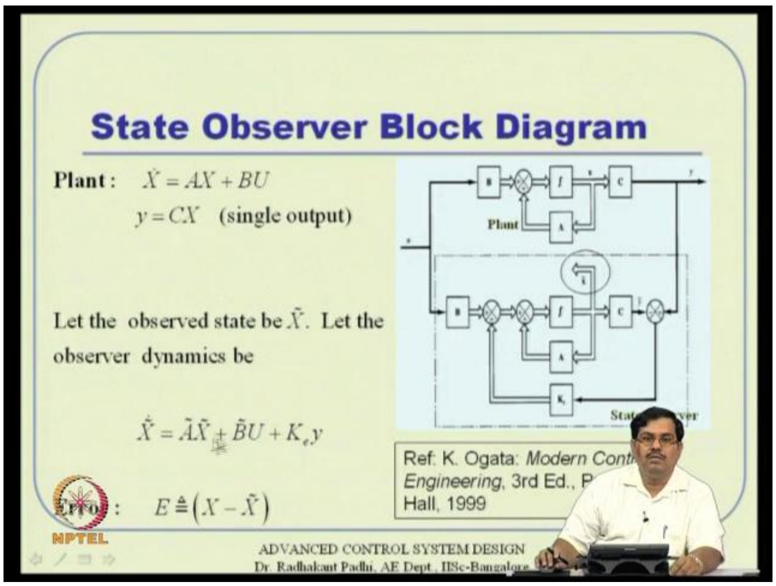
The philosophy of observer design is like this. In practice all the state variables are not available for feedback. Remember the the feedback state feedback control design assumes that, all states are available for feedback. In the in practice, all the state variables are not available; and possible reasons can include something like, some of the sensors may not be available there; and some sensors some sensors may be too expensive, so you possibly you would like to avoid those, if it is feasible to work with that. And then some available sensors are not acceptable either basically like, means it can give the high noise and probably it consumes high power and thing like that.

So, even if sensors are available you would like to avoid them. Primarily we can neither may be it is too expensive, privative cost sense or the quality itself is not good or the like the power consumptions are very high, if the trivial Reynolds vary fast like that actually. So, because of several reasons, so what we tell you what we think is all having all sensors in the system is actually not good in a way.

So, is it feasible to I mean design a control system and make it still waste on state feedback without all states being measured directly, we certainly need some feedback information; that means we certainly need to keep on measuring some information obviously, where we need not measure everything actually.

So, then at the state observer comes into picture; and the the state observer actually, what does it do? It estimates the state variable based on the measurements of the output over a period of time. That means, it does not measure only one time or something like that, you keep on measuring over a period of time, and that will contain sufficient information. So, that we can recover the state actually asymptotically; and for doing that, we what we need is that system must be observable. That is the condition that we need actually.

(Refer Slide Time: 03:32)



So, let us talk a talk first about full order observer design (Refer Slide Time: 03:28) and the something the full order observer something is like this I mean this philosophy and all that, you have a plant acting on that. And then a typically what you would like to have is this suppose X is measurable the that means X is available, then you do not need too much things, you can use that information; as well to probably you can put a gain out here and compute your U actually; U equal to minus K X.

So, for take the X and then compute U directly actually. And because that is not possible in this particular phase, we want to make this control operate on output feedback through state observation. So, the output is there, you do some mechanism something you go on here we will talk about details as we go along.

And then we all all that we need is recover of this this state information X tilde, X tilde is not X, but asymptotically X tilde will converse to X to... That is the whole idea basically. Then after the after this fellow converges that X tilde converges to X, you can work for all practical purpose you can keep on using X tilde actually in the…

We will see the details of how to recover that and all. And one that is there your control, we let this control what is here U we let on observed states that means X tilde actually. U U becomes not minus K X, but it becomes U equal to minus K X tilde now. So, that is the difference that you that we are talking here actually.

So, here in this particular lecture, we we are particularly interested how to recover these in a good way X tilde basically that is that is the whole motivation. So, this this lower side of the block that is the state observer. That is what you are going to talk here actually. Remember, we need all this all the system matrices A, B we are using here and we are also using the output information y, everything gets used here, one more time to recover this state information X tilde actually.

So, let us see I mean this plant dynamics is given as X dot equal to A X plus B U; and here we talk about y equal to C X, where we are especially interested in single output. To make it compatible with pole placement observer, which is like pole placement controller as we saw in the last class or nice, if you have single input system. Multi input systems are too much I mean extra things you have to do in then they are not unique and thing like that.

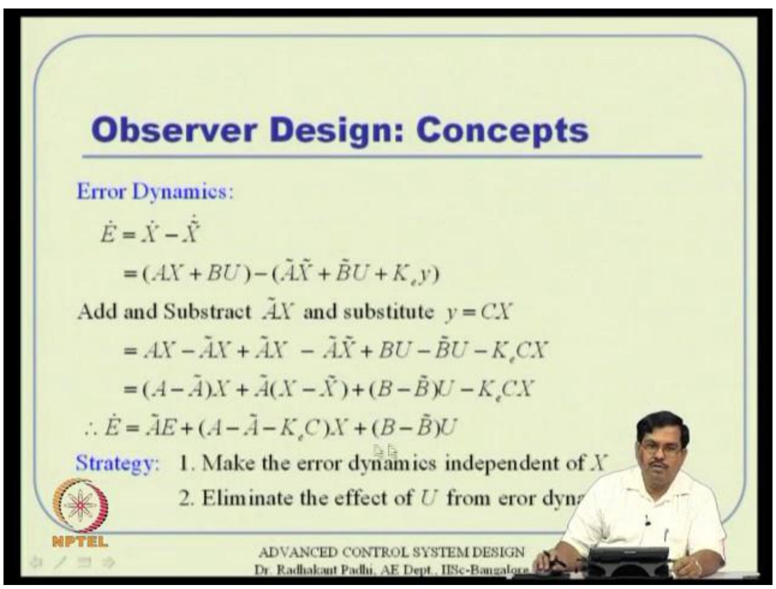
So, here we will confine our cells to single output systems. And then we are interested in estimating this X tilde basically. So, what we do? So, looking at this the system dynamics, we want to put almost like an artificial system dynamics let us say in this form (Refer Slide Time: 06:14). So, the whole X tilde is nothing but the observed states.

And then X tilde dot is almost same similar to that, but we have a K e times y term here actually. That is the output term where what we need to keep on in form I mean keep on using. Now, the question is, what is what is A tilde? And what is B tilde? I mean these are these may not be same as A and B, this is just that equation looks similar to what we have for the actual system linear system of course, but need not be exactly same as that.

And what is our what is our objective here? Objective is to drive this error E is E is nothing but X minus X tilde, we need to drive this error to 0 actually. And once the error goes to 0, then X X goes to X tilde that that is what we want to see actually or X tilde goes to X either way you can represent actually.

So, how do we do that? So, first we need to see you know in error dynamics, we want to see how this error dynamics we have; and then ultimately we want to kind of design of this K e that is our ultimate objective, how do we design this K e; and how do we select this A tilde, B tilde. Once you have done, then then the process is over actually.

(Refer Slide Time: 07:28)



So, let us see this is E. So, error dynamics sense E is nothing E dot is nothing but X dot minus X tilde dot; and x dot is this one A X plus B U, and X tilde dot is something the observe dynamics, observer dynamics that we are selecting. So, we will we will put X dot as that and X tilde dot as that. So, that is how we come to these.

And then you can add and subtracted a term A tilde X, and we can substitute y equal to CX. If you if you do this algebra, then it turns out to this is minus A tilde X plus A tilde X that is coming here. And then you have another minus A tilde X tilde that is that is indirectly from there, this plus B U is from here, then minus B tilde U from there, and then this term epsilon that actually. So, that is a direct substitution and thing like that.

Then you will kind of a combine the terms, once you start combining this is (A minus A tilde) X actually, what about these two terms you can combine; though those two terms you can combine and think like that. These two terms you can combine and leave out that one. So, ultimately what you are looking at is this E dot is nothing but A tilde E this this is coming from here (Refer Slide Time: 08:39), A tilde E plus this this minus this X is common to this term and the last term actually.

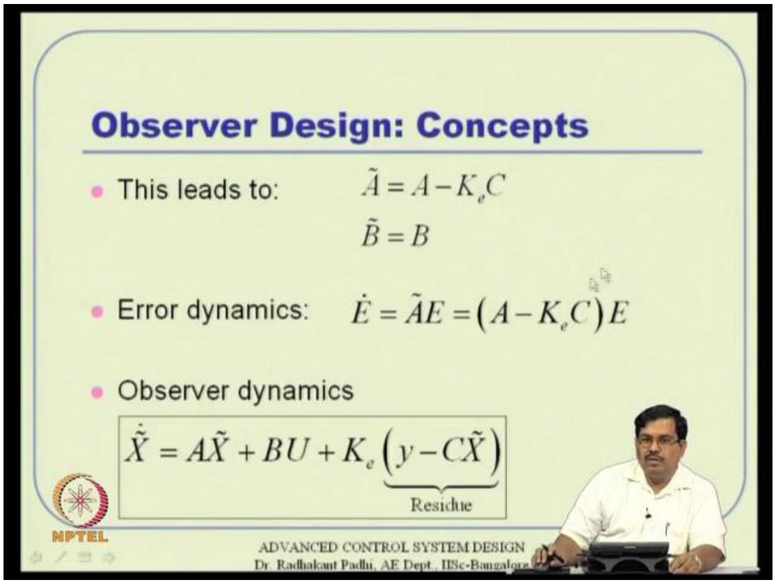
So, I will combine these two terms, A minus A tilde minus K e C into X plus this term extra term whatever you had left out actually. So, this this error dynamics turns out to be like that. Now, we have a freedom of selecting A tilde, B tilde that is that we have to kind of exercise our convenience here actually.

So, how do we do that? First you first thing, we want to make this generic make this as generic design that means this magnitude of E should not depend on magnitude of X. Suppose, for example, tomorrow it I mean today we talk about let us say locate a technology, where the distance is kilometers and two more I mean day after we talk about nanotechnology, where X is nanometers. So, those things I mean the our our E should not should be fairly independent of those things actually.

So, if in order to do that, what we want to do is these coefficients will forcefully make it 0, because we can do that here. If it is feasible to do, we will probably try to do that. So, then suppose we want to make this coefficient 0, and that coefficient also 0 to make it independent of control input also; the error dynamics should be independent of both, state and control, it should act by itself.

Then the selection of A tilde is A minus K e C, and B tilde is B. And once you select these, the B tilde equal to B this this coefficient is 0; and you select A tilde equal to A minus K e C, then this coefficient is also 0.

(Refer Slide Time: 10:14)



So, that you that is how we are left out with this this homogeneous system dynamics for error; as far as error is concerned, it becomes nicely a homogeneous system dynamics. And also let takes out the ambiguity that we had here while selecting these, because we now know what is A tilde, and what is B tilde? We we have that formula already actually. So, we have A tilde, we have B tilde and then we put put put it back.

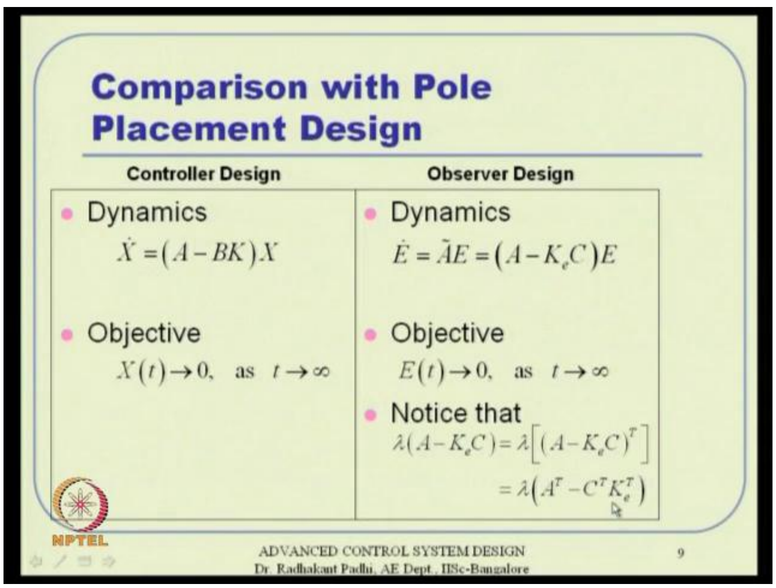
The observer dynamics ultimately becomes say this is our our formula remember that (Refer Slide Time: 10:43). So, if I if I substitute that it becomes that way, I have taken this minus K e C out here, because this is like y tilde C times X tilde is nothing but something like y tilde; and y minus y tilde is popularly known as residue information actually.

So, you have an observer dynamics, which is almost very same to what you have in state dynamic the original dynamics plus a gain matrix times the residue. So, using this residue information, we want to drive this error dynamics to 0 remember that; once you drive the error dynamics to 0, X will go to X will go to X tilde or X tilde will go to X actually. That is the whole philosophy out there.

So, of the the observer dynamics turns out like that. So, you can start with some initial condition and keep on propagating provided you know a value for K e, which will drive the error dynamics to 0; that is our objective actually. We select a K e such that the error dynamic goes to 0. And once you are once you have selected a K e, then this is observer dynamics and then we select a initial condition I mean guess an initially condition for X tilde, these error dynamics will will guaranty that the E goes to 0 actually.

So, that is how we kind of a propagate these dynamics to get X tilde of t which will asymptotically converge to X of t basically. That is that is the whole idea there. Now, there is problem out here I mean we almost see that this this terminology I mean if you if you just closely observe this error equation.

(Refer Slide Time: 12:18)



This is very similar to what we had it in controller design can see controller design what that the this was our dynamics, where U equal to minus K X, so A minus B K times X. And the objective was to drive X to 0. Here the dynamics is like that, that is what we got it actually, the dynamics is like that. And then, the objective is to go drive E to 0.

So, it is a very very parallel to what we have what you have already done; the only difficulty is here the gain matrix appear to the right hand side, but here it appear to the left hand side. So, we have to address that issue basically, there is not very same I mean it is very very close to, but there is a small matrix multiplication we see out here. So, that is in, but that turns out to be I mean that makes this error dynamics kind of incompatible to this. So, visually it looks very close to, but it is still not close actually that way.

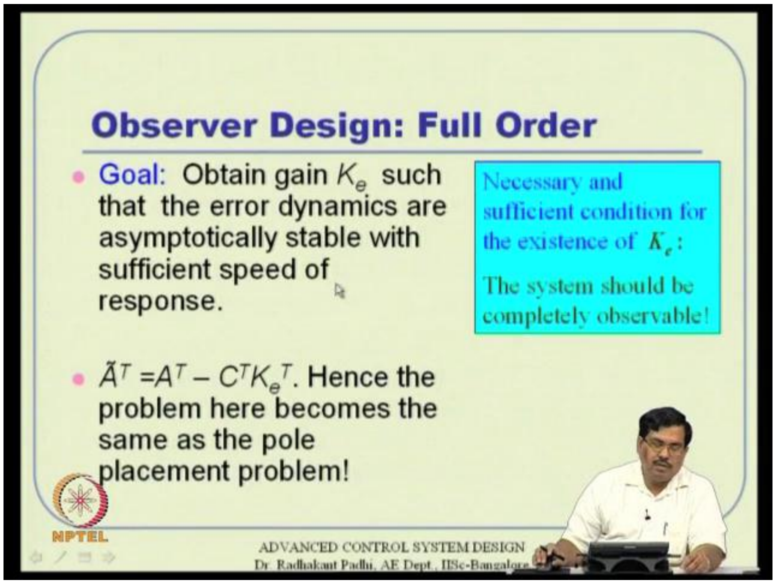
But the objective remains same, objective here is the X of t should go to 0, objective here is E of t should go to 0. So, for that, all that we need to do is this you need to assure that this this matrix that appears here, A minus K e C is what is called as (( )) matrix or the stable matrix sort of thing. That means the Eigen values of A minus K e C should should lie in the left hand side.

But the great observation out here is A minus if you see that, as for as the Eigen values are concerned like a A and A transpose of the same Eigen values the just making a transpose, the Eigen values are not perturbed. The characteristic equation lambda E minus A whatever you do, then lambda those terms are getting perturbed in diagonal elements only. So, A minus A transpose, the Eigen values remains same.

So, we take advantage of that and tell the Eigen values of A minus K e C is nothing but Eigen values of A minus K e C whole transpose just these two will remain same; and if I if I take out the transpose that is nothing but transfer I mean Eigen values of A transpose minus C transpose K e transpose. So, there is the it appears to be slightly different now, but the the good thing is this K e transpose is appearing to the right hand side as what happened in the controller design.

So, A has been transferred to A transpose, B got transfer B got transformed to C transpose; but the K whatever happened in the controller design something similar K e transpose appears in the right hand side. So, then we are kind of, because we know what to do actually. So, we will take advantage of that.

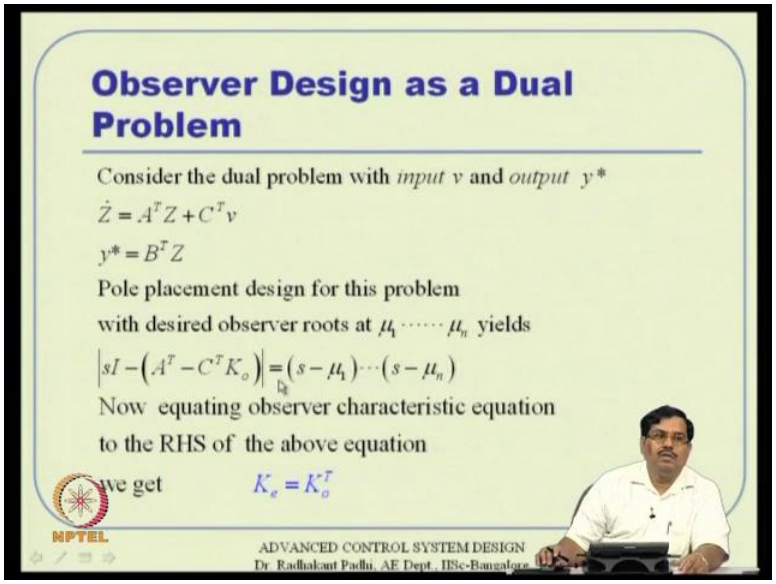
(Refer Slide Time: 14:41)



And as I told, the goal here is to design K e such that the error dynamics are asymptotically stable. And also remember that we have to have sufficient speed of response, because there is the whole idea is to decay this the error very fast; and and then you can do that, because we are not talking about a controller design, the control effort is a physical effort, here is all numerical actually.

So, you have to you can select the poles little for away, so that the error dynamics dies out faster. So, all this things taken into account, what we what we observe here is, we can really work with A minus K e C transpose actually. The whole transpose whatever you do here it A transpose minus C transpose K e transpose; that that is the term that that you have to work with actually.

(Refer Slide Time: 15:29)

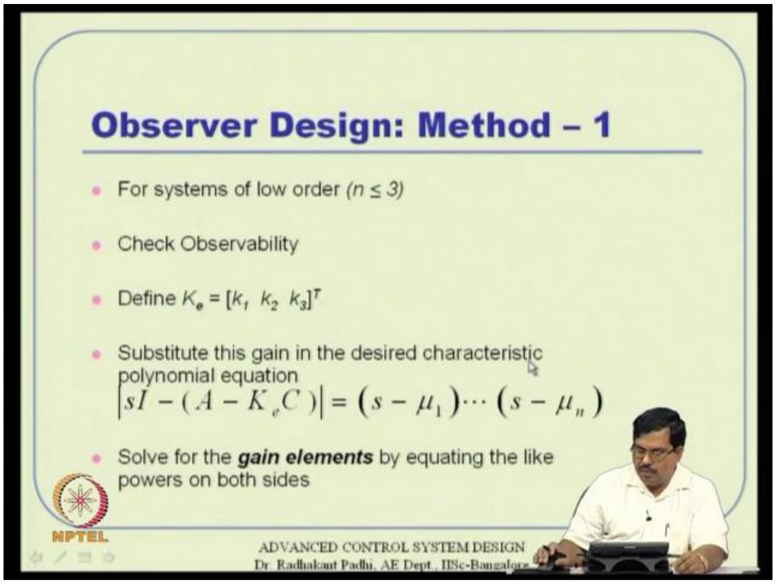


Now, as we discussed in the last class, there is a nice concept of duality we know that. So, if you consider this is a dual system of like this Z dot equal to A transpose Z plus C transpose v, where v is a virtual control input. And then y star is B transpose Z; then we know that, this the controllability of this turns out to be observability of the original matrix. And that is how it appears here also A transpose minus C transpose K e transpose actually here (Refer Slide Time: 15:53).

So, if it is A transpose minus C transpose K e transpose, so that is that is I mean kind of a helping us to design a K e transpose first and then we know K e actually alright. So, pole placement design for this problem essentially yields lies to this, the this I mean placing the poles of this matrix A transpose minus C transpose K e transpose, but then K e transpose we can visualizes something like K 0, where K e where K e is nothing but K 0 transpose I mean this is just an artificial matrix that you can introduce you need not also, you can directly solve for K e transpose actually.

So, the the characteristic equations suppose you want to equate it directly as method 1, what we discussed in pole placement last class. Then all that you need to do is you take out this this is a this A transpose minus C transpose K 0 transpose. These are this is the characteristic equation on the left hand side, and this is the characteristic equation on the right hand side, make it equal. And then select some like K 0 is nothing but K 0 1, K 0 2 like that and then equate the coefficients of the powers of both sides and then solve for K 0 thing. So, that is very straight forward actually.

(Refer Slide Time: 17:16)



So, that is what a method 1 tells about. So, if it is n is less than equal to 3, then first thing you need to do is check observability. And if the observability condition is satisfied and let us assume n equal to 3 here, then define K 0 K e is nothing about K 1, K 2, K 3. Remember the control gain, where control gain matrix was a kind of a row matrix I mean row vector. And observer thing, the gain is these kind of a column vector actually that will be (( )) from equation itself actually.

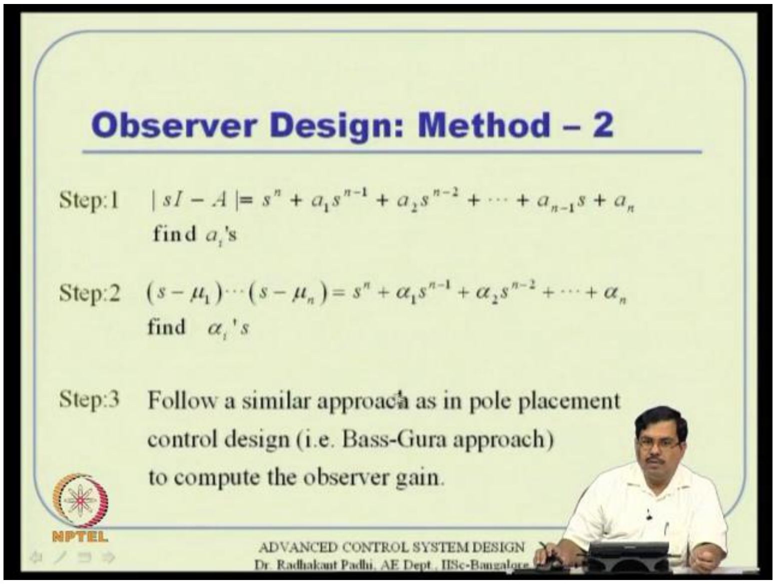
Now, if you substitute this gain matrix back in this equation, directly you can you can also work directly with this equation by the way, as far as you equating the characteristic coefficient I mean the equating the powers of the characteristic polynomial is concerned, you can directly work with that. That the Eigen values I mean that is what you need actually.

So, A minus K e C you can directly work with and sorry A minus K e C; or you can you can work with A transpose minus C transpose K e transpose, both will give you the same Eigen value anyway same characteristic equation that we get. So, then that make it equal to that and then you can solve for the gain elements. So, we are done actually. That is very straight forward for a provided you have a smaller dimension system actually.

And normally these things are useful for kind of a let us say you want to design a small control system for the actuator system independent actuator system, then probably these kind of things are not to get. Because, you have typically you have first order actuator or second order actuators actually most I mean third order may be some cases, but normal than that.

So, in a in some some sometimes if you want to design a controller, separate controller of actuators let us say these methods may be just handy for you. But in a good practical system for the entire flight control let us say you want to design, this may fail actually, and you may not you need better techniques that serve actually.

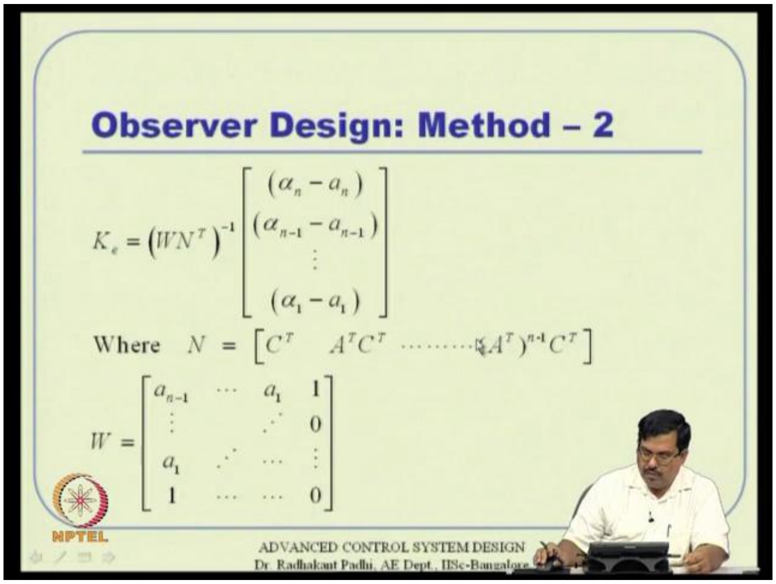
(Refer Slide Time: 19:14)



So, obviously we do have a better technique that is the method 2. That is what we talked last, that is a Bass-Gura approach of control design as in a pole placement control design. So, we will take the advantage of that approach. And then, first design K e transpose and then take transpose of that to get K e. So, the let us talk about that.

So, first is step 1 is s I minus A that that is the characteristic polynomial for the open loop system what you have. So, you can this will give you the the coefficients a 1, a 2, a of up to a n, then these are the pole locations. So, that will if you multiply them you will get some other polynomial with that will give you alphas.

(Refer Slide Time: 20:03)

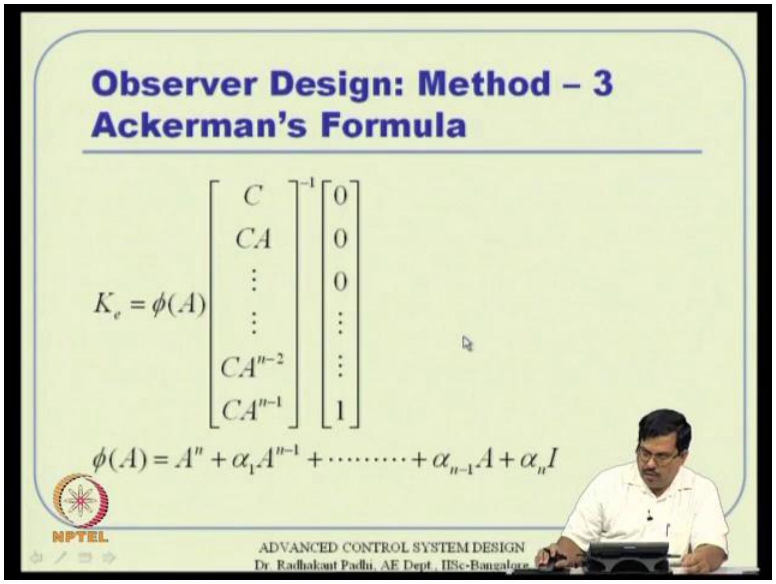


And then we can follow a similar approach for pole placement design to design a K e actually. Essentially you design a K e transpose first, and then take the transpose to get this final formula. This is actually nothing but the final formula actually what you get. So, where N is nothing but this observability matrix, now it will naturally pop up, because this this dual system what we are talking, this C transpose is coming here. So, this this will naturally pop up as we keep on applied the formula.

And W obviously we know how to how to form a W actually. So, N a N is nothing but the observability matrix which is given like that; W is given like that and K e is ultimately given like that. So, once you compute K e, the observer dynamics is like this, (Refer Slide Time: 20:53) this is your observer dynamics, where you know a value for K e now basically; and this K e is acting with error dynamics that way, which is driving the error to go to 0 again and again I am kind of a I mean emphasizing that. But you should never think that X tilde will go to 0, X tilde should go to X actually as soon as possible.

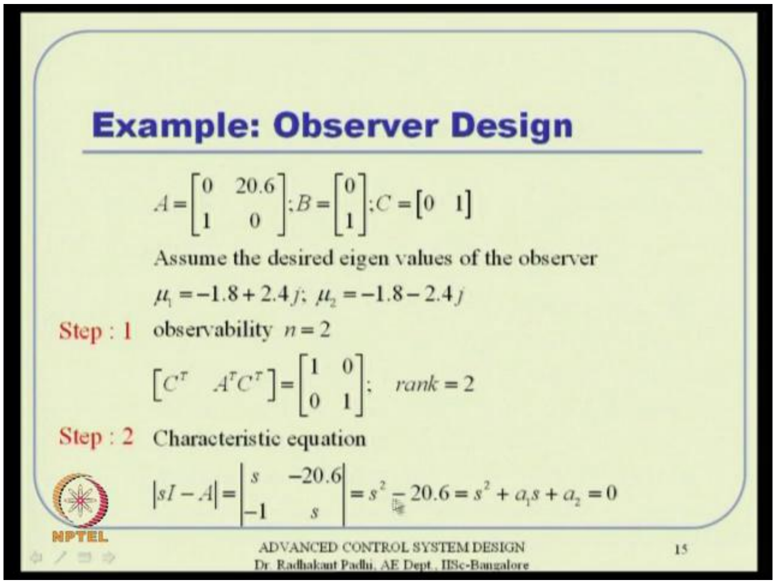
Now, if X driven to 0 X tilde will also go to 0 that is the different issue; but as far as observer is concerned observer design is concerned, we are interested to design some design this K e in such a way that, X tilde will approach X. So, that is what, so the method 2 is there.

(Refer Slide Time: 21:35)



And method 3 also we know I am not a kind of deriving or giving all the details again. But method 2 method 3 is like Ackerman's formula, so this essentially a formula which will give you that. And once you do the algebra little bit actually and then it turns out to be something like that, where phi A is nothing but this polynomial actually it call comes from Clayey Hamilton theorem, and all that if you remember the last class material actually alright. So, this particular thing is like this.

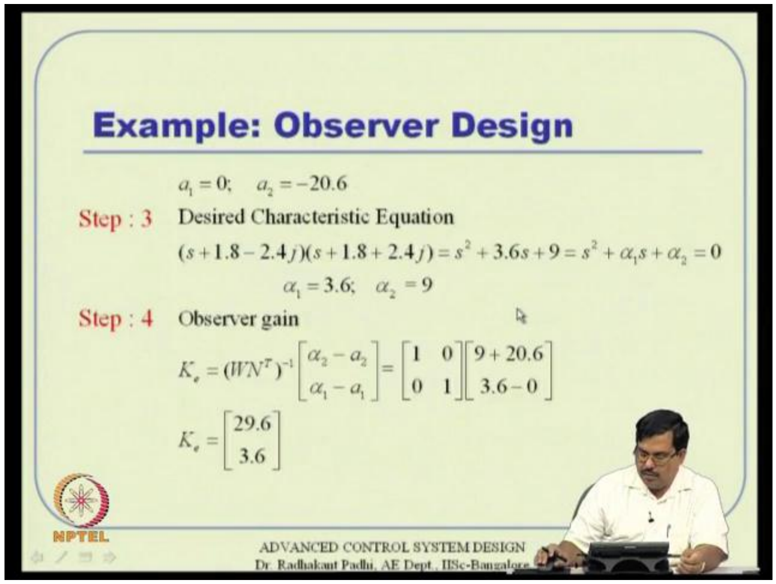
(Refer Slide Time: 22:06)



Now, let us talk about a small example to understand what is going on here actually. So, let us talk about a small 2 by 2 system, where the A matrix and B matrix, C matrix are given like that. And let us assume that you want the desired Eigen values to be placed somewhere here. And then we start with step 1, which is nothing but observability check n equal to 2. So, you have these observability matrix it turns out to be this identity; obviously, is a full rank matrix. And hence the system is controllable sorry observable.

I want the method I want the system is observable, you can design an observer and then you proceed with the next thing. So, next thing to proceed, we are applying method 2 out here. So, the next thing, we proceed is characteristic equation of the open loop plant. So, open loop plant is characteristic equations are like that. So, that gives us a 1 and a 2, a 2 is nothing but minus 20.6, and a 1 is 0 here actually.

(Refer Slide Time: 23:03)

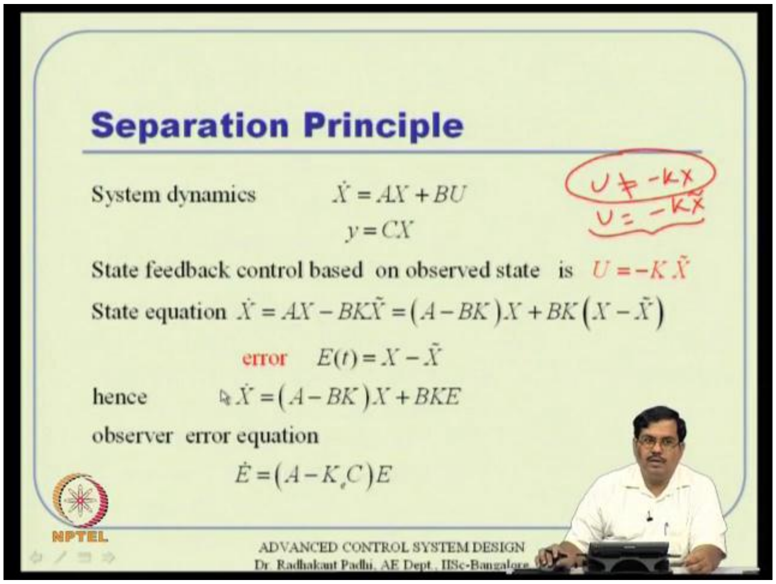


So, a 1 is 0, a 2 is there; and then the characteristic equation of the desired characteristic equation will give us alphas actually. So, we multiply use s minus mu 1 into s minus mu 2 you do expand that, it will give you some polynomial from which you can extract alpha 1 and alpha 2. Then the observer gain matrix is given to you I mean that are the formula ultimately. So, you just apply that right. So, that is how we design an observer, but remember in a in a examer class or implementation whatever is computing the observer matrix is not end of the story, you you also have to give observer equation.

And observer equation as I told is the is that one that we started with actually. This is the observer equation for which we have computed a gain matrix K K e. And if you want to implement it in your control design, then in a this observer dynamics you have to propagate in parallel with some guess value of initial condition.

And the nice thing about linear system is no matter what is your guess whatever is your guess value, it will converge I mean there is a universal convergence thing here, because this dynamics is as soon as this dynamics is stable it is independent of the initial condition. So, no matter whatever is your E 0, E of t is going to go to 0 as t evolves. However, it may take longer time to go to 0 in that that may not be good idea actually it may excite bigger transients before it goes to 0. So, it is also advisable to have an intelligent guess value for the initial condition rather than just blindly using something that is all that always helps actually.

(Refer Slide Time: 24:46)



So, this is where we are. And then let us study about a nice principle that that happens in linear system for which it is this approach is very popular actually, which is called separation principle it even have a is valid for Kalman filters. That is why these are very popular actually.

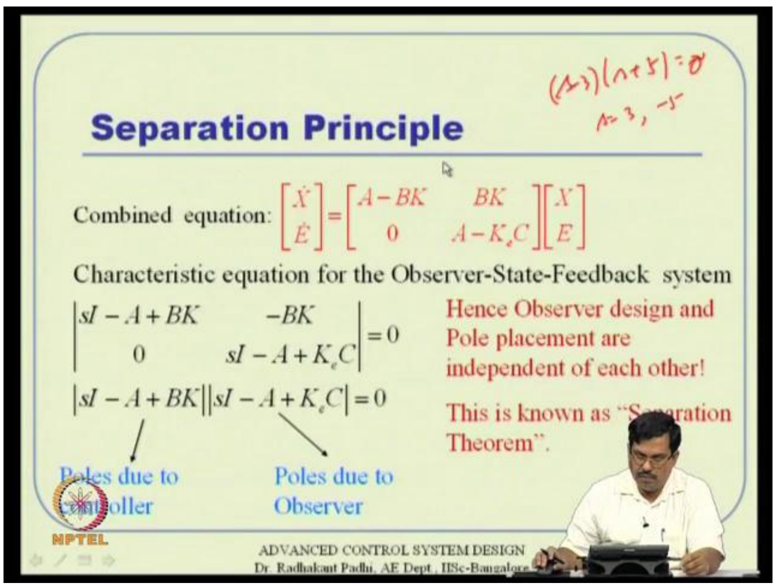
The question here is we have a system dynamics that way, but suddenly we have made U is not U is no more minus K minus K X, but U is equal to minus K X tilde now. And we know that, with this gone what is the guaranty that this U equal to minus K X tilde will give the overall system stability, it may not give that actually; that is the question. But we make it I mean we know everything that what when we design the design control system, we assumed that U equal to minus K X; and hence the we design a control a gain such the such that the feedback loop is stabilizing that to that we discussed in the last class.

Now, suddenly we are changing that to U equal to minus X K X tilde, X tilde goes to X asymptotic sense that is alright, but to transient it does not go actually right I mean during transient, X tilde is different from X. So, what is the guaranty that the overall system should remain stable that is the question actually? So, let us answer that let us try to answer that, so you have this U equal minus K X tilde, so substitute it; so that, what you get is X dot equal to A X minus B times K X tilde, because U equal to minus K X tilde.

And then is you can expand that, the X tilde info X tilde is nothing but that you substitute the A minus B K into X. So, what you are doing here is like add and subtract B K times X term. So, I am I am subtracting B K X and adding B K X also actually. If I do that, then it turns out to be A minus B K into X plus B K into this error term actually.

So, because this is error term, so the closed loop system now operates based on this actually. It is no more that, it operates with an additional term which is nothing but B K times E basically. But at the same time you have this observer error equation given this way, E dot equal to A minus K e C into E that that we just saw actually.

(Refer Slide Time: 27:13)



So, if I visualize a bigger system where X X and E are part of the states, then what do I get here is X dot and E dot I put it together and then I we visualize the bigger system operating in this way actually. So, the question here is will this bigger system Eigen values will also be stabilizing in the also be in the left hand then we have done, because X X will go to 0, E will go to 0 after that actually.

Now, fortunately turns out that, if you do a characteristic equation analysis for this matrix then this this being a block triangular matrix also squared a lot actually; this 0 works here there is a block triangular matrix for which you want to determinant actually; and these determinant turns out to be this one into that one actually. So, this is, so what do what does it gives us actually? Thus the poles of this overall system is nothing but the poles of poles due to controller, because this into that equal to 0 means either this is 0 or that is 0 or both are equal to 0 anyway actually.

So, I mean we want to find out the roots actually. So, in the roots of that of the entire polynomial will contain roots coming out of this equation, and root coming out of this is equal to 0. Suppose for example, if you talk about let us say s minus 3 into s plus 5 equal to 0, then the roots are s equal to 3, and minus 5 also. Because, that is how we are interested in actually. So, here also same thing, so this this this is actually like a determinant into this determinant equal to 0.

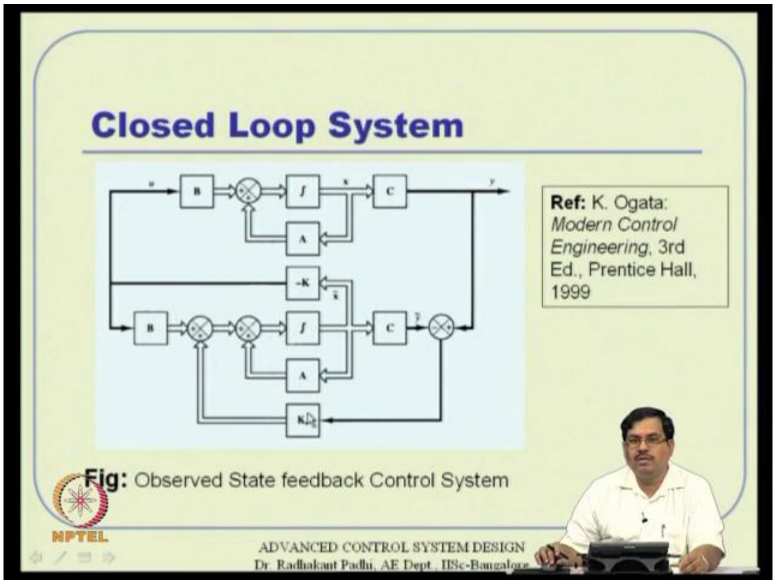
So, the roots of this entire equation whatever equations you have will will constitute the roots coming out of this equation being equal to 0. That means s I minus A plus B K determinant equal to 0, and s I minus A plus K e C determinant equal to 0. So, once I make this characteristic polynomial this will give me poles due to controller; and once I make this characteristic polynomial equal to 0 that will give me poles due to observer.

So, that means the overall system poles contain, poles due to controller and poles due to observer that we design separately. And obviously, we have designed the control system stabilizing and the observer system also stabilizing for the error dynamics actually. So obviously, nothing is going well the entire entire system dynamics is stabilizing in both sense actually.

So, what does it tell I mean we can essentially design the controller and an observer separately we do not have to worry about the interaction between them, because the interaction between them is guaranteed to be stabilizing actually. That is that is what is very popularly known as separation principle.

And that is why I mean it is also valid for Kalman filters also, Kalman filter design is an extension of this this observer, LQ observer. And then, we will see some of that probably philosophies at least. So, that is great theorem which tells us that the control design and observer design can be done separately we do not have to worry about the interaction actually.

(Refer Slide Time: 30:17)

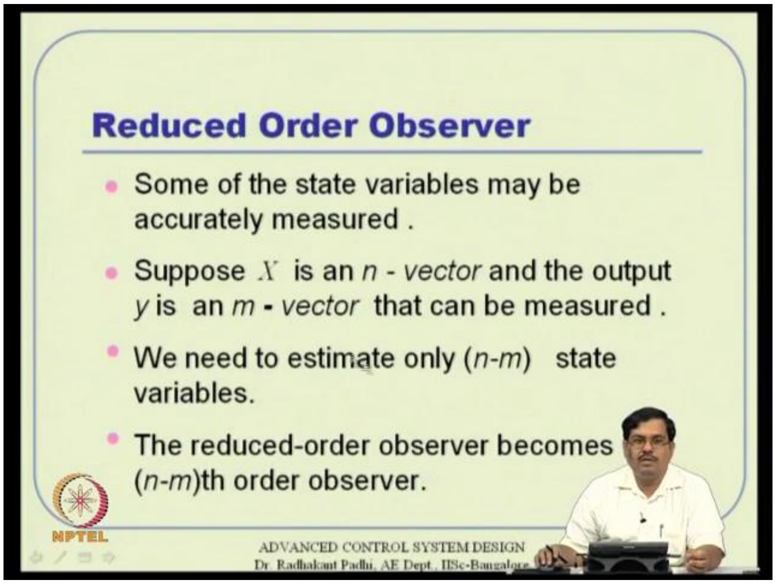


So, the closed loop system I mean block diagram sense it will operate that way. So, u will get computed from X tilde through minus K this is nothing but the control gain actually whatever gain you had. But this u will act both on the on the actual plant and it will also act on the observer plant actually. This this lower side of that is observer plant, the upper side is the control plant actually.

So, u gets computed from X tilde it does not get computed from X directly. But the way you are computing this K e make sure assures that X tilde converges to X actually. That is how it happens there. I will continue for further reduced order observer design (Refer Slide Time: 31:00). This is what we discussed here is, we are interested in estimating the all the states basically right. Even, if part of the state you directly measure from from a sensor or something even that becomes part of the observer dynamics.

So, even if the sensor information is actually very good for that particular state we are neglecting that and they are near mixing that of with everything else. So, transient sense entire thing is going bad actually. So, the very natural question of there is whatever you directly measure you do not really need to estimate. So, can you estimate rest of the things actually? So, that will that will take us to reduced order observer design.

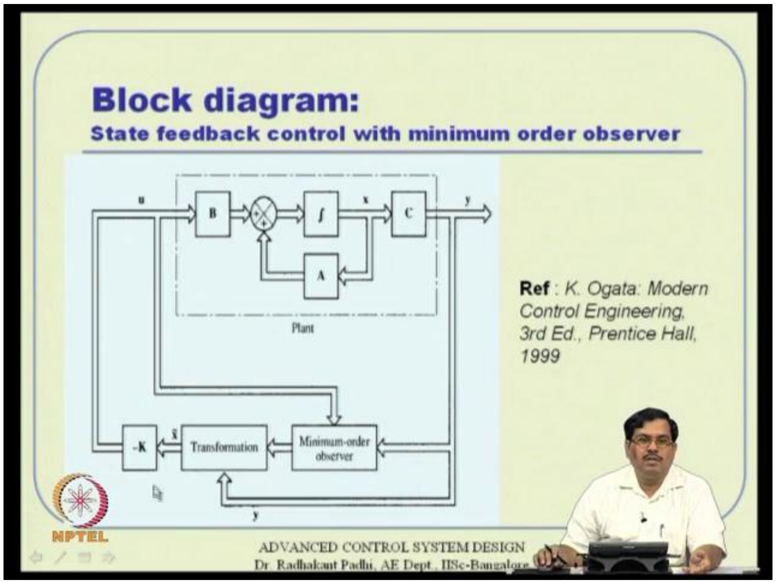
(Refer Slide Time: 31:39)



So, as I told some of the state variables may be accurately measured. So, you really do not have to estimate that again; and by doing that, you are not only saving computationally, you are also saving some you I mean you are also making the transient behavior good that is essentially does matter in the control design part of it.

So, suppose X is an n – vector, n dimensional vector, and output y is an m dimensional vector. Then essentially we we are asking the question that, can we estimate only (n minus m) state variables whatever we are we are observing we will just leave aside actually; whatever you are your sensors are giving directly good measurements and all that, we have do not want to make it as part of the estimation process actually. So, obviously the reduced order observer becomes n minus m th order observer. So, that is as I told computationally more efficient, and then we have I mean transient properties may also be better actually.

(Refer Slide Time: 32:38)

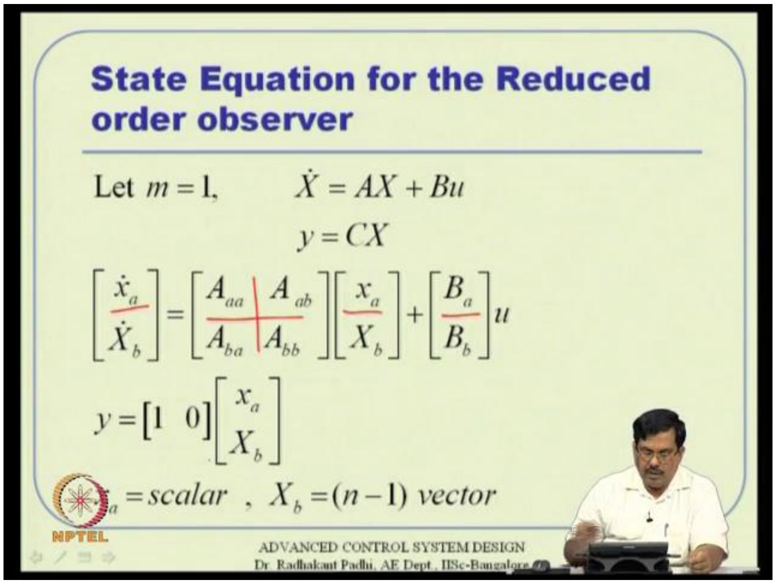


So, that is something something operating something like this. So, essentially you are having a minimum order observer now or the reduced order observer now. And essentially you also need a transformation thing to get it back into the the X tilde part of it. It is the it is nothing but just putting them putting the dynamics that you are putting the states or part of the states that you already know together with what you are estimating actually.

So, this is that transformation I will give you that, minimum order observer will give you the rest of the states, the outputs are giving you some part of the states. So, put them together and then you are getting X tilde, so you make it your control. And this is control gain, so control becomes minus K times X tilde actually here.

So, this transformation is not a very big deal I mean this is vary standard basically, we are just putting them together. If your y is directly measuring some states, then this is just putting them together. If your y is measuring some some combination of states, then you may needed the matrix transformation out here actually. So, that is why this is in general it is written like a transformation actually.

(Refer Slide Time: 33:45)

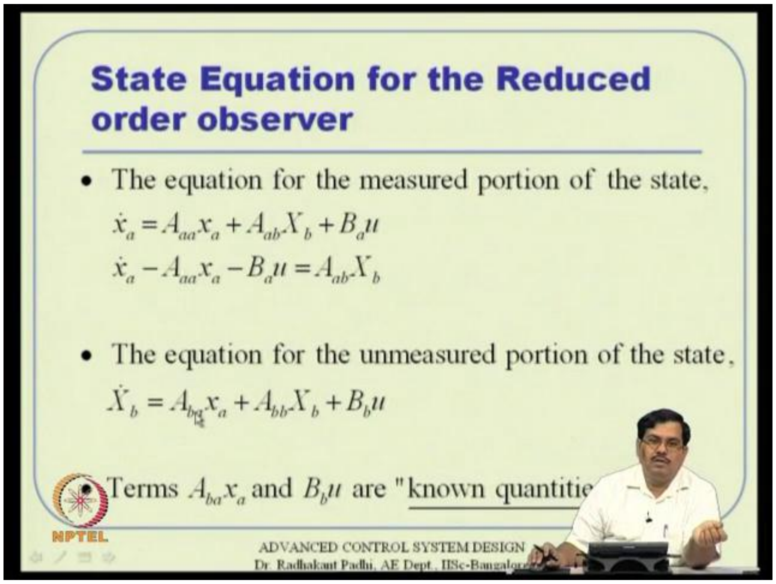


So, let us study this in slightly detail. So, what we what we are asking here is, we confine ourselves to m equal to 1 that means we have a like single output getting measured actually. So, that is what we want. And then without loss of generality we want to put that it is the first state in the state order, we just tell the we just we are claiming that, the state vector X is partition into this small x a, which is a scalar and the big X b, which is a vector.

So, we have this entire X we are dividing into just a scalar term which is directly coming as output. That is why this C matrix is 1, and then this is actually a 0 vector sort of thing. So, this is just a scalar that is getting measured; and then there is a bunch of 0's out there, so that will give you the output matrix actually.

So, essentially what you are doing? This is this X we are partitioning into x a, and X b, where x a is just a scalar which is measured. So, we are interested in estimating or observing this this X b vector that is our kind of objective here. So, the equation for the measured portion of the states; suppose you just take it I mean this partition matrix, so it is just the partition here, partition here, partition, partition here and then you talk like that; then you see that, that x a dot is nothing but this tiles A a a times x a plus A a b times X b plus B a times u. So, that is what you are writing here.

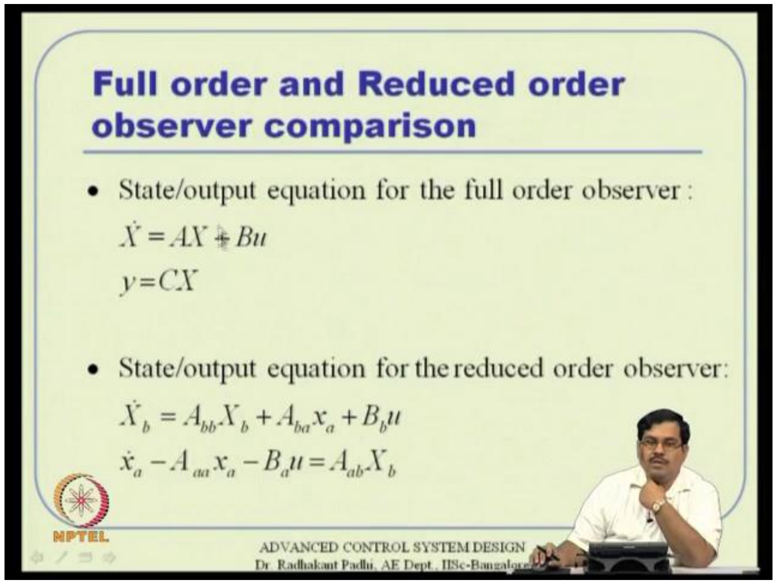
(Refer Slide Time: 35:20)



And so, once you write it then then x a dot I want to take everything that I know in the left hand side, and whatever I do not know, I will keep it in the right hand side actually. So, this we could this is just a different way of writing this actually. And then X b dot is nothing but A b a times x a plus A b b times X b plus B b times u actually. So, that is what we are writing here actually.

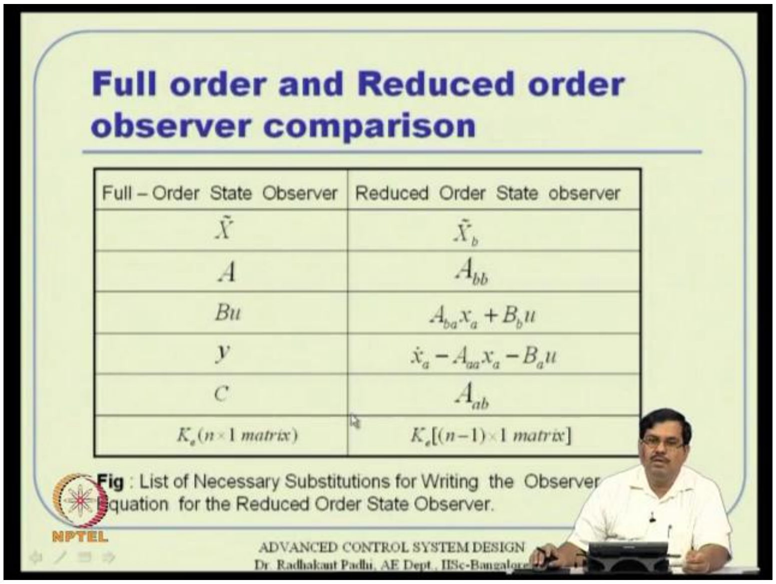
So, also remember that A b a x a and B b u this the is see this is essentially our in the our kind of interests here; and that is what we do not know, we want to estimate that particular thing. But in that but in that what you see is this particular term and this last term are known quantities. So, what you do not know is actually X b. So, we will try to see what we can do with that actually.

(Refer Slide Time: 36:15)



So then, what we have seen here? We have seen that, state and output equation for the full order observer is like this. But the state and output equation for the reduced order observer can be like this. This is my state equation. And I can consider that, the left hand side what I have is actually kind of known information even though I am measuring x a, but in a way I know the information of x a dot and then u also I know. So, this entire thing I can put it as some sort of a virtual output sort of thing I consider that is virtual output. And consider these, the state and output equation for reduced order observer is something like this, you want to drop parallels actually, that is why we want to put here.

(Refer Slide Time: 37:03)



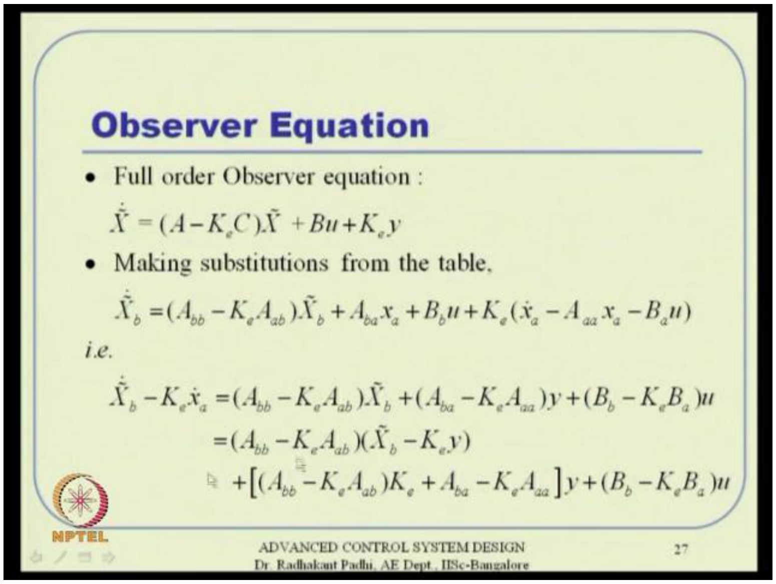
Full – Order State Observer	Reduced Order State observer
$\tilde{X}$	$\tilde{X}_b$
A	A_{bb}
Bu	$A_{ba}x_a + B_b u$
y	$\dot{x}_a - A_{aa}x_a - B_a u$
C	A_{ab}
$K_e(n \times 1\ matrix)$	$K_e[(n-1) \times 1\ matrix]$

So, if you observe this this slightly closely, then I can formulate this table nicely basically. This is this is the full order observer side, and this is the reduced order observer side. So, let us see one by one. So, you see this full order observer you are interested in X tilde, here we are interested in X tilde b; because X X tilde a we know already that is X a. Now, here we have A matrix, here we have A bb matrix. So, that is what we put here actually.

Here, we have B u only, but here we have this entire term well that is known to us. So, that is nothing but b A b a times x a plus B b u that that is what you put here. Here, we had only y which is like C times X, here it is we have there is y equivalent which is something like this. So, we put that one. Here we had C, here we had A a b I mean A a b. So, that is what we put C, and the A a b.

Here we had K e which is actually n by 1 matrix sort of thing, here it is just 1 lesser n minus 1 kind of vector basically. So, this is the this is the kind of comparison that you do with full order and reduced order. So, whatever or in the whatever things we know for the full order we can actually apply in the reduced order sense actually; whatever equation we know from this side, we can directly substitute the corresponding things from this side this table and we can design a reduced order observer. Let us see in a little more detail.

(Refer Slide Time: 38:29)



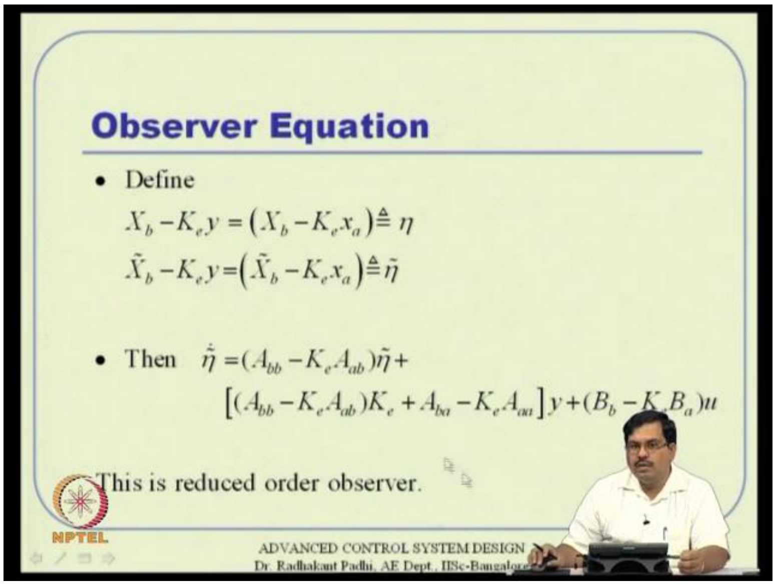
So, full order observer becomes like this, that is a full order observer equation written in a different way this is you can write this minus K e C out there if you want to; this is A times X tilde plus B u plus K e times y minus K e C times X tilde. So, that we have that is that is why I mean this K e y, so what I what I told here is, the what we saw there in this particular term is shifted to that actually that side I mean this is like this is. If you if I mean if you want to put it, then this will become y minus K e C, K e C times X times no sorry this is not K e is already there, so this is C times X tilde basically.

So, if you want to you can put it that way. But if you want I mean you can put it the other way also. So, you can just keep it that way actually, either way actually just keeps you that way same thing as that actually anyway. So, if you if you want to design an observer equation, all that you need to do is you would look look back to this table and put wherever A is there you put A b b; wherever B u is there, you put that one and things like that.

So, if you do that A, I can substitute all those terms. For example, y if I substitute y in terms of those things; so that is what I get it in terms of y here. So, B u is nothing but B b sort of thing; this B u I will substitute is that one actually. So, this B u I will substitute as those term actually that way.

So, X X tilde is nothing but X b tilde. So, I keep on substituting that and give getting I will I will get this observer equation for the reduced order observer. But I will let us try to do a little bit simplification out here. So, X b tilde when you minus K e times x a dot this particular term if I take it left hand side, I am left out with all that is which I can combine and try to put them together in this way I mean this is just math algebra simplicity actually.

(Refer Slide Time: 40:46)



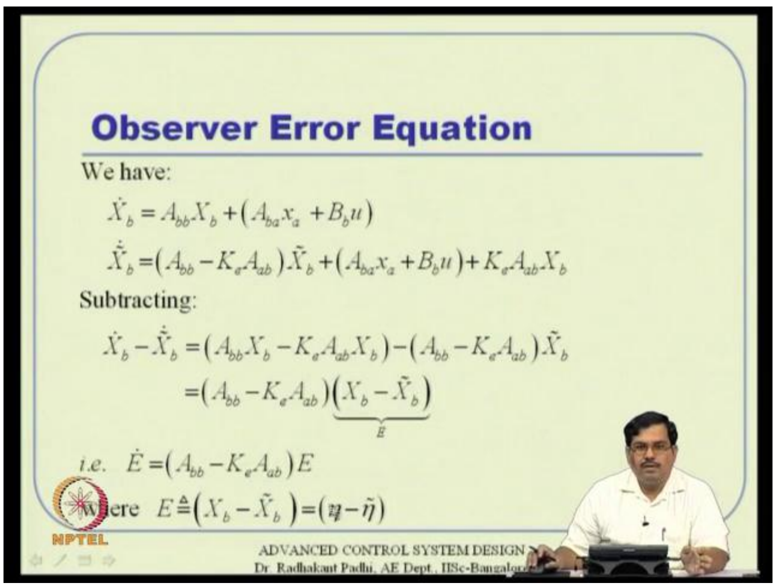
Now, it can define some sort of a term which is like a X b minus K e, remember x a is also y; y is nothing but x a. So, K e x a dot is nothing but this one suppose I want this dot out that means it will give me X b tilde minus K e times x a. That we that is nothing that is the reason why I am defining this this eta term, which is nothing but X b minus K e times x a; that is a new variable eta I am just defining that actually.

And similarly, eta tilde I am defining as K b I mean X b tilde minus K e y. And also remember that, when eta goes to eta tilde, then X b will go to X b tilde, because this term will cancelled out anyway. So, the whole idea here is we somehow do not want see this x a dot terms actually in a in a good observer equation actually. So, we are interested in observing eta instead of observing this I mean the working directly with this equation and all that. So, we want to eliminate this kind of x a dot term out here.

So, by defining these two terms, we can go away we can tell this is nothing but eta tilde dot now. This is nothing but eta tilde dot; this is y is nothing but x a same thing actually. So, eta tilde dot is this term is nothing but this right hand side what we had actually right.

So, all that terms will remain as it is actually. So, we have a term from y, we have a term from u, and these two terms are kind of known to us; and these dynamics is what we are interested in. So, this is nothing but reduced order observer equation actually. Now, I we are not yet done, because this K e we need to kind of design. So, until then we are not done anyway.

(Refer Slide Time: 42:31)



So, let us see that, how do we formulate and error dynamics and things like that. So, we have a, this let us go back to this equation (Refer Slide Time: 42:43), we have this term initially we started with this term. So, we put it back there and then X b tilde dot actually what is our observer that that part of the observer dynamics, what we have here, not that, here probably (Refer Slide Time: 43:01).

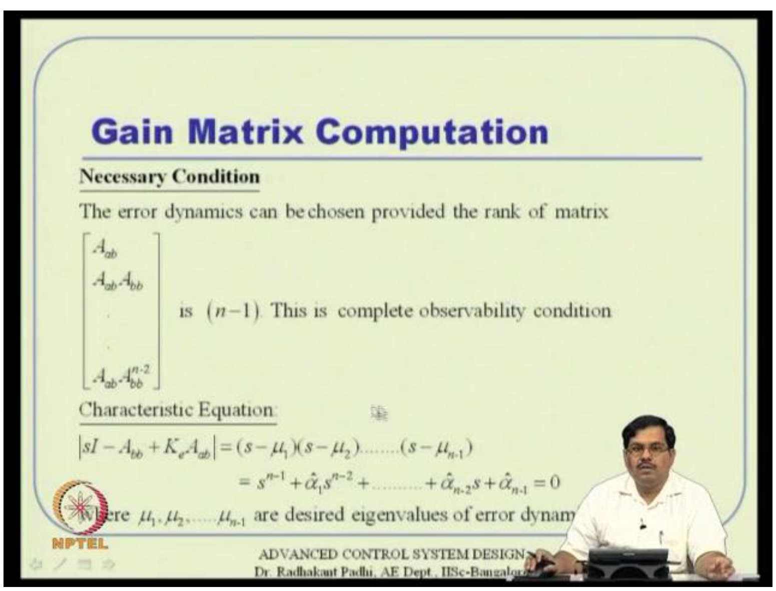
So, you put it back there and then I mean if you just look at this equations and try to subtract each other let me subtract this second equation from first. Then what you get is this error which popping up in the left hand side, and I am left out with all these, where these two terms cancel out actually, this is this is nice to see; this one and this term will cancelled out and you are left out with only that term actually. So, we can do some algebra here and then tell this is what what I have.

So, essentially it gives me E dot equal to this this matrix time E basically again, where E is defined as eta minus eta tilde now I mean E is like say this is same thing as I (( )) the way. This definition when you have this this difference is same as the difference actually, this this this is common to both, so it will cancel out anyway.

So, what I am having here is actually the same sort of thing, where the error is define in terms of X b now or the the error is defined in terms of X b minus X b tilde, but is also nothing about eta minus eta tilde; eta and eta tilde define that way for convenience that is all. So, observer observer dynamics becomes like that by definition.

So, if we know if we observe eta and we know x a, then obviously X b is known to us. X b is equal to eta plus K e times X a. So, once you design K e and we have a value for eta at any point of eta tilde at any point of time, and then we have a value for X b tilde at any point of time, because x x a is known us anyway. So, we can just do that algebra there that is the whole idea there actually.

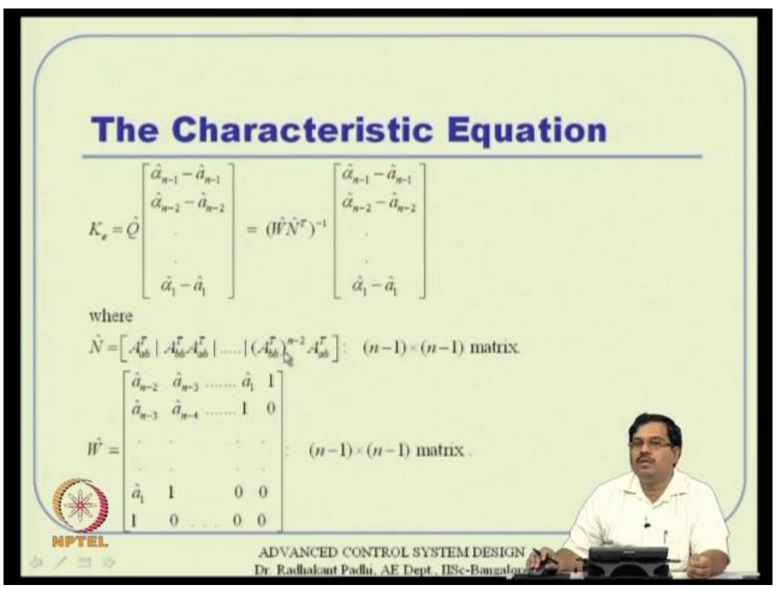
(Refer Slide Time: 44:47)



So, again the necessary condition is given in terms of observability matrix for the reduced order system in case. So, reduced order system observability matrix turns out to be like that; and this way rank has to be n minus 1, now because 1 is gone actually form there, (x, y) is not part of the output part of the state equation. So, what we need is, the rank of the matrix should be n minus 1 actually.

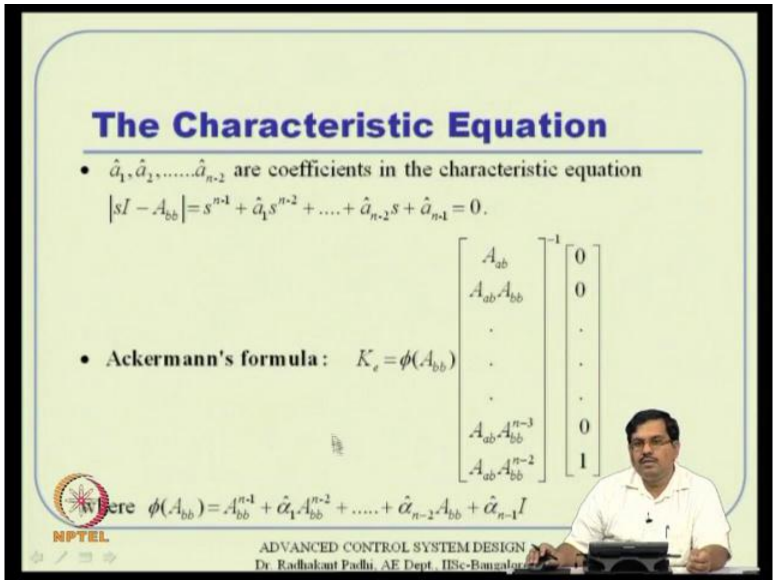
So, again you can follow this same characteristic equation, where the reduced order matrix is everywhere you have to work with reduced order matrices only. Actually go away to the same the partitions that we started with (Refer Slide Time: 45:22). So, we are all working with the bottom part of the equation now, top part is known to us anyway I mean the top part we are not so much keen, because x a directly observed. So, we are not interested to use that actually. So, that is how it is then again the same same approach you can try using your method 1, 2, 3.

(Refer Slide Time: 45:47)



And you get this is method 1, you can directly equate to the characteristic polynomial. Method 2, you can equate through the Bass-Gura formula, where W is given like that and tilde is your reduced order observability matrix or you can have a Ackerman's formula also we can we can directly use it in the formula sense.

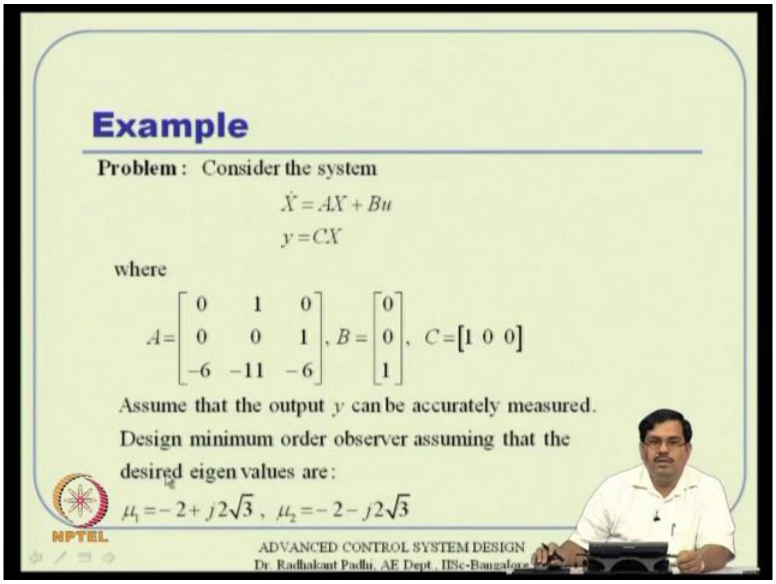
(Refer Slide Time: 45:57)



So, all these tools are available to us say already to us actually. Again the same separation principle holds good nicely (Refer Slide Time: 46:08), where you can show that the the poles of the overall system are nothing but about poles of the poles due to pole placement controller, and that is poles due to reduced order observer.

So, that is how we in case, so let me just for complete circles a controller and this is this is observer part, this is controller part. So, that is on the overall system poles are given as the the poles are nothing but the same poles which is coming from the controller design and which is coming from the observer design. And therefore, the pole placement design in the design of the reduced order observer also can be made independent of each other. So, we can do that design independently actually.

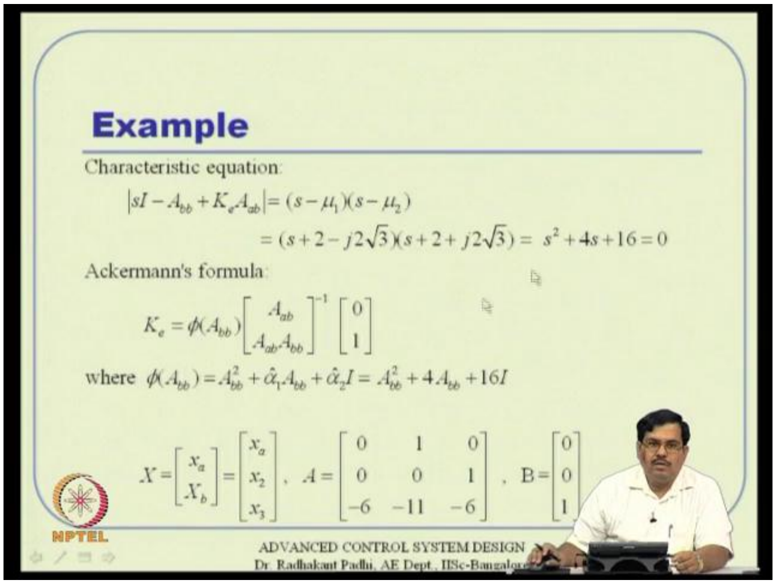
(Refer Slide Time: 47:04)



Again another small example I mean these are all just substituting and getting that for right answer sort of thing. So, we have a third order system let us say and you have a first first state is kind of known to us let us say. That is C is 1 here actually, so first state x 1 is known to us directly. So, we are interested in kind of observing the, what is x 2 tilde and x 3 x 3 tilde basically.

So, we want to design a minimum order observer. Assuming that the desired Eigen values for the error dynamics again are like that (Refer Slide Time: 47:38). So, the design value desired Eigen values for the error dynamics are something like this actually.

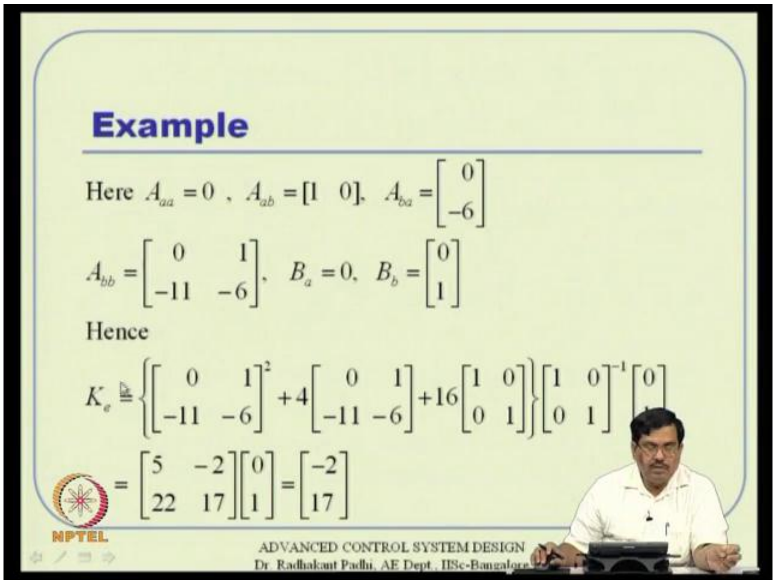
(Refer Slide Time: 47:45)



So, then you can go back and substitute your characteristic equation turns out to be like that, remember these are all partition matrix, small smaller dimensions than what you had originally. We put that in characteristic equation, then expand it and you will get s square plus 4 s plus 16 which is equal to 0. That means you you collect the the A 1, A 2 from there and then you have a Ackermann’s formula you can directly use using this way.

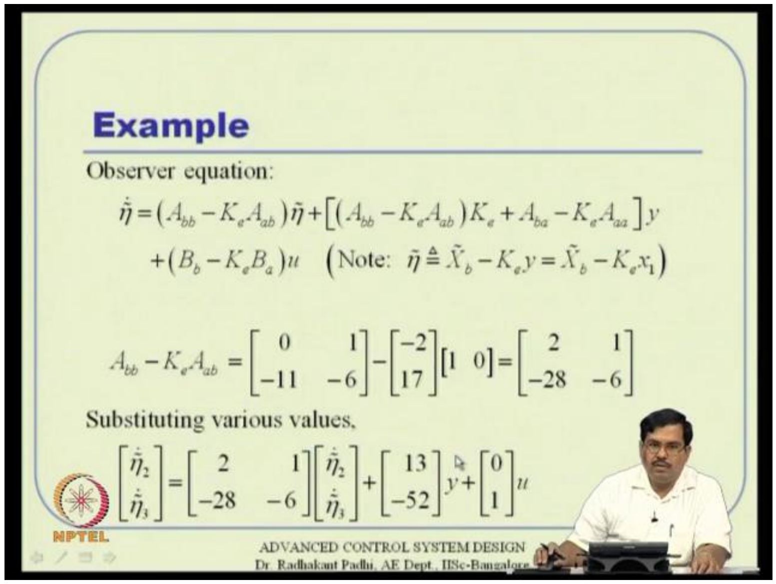
So, once you use this the once you know what you are I mean this this small matrices and all that, we need to compute for this gain matrix I mean ultimately is that is what you are interested in; and that will be directly given to you, you can use any method they that you want really. And given method 1 will be ok here, because you are working with a 2 by 2 matrix out ultimately basically.

(Refer Slide Time: 48:33)



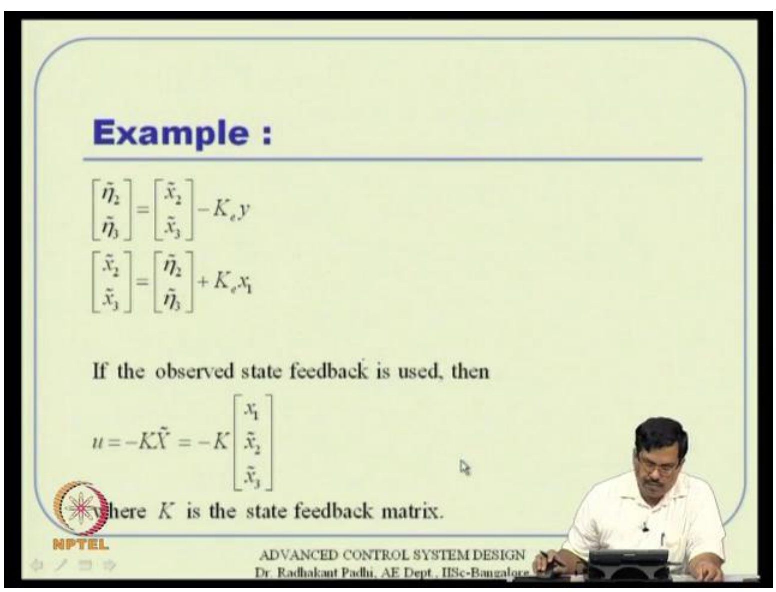
So, this is if you substitute all the things that you that are available to you, the various partition matrices and all that a for A and B. Here A a a is 0, A a b will be a b is that, A b a is that, A bb is that. Then B a, B b A bb is also like that. So, you can simply substitute and get a get a formula for get a value for the gain matrix that is acting with a reduced order observer actually; you know again the computing the gain matrix is not everything we also need to worry about the observer equation that is about that is what we need to propagate actually. So, stopping here is almost done, but only about 70, 80 percent actually without that this gain is actually useless.

(Refer Slide Time: 49:19)



So, we have to find out an observer equation also and this is the formula again we substitute various values. And of ultimately we see that, this is my reduced order observer dynamics. So, I start with some guess value for eta 2 and eta 2 tilde, and eta eta 3 tilde; and I keep on propagating this equation along with some I mean some control which is like minus K times X basically.

(Refer Slide Time: 49:47)

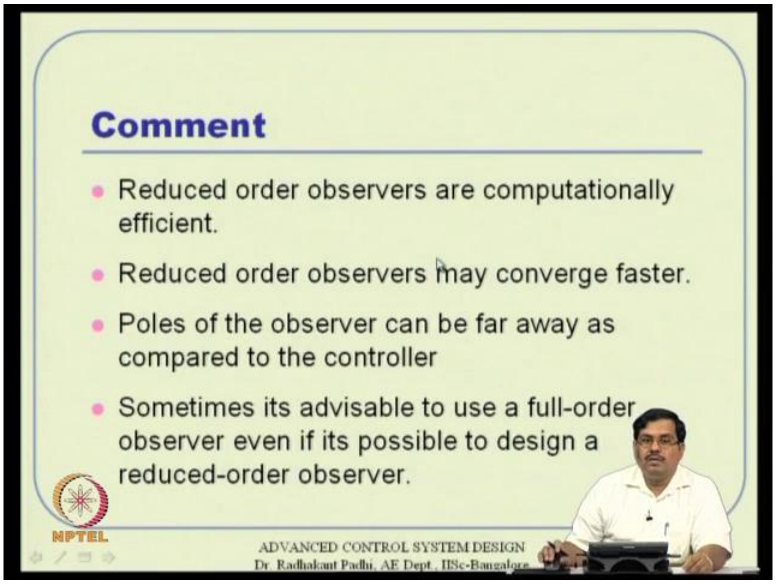


And this control is minus K time X tilde now, where X tilde is x 1 is directly known to us, and x 2 to x 2 and x 3 are something that we are observing actually. So, x 2 instead of x 2 and x 3, we will substitute x 2 tilde and x 3 tilde, but x 1 will remain as it is actually. So, that is how the the the state feedback control will act as far as control formula is concerned actually.

So, again they by substituting various values you will get, the you propagate this dynamics, and probably this dots are not there, again a small print mistakes probably these dots are not there here (Refer Slide Time: 50:26). So, this is a basically system dynamics that you need to propagate. And then once you propagate that at any point of time you are getting values for these and by definition this is this is that and that is how we have defined actually.

So, we can recover this x 2 tilde and x 3 tilde from there, once you know this actually, we can recover from there actually. So, the observer state feedback equation turns out to be like that actually. So, now where we are actually we we need as far as operational things are concerned, we really do not operate on x a dot or y dot; y dot is let us just not use by redefining this this eta that way I mean that is the reason why we redefine eta variable that way (Refer Slide Time: 51:12). So, we kind of eliminated this necessity of this x a dot and all that; that is that is the key observation there. So, that is how we act actually.

(Refer Slide Time: 51:26)



And then before we conclude there are certain comments out there as I keep on telling this reduced order observers are some computationally efficient; certainly because you are acting with smaller dimension matrices actually that is the reason. And the reduced order observer may converge faster. See, the word may is important, because it is not now it is not universally guaranteed that it will converge faster.

And that is the reason why sometimes eta locate it is also advisable to use full order observer, even if the design of a reduce order observer is possible actually. If the even if it is possible to design, if reduced orders observer is always a good idea to design a full order observer also together. And then see compare which is better, then based on that you take a decision whether you buy the reduced order observer or full order observer actually.

So, there are (( )) for that as for as implementation is concerned. And also one more last comment is one more time, this this poles of the observer can be far away as compared to the control, actually it should be far away basically. Because whatever whatever this controller gains that you are computing based on the certain recommendations that we saw last time, that the closed loop Eigen values of poles should not be too far away from the open loop poles; and the closed loop poles should not be too far away from imaginary axis all those things that we discussed actually last class.

Ultimately, these are realized through a control system. That means you are paying the price for for realizing that controller, but as far as observer things is concerned you are simply computing it I mean this is nothing but computer computation. So, only thing that you have to worry about is the noise amplification part of it, because ultimately the no known measurement is error free, all measurements will have some error rather; even the they are the small errors and things like that, we do not want to amplify them further actually.

So, you have for only because of that, you have to restrict your against it some value I mean your pole locations at some some location, but other than that there is virtually no restriction actually, they are not realized to any physical mechanism. So, the poles of the observer should be far away in the thumb rule is about at least about four times further from control pole locations about three I mean at least three four times further away from the control loop controller poles; so that, the error decays faster.

That is what we ultimately want, X tilde should approach X as soon as possible. And that you can you can do that by selecting the pole locations further and further away and only go to that level from for which is the noise amplification properties will not amplify basically, but other than that there is no restriction you can do that actually. So, that is the recommendation that is a that is universally there all right.

I think for last two classes, we discussed about pole placement controller and observer along with reduced order observer. And you can see many many places these are all applicable and these observer concepts are also applied I mean kind of rigorously in in Kalman filter design. So, which is lot more practical thing compared to what we saw today.

So, this will this is just a pole placement observer, then people have thought about how do you handle that very efficiently, when you have multiple output information, because that is what reality is. We certainly need multiple sensors multiple output and all that actually; and that will relates to the multiple or multiple I mean this multiple input control designs sort of thing and which is very need to as for as this LQ controllers are concerned, Linear Quadratic controllers and all. That will ultimately lead us to LQ observer also in parallel and then, L Q observer concept you will be very tightly related to Kalman filter actually; that is the way it develops that way.

So, with those comments I think I think I will stop here for this particular class. References are like this (Refer Slide Time: 55:22), largely I have taken from first reference for some good information is also contain in the second one, if you are interested. I think many of my material that I covered is from first book actually as far as control design, control ability, observability and observer design using pole placement ideas are concerned actually, but some good information you can also see in the second one actually, thanks a lot, I will stop here.