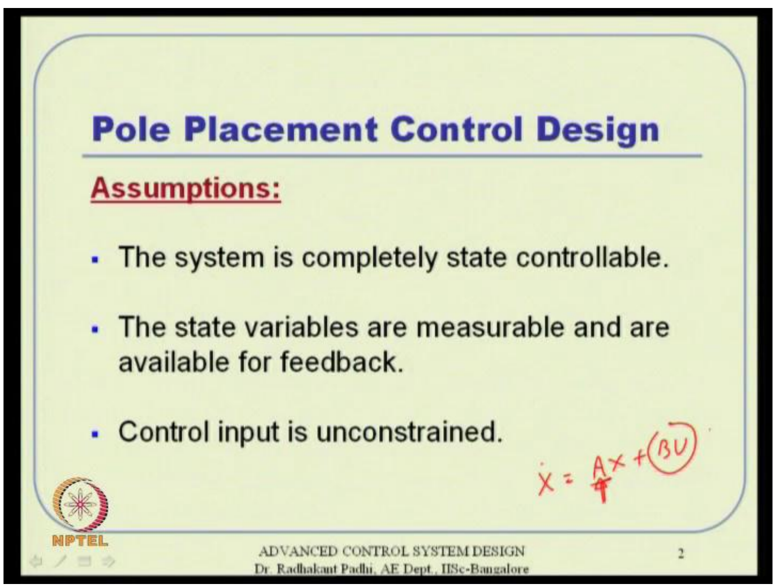
Advanced Control System Design
Prof. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Science, Bangalore

Lecture No. # 21
Pole Placement Control Design

Hello everyone, we have completed this analysis tools so far, like stability, controllability observability all that. Now, the question is can we make use of some of those designs and some of those tools for control design? So, it is also called as control synthesis actually.

So, analysis part being over we will continue with synthesis part next couple of classes. So, first one in the series we will see this pole placement control design in this lecture 21. So, let us start with the concept. We are given a kind of plant for which the dynamics are known to us and it is a linear system of course. X dot equals to X plus B U and then can we design a control system U, I mean that is what we are interested to ask in this lecture.

(Refer Slide Time: 01:05)



So, the assumptions behind this particular design, first thing is the system is completely state controllable. So, if it is not controllable obviously, you cannot do anything and hence we cannot do pole placement design either.

And the second one is the state variables are measureable and are available for feedback. So, in other words you can actually implement this control. So, if you design a formula for U in terms of X, then X would be available for computing U. I mean that is the if part of the X is not available, then you can have your mathematics ready, but you cannot implement the controller. Because information is not available otherwise.

We also assume that the control input is unconstrained. That means when normally the control inputs are typically constraint that means the magnitude of control, and the rate of control all that are not really infinite. They have to be constraint by physical constraints, they have to be constraint like energy input all sort of things.

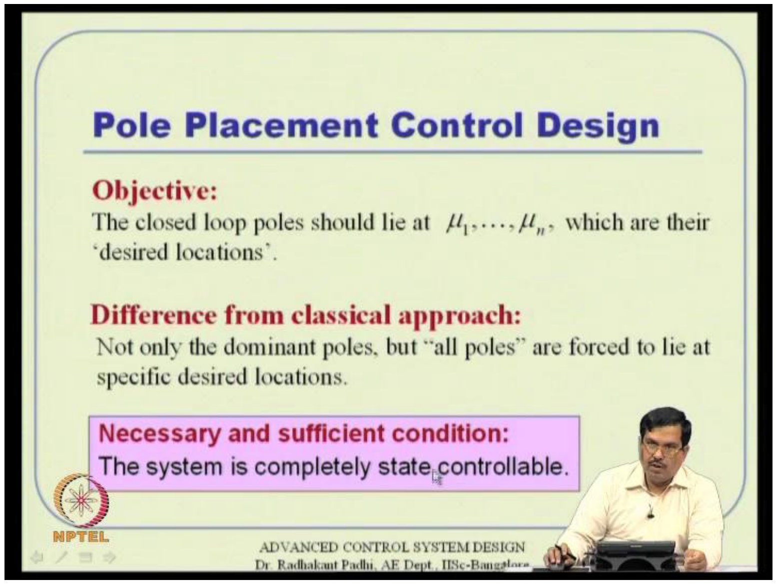
But here we will assume, as far as the theory development is concerned, that the control input is unconstrained. So, these are the primarily three assumptions that we will rely on actually. First is system is controllable. Second is its states are available for computing the controller. And the third is the control is unconstrained actually. So, let us see what is the philosophy here. The objective here is we, the original system that we are talking right. So, that is the X dot equal to A X plus B U. Original system what we are talking here, is something like X dot equal to A X plus B U.

Remember this system is controllable, but it is by no means it is need not be stable. So, the system can be unstable or even if the system is stable, the poles of the A matrix that means the Eigen values of the A matrix need not be at the desired location. That means even if you ignore this control input for a second, and even if the system is stable you may not be, the system may not be performing the way we want it to perform actually.

So, we want to enhance the response sometime. We want to increase dumping sometimes like that actually. So, that just because the system is stable does not necessarily mean, that we do not need a control system design you may still need that.

So, that is what you are telling. So, what you are telling here I mean unlike our classical control system. Typically in classical control system we will be bothered about dominant poles, that means N P R design we will worry about only two dominant poles and try to locate them.

(Refer Slide Time: 03:50)

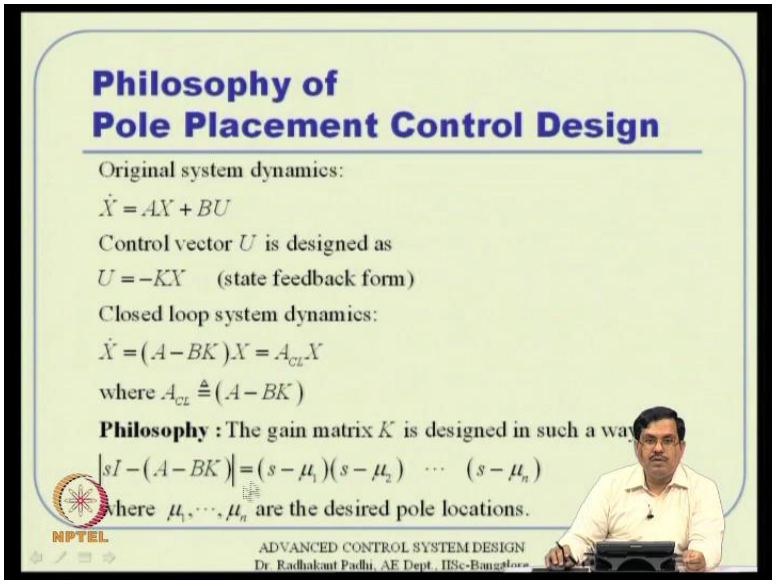


Now, we need not worry about only two, but we can talk about all the pole locations. So, we would lie at our choice the locations should be dictated by our choice, which will eventually come from either time domain specifications, most of the time or sometimes partly from frequency domain specification also.

So this mu 1 to mu n are the desired location remember that. So, in other words the original system poles need not be at those locations it can be at somewhere else. But we want to design a controller in such that, the close loop system pole should be located here.

So, as I told the difference from classical approach is not only the dominant poles, but all poles are forced to lie at specified desired location that is the difference. I already told necessary and sufficient condition turns out to be the system needs to be state controllable, completely controllable. So, by assumption this is controllable anyways, so we satisfy this actually.

(Refer Slide Time: 04:58)



Now philosophy of loop what is the idea there? What is the philosophy of pole placement control design? We have this plant which is I mean X dot equal to A X plus B U linear system dynamics. And we want to design a control vector U, in the state feedback form. That means U equal to minus K X.

So if you design that U equal to minus K X and then the close loop system dynamics becomes X dot equal to A minus B K into X. And we typically tell that this is nothing but the close loop system matrix A minus B K.

So, we have an idea that, suppose somebody kind of gives us this K, then this is our close loop system matrix. But the problem here is this is the reverse we want to design a K. The formula is given to us. U equal to minus K that is the formula we want to kind of use for the control design.

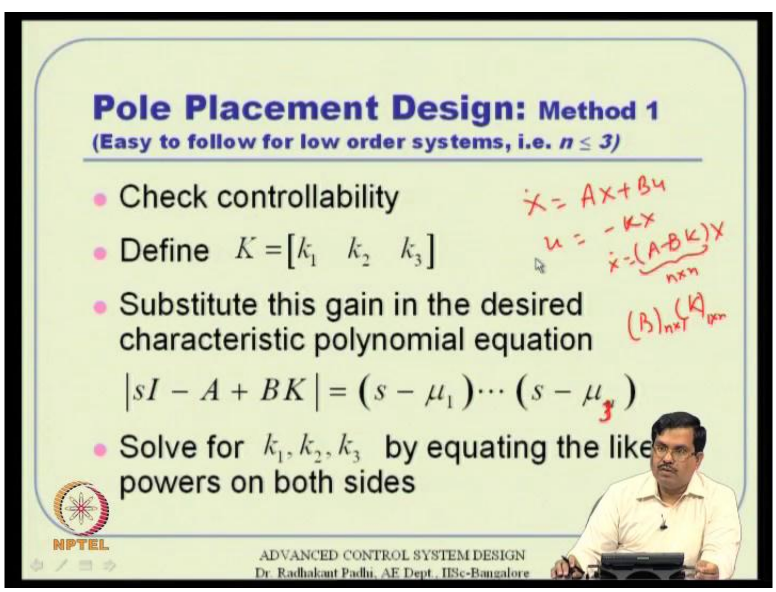
So the control freedom of control is actually works down to freedom of designing this K matrix. So the gain matrix needs to be designed. So, after designing that the close loop matrix becomes A minus B K. And the philosophy is like we want to design this gain matrix K in such a way, that the close loop poles or the close loop characteristic polynomial rather, which is dictated by the s I minus of close loop A matrix determinant of that.

So left hand side is close loop characteristic polynomial is nothing but the desired characteristic polynomial. Because of the pole locations are known to us. So, it is this right side is nothing but the desired characteristic polynomial. So, if these two becomes equal then obviously the close loop system matrix will have these poles. That is the whole idea.

So, once again this mu 1 to mu n is known to us. So, obviously our desired poles are known to us, the desired characteristic polynomial is known to us. Hence, all that we are doing is the closed loop system matrix which is A minus B K and the characteristic polynomial is given by that s I minus this A minus B K determinant. So, we are just making them equal.

Once we make it equal, then obviously our job is done. Another question is how do we make it equal? There are various ways of doing that, we will study method one two three here. Method one is first, but before we go there, before we would develop further things, first we will study single input system. Pole placement design has unique solution if you consider single input system, states can be many, but the control is just single in that situation pole placement technique design is unique and there is no confusion for that. So let us study about that particular thing. I remember one u is scalar then B is a vector, it is no more a matrix.

(Refer Slide Time: 07:56)



So method one: That is what we talk about so far. The very first step of any control design, not necessarily pole placement is check controllability. Once the controllability condition is satisfied, then there are further is hope that you can actually design a controller otherwise no.

So first thing we need to do for any control design even if it is not told to us explicitly, is to check controllability. Now we are telling method one is very straight forward just derived from this condition that we just discussed here. So, this particular method is not very neatly kind of suitable for higher order system. It is easy to follow for low order systems in general.

So if n is less that equal to 3 then probably you can do it manually. This is the manual method basically. So, what you are telling that we have a single input system. That means you have this X dot equal to A X plus B u and u is scalar. And u you are telling minus K X so, X dot equal to A minus B K times X that is what we are interested here. That means the dimension of this A minus B K, A minus B K incase that is the dimension has to be n by n. And remember B dimension is nothing but n by 1. So K dimension has to be 1 by n.

So, that is what you are telling here, that we select a matrix of let us say, that is n equal to three here. That is what we are talking. So we select a gain matrix k 1 k 2 k 3 one by three here. It is just a row vector actually. And then we substitute that in to the formula that we just talked about. This is the left hand side of characteristic polynomial that is what we substitute here. So s I minus A plus B K determinant, that should be equal to s minus mu 1 into s minus mu 2 into s minus mu 3 really, I mean you do not have to continue up to n you have 3 poles only.

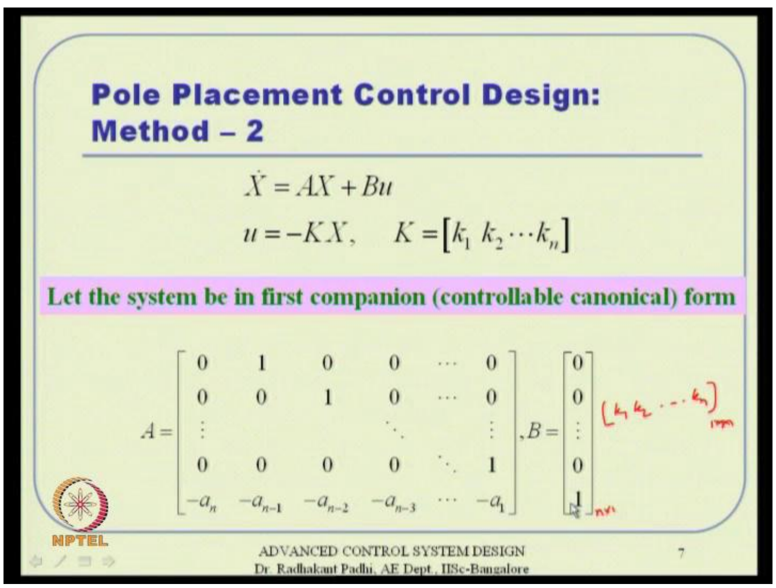
So in n principle, this is third order system then this has to be three basically. s minus mu 1 into s minus mu 2 s minus mu 3. So we have a third order polynomial in the left hand side, and we have the third order polynomial in the right hand side. So just equate the coefficients and solve for k 1 k 2 k 3. So, the third order polynomial will give three coefficients and this, well four coefficients in general. But s q plus that I mean zeroth order term taken into account. You just take this first order coefficient is s q which does not give us anything it is just an identity. But, afterwards s square though will have a coefficient s will have a coefficient and we have a constant term. That is also kind of coefficient for zeroth power of s.

So, if you equate these two from these three coefficients, we will get three equations. And our entire equations had freedom of three the k 1 k 2 k 3. So, we can actually solve this. So you see I mean I am not giving an example here, but if rather fairly easy to do that all that you are doing here is just equating the coefficients and solving for k 1 k 2 k 3. Once you solve for k 1 k 2 k 3 your gain matrix is ready. Hence your control formula is also ready u equal to minus K X so, that is how it is.

Now method two so, method one is in general it is for n equal to 1 2 3. But if it is more than 3 4 5 and all that, it becomes very cumbersome to do that manually. And even if you are able to write the coefficient equations manually, then probably you need to do symbolic computation anyway, for solving this in a computer manner or may be by numerical methods

So, if this not a very neat way to design pole placement controller for higher order system. So, obviously people have thought about how to circumvent that by various other alternative approaches.

(Refer Slide Time: 12:06)



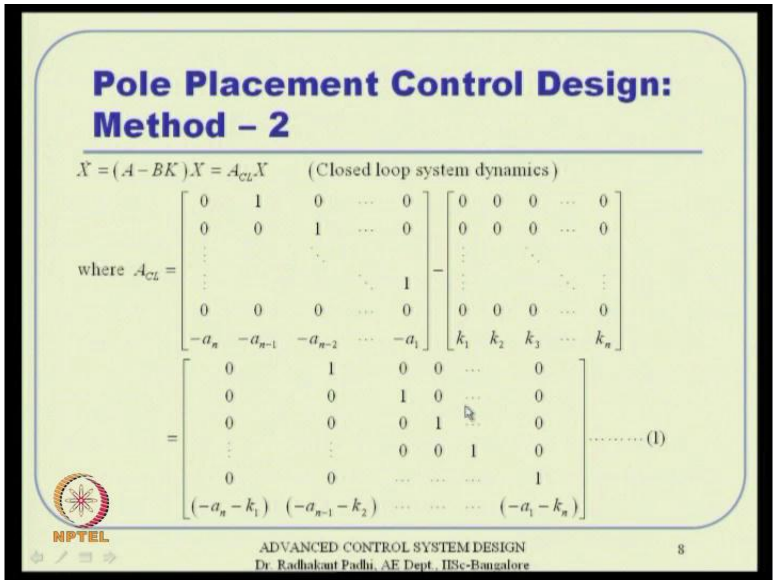
So let us study that method two which is little more general. So, what you are having here is X dot equal to A X plus B u in general this is not third order system, it can be any order system. But, u is still a scalar. So, you have u equal to minus K X, and K is nothing but 1 by n row vector.

So, once you have that then the obviously A is that and B is that, what we are assuming here is the system is in first companion form or controllable canonical form that is the first I mean we will also see if it is not in this form what to do later. Or let us assume that the system is in controllable canonical or first companion form for which the A matrix is given like that. We have studied that before, all that n minus 1 rows will have this particular structure, B will have all 0 0 0(s), last entry is 1 and the last column last row here is some numbers, that is the first companion form.

Once you have that then and the K is that way. So, what we have X dot equal to A minus B K into X or A C L X. Where A C L can be computed that way. Remember B times k when you put K, B is this and then K you put it here, B times k you multiply k 1 k 2 up to k n this is a row vector. So, this one has n by 1 and this is 1 by n so, will result in n by n matrix. So, what you would I mean is rather easy to see, that this is like if you multiply first I mean first 0 to k 1 is 0 0 to k 2 is 0 everything else will be 0.

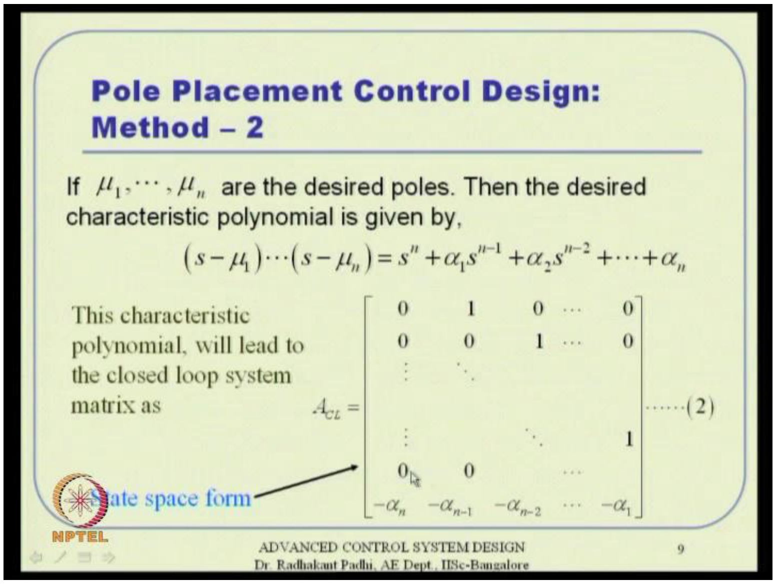
So, the first n minus 1 rows will be typically 0, and the last row will have some numbers which is nothing but minus k 1 minus k 2 and all that, minus B K term we will have result like that.

(Refer Slide Time: 13:56)



So this is A matrix minus B times k is nothing but simply that actually. First entries are all 0 and the then the last entries are k 1 k 2 up to k n actually. And the need part is the because these are all 0, these first columns are I mean first n minus 1 rows are not disturbed at all, they are just remaining as it is. So, we will have A minus c I mean that A minus B K or A C L which is like this also in the first companion form, that is the 0 1 0 0 all sort of things. So first n minus 1 rows will not be perturbed, and then only the last row will be disturbed. So, that is what we will get here.

(Refer Slide Time: 14:43)



Now if we have the close loop pole locations known to us and that is s minus mu 1 multiply s by s minus mu 2 all that up to s minus mu n. Then, it will result in a characteristic polynomial and this characteristic polynomial will lead to a closed loop system matrix in this form.

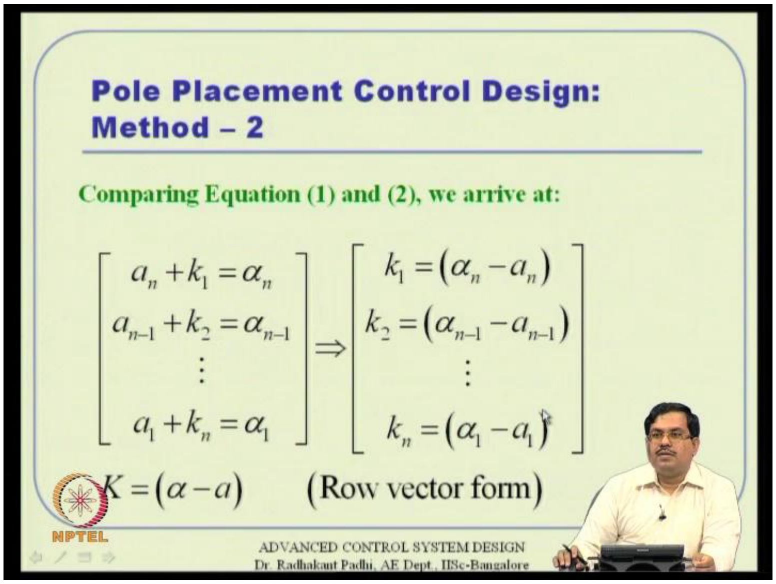
Suppose you take any output by input sort of thing I mean if you want to convert it to state space form, then this will be the resulting A matrix in the control level canonical form. And that primarily comes from this characteristic polynomial being this way.

So, what you have here instead of equating the characteristic polynomial coefficient by coefficient we are formulating a system closed loop system matrix. And then we are equating the matrix as such. This is the matrix for the closed loop system matrix and that is the closed loop system matrix which will result from closed loop system poles.

So by equating the element by element matrix form the task become much simpler. Because you do not have polynomial expressions to equate as a simple simple expressions to express, to equate in the last row only. So, because the first n minus 1 rows are identically same so they are just identities actually. Only equations are formed only from the last row of this matrix, and last row of that matrix.

So you will just element by element we will equate and then that will result in this. For example, this first one minus alpha n minus k 1 will result that is equal to minus alpha n. Similarly, the second element minus A minus 1 minus k 2 is nothing but, all minus alpha n minus 1 like that.

(Refer Slide Time: 16:19)



So then you result this set of equations will pop up and from which we will be able to compute k 1 to k n very easily rather. So in a vectorial notation K is nothing but alpha minus a where, alpha is a row vector containing alpha 1 to alpha n and then a is a row vector containing a 1 to a n actually.

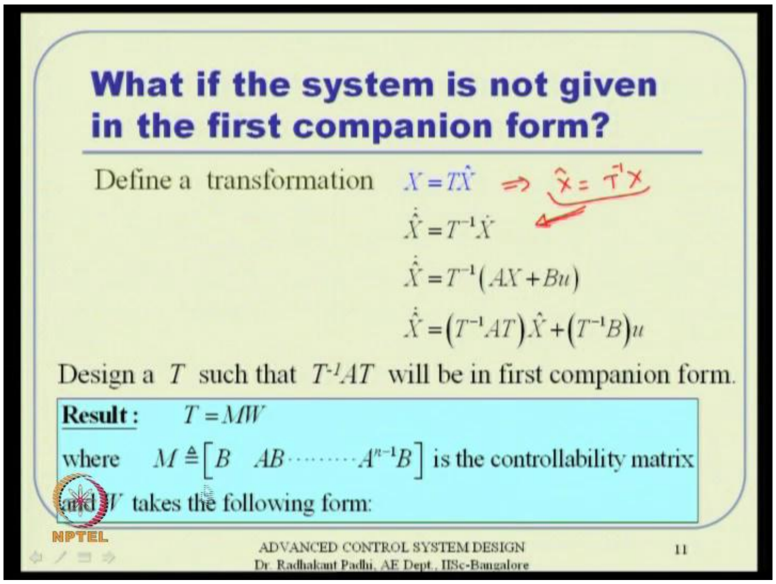
So, but remember it is a reverse order. So, this you have to be slightly careful while using this actually. What we are getting here is k 1 is equal to alpha n minus a n and k 2 is alpha n minus 1 minus a minus 1. Whereas, k n is nothing but alpha 1 minus a 1 so, it is kind of a reverse order.

So this vector alpha and n is to be defined properly. So, do not just start defining alpha 1 to alpha alpha n rather it is, alpha n first and then alpha n minus 1, then alpha a alpha 1 at the end.

Similarly a vector is also defined that way. Now, what after you remember that that is very easy because you have these coefficients ready any way right. So it is just a formula. So, what is the design procedure then, design procedure is like let us say alpha these are the closed loop pole locations.

So we just expand and collect these alpha 1 alpha 2 up to alpha n, that is where you will collect. And then we already have this matrix this this coefficients a n a n minus 1 all that already known to us from the system matrix. So, then you just use the formula and get the answers sort of thing. This is just whatever these gain coefficients are given like that and you are done. The design is u is equal to minus k X anyway.

(Refer Slide Time: 18:01)



Now the question is what if it is not in the first companion form. See these are all valid this system matrix nicely popped up because it is a first companion form that is also first companion form. So, we are kind of lucky to get all that. Now the question is obviously life is not in first companion form. So, what you do in general basically. Then what you I mean then people have thought about that is the probably one of the great contributions, what people thought about it instead of starting directly which will lead me to complicated expressions and all that. Is there a transformation which will take me to first companion form?

So starting from any general form can I take it to first companion form? So that is what the idea here is. So, let us define a transformation X equal to T times X hat obviously, T is to have a is a it is a full rank matrix otherwise, the inverse will not exist.

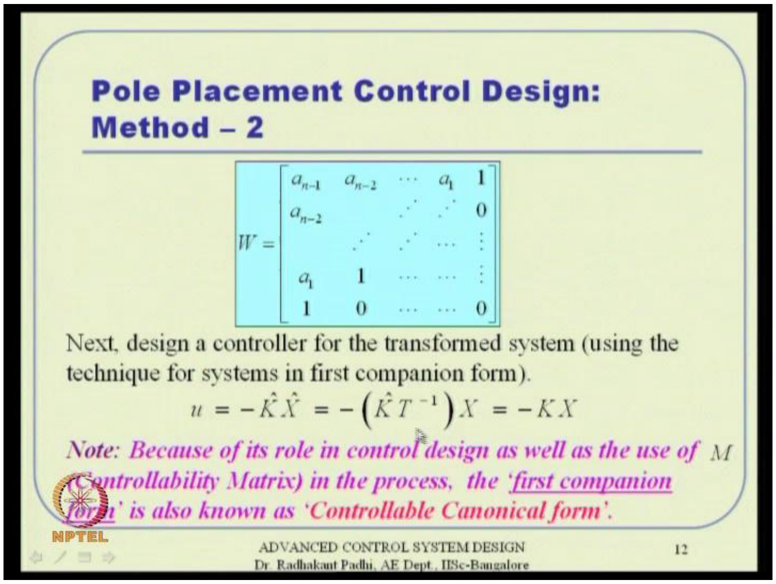
So T needs to be full rank this a transformation so, that means if I know X hat I can get X ,if I know X I can also get X hat that way. So then we want to kind of visualize the problem in terms of X hat variable let us say. So X hat is that way, I mean X hat is nothing but T inverse X so, X hat dot is also T inverse X dot right. So this expression gives like X hat is nothing but T inverse X, and T is a constant matrix so, this will give us this X hat dot is equal to T inverse X X dot.

So then this X dot is nothing but A X plus B u that we know, and again we want to convert X so, X is nothing but, t X hat actually. So what expression you are getting is like T inverse A T times X hat plus T inverse B u. That is very similar to what you get in similarity transformation. Similarity transformation for A matrix is given by P inverse A P and P is nothing but, Eigen vector I mean the matrix consisting of Eigen vectors and all that.

Here it need not be Eigen vector, it can be I mean there will be some other matrix T but, it is T inverse A T that is like kind of I mean similarity transformation basically. So, what is that I mean the philosophy here is design a T of the transformation, see you have to pick up a T such that, this T inverse A T now, will have first companion form.

So if this is first companion form then we are ready to start our problem basically. Earlier we started with the assumption that this is in first companion form. Now, we are interested in transforming the system into X hat variable. So, that in X hat coordinate system, the system matrix is in fact in first companion form. The question is can we do that? is it possible to do that? And that I mean the great thing is it is certainly possible. And T takes this form M into W, where M is nothing but controllability matrix, and w is something given like that which is nothing but coefficients of the open loop characteristic polynomial. I mean that is what the a matrix coefficients basically.

(Refer Slide Time: 21:10)



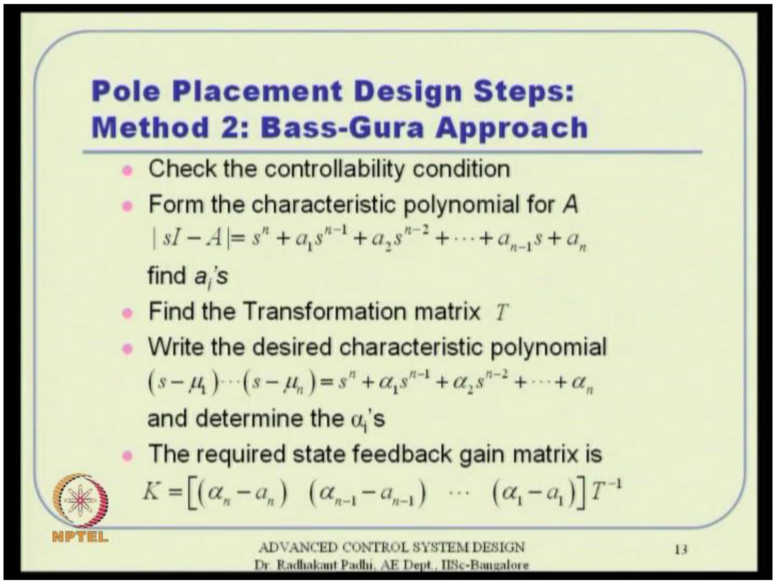
Initially you have a matrix I mean coefficients which are nothing but, coefficients of this characteristic polynomial if it is in the kind of first companion form, or the in general otherwise it will have this a 1 a 2 all this a n will come from the open loop a matrix actually. We will see an example to see, how do we select this and all that actually.

So, open loop characteristic polynomial will give us these coefficients and this M, I mean is nothing but the controllability matrix. So you have the T T is nothing but M time W and once you apply this then you are run actually the T inverse A T will be ready now. So, this will be in first companion form and hence you can compute your gain matrix in terms of X hat variable remember that.

So u will be in terms of X hat, what X hat and X you can again use that X hat equal to that and then you can write it in terms of X actually so that is not a problem. So, that is what I mean written here u equal to minus K hat X hat. But, K X hat is nothing but, T inverse X so K what you ultimately you are interested in K. So this K is nothing but K hat times T inverse that is your gain matrix.

So just a comment here. Because of this role in control design, I mean you can very clearly see that there is a need for controllability matrix something like that. The first companion form is also known as controllable canonical form I mean this is just a comment sort of thing. First companion form is heavily used in the control design really So, that is why the term comes from there itself.

(Refer Slide Time: 22:57)



So, what is the pole placement design steps in general? This method two is also called as Bass Gura approach for their contribution. So first thing to do is controllability condition check, you have to check controllability condition. Then you can form the characteristic polynomial of the open loop A matrix. So s I minus A characteristic polynomial is one third of polynomial. So this is given in this form so, you can collect this a 1 a 2 up to a n which will be necessary for forming this W matrix, that is where these coefficients will come from.

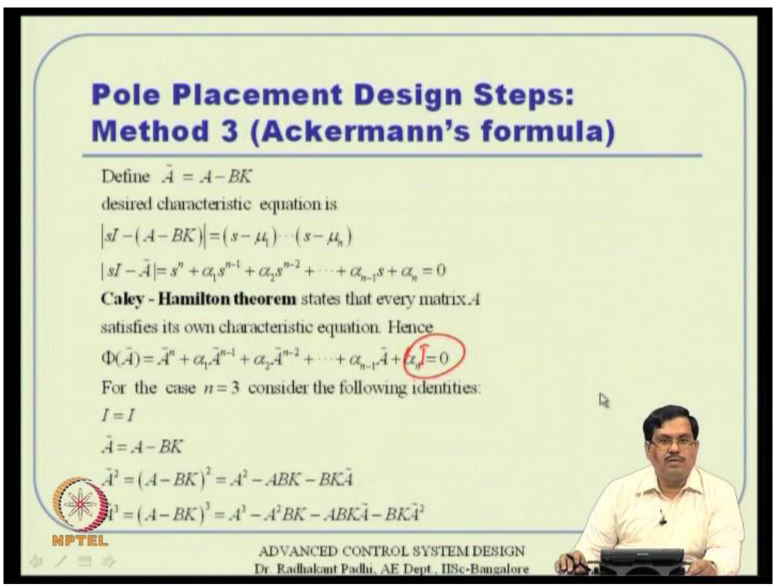
So you can collect these coefficients. And then the transformation matrix is ready where T is nothing but, M times W where M is controllability and then W is also ready now. Now we can write the desired characteristic polynomial which is in this form. And then this will give us a method of selecting this alpha 1 to alpha n now. These are the coefficients of the desired characteristic polynomial.

So a 1 to a n are ready from open loop characteristic polynomial, alpha 1 to alpha n are ready from desired characteristic polynomial. And the desired gain matrix is nothing but this one right, K hat times T inverse and K is nothing but that what we discussed before, this time this will be k hats now basically.

So once you substitute all that then this is nothing but that actually. So you have a gain matrix which is ready for you now. So, u will be ultimately minus K times X, after multiplying with T inverse this K will act with X actually. So that is the method to approach.

So method one is just equating the coefficients of the two polynomials. These two polynomials whatever, one polynomial is here one more polynomial here, you will directly assuming a gain matrix form k 1 k 2 k 3. And then if this characteristic polynomial for the closed loop will contain k 1 k 2 up to k n all that and then equate the coefficients solve for those equation. Method two is little more systematic and hence simpler, and then this procedure once you follow this procedure the gain matrix will naturally pop up.

(Refer Slide Time: 25:19)



Now there is a method three which is personally I like the most actually, because it is a direct formula that is my personal preference probably. Which is called as Ackermann's formula it is in fact, it will lead to a formula ultimately.So let us try to understand that. So, we instead of A C L for notational simplicity, we will just write it as A tilde here, I mean both are same anyway.

So the close loop A matrix A tilde is nothing but A minus B K. And the desired characteristic equation is like that this already we know. So, suppose you want to expand that and all .So the characteristic equation will turn out to be like that. So, this part is polynomial if we equate it to 0, it is characteristic equation.

Now, this characteristic equation we will call it as phi of A tilde. I mean phi not A tilde really but it is phi of s sort of thing. Because it is a scalar equation. Now there is a great theorem in matrix theory which is Caley Hamilton theorem. Probably you might have not seen its usage so extensively but here is a direct usage actually.

So what it tells this Caley Hamilton theorem is that every matrix will satisfy its own characteristic equation in matrix equation sense basically. See characteristic equation is typically a scalar equation. But once you apply Caley Hamilton theorem the scalar expression that you have, if it is coming from some matrix say, then the same matrix will also satisfy its own characteristic equation which is a very very great theorem.

Because, we have information about only a one equation one scalar equation information. And you are able to conclude a matrix equation directly from there. The n by n matrix equation that is the beauty. So, if you see this usage of Caley Hamilton theorem let us say, try to we want to apply. This is nothing but the characteristic equation for the close loop matrix A tilde. So obviously this A tilde will satisfy a matrix equation that way. So, if you just see this equation, whatever s to the power n plus alpha minus to the power n minus 1 all that. And the exactly similar equation you can write it here.

Probably there is a small print mistake basically, probably. So, alpha n times I basically so you cannot write a scalar equation there actually. So alpha n times I will happen there actually that is equal to 0. Anyway, so without complicating the matter we can also take n equal to 3 for understanding the concept and it will be valid for n equal to 4 5 any order basically by the way

So without loss of generality we will consider n equal to 3 here. And we will also let us before applying this Caley Hamilton theorem and proceeding further, we also note that these powers of A tilde will be necessary for us. So we will also see that I equal to I that is a identity anyway. But, A tilde is nothing but, A minus B K and A tilde square is A minus B K into A minus B K. So, if you expand then that will turn out to be like that.

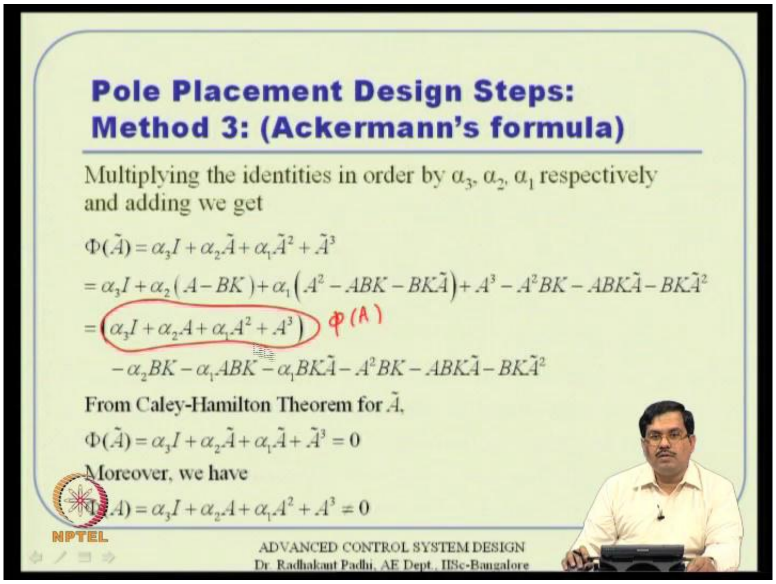
So partly I am not expanding everything, I mean what we are telling, we are expanding only the terms that are necessary and leaving out with A tilde expressions that will that is all right with us actually. We do not have to write all this in terms of A B and all that. Similarly, A tilde cube will be A tilde square into A tilde. So, A tilde square U already we have, into A tilde is A minus B K. So we expand whatever we have and then leave out the polynomials in terms of A tilde for simplicity.

So A tilde is that, A tilde square is that, and A tilde cube is that. So, what you observe here nicely see A tilde 1 A matrix pops ups with some other matrices. A tilde square 1 A square matrix pops up with some other polynomial. A tilde cube A cube pops up with some other things actually.

That is the fact that is going to be used later actually. Now the Caley Hamilton theorem will excite and tell, okay this is nothing but, this phi of A is A tilde is 0. Remember this is valid for phi of A tilde not for A, because A tilde is the closed loop characteristic polynomial, I mean close loop system matrix. And this is what we want A tilde to satisfy the characteristic polynomial ultimately we want it to satisfy this equation, that is why we started with.

So the close loop A tilde matrix will satisfy this equation, not phi of A. So that is what we tell here. So, we want to formulate this phi of A tilde first. So phi of A tilde for n equal to three, we have only three terms actually, three terms from that side remember, that not from that side or that it turns from that side right hand side.

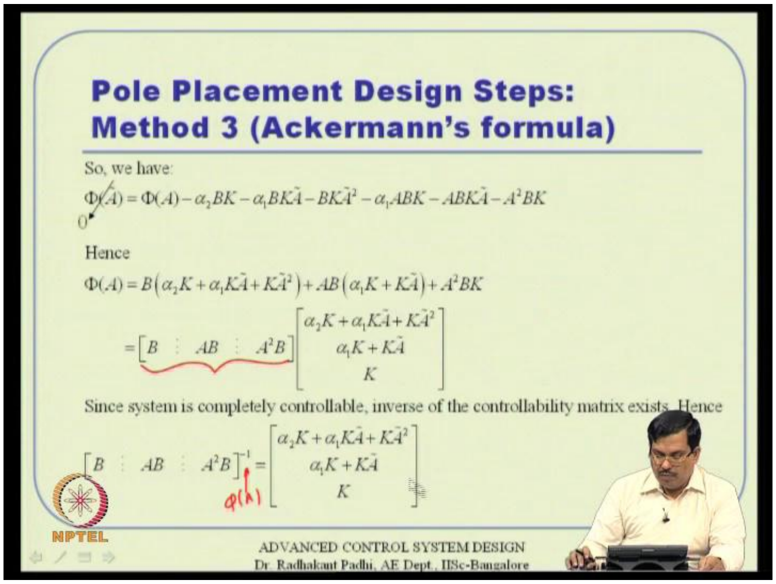
(Refer Slide Time: 30:01)



So the three terms from right hand side are like that alpha 3 into I plus alpha 2 A tilde plus I mean alpha 1 A tilde square plus A tilde cube. And all these expressions we have just derived. So we substitute all that whatever expressions we had, and it I mean we can collect the first coefficients, where first co efficient is from here and first coefficient from here first coefficient from here like that. And it turns out that this particular thing that you are talking about, this particular thing that you are looking at is nothing but phi of A matrix, right then we have a bunch of other things actually.

So that is what we want to excite and then tell if, the phi of A tilde is nothing but 0. But phi of A what we have here is certainly not equal to 0, phi of A A is not equal to 0.

(Refer Slide Time: 31:01)



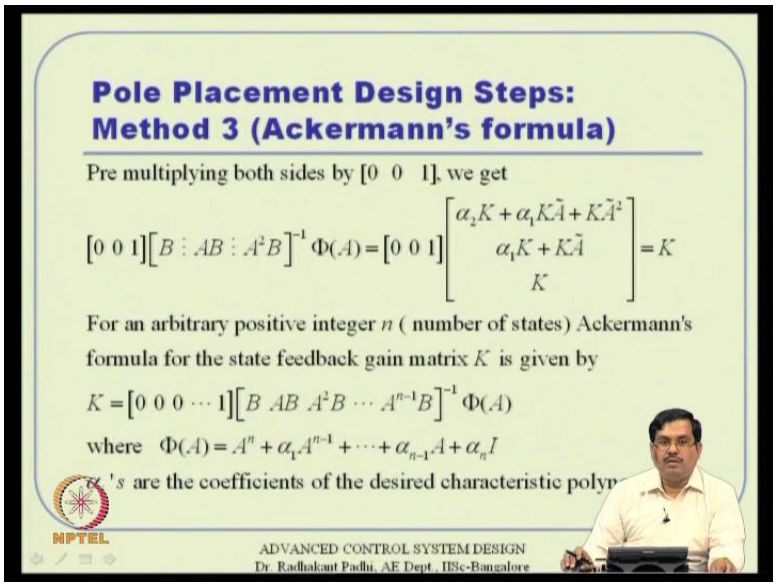
So let us use this property and tell, what we have ultimately this equation tells us that phi of A tilde is nothing but phi of A minus, all these actually whatever terms we have.

So but, what we know is this is 0. Because that is Caley Hamilton theorem, phi of A tilde is 0. So what we are left out actually we can actually get an expression, solution for phi of A basically. Phi of A is nothing but all that, and in a partition matrix sense I can write it that way. Where the beauty is it is again this controllability matrix is popping up very neatly actually.

So what you are having here ultimately is I if I want to write it. So the system is kind of completely controllable we understand. So, this matrix this controllability matrix is a non singular. And hence this is a well I think this is a small print mistake again here, times phi of A here. I will right that actually. So here this phi of A will multiply here. So if I take the though I just want to solve it for this particular matrix or kind of a whatever I see here is a matrix right .So this matrix into this matrix is nothing but phi of A. So, we want to solve for this matrix and this matrix is nothing but, this inverse times phi of A. I pre multiply both sides with this inverse actually.

So what I am getting here, I am getting so this particular matrix consists of several rows and all that. But what I am interested in I am certainly not interested in first two rows actually. I am interested in only the last row, that is my gain matrix actually.

(Refer Slide Time: 32:51)

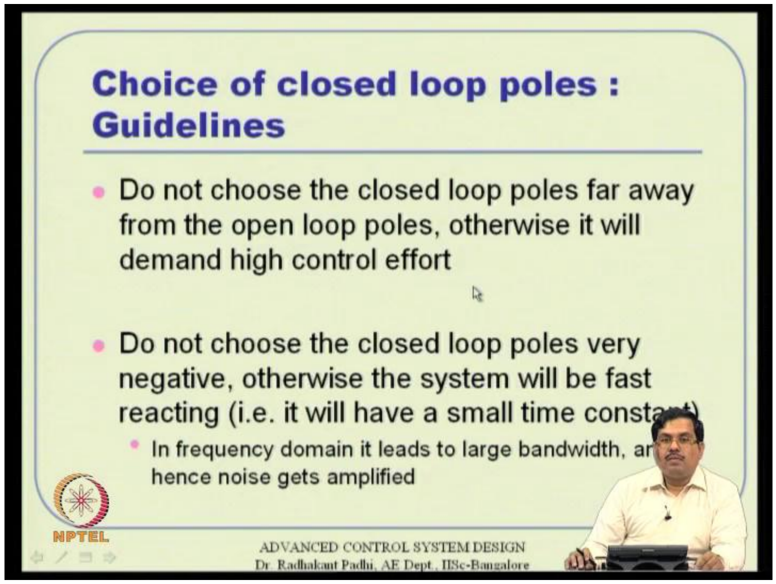


So if I want to extract my last row then what I do is I simple multiply, pre multiply by 0 0 1. If I multiply 0 0 1 it will give me the first row basically. So my ultimately what you see here K gain is nothing but a formula, directly it is a formula. So 0 0 I mean 0 0 1 with a controllability matrix inverse times phi of A. So that is the formula that you are talking about is Ackermann's formula actually.

So in general if it is of n states like n number of rows, I mean n number of states. Then you have to have a little general formula for this which will tell us that this is 0 0 0 all the way up to n minus 1 then last one is 0.Then, you have a controllability matrix inverse then phi of A actually. Where phi of A is nothing but, the same expression is phi of A tilde where A tilde substituted by A basically. We started with phi of A tilde right, Caley Hamilton theorem. We started with all that actually phi of A tilde.

So this is like that actually. So, this is of the method one two three that we discussed actually, method one is just equating the coefficients, method two is Bass Gura formula. We have steps to follow and method is just direct is just have a one single formula for designing the K matrix actually.

(Refer Slide Time: 34:12)



So let us talk about little more things. So, all this method one two three that we discussed it only gives us some mathematical formulas, provided we know the closed loop poles. Nobody has told us how to design closed loop poles and all that and nobody will ever tell us like that these are all problem dependent anyway. But, there are guidelines available and the first guideline tells that do not choose the closed loop poles far away from the open loop poles, and change only those poles to a different location whenever, it is badly necessary otherwise do not actually.

Changing the pole location is not free. We can change the pole locations only by using control. So that means the difference of this open loop and close loop pole locations is done by control effort. So the moment you change these poles different I mean far away to each other, then the control demand suits off actually, control effort suits off actually.

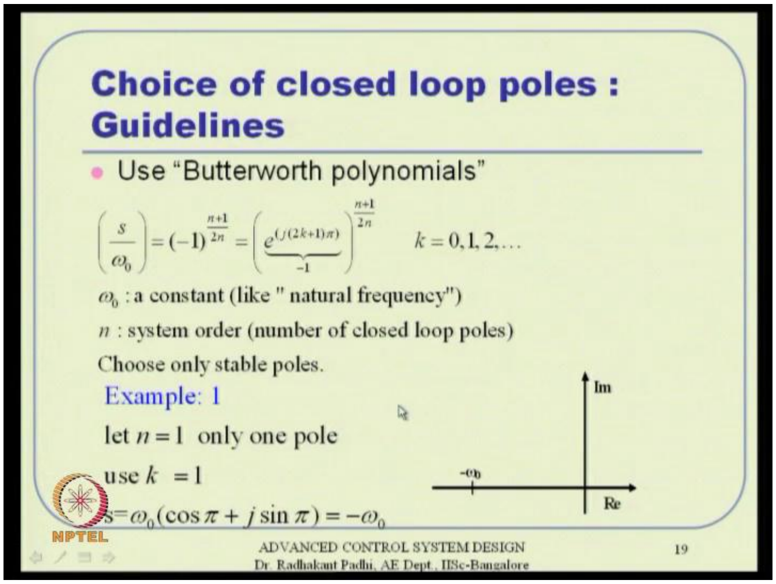
The second is do not choose the close loop poles very negative, that means you are telling if your negative side is stabilizing. So, I will go negative negative far away from the negative half actually in this plane, that is also not good because you remember if you have a negative I mean negative pole location then there is a exponential A to the power minus sigma t term actually, sigma is the real part of the pole location.

So if the sigma is large then that term is very large that means it excites fast decay and things like that. That means in frequency thing, in frequency domain it leads to large bandwidth sort of thing, the response becomes very very very fast. That means for a little input system will try to respond very fast.

So that means if you see it in a negative point perspective, then it tells us okay what about noise input, noise is also an input. It is a low magnitude high frequency input but, even if it is a low magnitude the system reacts for every little thing. So that means it will also try to react to noise actually. So, the noise amplification property becomes, I mean predominant if once you have this far away poles being there.

So unnecessarily do not get too much kind of ambitious to put the I mean closed loop poles far away from the imaginary axis. That is not good either. So, looking at a little more in depth, suppose the open loop pole one of the pole is located far away in the left hand side. Probably we would like to bring it back closer to the imaginary axis. Some applications are like that. So just because it is far away from the imaginary axis in the left hand side does not mean the system properties are really good. It may be less robust to the noise. So with increased robustness and all we can infer bringing the pole locations little more closer to the imaginary axis. So these are just guidelines

(Refer Slide Time: 37:12)



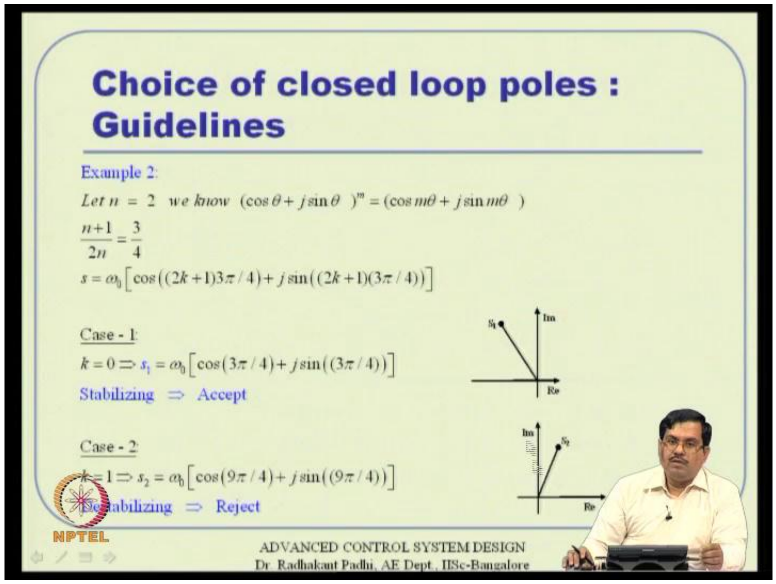
And there is another guideline. How do we select? if you are kind of confuse, how do we select? how do we place the pole locations in a good way?

And one suggestion is you can use whatever polynomial, which is given some form like this s by omega 0, where omega 0 is like natural frequency some design parameter constant. So you select some omega 0 and then s by omega 0 that is a close loop pole locations you kind of distribute, attempt to distribute this formula, where minus 1 we can use de moivre(s) formula sort of thing, we have to write expand in terms of complex exponential.

So this complex exponential and n is nothing but system order basically. And this will churn out a bunch of poles. Because k is 0 1 2 3 it is an infinite series anyway basically. So you have to select only n of them because that is what we were interested in. And obviously you will we are not a fool to select unstable poles. So, we will certainly select only poles those are positive actually.

So, for example if n equal to 1 we have only one pole so, you can use k equal to 1 here and then this will give in like this formula where sin pi is zero. So we are left again cos pi is minus actually minus 1 so, we are left out with S equal to minus omega 0. So, that is just one thing and we are ready with direct application equal to k equal to 1 actually it will give us that.

(Refer Slide Time: 38:39)

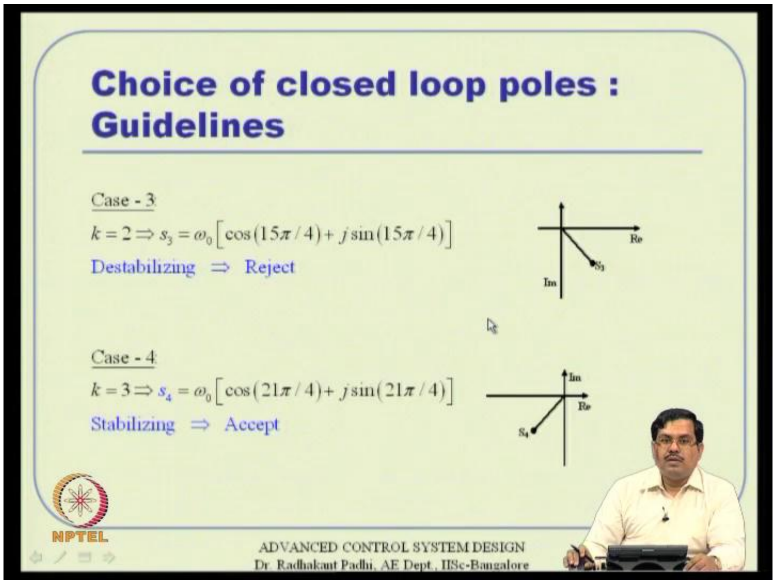


What about little more? Suppose, you have like second order thing actually how do you do that? This is for the first order system, what about second order system? So let n equal to two in that case so, first is like cos theta plus j sin theta to the power m is nothing but, cos m theta plus j sin m theta.

So if you put n equal to 2 here, this is n plus 1 is 2 plus 1 3 divided by 2 into 2 which is 4 so, that is the fraction is 3 by 4. So n plus 1 divided by 2 n is 3 by 4. So you have a formula like this, and this is direct substitution of this formula whatever you have.I mean this complex exponential we expand it in terms of cos and sin basically, that is all we are doing there. So case 1 is like so we put start with k equal to 0 so, in this formula. Whatever, formula we have we start with k equal to 0,and then s 1 turns out to be like that which is somewhere here actually. And obviously this location is left hand side of the I mean this s plane

So, we will buy it we will tell it is right we can accept. So this particular length omega 0 will dictate and that will design I mean we will select a omega 0 and then tell this is good or bad. That is again dependent on several factors like for example, your jumping ratio or our omega I mean what is that? The sigma partial part of it the percentage over showed like that. So these are this will give us s 1 which will we are ready to buy. Now when you put k equal to 1 because k equal to 0 is there, now next you put k equal to 1 this will give a pole location which is actually in the right hand side. Obviously it is destabilizing so we I mean obviously we do not want to take that. So this one we are accepting, this one we are rejecting.

(Refer Slide Time: 40:28)

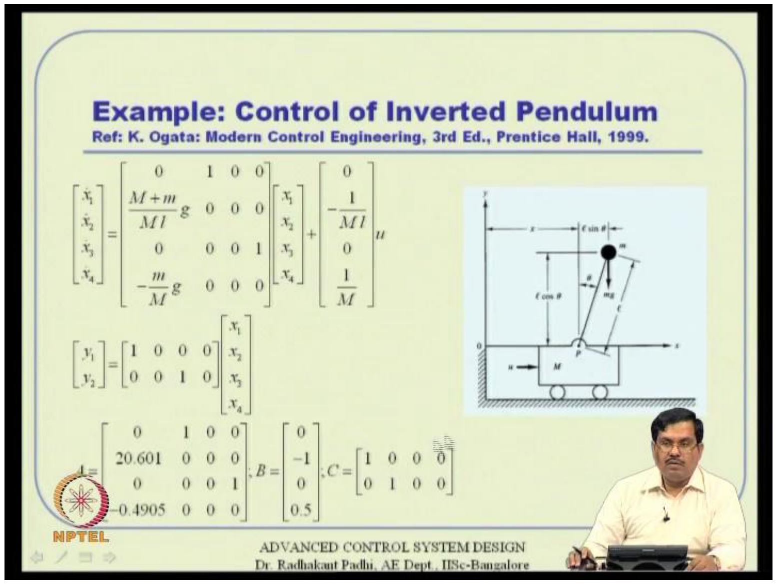


Now the next one k equal to 2 it is again it will pop up in the fourth quadrant and hence it is rejection also. k equal to 3 it is in the third quadrant which is again positive I mean this is stabilizing. So, third anything that is second quadrant and third quadrant we are going to buy. So this is going to be third quadrant so, we accept it.

So if you see this line, this line starts from here and nicely take some sort of a clockwise evaluation. So this attempts to kind of distribute the poles in a fairly kind of good manner basically. This because the arm length remains same it is just a rotation which happens so, it tries to place the poles in a good way rather.

So this is whatever polynomial is all about. If in a higher order case you can also keep on doing that you can change your omega 0 and then again excite this one more time, and things like that. So these are just guidelines by the way, I mean these are not (( )) these are not design procedures, these are simply guidelines to start with actually.

(Refer Slide Time: 41:36)



So let us now take a little time to I mean apply all these to a example this example I taken from Ogata again. So we are talking about an inverted pendulum stoned on a cart sort of thing and then the dynamic equations derivation and all we are not interested that much. What you are telling is that part is done already and this linearization and other things have already been done. So we have a linearization equation that way about the inverted position basically.

So we want to stabilize the valve I mean the inverted pendulum in its vertical equilibrium. Vertical top actually though that is what we want to do so, obviously unstable equilibrium point we want to stabilize that using control system. So if you see this and remember these y 1 and y 2, we are taking x 1 and x 3 only.

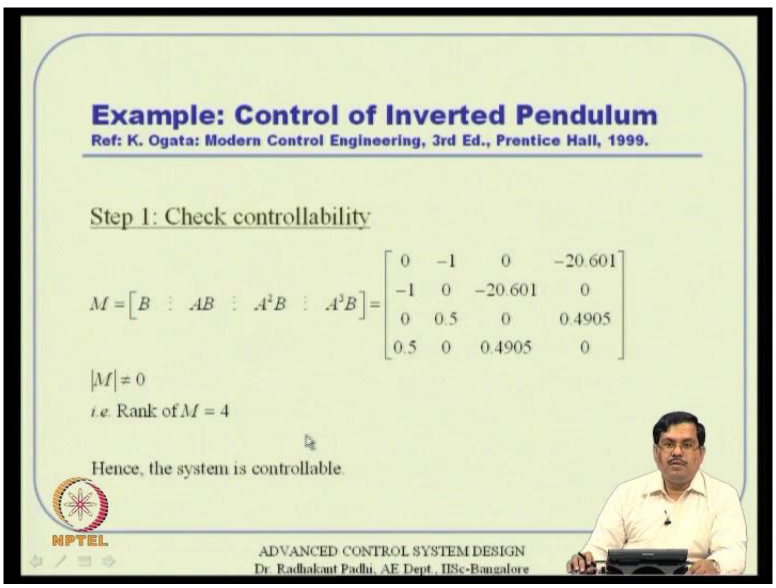
And it is also, this x 1 x 2 x 3 x 4 is just not the pendulum it is also the moment of the cart. So cart is moving and that is where your control input is acting. So by moving the cart forward and backward in a appropriate manner can you stabilize the pendulum on the vertical equilibrium position, that is the problem actually.

So the movement, the control conditions are all related to not only this pendulum swing but, also the cart movement actually. So they are all propelling system that way. So what you are measuring here is x 1 and x 3 which is like position measurements only, x 1 is probably this theta and x 3 is this position movement of the cart actually.

That one we are not that much interested here, because we are designing an observer for this I mean that is that probably next class we will see observer design and all. So this equation is not that relevant as of now. But, here we see this is our system matrix A matrix, this is our B matrix, this is our C matrix B is 0 obviously.

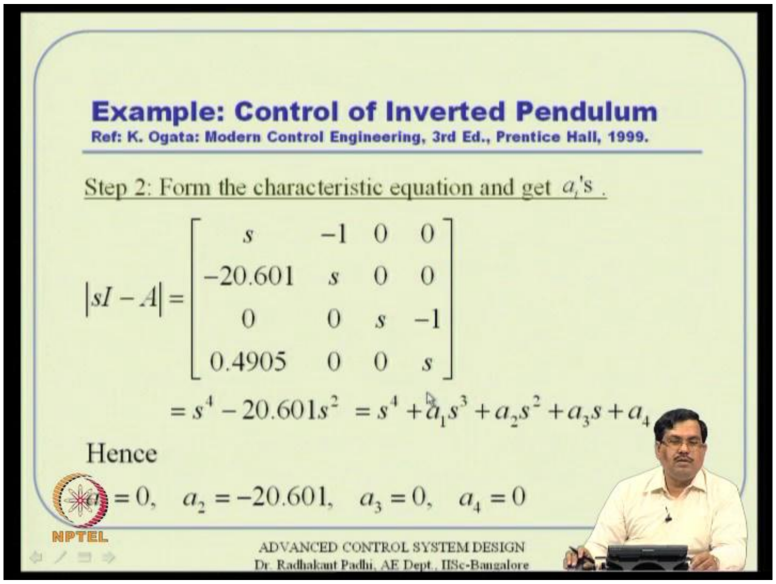
So and then just to work with some numbers we select these numbers appropriately. Obviously so, these numbers A B and C will have numerical values now. So first thing is to check controllability design step one.

(Refer Slide Time: 43:36)



So we form a controllability matrix, it turns out to be like that and then determinant is not equal to 0 and hence rank is 4 is a full rank matrix. And hence the system is controllable and that is the reason why there is hope we can actually attempt to design a control system.

(Refer Slide Time: 43:56)

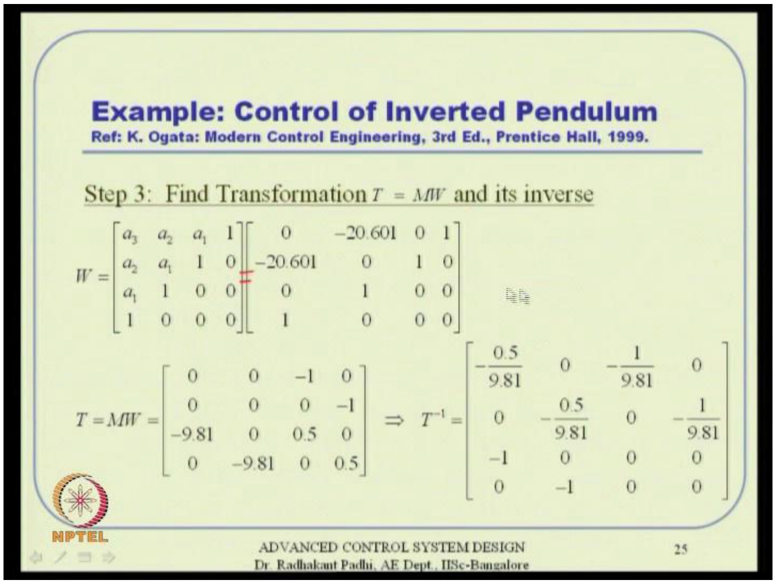


So next go to the step 2. This is what we are talking about, like a practical example of a using our theory what you know. So s I minus A is nothing but, the open loop characteristic polynomial A matrix is like that, A matrix is given like that. So s I minus A turns out to be like that. And s I minus A determinant turns out to be just these two terms s to the power fourth and s s square.

So we want to write it in general s s 4 s q s q all sort of things and then equate the coefficients properly actually a 1 a 2 a 3 a 4. So obviously a 3 and a 4 are 0 0 a 2 a 1 a 1 is nothing but, this cube. Cube is not there in this polynomial. So a 1 is 0 and a 2 is s square s square is nothing but, minus 20.601.

And normally the highest power coefficient is always 1 we do not require that. So we have collected these coefficients a 1 a 2 a 3 a 4 this exercise gives us that. And by the way in general, all this determinant evaluation can easily be done by using symbolic software. If it is too much, I mean high dimensional and all that actually that is not a problem.

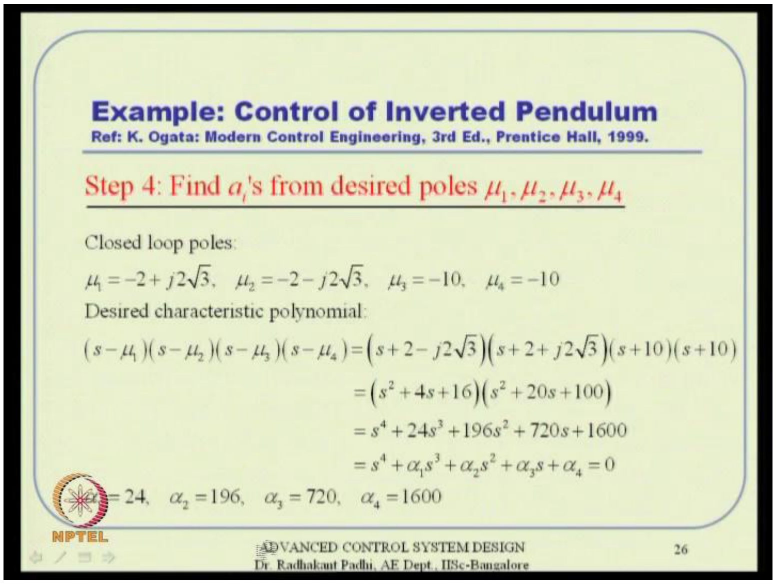
(Refer Slide Time: 45:13)



Now find the transformation that is what we are interested. We are interested in applying method two here the Bass Gura thing. So T is nothing but, M times W where m is already known to us M we have already evaluated that is also dual purpose by the way. So it serves the controllability check as well as we will use it in the T matrix. So M times W where W is like this that is the form. So that is well that is again a small mistake probably, this is equal to that obviously. W is the substitute the coefficient say a 1 a 2 a 3 and you will get that one actually so, that is equal to that actually.

So, now we can see that T equal to M W and we can very easily compute. W is already there, M is also there and so M times W we multiply and we get that. Once T is there T inverse can be computed, that is how it is. And then I can find we are not yet done because you have simply computed T and T inverse yet.

(Refer Slide Time: 46:18)



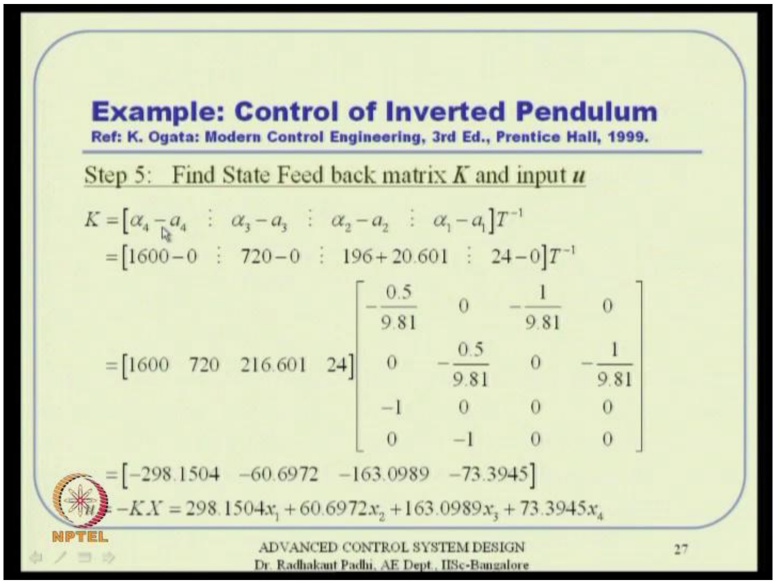
We have to find this; we have to also cater for this desired close loop like a poles and all. And desired close loop poles are given like that. Also remember for real characteristic equations are typically real polynomials. So the closed loop pole locations, the only restriction that to select is they have to be complex conjugate. If they are complex number they have to the conjugate pair it should exist actually.

These two are real number double pole at minus 10 that is, or these two are complex conjugates of each other. So, thus the restriction how you want to select. By the way whatever polynomial naturally gives you that. Because whatever, we saw that second quadrant and third quadrant naturally gives you that actually.

Anyway so it happens like this mu 1 mu 2 mu 3 mu 4 that way. So the desired characteristic polynomial happens to be like this all these four multiplications and you expand all that, and then it will give us that the coefficients alpha 1 alpha 2 alpha 3 alpha 4 that way. Once you expand, the good thing about complex conjugate polynomial is once you multiply them together, the resulting polynomial is real.

So basically these two together will give us this real polynomial, and these two together will give us that polynomial. So again you multiply and then collect these coefficients alpha 1 to alpha 4 so, that is how what we do here.

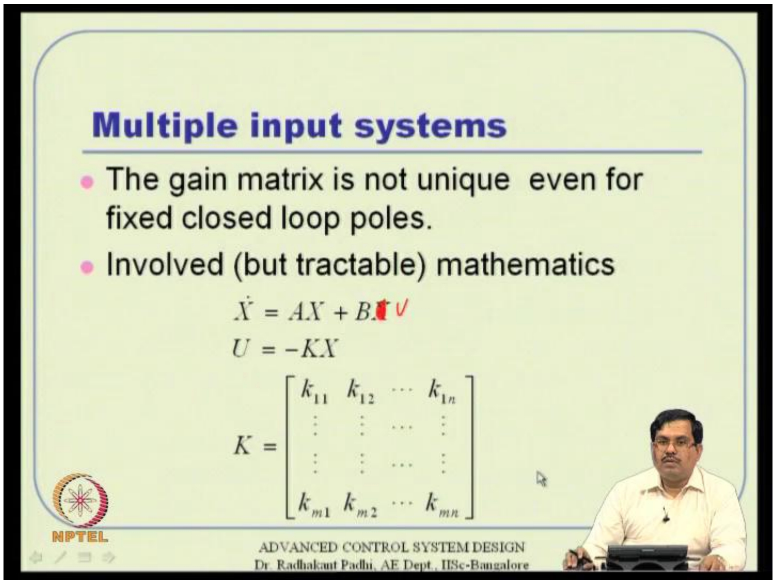
(Refer Slide Time: 47:40)



So, then find the state feedback n matrix which is directly available. These are the this is the row vector times T inverse. T inverse we have already computed, this row is available now. So just compute it and then u equal to minus K X. this is what actually.

So this is all about some examples and all that with single inputs systems. Now, what you do when the system is actually multi input system. So in general U is this, U is R m and where the problem here is this does not admit kind of a unique solution. Full placement technique is good only for single input systems in a very good way basically. But there are techniques, tricks and techniques to handle this issue to a limited extent.

(Refer Slide Time: 48:27)



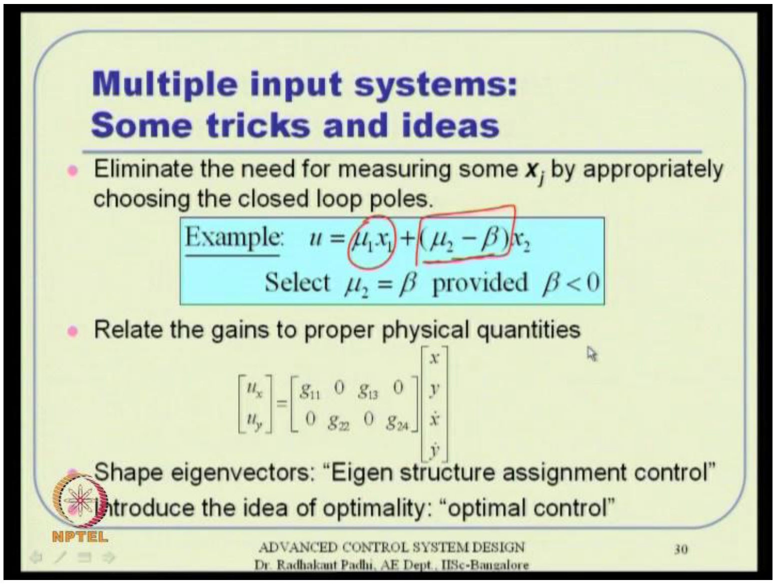
So what I told here that is the same thing written here. The gain matrix is not unique even for fixed closed loop poles actually. Say essentially the gain matrix is not unique that is all actually.

Then, the matrix where the mathematics becomes kind of involved but, it is traceable it is not becomes kind of untraceable mathematics, I mean they are linear systems. That is why they are kind of popular actually.

So, it is traceable mathematics but, it becomes little more involvement and hence it is not that good. Oh again there is a small mistake (( )) this is certainly U. So X dot equal to A X plus B U and U equal to minus K X. So gain matrix is like that in general.

Now the gain matrix is not a row matrix, I mean not a row vector but, it is a matrix in general. In that sense what you want to do?

(Refer Slide Time: 49:35)



So one idea is you can eliminate the need of measuring certain x certain states by purposefully selecting some sort of a combination. Appropriately, choosing the closed loop poles. For example, if your control formula in a symbolic sense is like this where, mu 1 and mu 2 are pole locations. Then purposefully I will select 1 mu 2 equal to beta. So this term equals to 0 and hence I do not need to use x 2 I do not I can eliminate sensor.

Now this is all subject to the condition that beta is actually good. Beta is in the negative side and all that negative side of the plane basically. But, if it happens that way some polynomial, some coefficient if this one need not to be mu 1 it can be anything actually. But just a function of x 1 and this one, right term this one is to be certainly mu 2 minus beta. This need not be mu 1 actually it can be anything that is okay.

But this is of importance; this has to be like that. Then certainly I can select mu 2 equal to beta provided beta is negative, that is the only condition. Then the beauty of it is I do not need to measure x 2 anymore. So I can eliminate sensor. But again you have to see whether that is the design is good only with x 1 information, this is this will excite this output feedback controls sort of ideas basically.

Then the second thing is can you relate the gain matrix to proper physical quantities? That means suppose you have u x and u y. Now u x which is like let us say the position sort of ideas or u x is related to object x primarily, u y is related to object y primarily. They are coupled though the x and y systems are coupled is obviously but, you want to preserve that independent feedback sort of ideas.

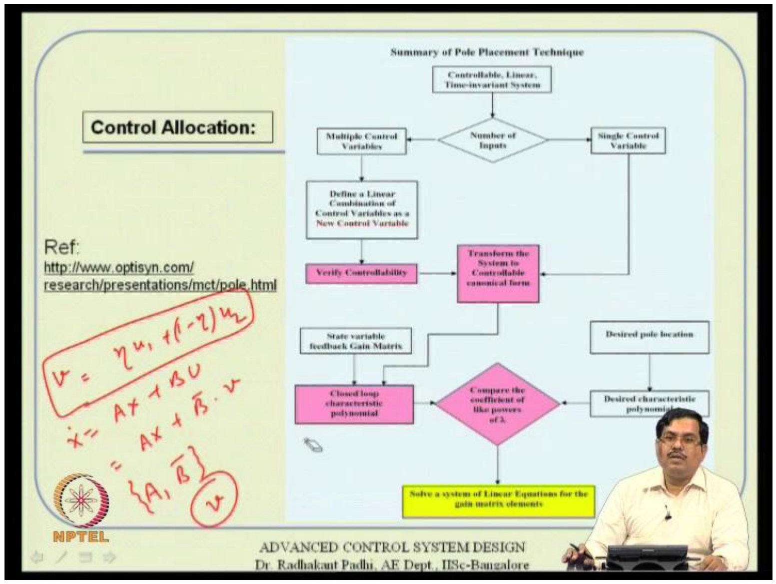
So the plant one will act on the measurements from plant one, and plot b controller will act on measurements from plot b itself. The information exchange from plot a and plot b, even though they are weakly coupled so, need not be considered actually. Sometimes this power plant separating a different location, power system control and all in electrical engineering, these kind of ideas are useful actually.

So then you talk about these elements are purposefully, I mean enforced to be 0. So your entire freedom of 8 entries have reduced into 4 entries actually. So these are some of the ideas you can involve for even better simplicity. But these are more or less heuristics actually, they are more or less it is a kind of intuitive .But it is not mathematically rigorous actually.

So, if you want to have mathematical rigorous thing then there are techniques available which is one of that is Eigen structure assignment control. Where we are not interested only in Eigen value placement but, we will also interested in Eigen vector directions. They have certain meanings and all that. So we are not going to discuss that particular thing here. But another alternative thing is we can introduce the idea of optimality, which will lead us to optimal control design for linear systems. Especially stabilizing controller design will fall in the frame work of linear quadratic regulator that we are going to do here.

So that is and you can also visualize that optimal controller, the L Q R controller is kind of a Eigen structure assignment controller. And ultimately it will give us a formula for which it is valid from multi input system and all that. Rightly in general it is valid so, it implicitly it is doing some sort of a Eigen vectors shaping. But, you are not enforcing that to begin with, we will enforce it indirectly sort of thing.

(Refer Slide Time: 53:22)



So these are some other tricks and ideas that it can follow. Another idea that is I have taken from the website which say a very good company of course, dealing with control system estimation and all that. They have put that idea in a very neat way. So, this is like you may not be able to read here. So, I will let me explain this concept in a little detail way sort of thing. So, what you are telling here is like you have this number of inputs. So, we have this entire pole placement technique given in this diagram sort of thing. So, either you have a single control variable or you have a multiple control variable. Single control variable nothing too much I mean to worry about actually things are in order.

When you have a multiple control variable all that we are doing is, you can define some sort of a linear combination of those variables. Suppose you have let us say u 1 and u 2 then you talk about the vector v which is like partly u 1. Let us say mu times u, mu is not a good variable probably because that is already taken. Let us say eta times u 1 plus 1 minus eta times u 2. Where, eta is a number between 0 and 1 actually. That means you are formulating some sort of a convex combination of u 1 and u 2. You are actually kind of distributing the load in a percentage sense. Let us say one eta is the 0.3 then it is 30 percent u 1 and 70 percent u 2 like that actually.

So you do that and then because you have reducing this way, you are not sure whatever B matrix pops of with v. So, ultimately you started with X dot equal to A X plus B U. So, this will we are interested in writing A X plus some sort of a B bar times v. So, the system was certainly controllable with respect to A B pair. We are not sure whether it is controllable with respect to A and B bar pair. So, that we need to kind of do that actually. So you want to verify whether A and B bar this pair is controllable or not.

Now once this satisfies these, then you can transfer the system controllable (( )), I mean what essentially tells is once this condition is satisfied you design a v actually. Once you design a v then again you distribute that in terms of u 1 and u 2. That means in this particular case eta times kind of I mean you can this... Once you design this v you know what you are doing actually, that means this v consists of let us say 30 percent u 1 70 percent u 2.

So you can kind of try to redistribute back and tell this v I will kind of partition that. And tell 30 percent load will be taken by u 1 and 70 percent will be taken by u 2 I mean that is the idea there. Not a very neat approach but, certainly it does work in many cases especially for linear systems this because of the super position principle comes to your rescue. So, that way it becomes helpful in this particular thing.

But, this method need not necessarily guarantee to work. Suppose, you select a different eta it may be controllable suddenly, for a different eta it may not be controllable like that. So you have to do some iterative design for making it work actually this particular approach. But then I mean for single input systems it is certainly a definite good tool though what we talk, this pole placement design.

So, the in summary what you are telling here is, in this particular class we talked about pole placement control design for single input system in a good way. Multiple input systems you can do you can do a job but, it is not very straight forward exercise, because of non uniqueness of the solution. But forgetting that effect there are ways to handle that, there are I mean may not be very mathematically rigorous way but, there are ways to handle that. But, if you want to be mathematically rigorous then you have to use this either Eigen structure assignment or some sort of optimal control L Q R thing.

But other than that, we discussed about like method 1 2 3 how do we make use of these ideas of civilizing a I mean stabilization and all that. To come up with a controller design, and primarily we have focused on state feedback controller of the form of U equal to minus K X.

So, entire thing was boiled on to, how do you design a K matrix or K properly basically. So for that we discussed about three methods. And all these three methods, we also demonstrated I mean one of the methods we demonstrated using this inverted pendulum and then we will continue further with our discussion on like how to make use of that, these concepts that we discussed here for observer design in next class. With that I will stop probably, thank you.