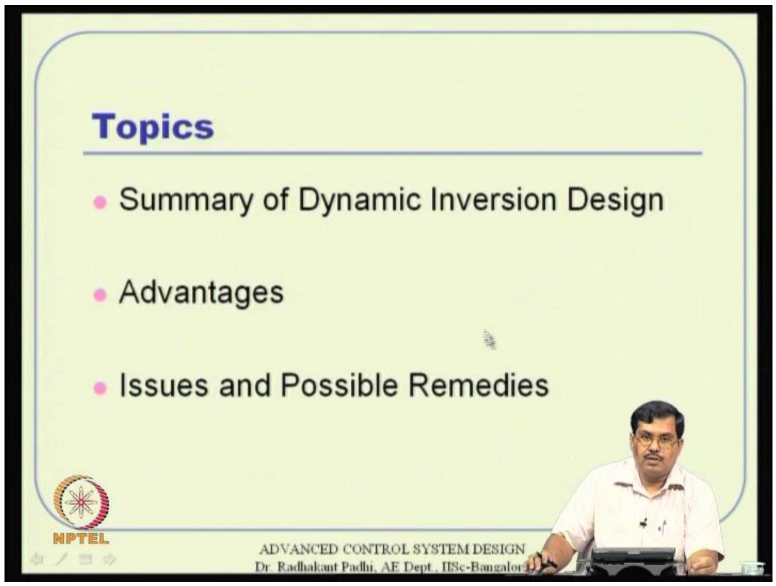
Advanced Control System Design
Prof. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Science, Bangalore

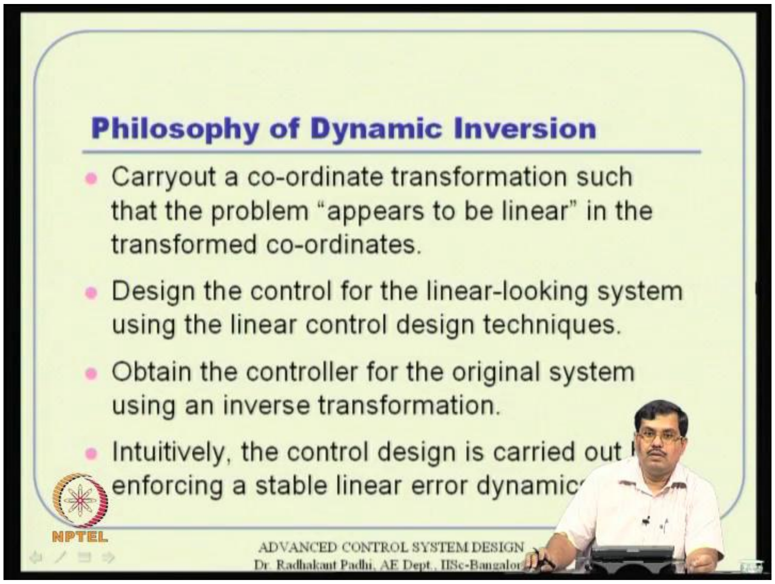
Lecture No. # 35
Dynamic Inversion – II

(Refer Slide Time: 00:35)



Hello everyone, we will we were seen some dynamic inversion concept last class. We will continue for discussing further on the topic, and then try to see some of this difficulties and possible remedies, and things like that. But before that will I quick review of what we discussed last class, and that is how it the topics are organise something like that I will quickly review some sort of a summary of D I design or dynamic inversion design. We will have again list out the advantages associated with that, and because of these advantages we do not mind for this these issues, and rather hunt out for possible remedies instead of we will actually. So, that the that is the that is the way we will discuss in this class.

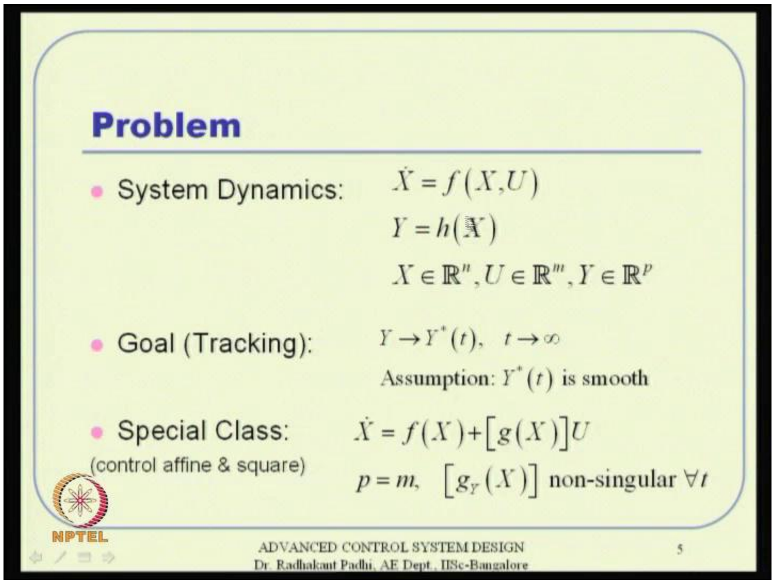
(Refer Slide Time: 01:03)



So, quick review of what we have discussed already last class, this the summary part of it. So, philosophy of dynamic inversion is is like this, we carry out some sort of a coordinate transformation, such that the non-linear problem appears to be linear in the transformed coordinates. It is not linearization per se, but we just do some sort of a non-linear function mapping business, so the so that in that new coordinates and problem appears to be linear. Then once the problem starts appearing to be linear, then we know linear design control linear control design technique. So, design the control for the linear looking system again this not linear system what we just appears to be linear that is how actually.

So, using that linear looking system we design a controller using the linear control design techniques and once you have done in that coordinate frame we want to see what is the controller in the original system. So, we do some sort of a inverse transformation actually. So, obiviously this design philosophy holds good if and only if the inverse transformation is possible actually. When there is there is a unique inverse transformation that is also this basically. So, anyway so this so intuitively what is the inverse transformation is the control design is carried out by enforcing some sort of a stable linear error dynamics in the transform coordinate frame that is the is the bottom line actually.

(Refer Slide Time: 02:28)

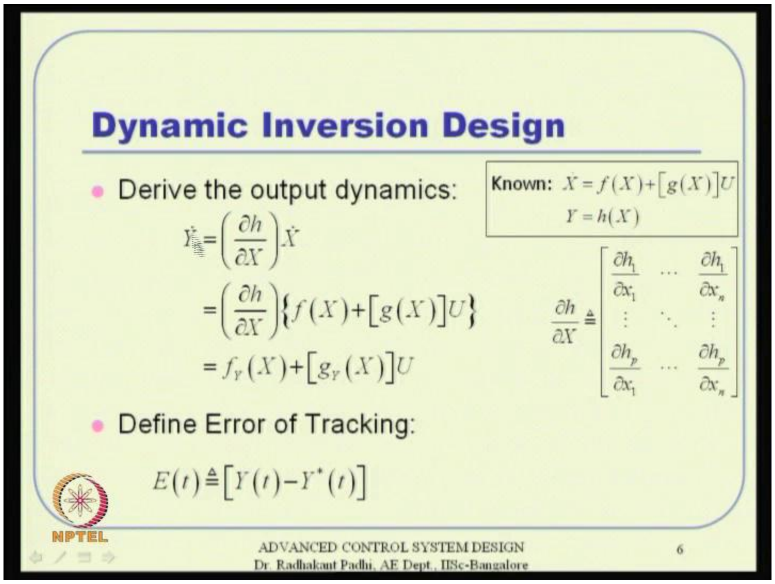


Now, how do we do the mathematics part of it something like that we discussed last class in general the non-linear systems can be described systems state equations like this x dot is f of x u y is h of x, where x u and y can be of different dimensions actually, x is state, u is control and y is the performance of output actually. Let say which also non-linear function of states. And goal objective we want to essay of that this performance outputs, so tracks some we get outputs that is y should go to y star rise to u goes to infinity actually. So, when the assumption here is y star is a smooth vector actually; that means, y is continuous and y dot is also continuous at least.

And then special class I mean this itself is a little convenient I mean kind of complicated system dynamics and all that. So, we do not I mean it is possible to deal with of concentration a lot of attention for designing controllers for this class of generic non-linear systems. But in this particular lecture will confine ourselves to class of problems where, we I mean we take this control affine system where control variable appears to be linear, and we mentioned to that p is equal to one; that means, u I mean the dimension of u and y are same; that means, an input output same process of the system is a square system basically.

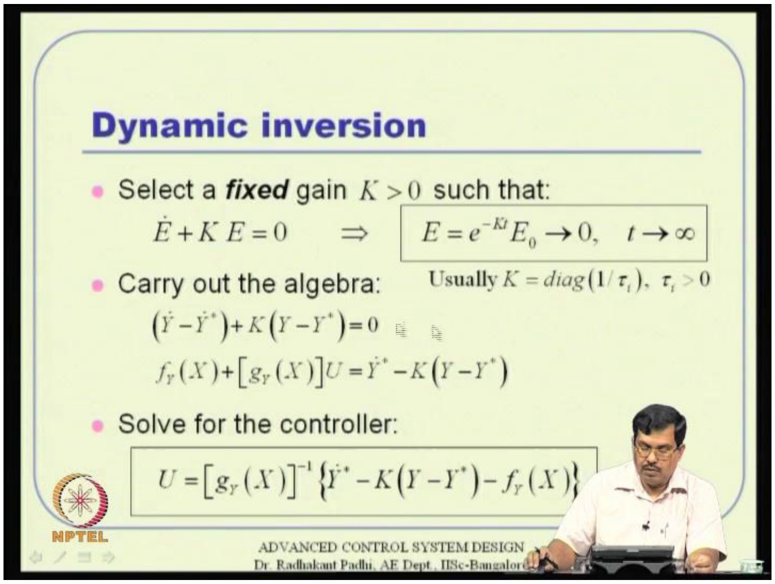
And this g y of x which will define in the next slide is non-singular for all time, because we need as matrix inversion for this matrix, g and g y are different of course, their relative, but their different. So, what is g y we will see that in the next slide anyway. So, this is our objective y should go to y star it tracking objective and we concentrate ourselves to control affine square and for which g of x u non-singular for all time.

(Refer Slide Time: 04:16)



So, how do you do that? So, for receiving that we will first what we did is we derive this y dot and because y is h of x, so y dot is del h by del x into x dot and x dot is that so, substitute of that and then define what is f y and g y. So, f y is del h by del x into f and g y is del h by del x into g, so that is g y actually. And del h by del x you define in this matrix actually, so that is Jacobian matrix set of thing. So, this matrix g y of x needs to be non-singular, basically we will see that in next slide why it is need to be like that.

(Refer Slide Time: 05:00)

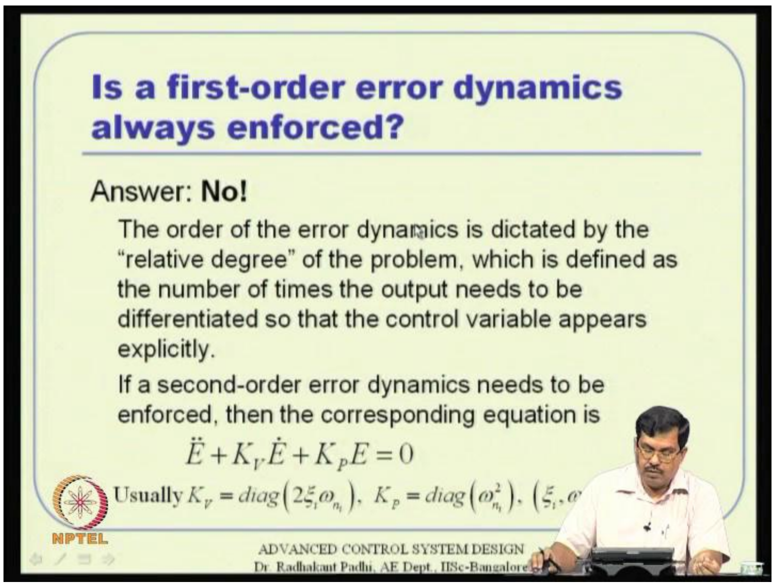


So, we our objective is that y should go to y star so, we define in error vector which is y minus y star and then we must say that e goes to zero asymptotically. And how do you do that? We do that by selecting sort of a fixed gain k is a positive definite matrix such that this sort of dynamics is unforced, and we know that once this sort of dynamics unforced, the solution is like this e equal to exponential minus k t into e 0. And because k is a postulate by matrix by selection it makes to go to zero actually e goes to 0 infinity. Remember that this first of all this t goes to infinity is mathematical notion and for practical program, this t goes to I mean this infinity and all that is large dictated by the settling term that we select actually.

So, if the settling term is small that means, the traffic reference is small, and think like that and also second point is this fixed gain matrix need not be constant in some applications you can have limited gain k reeling as well that is that is also inversion. Well we really need that most of the time it is happens that we will find out need that actually. Than this is what the objective was, so what we do to enforce these are dynamic we substitute this equal to y minus y star so that is y dot minus y star dot e y y minus y star so enforce these are dynamic, and then y dot is something that we derived here. So, y dot we substitute and then try to solve for a controller u. So, that is how we were to control thing control I means, so it is a non-linear function for control that is what we obtain with a fixed gain k.

So, this control; this controller is I mean this formula what you see here is there is once you start using these essentially satisfy this equation hence this equation and hence that one; that means, this e dot plus k e is continuously enforce actually. So, that we have this control design works we discussed many details on that last class actually. And usually the way to select this gain matrix for the first set of dynamics is like this, you select k equal to diagonal 1 by tau where is if k is diagonal matrix then these are diagonal elements, and then where certainly positive for positive tau and then it needs to be postulate that actually. So, this is one of the how do you design that.

(Refer Slide Time: 07:17)

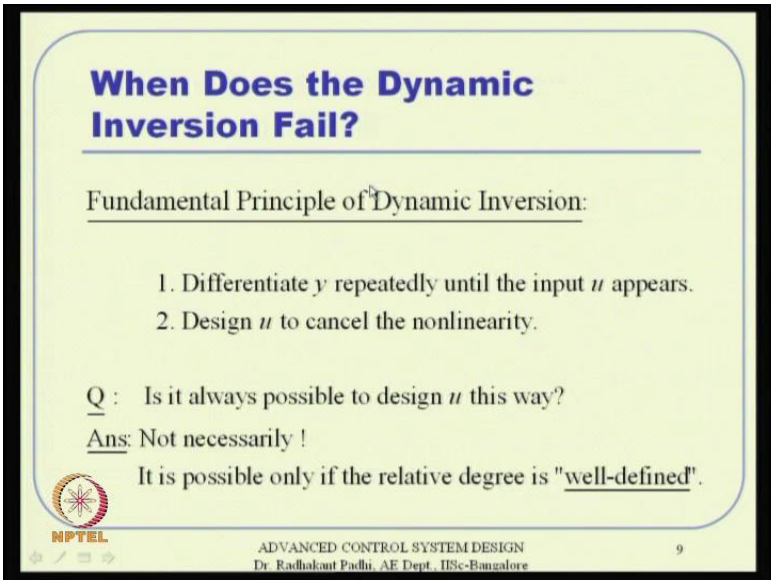


Now, the question here is do you always enforce this first order error dynamics certainly not. So, this is dictated by what is called relative degree and the relative degree is defined as the number of times the output needs to be differentiated so that the control variable appears explicitly. That means, each of the output vector I mean each of the component of the output vector you can keep on a taking derivatives. And In each of the component have two vector we will at some point of I mean derivative I will I mean this control variable start appearing, and that becomes relative degree for that particular component of the error vector I mean this output vector. And that is called the relative degree and the total relative degree just addition of those individual relative degrees by the way.

So, what you are assuming here is each of the components of this e satisfies the same order if it happens, then you can (( )) represent this equation that ways. So, instead of first order error dynamics, you can you can try to I mean enforce this second order error dynamics in state, where k v is can be selected that way and k k p can be selected that way, so that each of the individual component wise it becomes a standard second order error dynamic in the form of e double dot plus 2 zeta omega n, and the e dot plus omega m square e equal to 0. That is the very standard error dynamics or linear system, that for which we know the solution behaviour and think like that.

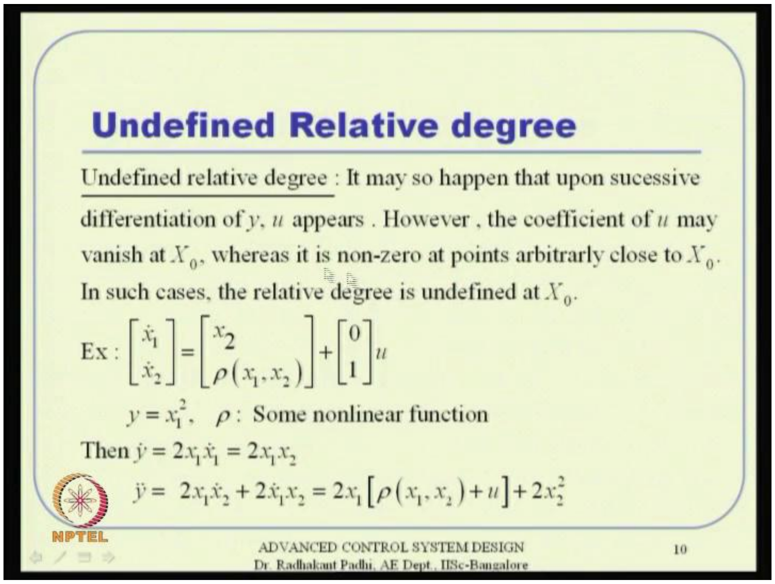
So, obviously, this k p and k v are can be selected this way diagonal I mean diagonal matrix manner, and this zeta and omega can always be selected from performance specifications like again like settling term performance over state like that for this individual errors and actually.

(Refer Slide Time: 08:58)



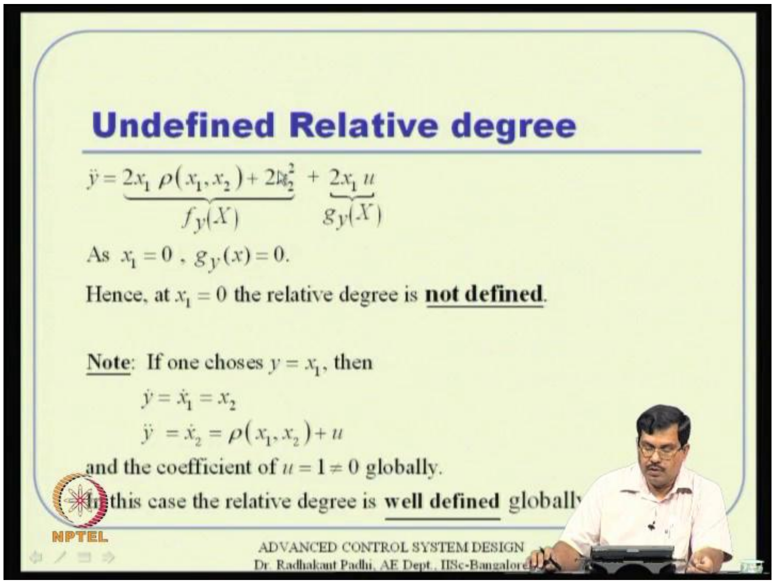
Now, we also discuss when the dynamic inversion does not work when does it fail, because for understanding that we also discuss that this the fundamental principle of dynamic inversion, we used to differentiate y repeatedly until u appears in the single input and single output sense. And then design u to cancel the nonlinearity all that we do and then simultaneously it enforces some sort of stable error dynamics actually. So, is it always possible to design u this way the answer turns out to be not necessarily true, and it is possible only if the relative degree is well defined actually.

(Refer Slide Time: 09:36)



And to understand what is well defined we need to see this what is undefined relative degree, and this statement may turns out to be like this it may so happen that upon successive differentiation of y, u does appear u appear. However, the coefficient of u may vanish at x whereas, it arbitrarily close to x 0, let us say it does not vanish, it only vanishes at selected point x 0, but very close to x 0 does not vanish actually. So, however, we just understand this so, also small example this x 1 dot is x 2 dot is something like these this if you if you take it that way.

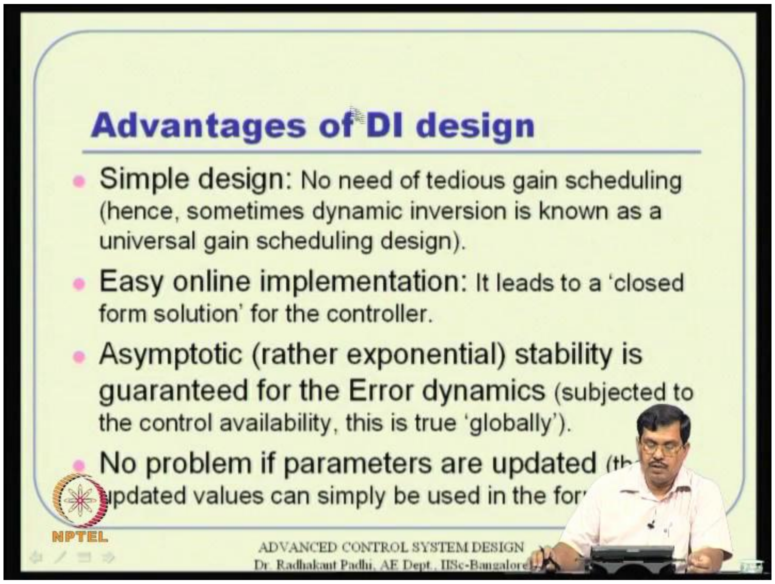
(Refer Slide Time: 10:21)



Then, y let say x in x 1 square then y dot like this y double dot like that so, what happens? If you can rearrange this term and it turns out the f y like this and g y like that. So, as x 1 is 0, then 0 f x turns out to be 0. So, the entire x 1 0 like that the entire x 2 f 3 this term x is actually 0, but the another term little bit away from that 2 x 1 u by g y actually. That means, at x 1 equal to 0 the relative degree is not defined obliviously. So, what do I do circumstances this problem probably you can go back and think let me select a different way, y equal to x square x 1 square is not good. So, let me select x 1 then our reference is if you do the double relatives and all that with the coefficient of u is 1 which is non-zero throughout actually globally.

So, if you select y equal to x 1 for the same problem the relative degree is well defined, but if you select x 1 square it is not defined actually. So, this ho you can think that the relative degree is defined well defined then we can talk about doing dynamic inversion and there is a problem that then I may not be able to do that actually.

(Refer Slide Time: 11:25)



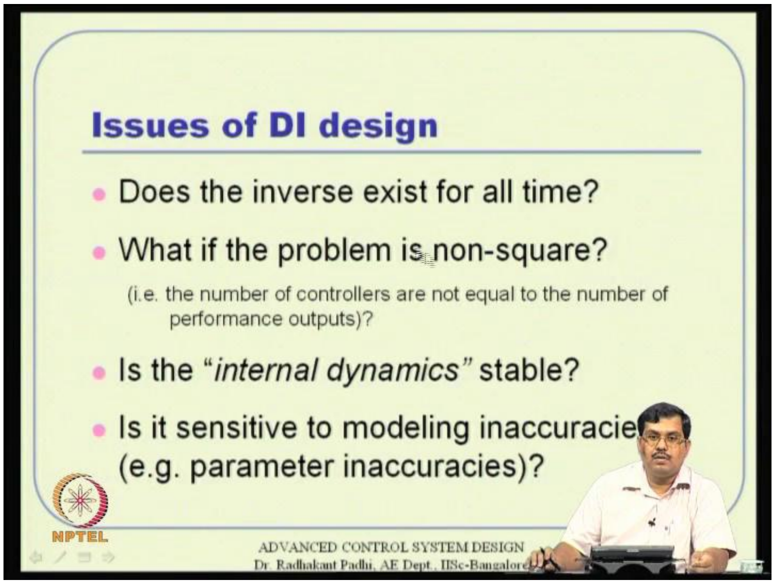
So, the advantages what we summarized last class it is turns out to be very simple design there is no need of tedious gain scheduling in general, sometimes you may knew where which particular application you may knew it a small amount of scheduling. For example, let me summarize something sometimes people do limit it is scheduling within based on some sort of manoeuvrability actually manoeuvre capability. We calculate how much we can manipulate and find the type and make the again as the function of that, but several I man only when it is absolutely necessary otherwise it turns out that gain scheduling linear not required actually.

So, it is also sometimes called as universal gain scheduling design, because just by just by selecting one gain you have taking care of the entire domain of operation actually. It is easy for online implementation; because it is close from solution after all there is no iterative solution involved here you just keep on evaluating a formula once you will know the information about the state. Then always these are asymptotic rather exponential stability is guaranteed for the error dynamics as far as output tracking error is concerned, that has to go to zero. I mean rather asymptotic or exponential way and instead it will that is to globally also, because what you are doing here is linear system error dynamics and the that is nothing to with local stability.

So, global exponential stability for error dynamics is guaranteed as far as output tracking is concerned subjected to obviously, control availability if your control as rated that formula is no more valid anyway. And we also discussed last class there this no problem if the parameters are updated that means, if your design changes are goes through modifications and some some values of the parameter are same, if you have updated for purely linear design for note by the standard classical gain for the link then you have to go by can start from the beginning. Because your potentially your updates are lower or same actually, but here it does not happen that you have all that you have is a formula and the formula needs to evaluated with respect to the new set of parameters basically.

So, you may still need to little bit in that, but normally this does not requires too much of exercise. So, that scalability design or that what you called rapid prototyping and think like that it is very easy actually that means, once you the very comfortable with this kind of design then you can go from I mean you can very quickly synthesize your controller for the entire system for the entire flight envelope actually. So, these are the advantages of the dynamic inversion design and hence we looking at the possible issues I am trying to hunt out for remedies actually. So, those are the things that we will discuss next in this class up to now, we learned summary of what we discussed last class in detail.

(Refer Slide Time: 14:20)



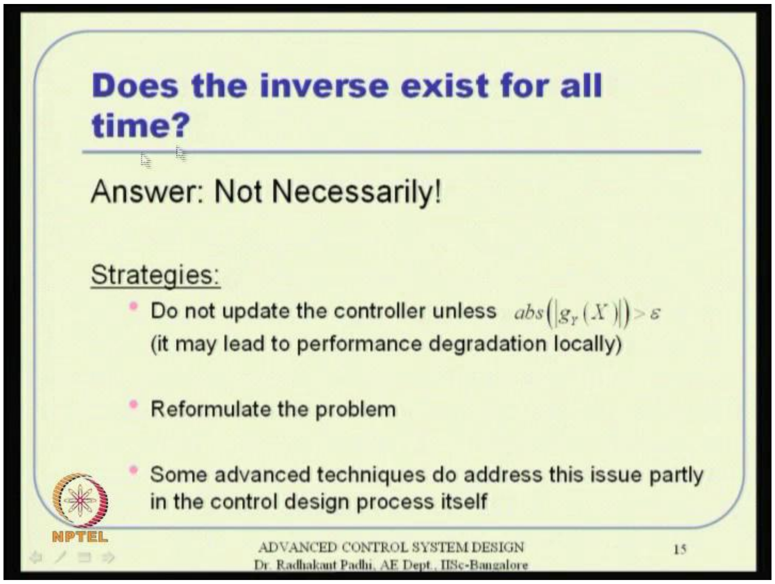
So, you see some remedies in D I design so, let us talk about that this two summary is there are potentially, where we can think about for five issues and in first issue that comes to mind is we does the inverse exist for all time. We always have if you go back to this final formula there is matrix inversion and evaluate the matrix can matrix may exist the inverse may not exist actually. And what we are demanding here that this matrix given one singular for all time, because x is time varying function I mean x is varies with time so, g y of x is time varying matrix actually. So, will it remain non-singular for all time that is the first issue.

Now, what if the problem is non-square? So, conveniently assume before that p equal to 1 that the number of outputs are equal to number of inputs that is why g y of x turns out to be square matrix, if does not become square we may not even talk about inverse actually. So, what you will do in the in those situations actually. Then there are two biggest issue in D I design in general even if these two are taken care where the third issue is what about this internal dynamics that means, what we have done is I can see that y goes to y star in other words y minus y star arrive that goes to zero that is all right actually, but also remember that dimension of y is not lesser than dimension of x normally.

So, I mean this does not make sure that everything remains very good in the output space that means, the in the sub space of the in the problem basically. So, what about those n minus p dimensions actually? So, the dynamics of the system do exist in those directions I means those dimensions as well so, what about what about that dynamic what is something called internal dynamics. Because that is not coming into picture of this y dot expression output dynamics actually, we will see that that is internal dynamics. And another big issue turns out that it is I mean the any design that you propose any control design should have good amount of robustness with respect to modelling errors, because no modelling can be very perfect we all know about there will be modelling inaccuracies and think like that.

And it unfortunately it turns out that this particular design is sensitive to modelling inaccuracies, that means, if you have parameter inaccuracies or if some terms that we missed out in modelling and think like that, then it is typically not very robust actually. So, these are the reasons why I mean this particularly this is probably the most important reason why we for many years this reason was not very popular actually, but it will do you know how to handle this anyway. So, we see this issues one by one and try to see some sort of remedies and all that philosophically some something’s will tell precisely something’s we leave it for the discussion in subsequent classes actually.

(Refer Slide Time: 17:14)



First thing you does the inverse exist for all time as we discussed not necessarily. So, this g y of x inverse that you have taken care, I means that we have used need not admit an inverse for all time. Now, there are two issues one thing is it may not admit I mean it can become singular for a longer time or it can singular for a for a smaller time inter return times actually. So, if it represents to be longer time then probably there is something done formulation itself we need to go back and see whether that is what we have actually formulated is right and wrong actually. And most likely it will turn out that the entire formulations somehow it is it is wrong so, we need to do something else for that.

For example, if the control ability is not there in the plant then it will turn out pop up that this kind of issues will be quite large actually. Whenever if your controller is less effective for the output objective whatever objective is there then it may not it might be struggle to do that and hence it will not able to do that actually that way. So, I mean for example, if you want to airline aircraft we know by limited coupling we can also use (( )) aircraft, but primary controls are process are airline actually. So, for some reason if you airlines are from properly the one thing only for very small amount of manipulation you can do using actually. Otherwise, if you really want to real it very large quantity what will not necessarily and then it will this matrix and Inverse non-singular I mean this symmetric will go singular and think like that. So, that is this that is why fundamental problem there.

Now, what if that matrix inversion represents to be I mean singular for small intervals of time and for that the engineering solution tells that we do not have to update your controller, unless otherwise this condition holds good actually. Because that means, I take the determinant of geometrics g y of x I keep on taking interval point of time taken absolute value of that and if it is beyond certain I know for c that the matrix is non-singular. And then only I will update the control otherwise I will not. So obviously, when I am not updating the control I am just learning previous control or something it may lead to performance degradation locally. So, will not able to track your commanded variable and think like that, but that will appear only locally.

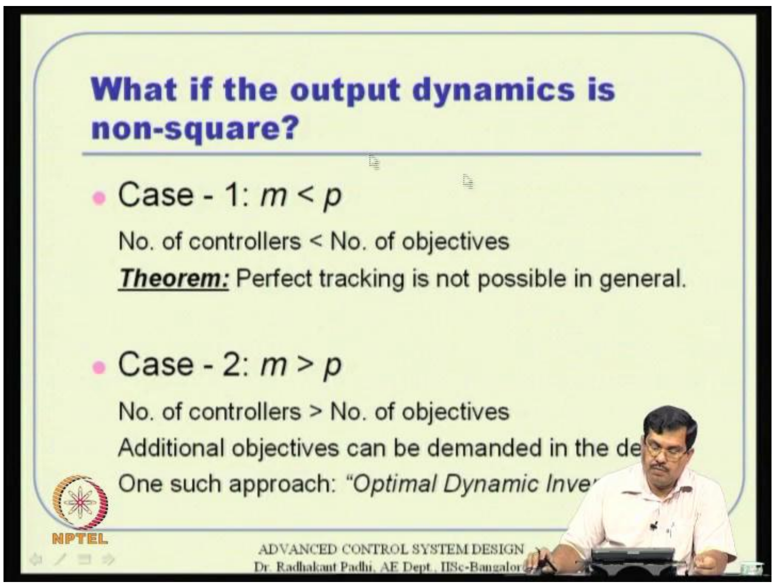
And then after you are out of that, that means, after you are out of this similarity domain then again this will this condition will hold good and hence you can still be able to update your controller actually. So, what my recommendation is like whether it happens or not keep it as an integral part of your programming actually, whenever you want to implement a controller just before updating the controller have this check actually. Only when this your check is passed through then only you update your controller otherwise hold the previous control. I mean that will be true after your first times to, but first times to usually the first time to itself singularity then you have to really go back and think something else actually.

So, absolutely we always have to reformulate the problem that means, if I have something like that I can lead to lesser I mean kind of not so, highly demanding anywhere. That means, I can do a nice terming instead of a soft terming out of thing these are like reformulating the problem actually or I can even think of some sort of adding some control surface in addition to what I have I mean that is design sense fundamentally and thinks like that. So, that they are like subject to problems, subjected to capabilities of the vehicle and capability of the system like that actually. But apart from the this these engineering issues and philosophical issues and think like that there also some advance technique to address this issue partly in the control design process itself, that means, we do not when I have been talk to more in that in this class, but there are there are techniques available which will prevents this from locally.

So, here we are just assuming we are just putting some sort of condition and then working for the best. But this techniques what we are talking here, we will purposefully tried to avoid this being I mean this similarity issue coming into your system actually. So, they will keep on monitoring some sort of determinant rate and think like that philosophically and then they will try to modify I am telling that, if your approaching determinant approaches to zero then probably that not the right way to go right directions to go basically.

So, I mean, but more probably you can recall some literature or something in try to see that actually, but fundamentally there is no controllability in the plant locally you cannot do anything actually. So, that is an issue that we should be aware of actually anyway so, coming back my suggestion is immovably have this check in your implementation of the dynamic inversion controller, before updating your control actually.

(Refer Slide Time: 22:37)



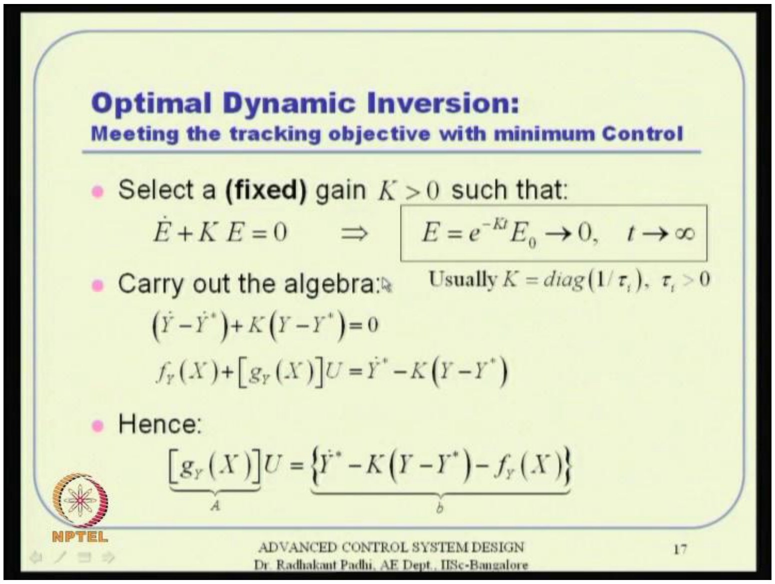
Next issue what if the output data means is non-square, that means, dimension of y is not equal to dimension of p in that sense g y of x is non-square actually. So, obliviously we cannot talk about inversion of that matrix. Now, there are two cases may happen, that means, first case is m is less than p and the second case is m is greater than p. So, m is less than p means number of controllers are less than number of objectives, number of independent controllers are less than number of independent objectives here actually.

So, in that situation there is there is a great theorem is tells us that perfect tracking is not possible in general, that means, only for limited objective; that means, like regulated design and things like that you can do some sort of a manipulation of the variables you select x 2 instead of x 1. So, that you can attain your objective and think like that one example we saw that in the last class also that how do I handle this for a second order system and all that actually. But in general for arbitrarily signal tracking the theorem tells that is not possible actually, you cannot demand an arbitrary signal and then tell me to track it unless and otherwise, I have at least the some number of independent controllers as the number of objective that you want me to track actually.

So, that is I mean that is a theorem that will we will accept it without proof actually. Anyway, so the case two is really interesting it tells you what if the reverse case. So, I by that, because I have not given enough capability here when m is less than p, but suppose I gave you more capability I mean greater than p then what? So, obviously, the large more controller capability then the demand actually so, certainly I mean intuition tells us that it is it is possible the tracking objective has to be met actually.

But in addition to making that tracking objective we can always put additional objectives, because you have larger number of controllers where more number of controllers anyway. So, we can put additional objectives can be demanded in the in the design in general actually, in once it is approach is what we call as optimal dynamic inversion. There are various ways of interpreting this problem obviously, and I will take you through one development that we in our lab we were using that we have propose this and then trying to use it actually anyway.

(Refer Slide Time: 25:00)

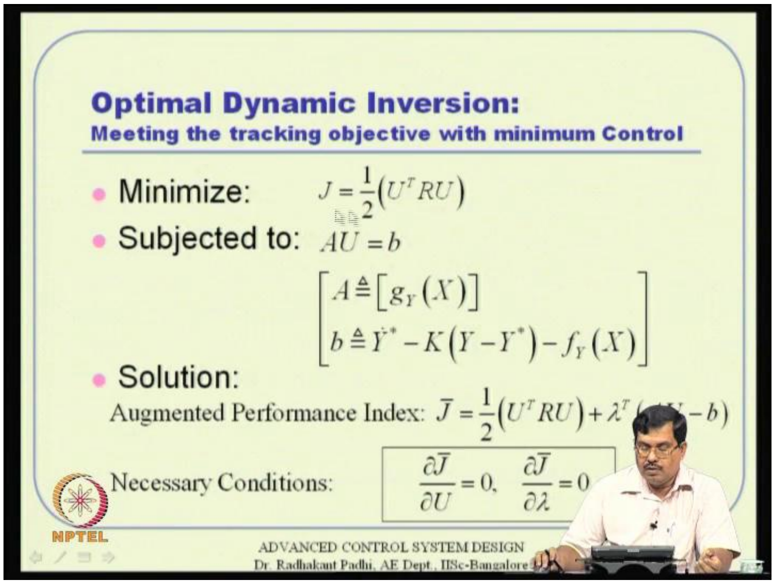


So, this is this is something like m is greater than p so, what let us let us go back to this entire design process and try to see where we are invoking this issue actually. So, if you go back and see this slide that is we are started with and that we have to enforce this e dot plus k equal to 0 and theory of the algebra. So, we substituted e equal to y minus y star e dot y dot minus y star dot y dot is like that we derived before. And then we will tell this from here to here we cannot do that, because this I mean here to here is, but the step that we discussed last time, but we cannot solve the control directly taking g y of x and inverse, because g y of x is no more square matrix actually.

But I can always interpret this equation it becomes to end equation. So, I have to find a solution such a way, but this equation is satisfied actually for whatever reason. So, this equation I will rewrite that way and I will define this a and b for notational simplicity. So, that this time varying matrix a this time varying vector b. So, the equation turns out to be something like a u equal to b let I am say a equal to b just remember that a and b are no more constraint matrix and vector they are time varying matrices and vectors actually. But this equation is still valid actually, because just that from this equation we will not be able to solve for u directly, but this equation itself is valid so, I will interpret these as a constraint equation.

So, if they if I somehow find entire solution satisfying this constraint equation that means, this equation again satisfied this equation is still valid and I will have the tracking objective, but anyway. Now, how do how do find some sort of solution for this u remember this under constraint problem, where number of controllers are larger than number of outputs that means, number of free variable that you were talking about larger than number of constraint actually. So, this under constraint problem.

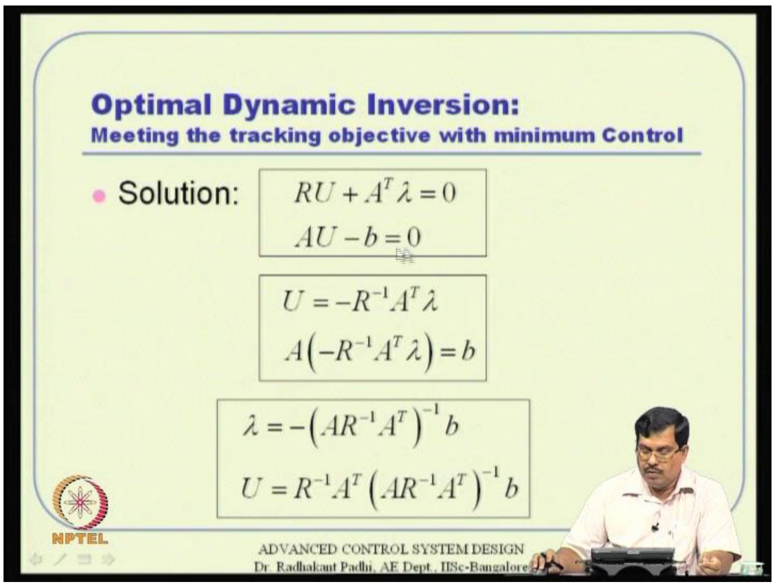
(Refer Slide Time: 27:16)



So, let me put additional objective for which I will tell you let me if something is given to me or some problem demands that I will do something justifiable something meaningful then I will put that also the tracking objective. But without that one I will always I can always formulate some sort of control minimizing problem actually, I will try to get control solution for which this performance index needs to minimize at all time, this quadratic performance index. So, I will try to minimize this control effort subjected to this constraint equation that I am talking actually, this constraint equation guarantees tracking and this objective what I am talking about this guarantees minimum control effort for that tracking basically.

So, now we can say that this for any point of time this is standard kind of quadratic performance index with linear constraint equation and hence, it is very rather standard to solve it with static optimization ideas and all that actually. So, we will interpret this is my cross function to minimize this is a constraint equation. So, that augmented performance index turns out to be j plus lambda transpose, a u minus b and then I have to satisfy both the equations together that del j by del u equal to 0 and del j bar by del lambda also equal to 0. And lambda is the lambda dimension of lambda is the dimension of y obliviously, there is coming from this constraint equation actually.

(Refer Slide Time: 28:42)



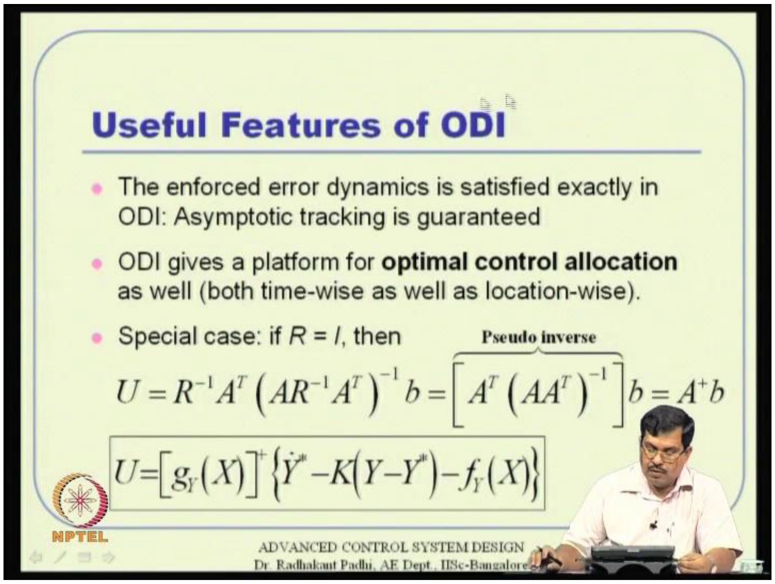
So, how do you do that? Suppose, del j bar by del j bar by del u let us do then this first term you get r u the second term you get a transpose lambda so, r u plus a transpose lambda is to be 0. And second one, if you do this turns out to be a u minus b equal to 0 that means, a u equal to be this is nothing but the constraint equation again. So, here I have given here so, first r u plus a transpose lambda 0 a u minus b equal to 0. So, these two equation need to be solve together. So, let me solve for u from here so u is nothing but minus r inverse a transpose lambda and then, if I substitute that u here by actually what I am getting here is a times that is nothing but b A times u whatever u I have a equal to b.

So, let me solve for lambda from here and lambda solution turns out to be like that and once I will to lambda solution I will put it break here in the control solution. So, my control turns out to be r inverse a transpose this entire matrix inverse times b. And this is possible, because why is it possible because the if you see this if a lambda equal to b and think like that this particular matrix a times r inverse times a transpose, if that only a certainly some sort of square matrix and more than that r is supposed to be positive definite here. That is requirement for some sort of convex cross function and all that what your telling here, r must be possible definite then again by standard (( )) sort of thing you just take r is a diagonal matrix to econometrics actually.

So; that means, I means this particular r being a positive definite matrix r inverse in also positive definite matrix and because of that a times r inverse a transpose is guaranteed to be positive semi definite matrix are going to be square is going to be symmetric it is also going to be positive semi definite. So, that is why we can I will be able to talk about an inverse, all the time assuming here is there know is positive semi definite means there is chance of getting singular again also. So, that one what I am assuming here is this particular matrix is non-singular here by assumption, but this square matrix certainly and that is why this is your taking inverse of that square matrix actually, that is the philosophy.

So, anyway so coming back this constraint equation satisfy guarantees there are dynamics, there are tracking goes to zero and then this objective function that I am selecting also, guarantees that control minimizing solution for that. And the way I select for r that, that means, let say r select r 1 r 2 r 3 as diagonal elements and all that selection of this r 1 r 2 and all that will also give me some sort of a allocation technique basically control allocation technique. That means, if I select r 1 higher than out to be rest of the things then this design will assure that u 1 turns out to be less of them rest of the things basically. So, that way we will able to tune my control components using as proper selection of r actually.

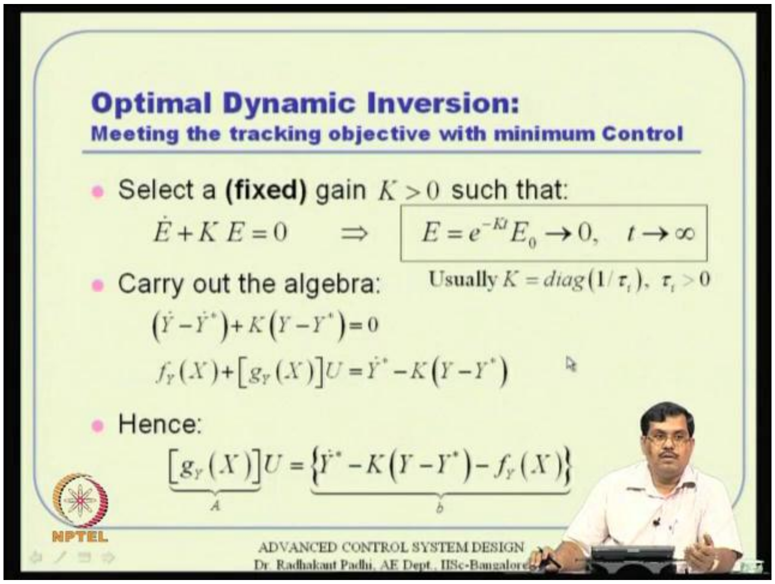
(Refer Slide Time: 31:32)



So, this is what we call is optimal dynamic inversion, because there is some sort of a optimal control allocation actually that that is the reason for that, and some useful features of O D I optimal dynamic inversion is as I told you already. Therefore, the enforce error dynamic is satisfied exactly so, asymptotic tracking behaviour is not compromised. O D I also, gives platform for optimal control allocation as I also told you now. So, if this can be done both time wise as well as location wise or component wise actually, because this r matrix what we are selecting here need not be kind of constraint matrix it can also function of time.

So, as time goes you can you can enlarge one component and one component and think like that initially, you want to use some particular component larger later you do not want to use it larger and thing and all that thing can be incorporated by making r is a function of time also. The function of components of the I mean various components actually, like r 1 r 2 r 3 is turns to essentially stands for u 1 u 2 u 3 I mean when set of that provided R is a diagonal matrix.

(Refer Slide Time: 31:32)

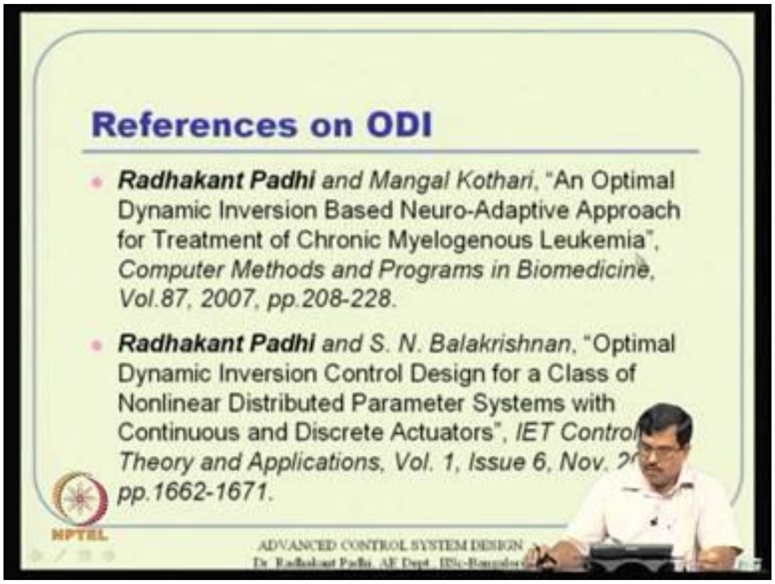


Now, one interesting observation turns out that what if I select r is identity matrix now, if you whether can see this expression and R is identity means R inverse is also identity it means I will end up I will end up something like a transpose a a transpose inverse time b. So, this a transpose a a transpose inverse is nothing but pseudo inverse actually. So, if I take pseudo inverse b so without I mean without all these I mean algebra and all I just end up here this is a non-square matrix so, I just talk about pseudo inverse and that is also, an optimal solution in this in the sense that R becomes identity; that means, equal distribution of control effort and also an actually.

So, without understanding all the details if somebody takes a pseudo inverse for that results, because that he talks about is some sort of a right pseudo inverse and think like that that you got, because this is under constraint equation. If you represents to be over constraint equation pseudo inverse equation is never got actually we will discuss that in matrix theory also.

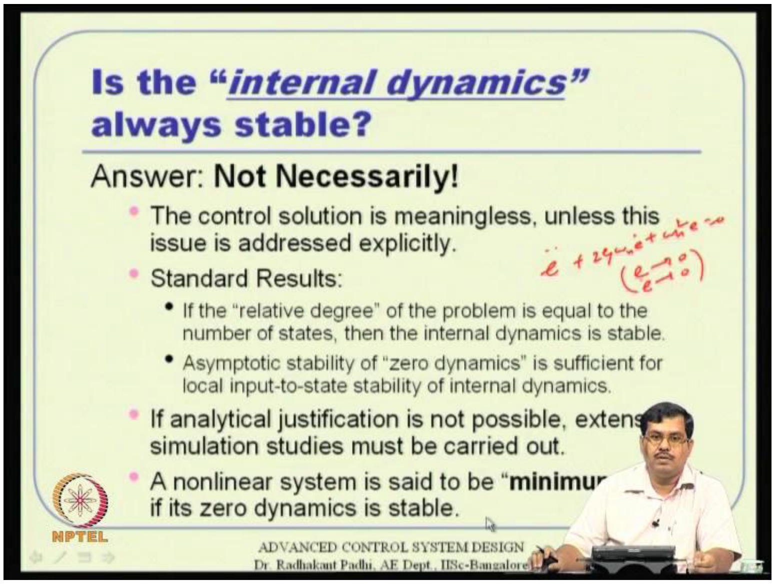
So, is that you forget that you can go back to the matrix theory lecture and then try to recapitulate some of that. Anyway so, this is what I talking this special case r equal to I then you can we will end up with a to pseudo inverse constraint design. But anyways we written more generality here, because you can the fact is you can it gives us a platform for optimal control allocation both time wise as well as location wise or component wise basically. So, that that is the beauty of this O D I basically.

(Refer Slide Time: 34:30)



And also, just a coming sort of thing this O D I idea can be extended to distributed parameters system control, because that that is why we have this infinite modes and thinks like that so, invariably we will end up with under constraint problems and all that actually. Well I talked too much on that, but so those of you want to study more the other reference is from our research output. And first one is it we talks about some sort of biomedical problem and all, but there is generic session on this paper, which talks about this theory is a is a generic frame work actually. And then this second one is a you can think of as some sort of extension to distributed parameter system how do you use this O D I concept for that actually.

(Refer Slide Time: 34:54)



So, let us move on to the next topic the next big topic the question here is, is the internal dynamics always stable that is a major issue. So, the last issue is where the primary big issue that is that was like why this D I design was not very popular for a long time. So, the internal dynamics always stable answer turns out to be not necessarily so again and obliviously, the controls are given if the entire dynamics non stable and one stable then obliviously, it is somewhat like you know present success for present day sort of thing. That means, the control solution is meaningless, because internal dynamic the dynamics in the rest of space is going on stable that means, your system as a whole can (( )) actually your output tracking may happen, but your system may even turn out even cross and think like that.

For example, if you talk about let say turning I mean turning an aircraft and think like that lateral dynamics and if lateral dynamics is simply turn circle sort of thing. And while taking a circular turn your longitudinal dynamics become unstable, that means, height can be negative that means, you are I mean height can be unstable it can come the entire aircraft can cross actually. So, as far lateral dynamics is concerned I mean that you can I mean, if you present the trajectory probably it will still be circle, but if project this I mean trajectory from top actually it will still be a circle. But the aircraft will essentially going some sort of a helical path and then altitude (( )) and finally, it is crossing actually.

So, no point in having such sort of design where we can only concentrate about output actually. So, invariably what it turns out that the moment you talk about dynamics inversion design I mean all that you are discussing here, is input output linearization by the way. And there is another problem concept called input to state linearization which we are not discussing at all actually. So, for that design the internal dynamics is not necessary, but input output version is very practical, that is what lot of people use variety application and that is where, internal dynamics is to be explicitly addressed. That means, any design that you any design that you propose based on dynamic inversion you should take extreme care and to explicitly demonstrate that internal dynamics remains stable and this will never be assumption actually.

So, coming back the control solution is meaningless unless this issue is addressed explicitly. So, there are some standard results first that tells us that if the relative degree that means, the total relative degree that is what I am talking. If the total relative degree of the problem is equal to the number of states then the internal dynamic is stable and again what is the total relative degree like, relative degrees of each of the component of the output vector is actually. If that total relative degree turns out to be equal to number of states that means, there is no internal dynamics actually every dynamics every, I mean dynamics and every kind of dynamics that is visible in the design.

So, the internal dynamics is not there otherwise internal dynamics is always stable actually everything goes to the output vector actually. Now, the why it happens by the way because if I enforce a stable second order dynamics then not only let say like e double dot plus something like one times e dot plus I mean, if I write it somewhere like. Let say, if I assume this e double dot plus 2 zeta omega n e dot and things like that zeta omega n e dot plus omega n square e equal to 0, then this design not only enforces that e goes to 0 it also, enforces that e dot goes to 0 two to that is happens actually.

So, the moment I take the second internal dynamics I have taken care of two dimensions actually. And similarly, if I take care of those many dimensions then I am done actually that way the total number total relative degree turns out to be equal design the number of state then I taken care of stability of all the dimension actually. But that is a very luckily sort of situation that means, many times it will not happen that way. So, cases only e will end up with those situation where, we have enough control availability for example, actually. So, for example, if you talk about only let us say attitude stability of satellites with three axis control independent control. Then for each of the axis you can think of some sort of double integrator and then theta omega each of the axis will go to 0, I mean some stable values actually, but that is the very rare situation every problem will not admit that kind of solution actually.

So, what next? The next case there is something called zero dynamics and the theorem tells that asymptotic stability of zero dynamics is sufficient for local input to state stability of internal dynamics. I mean these are cancel, I mean more detail if somebody wants I can always see (( )) there is a next chapter for this discussions and all that actually. But anyway zero dynamics means, like what we define as zero dynamics is something like that you purposefully take that y star that we discussed as zero, zero for all time actually. So, irrespective to whatever, command that your giving you formulate a parallel artificial problem for which y star is zero y star dot double dot all that are zero actually.

And then you do all control so, with respect to that control because what is that I mean that has become complete independent of the output command that you want to track. So, and hence this becomes some sort of intrinsic property of the system actually. So, that way it is that is what we want to call this zero dynamics. And if the zero dynamics happens to be stable, because you know after the tracking dynamics are goes to zero and think like that the I mean what your assuming is that zero command is already been tracked actually.

So, whatever, is left out is nothing but what is called as a zero dynamics actually. The difference between zero dynamics and internal dynamics is for internal dynamics we are actually giving the desired command, for zero dynamics we are artificially giving zero command and hence, it becomes some sort of a homogeneous system actually. So, the zero dynamics remains the theorem tells us that the zero dynamics is stable, that means, asymptotically stable and think like that which can be kind of assured using Lyapunov theory, because it is some sort of a homogeneous system dynamics. And these Lyapunov theory and all we are going to discuss next class actually, next class onwards then we will see that actually what are the implications.

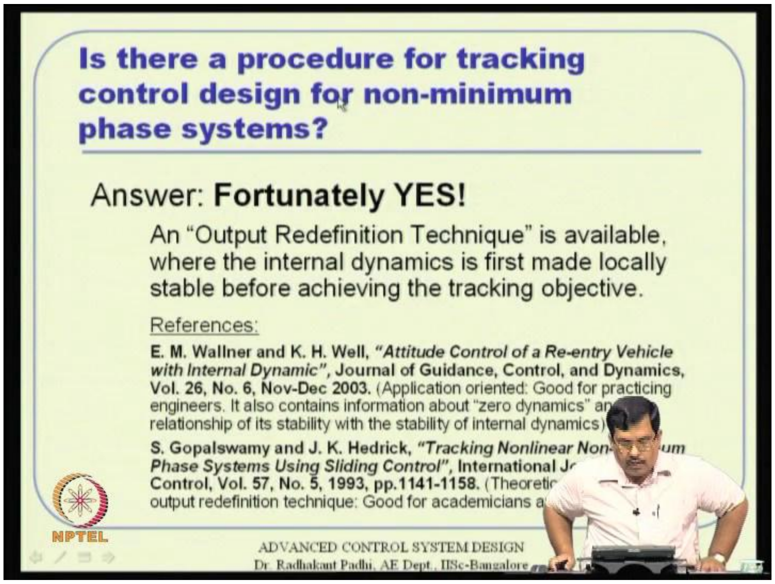
Anyway, if the zero dynamics happens to be asymptotically stable then it is sufficient for to still that the internal dynamics is stable and the input to state sense that means, internal dynamics is guaranteed to stable actually. So, all right so that is what it is. But if that is sometimes also, it is difficult for complicated problem, because we will end up with some sort of complicated zero dynamics how do we solve that? One thing to say that is the zero dynamics can be Linearized about whatever you have written the terminal trajectory and keep on solving that the Linearized zero dynamics will remain stable. And then invoke this what that indirect Lyapunov theorem which tells, if the linear system around some point is stable then non-linear system is also stable locally I mean that that is the way to do that.

Otherwise directly you can invoke this Lyapunov direct theorem then it is the best, but even if do not again I mean if you may not be able to do all that, then and in other words if the analytical justification is not possible, then extensive simulation studies must be carried out actually. So, at least from the simulation studies if the so, I mean that is the Monte Carlo simulation, that even if I take I mean the various signal should track and think like that then internal dynamics still remains stable actually. This zero dynamics conceived also gives us this idea of what is called minimum phase systems. And then by definitions this minimum phase systems or non-linear systems for which the zero dynamics is stable.

So, if the zero dynamics is stable that means, the internal dynamics will stable will be stable and hence, there is no problem actually and this minimum phase non-minimum phase and think like that that we study in classical systems. That means, the right hand side zeros and think like that that those of you know and we have also, discuss some of that in the very beginning classes actually. So, there are there is a right hand side zero sort of thing it represents some sort of a non-minimum phase behaviour and think like that actually, they are concepts are related to that also.

That means, what happens there is in the right hand side whatever, zeros becomes right hand side pole for the internal dynamics actually, that is where the connectivity comes and all that. So, this definition what you see here is not very different from what you have already know actually. So, if the system is for. So, there is the system for which the zero dynamics is unstable that means, that is not a very good candidate for dynamic inversion actually so, there are remedies to do that actually.

(Refer Slide Time: 44:27)

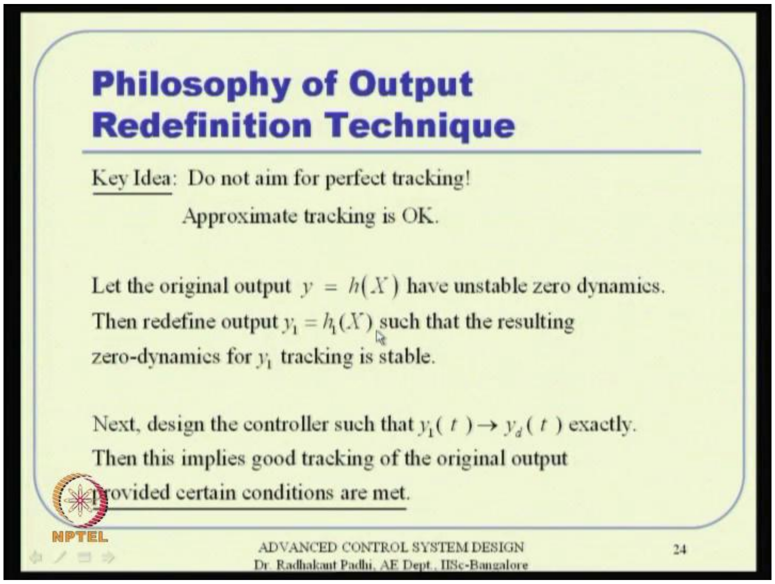


So, is there procedure for tracking control I mean the next question is obviously, is there procedure for tracking control design for non-minimum phase systems that means, for which the zero dynamics is not really good and answer turns out to be fortunately yes, you can do that actually. And this is something called output redefinition technique and this technique is available for which the internal dynamics is first made locally stable before achieving the tracking objectives. So, first we make sure that the internal dynamics remains stable and then on that system you try to design some sort of a tracking controller actually. So, your control objective will be split into two I mean two sort of roles actually.

Sometime that discusses that also like while I mean while developing aircraft your make sure that it behaves like another aircraft and then design a controller for that aircraft and that kind of idea basically and philosophically. So, first you make sure that the controller will first make sure that the internal dynamics is locally stable and then try to observe attend the tracking objective. And in that attend you may not be able to do a perfect job also by the way, but approximants job which is very close to what you want is also all right actually.

And that is what more on that you can study on those these two paper this is the second one probably the kind of seminar paper watch first proposal that. And this is probably a little bit math oriented and good for academy sense and further research think like that, but for engineering things and understanding these techniques and all I will probably suggest the first one actually.

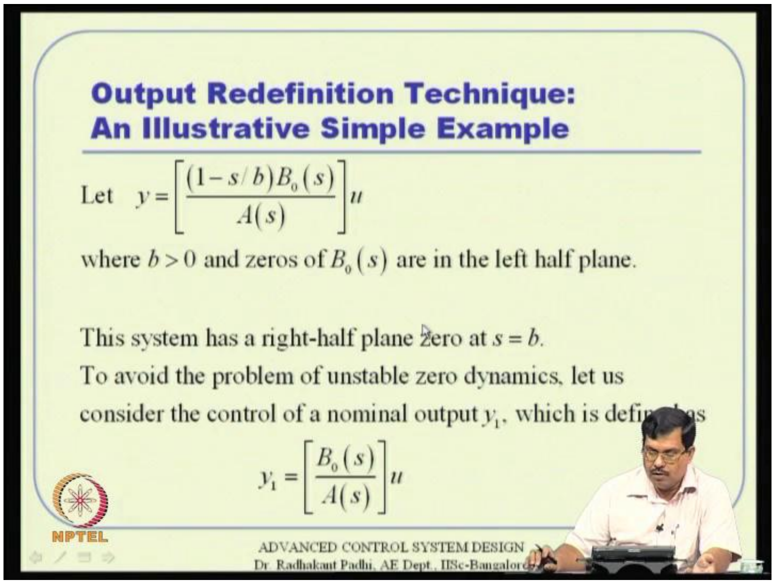
(Refer Slide Time: 46:10)



Anyway, so what is philosophically let us try to understand this little bit at least and then before stopping this discussion. So, the key idea for this output redefinition technique is do not aim for perfect tracking, but approximate tracking is ok that means, but approximate should be very closed to what you really want, I mean that then that sense actually. And so, what is the idea then let the original output be some y equal to h of x which is an unstable zero dynamics. So, what you do then? We define y 1 which is h 1 of x such that the resulting zero dynamics for y 1 tracking is stable, we do not we do not aim for y tracking we aim for y 1 tracking instead. And what you do after this objective is met we go back and see that the design I mean design the controller such that y 1 goes to y d of t r or y star of t exactly.

Then this implies good tracking of the original output provided certain conditions are met actually. So, what is going on here is something like this, we make sure that y 1 tracking is assured then go back and analyse how much error will make with respect to y actually. Instead that particular design what your proposing for y 1 tracking is it really good or bad for y tracking and that that is the way and it turns out to, if the y 1 is selected properly in the judicious manner, then y and y 1 may not be toward actually this of course, may not be toward.

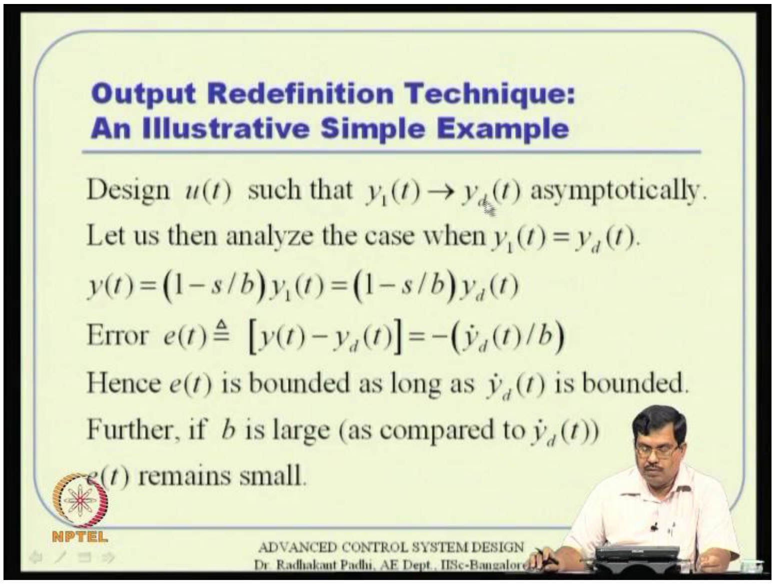
(Refer Slide Time: 47:38)



How do you do that? Let us see here small example, example is again I have taken from certain level apply it non-linear control certainly, anyway let us talk about some sort of linear single input single output system for which we can represent it using transpose function actually. So, if y and u are elected by this transpose function then, obviously, there is right hand side zero here s equal to b for which and clearly there is a problem of internal dynamics zero dynamics is unstable obviously. So, the system has right side right hand side plane zero at s equal to b and then, if we avoid that then what we will do that we will simply take out this and formulate A y 1. Remember B 0 does not contain a right hand side zero actually that is our assumption.

So, we define y 1 and make sure that y 1 tracks our objective everything is met actually now, what if we will analyse what happens to y basically that is the objective. So, y 1 has gone to y d already y d turns out desired actually this is the y d and y star are same.

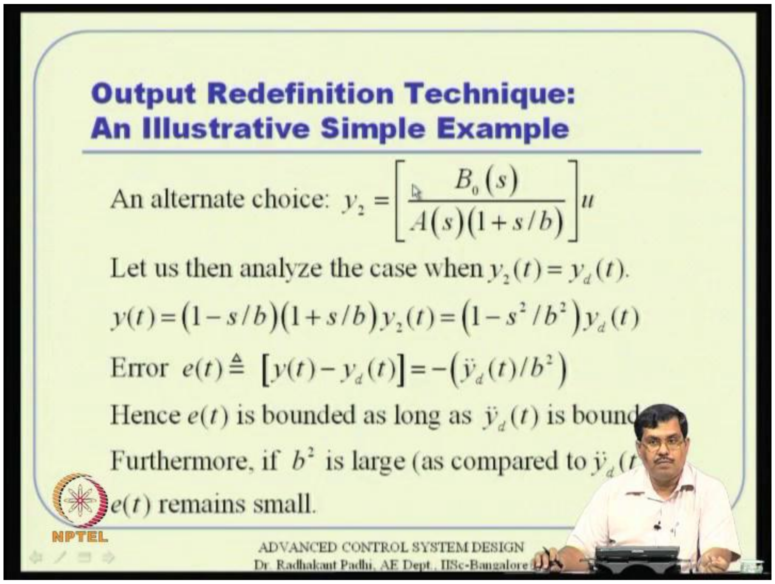
(Refer Slide Time: 48:33)



So, y 1 has gone to y d and so what you do and then what we do is we analyse, the case when y 1 is equal to y d, y one has gone to y d that that is the some sum actually. Now, what is y? Y and y 1 are defined like that so obviously, y and y 1 are related like that y 1 defined like this so, y defined like that so, y is this times whatever this times y 1. So, that is what your written here y is nothing but that one minus s by b times y 1 so obviously, this y 1 has gone to y d already so, instead of y 1 I can substitute y d and then error is defined that way. So, that means, this I can define this see this derivatives and all if you see that actually This is so, remember this one is y d minus y d dot y b this expression if you expand it nothing but y d minus y d dot y b, I was multiplied by s is nothing but time derivative actually.

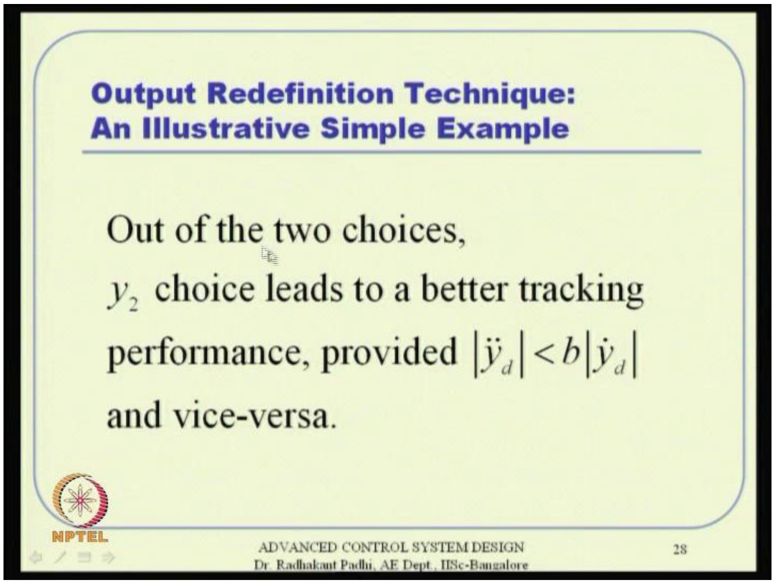
So, y d minus y d dot y b minus y d; that means we will end up with this expression actually. So, what it tells us the error with respect to y and y d will remain bounded as long as y dot t is bounded where b is a constant number actually. As long as y dot t is bounded then this error will remain bounded actually, further more if b is large that means, this bound is going to smaller and smaller actually so that is one thing. Now, let us reinterpret this same problem defining y 2 instead, y 1 was defined like that let us perambulate a problem two with y 2 we are not only will cancel this, but we will add 1 minus 1 plus s by in the denominator.

(Refer Slide Time: 50:30)



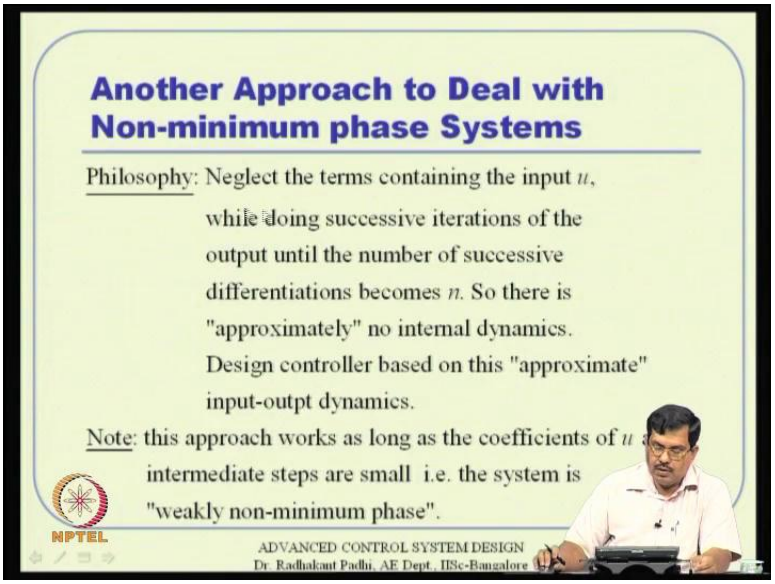
So, that is what we have not taken the zero part, but we have added a pole identically in the in the left side actually, then what actually. So, we can again do the same analysis again, but this time it is y double dot, because s square will pop out. So, this e t e of t becomes a function of y double dot now actually. So, this e of t will remain bounded as long as y double dot is bounded and if b square is larger this time then y d again obviously, e t e will be smaller and smaller.

(Refer Slide Time: 51:15)



Now, naturally turns out that this selection probably greater than that, but what non-linear control lead to be slightly more careful it, I mean kind of see this expression very carefully and this analysis is true. So, out of these two choices y 2 is better provided this condition holds good, because it may replace so, happen in some problem that y dot in this condition may not be valid actually. Then your y 1 selection is better, but if this condition holds good then your y t is better actually. So, we are like some of this idea of that we can exploit.

(Refer Slide Time: 51:41)



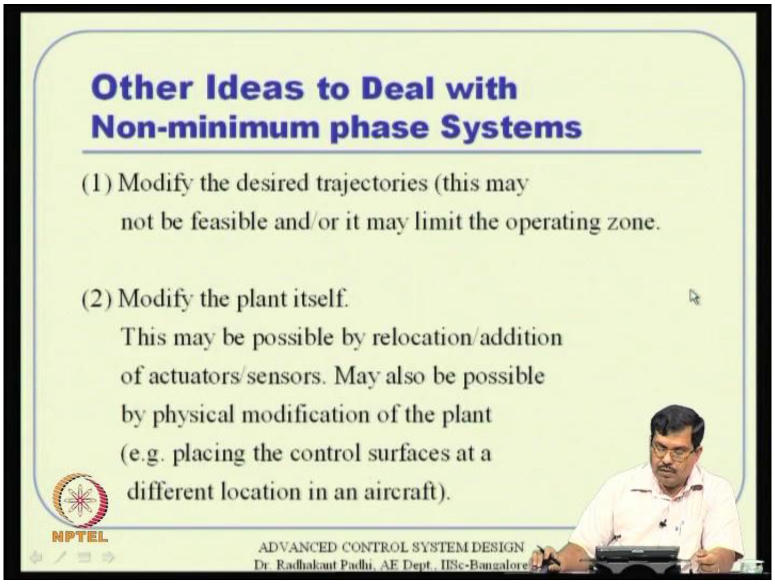
Another approach to deal with non-minimum phase first of all which like neglect terms and containing the input u while doing successive iteration of the output until number of successive, differentiations becomes n actually. This is the trick that we regularly exploit in aerospace design by the way, because if you see the equations, that we discussed we derived also and think like that your v dot w dot and all that will contain a term from control surface reflex also as well, but these are not very powerful. So, if you really want to synthesize your controller directly from the v dot w dot for the guidance purpose and think like that, then your rotational dynamics will remain stable actually.

So, how do you do normally we take some instead of v dot we take v double dot instead and one of the design we will discuss, in the one of the classes later probably. But we will try to end up this Neuro Adaptive design with aircraft control and all that actually. So, instead of v dot we take v double dot and then again the control dot either, they simply tossing the that is zero and think like that control effective zero. Then you go to the next level of differentiation and then v dot v double dot we contain r and think like that. So, w double dot we contain expression of q basically and for I mean that q dot and r dot and all there is still control will appear and because of that the control is powerful for rotational dynamics and all that.

So, that that will we want to use a dynamics inversion we want to invert the matrix actually. So, these are concepts that you have to I mean especially in aerospace applications we use it regularly. So, what is the idea here? If so, at particular output level let us say your control appears so you so, do not get unnecessarily happy, because that particular channel the control may appear, but it may be very less effective. So, in that sense you can very well neglect the control variable and do one more derivative and then hope for the best in the sense the next level of derivative control can still appear one more time. Because your derivatives of other state variables and all that and at that level the control may become powerful actually so, we will have good control availability at that that level actually.

So, the then you use the dynamic inversion at that level of differentiation so, what we were telling here is exciting more level of I mean more and more output derivatives actually. So, that is philosophy that we use regularly actually. And as I told this approach works as long as the coefficient of u at the intermediate steps are small that means, the system is weakly non-minimum phase. If the system turns out to be weakly non-minimum phase then this idea happen an works on hopefully normally, this aerospace problems it works very well very well actually.

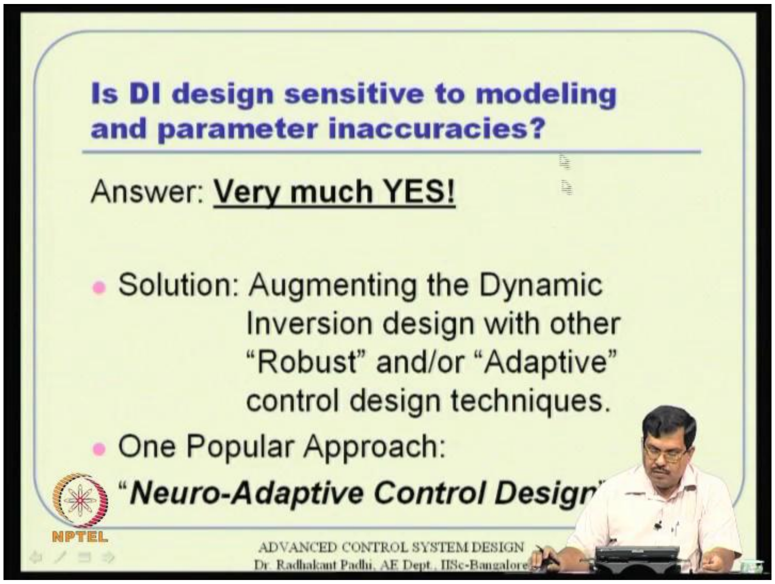
(Refer Slide Time: 54:27)



Other idea is modify the desired trajectories that means, if you really assumed sharp manual do not assume sharp manual basically it not capable. So, you use some sort of benelli manual or think like that. And then the other ultimate is modify the plant itself that means you go back and design redesign the plant itself actually. This may be possible by relocation of the actuators sensor or by adding more number of actuator sensors in the in the loop actually basically. So, you are altering the transform function or you are altering the system dynamics that you are looking for actually.

So, it may be also possible by physical modification of the plant that means, for example, placing the control surfaces at different locations in an aircraft. So, these are like bigger issues and all that so, in that sense we have we have plant itself you are changing and so, then this first you are your modifying your objective, the second one your plant itself your changing actually which is the way to do that.

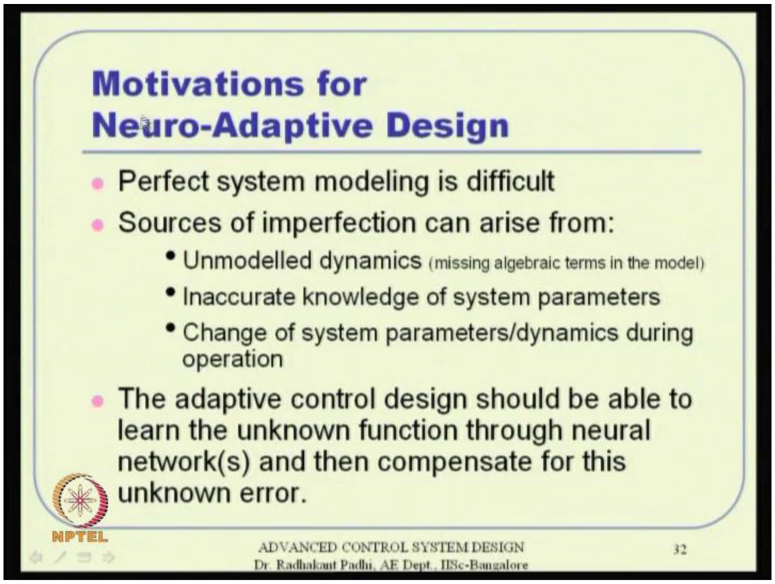
(Refer Slide Time: 55:24)



The next big issue that is the fourth one what we discussed before, is the dynamics inversion design sensitive to modelling and parameter inaccuracies and the answer turns out to be very much yes actually. And the solution is because of the this particular big issue this design was not very popular for a long time again and now, people know how to handle this actually. So, if it becomes big issue in a particular design then they tell let us not use dynamic inversion as it is, but let us try to augment this using more robust control design or adaptive control design techniques. That means, you will see D I plus such infinity D I plus adaptive control D I plus Neuro Adaptive like that actually those ideas.

And one popular approach that we propose and we use in our lab also like that is what is called Neuro Adaptive control design, this adaptive control design philosophy however, level network is also in the loop actually.

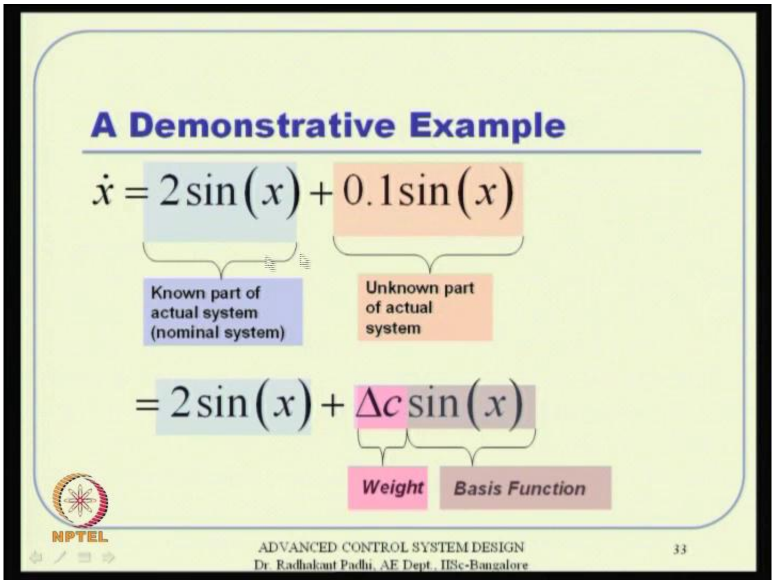
(Refer Slide Time: 56:34)



And this particular design we are going to discuss extensible anyway in subsequent lecture. So, I will not going to talk too much on that, but philosophically speaking why do we need this thing this issue comes why we need it parameter inaccuracy modelling inaccuracy and all that. So, motivations for n a design Neuro Adaptive design is already told the perfect system modelling is difficult and source of imperfection can arise from Unmodelled dynamics that means, missing algebraic terms in the model or inaccurate knowledge of the system parameters. You really do not have exact number of I mean values of the parameters the parameters can keep changing also.

For example, mass of the aircraft keep changing because at least fuel keep changing out actually or mass of the rocket changes very rapidly rather that way. And you can also, change of systems parameters systems dynamics during operation, I mean it may not remain constant forever actually. So, the adaptive control design should be able to learn the unknown function through the Neuro networks and then compensate for this unknown error. So, the entire plant you know large part of the plant some person you do not know so that whatever, unknown plant unknown person also that person needs to be learn and these that unknown, I mean that inaccuracy whatever, we are talking is going to throw some sort of unknown function in the system dynamics. And that unknown function makes to the learn through the Neuro network and then once you learn it you can comments it for that actually.

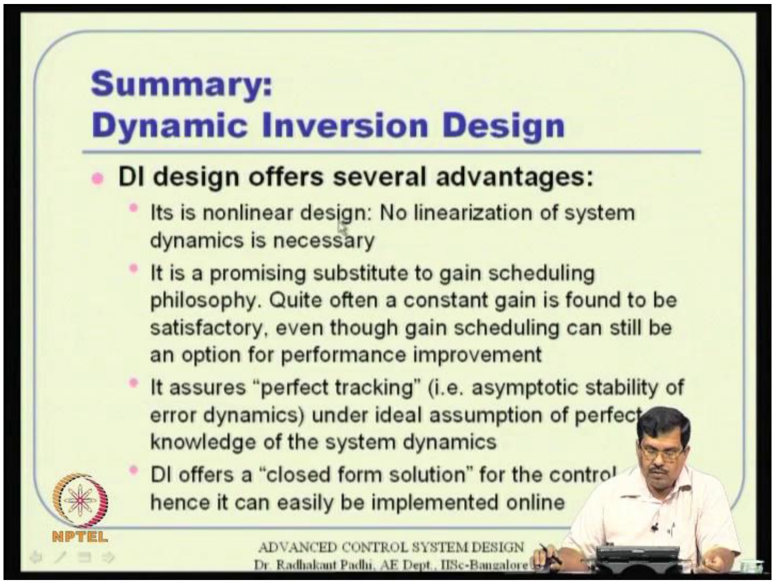
(Refer Slide Time: 57:51)



So, philosophically speaking that sees something like this. So, let us say x dot equal to 2 sin x that is actually known part of the systems dynamics, but 2 is not 2, but for some reason at least point one, that point one we are not knowing what is this point one or minus 1 or point I mean minus 1 or whatever, it is that is not known to us. So, because this two is become to two point one and x dot x become like this. So, this additional non-linear function is evolved, because the parameter inaccuracy, because of parameter inaccuracy, we have done unknown function and all that.

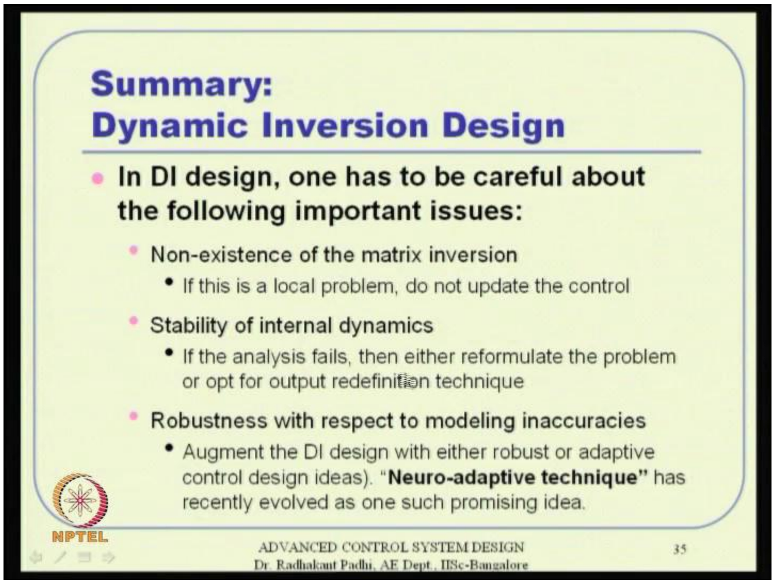
So, this particular function if I analyse little I can tell this part sin of sin x can be some sort of basis function and delta c turns out to be some sort of unknown coefficient. So, if I if I will learn this function using this basis function sin x or some other basis function are also hold good then I actually know this unknown part after learning. So, after learning will not use only 2 sin x, but I will I will able to use entire thing actually. Because this systems dynamics will no more be complete unknown basically, that is all idea there we will talk about that in subsequent classes anyway.

(Refer Slide Time: 58:59)



So, summary, so summary of this entire dynamics inversion design D I offers several advantages first of all I mean let us try to recapitulate again it is a non-linear design no linearization of system dynamic is necessary. It is a certainly promising substitute for gain scheduling philosophy and quite often a constant gain is found to be satisfactory even though limited amount of scheduling can still be done actually. It assures perfect tracking that means, asymptotic stability of error dynamics under the ideal assumption that perfect knowledge of the system dynamics is available. And the D I offers a closed form solution for the control and hence, it can be easily implemented for online applications.

(Refer Slide Time: 59:40)



But when you implement D I design, one has to be careful about the following important issues first of all the non-existence of the matrix inversion. And if this is a local problem do not update the control and there is a stability of internal dynamics issue if the analysis fails, then either for reformulate the problem or opt for output redefinition technique these are options available to you. And robustness with respect to modelling inaccuracies and all you can augment that D I design with Neuro-adaptive technique actually. So, with that I think, I will stop this lecture, thanks a lot.