Optimal Control, Guidance and Estimation

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

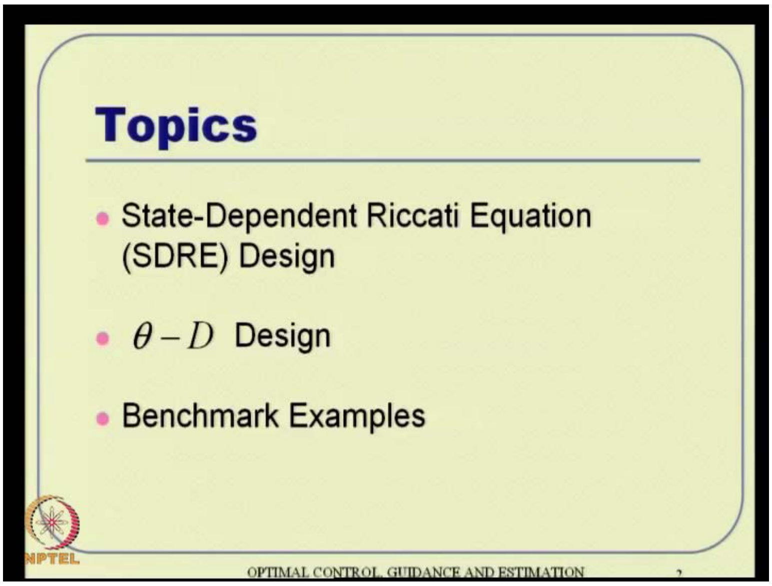
Lecture No. # 19

SDRE and θ – D Designs

Hello everybody, we will continue our lecture series on this optimal control guidance and estimation course. So far, we have talked about I mean lot of lectures on so called L Q R theory that Linear Quadratic Regulator Theory, and various extensions and that in everything we discussed.

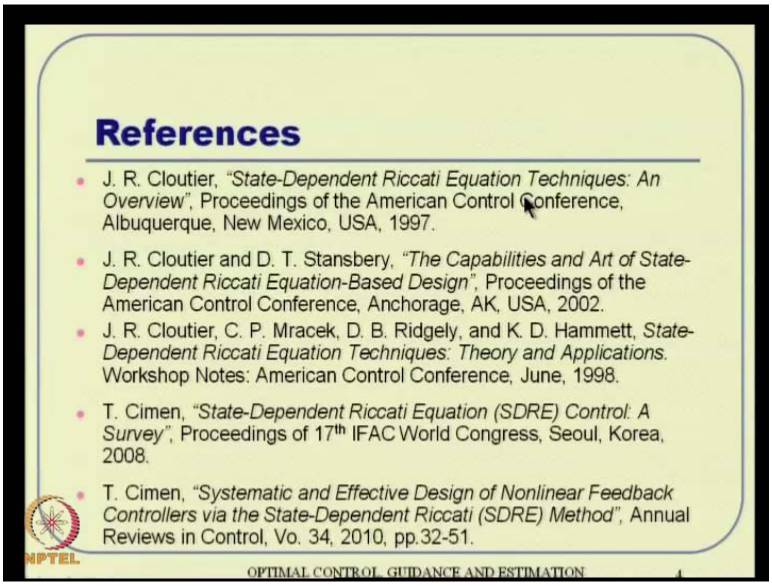
Now, we will go slowly start moving on to the non-linear systems and all. So, there are the two techniques that has appeared, not very far of about 10-15 years like all this development happen, and then we will we will see how these are applicable to the non-linear systems, actually. So, one is called SDRE method - State Dependent Riccati Equation method, and the other one is also called theta d design. And both of that are are applicable to a class of non-linear systems with same regulator in mind, actually.

(Refer Slide Time: 01:10)



So, let us see that and the topics and in this particular lecture, first we will give a good overview, I mean fairly good overview of State Dependent Riccati Equation design, and then we will move onto theta D design. On the way, we will also talk about this consult of benchmark example problem to make the ideas clear, actually. Let us start with SDRE design that is become quite popular, and then we call it as a simplest form of non-linear control design, non-linear optimal control design that I can think about. The whole idea here is it is somehow kind of repeatedly used the l q r solution that we all know, actually. So, let us see how things proceed and things like that, it will very different and very clear, very quickly actually.

(Refer Slide Time: 01:55)



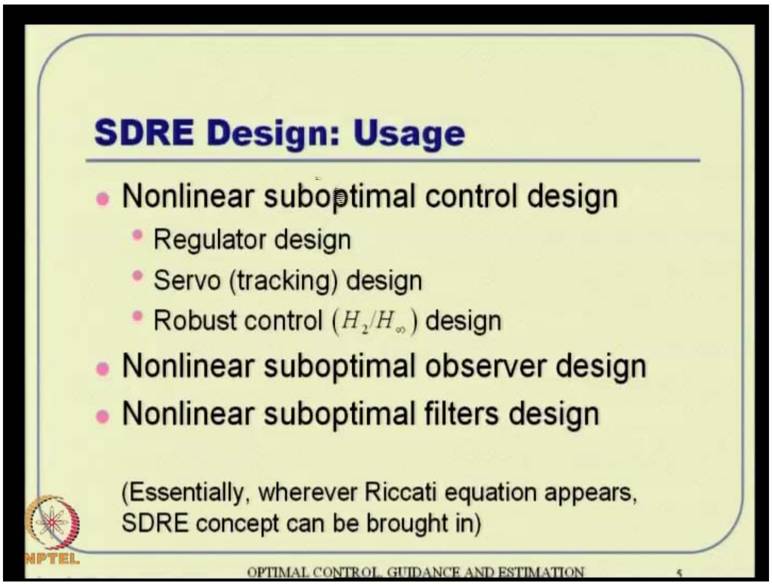
So, these are some of the references that I have used and I will be using. Some of these things, you can you can think of using that itself, actually. And of late, I think if you really want one difference, then the last one is something I will recommend. This is something like review paper. It it appeared in annual reviews of control which is actually a reviews journal, about I mean in 2010. And then, a little bit prior to that, the same author has presented this and it is available in conference proceedings. Also in the 17 th ifac world congress, he has presented the same before, but genuine version happens to be little more complete and little more extensive and all that, actually.

However, the conference version should be available freely and all that you need to do is probably register yourself, and ifac papers kept online and you can download it. (( ))

Anyway, but the whole idea started with sometime, maybe it was there when the people are talking about lqr and all that; however, J R Cloutier, who who happens to be a scientist in a nuclear air force based in US, took it to a very high level. In other words, he he started all these deep research is this method and then, lot of good nice theoretical result also he proposed.

So, anyway a lot of credit goes to him and his coworkers actually. So, with that let us see what is SDRE design, its it is usage and all that actually.

(Refer Slide Time: 03:22)

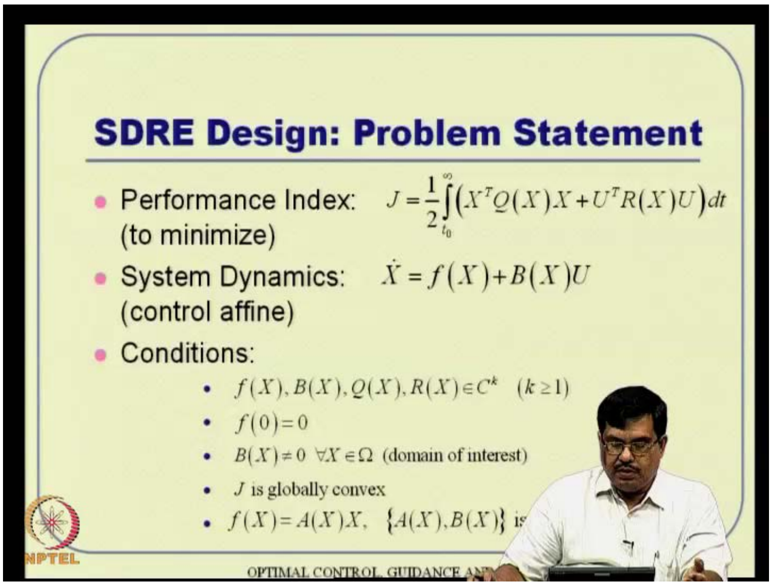


So, before talking about SDRE design, the the utility part of it largely falls on into these categories actually. First thing is non-linear suboptimal control design. (( )) I am not talking about optimal design per say, because the system dynamics will have. I mean you you will understand as you go along. The system dynamics is not used as a non-linear system. It is somewhat approximated as a linear looping dynamics and all that actually. So, that brings in the issue of sub optimality. Also, we will see theoretically why it is suboptimal actually.

So, it is used, first first thing is a regulator design and then, something like servo design and tracking problem. You can also use it for a steam as inherent design thing like that and also you can use it something like non-linear suboptimal observer design, like like filter design is also visible.

So, what happens here is, whenever you see Riccati equation, in other words, all these things we will talk about some sort of a Riccati equation. So, whenever you see Riccati equation, where there is design, then the state dependent Riccati equation technique can be brought in actually. So, that is where the utility is fairly wide.

(Refer Slide Time: 04:40)



Now, talking to, I mean this SDRE design problem statement and all that, this first thing is what we are looking for is a performance index in quadratic form. So, very close to what we know in lqr design, especially infinite time lqr problem t of f goes to infinity.

So, but in addition, we will also see that this this Q and R can be function of states also. It may not be constant matrices or it will not be only, I mean, a priory fixed constant number and then, matrices and all that. Actually, you can you can have a design based on Q, which Q itself can be a function of X. Similarly, R itself also can be a function of X and these are additional freedom is this design basically. But more important is, you should have a system dynamics in this form. This is called as control affine form or linear in the control form typically known.

So, you have all the non-linearity's in the f of X and b of x matrices f of f is a vector and b is a matrix, but U which is a control variable appears linear. Actually, that is the Q equal U. That is the key requirement. But in addition to these, this is what you see here, this f of X and B of X something. Well, these two things will need to satisfy several other conditions as well. The very first thing is, this f f of X and B of X as well as Q of X and R of X, they should belong to class C K, where K should be at least 1. But this, when you talk about class C 0, it is is continuous function. Class C 1 is something like derivative is continuous and thing like that actually, the first order derivative continuous class C 2, when K equal to 2 is the second order derivatives or continuous thing like that actually.

So, what you demand if at least first order Methodius of all these functions actually functions and matrices. In addition to that thing, you must also have f of 0 equal to 0. Remember, when the whole idea here is to minimize the derivation. I mean minimize this I mean that take X to 0. Basically, that is the whole idea here and once X goes to 0, then X dot should also go to 0 because x goes to 0 asymptotically then, X dot should also go 0. In that situation what happens is, U should also go to 0. So, if if this 1 X dot goes to 0 and U goes to 0 and f of X is non 0, then there is an incompatibility issue. (( )) In other words, let me explain that little later, probably when X goes to 0 here, U should go to 0. Then what happens if this goes to 0, U goes to 0 and this is this is non 0, let us say. Then, this will not be non. This will also be non 0. That is the problem actually.

But if it is, if this also goes to 0, if this also goes to 0 then this also goes to 0; that means, this is a compatibility problem. Actually, I mean this may be compatibility sort of thing actually. So, why you want that because once you once you go to the steady state, you do not want to deviate from the steady state actually. So, that is thus the whole reason why under steady state control should go to 0.

Once control goes to 0, then the state state derivative should also remain at 0 actually. So, that is the reason why we want that actually (( )) alright. So, the next condition is b of x should be non 0 for all x actually and that is directly visible, because b x b of x happens to be 0, then no control region invoke. Actually, it leaves to the loss of control ability actually.

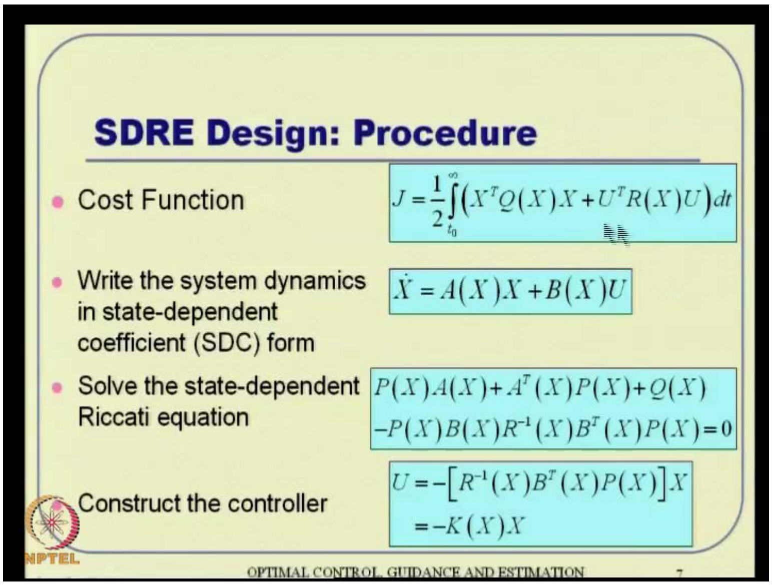
So, if your B of X has to be has to be non 0 for all x actually, then, in addition to that, J has to be globally convex because we are all talking about Riccati based design derivative based approach and thing like that. So, J has to be a convex function actually. And then the key technique here is the f of X has to be decomposed into A of X into X. That is called state dependent coefficient form. Remember, we are not talking about

Taylor's series linearize and all that actually. We just see the function and just try to extract or just simply algebraically write it this way.

So, if you write it that way, there are various approaches one can write. It is not a unique approach and whatever A of X you select, then this A of X and B of X should be point wise stabilizable. That means, if you put a value if you put a value for X, then these two becomes some sort of matrix or some A matrix and B matrix will have numbers now and at every value of X that you put, then this A B pair should be stabilizable.

So, these are the requirements before proceeding further. In summary, what it means is, you should have a control affine system and you should incur something like a quadratic cost function. In other words, the the aim is to design a regulator where X X should go to 0 actually. Then, the conditions required are like this. All the all the things f of X, B of x, Q of X and R of X has to be at least class C K, F of 0 has to be 0, b of X has to be non 0 for all X. And and then the j has to be convex, and when you write f of X like this, A of X into X, then A of X and B of X has to be point wise stabilizable.

(Refer Slide Time: 10:21)



So, under these five broad conditions, let us see how to proceed. The idea is extremely simple. You have a you have a J already quadratic looping form. It is not strictly quadratic, because once you if you put Q of X and R of X, then at least this part is not really quadratic. But assume that Q of X is just a number after substituting from values of values of X, then it happens to be something like a quadratic cost function and here also writing f of X here as A of X times X.

So, once you write it that way, then X dot is nothing but A of X of A of X times X plus B of X times numerically. So, what it happens, it is like, something like a linear looping form or or I mean it is known as something called SDC form or State Dependent Coefficient form.

So, once you have, once you see this and just imagine for a second that we know the information about state. So, if you once you substitute it, I mean once you know the value and substitute it here then, Q R A and B takes the form of some numbers and hence, you can interpret that a some sort of A standard lqr problem actually.

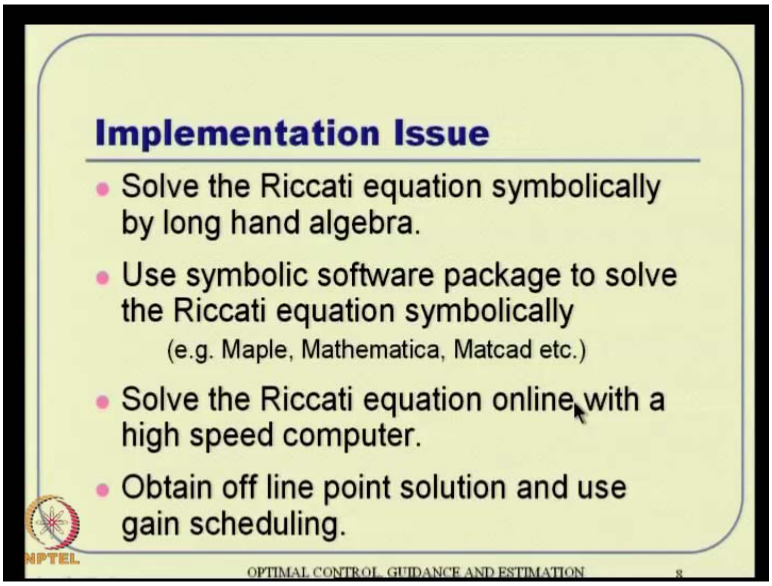
So, if a standard lqr problem, then you know the solution for that and the solution happens to be something like this, U equal to minus R inverse B transpose B into X all or minus K, where P comes from this Riccati equation actually.

Now, the only difference here is, this Riccati equation is a function of state now. Hence, this you have to really see whether, I mean this p of… It it is not like a constant Riccati equation actually, in other words every time an x changes, this the the equation itself changes and hence, every time we need to keep on solving this equation online actually.

So, the procedure is like that. You use somehow selected Q of X and R of X and write f of X is something like A X times X and this is anyway ready, this part. So, then you should solve this Riccati equation and get a value for P of X. Once you get P of X, then compute the control that we are looking.

So, as I told, if the solution procedure demands that you were, that you have to solve this, keep on solving this Riccati equation every time even though it is a finite time, even though it is a infinite time regulator problem actually. You know in infinite time lqr, you are you just need to solve this Riccati equation one time and that can be done offline and all that. But here is if you have to solve this equation repeatedly, and it has to be solved online actually, and then you have to construct the controller exactly like using the formula for what you know, what you know for lqr actually.

(Refer Slide Time: 12:58)



So obviously certain certain implementation issues, and all that the first implementation issue is how you solve this Riccati equation? That is that is what it (( )) down to most of the difficulties, numerical problems actually. So, first thing you can think of solving the Riccati equations symbolically by long hand algebra and if the problem dimension happens to be smaller, we have given some examples before in this in this lecture series. It is possible to solve the Riccati equation symbolically. So, that the best thing to happen if you can do that.

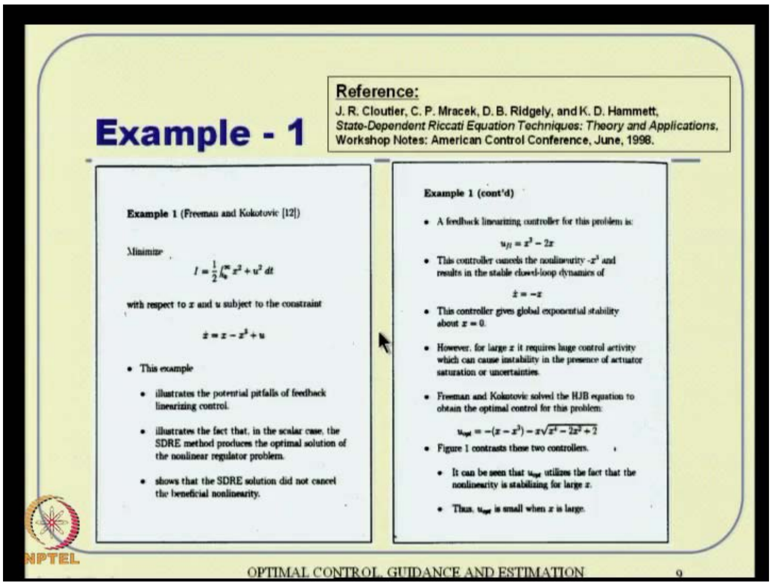
The second best is, probably you can think of using some sort of a symbolic software package, something like Maple, Mathematica, Matcad lot of symbolic software are available nowadays, and you can solve this, try to solve this Riccati equation actually. If you have prolong dimension is not very high then, this software will try to probably give some solutions actually.

So, this solution, you will use in this is this heading. So, that is next best actually and if is that too is not possible, then it is like a numerical solution sort of thing. So, you have to take a fast computer and try to solve this equation online actually.

And however, Riccati equation is now heavily studied. In other words, they are fast efficient algorithms also available. So, if you use that algorithm in the fast high speed computers, then the online solution is still possible. If that too is not possible, then the last of solution (( )) probably you take lot of points, lot of different different X and all that and then try to get a solution for K. Then, you have a set of k’s and hence you interpolate from there. Now, this is a concept of gain scheduling actually.

So, you can bring in the concept of gain scheduling and then, do this offline solution of this Riccati equation in many places, and then schedule the gains actually or interpolate the gains. So, they are the things that people suggest, I mean for using this SDRE online actually.

(Refer Slide Time: 14:55)



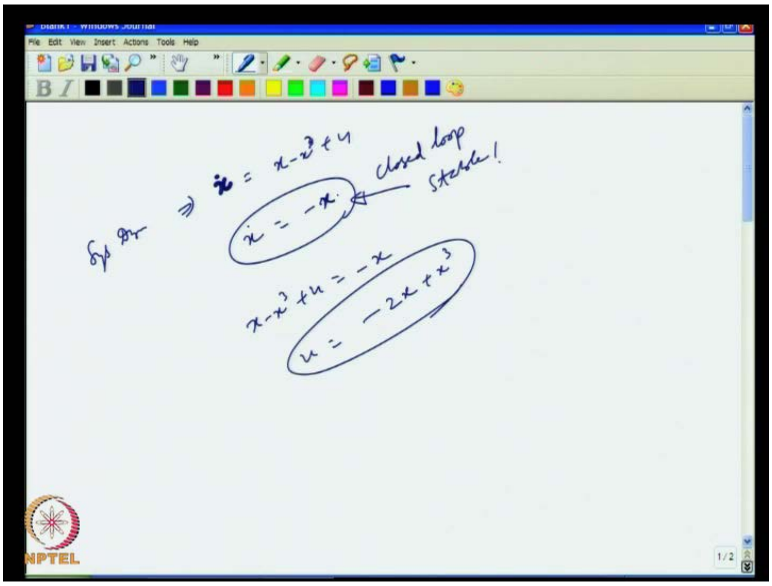
Now, we will see a nice example to to show what is going on and and thing like that. This example, I have taken from this one actually. It is a nice workshop in an American control conference in 1998 actually. Finally, this is a very standard functional problem. We have a quadratic regulate, I mean we have a quadratic cost function and we have a non-linear side equation. I remember, I think you will be able to download this this power points as well. So, you can see this equation very clear actually

Anyway, so this is X dot equal to X minus X cube plus U. So, that is thus the standard benchmark example. We will also compare to this equation again and again and in the domain programming lecture and all that. I mean the domain programming lecture, we will actually derive the solution as something like this (( )). In other words, this is actually a particular problem what you are looking at. Even though it is a non-linear problem, there is an exact close form solution available for solving this actually.

So, that gives us some sort of a benchmark control solution and using this benchmark control solution, we will be able to demonstrate certain certain good things. In other words, if you are proposing a new method and you will be getting a solution from there, then does it not really give this solution or at least very close to this solution actually.

And this also has another point; that means, if you really think about another non-linear control design, the very thing that comes to mind is, this is also called dynamic inversion design or feedback inversion design actually. Those of you are interested, you see my other lecture, one advance control where I have taken one or two lectures on feedback linearization or dynamic inversion design actually. So, what it happens is, dynamic inversion is easy. In other words, you can just think about designing a controller such that the closed loop system will appear something like that actually. So, you have this X dot equal to and then like probably do that part u (( )) So we have this.

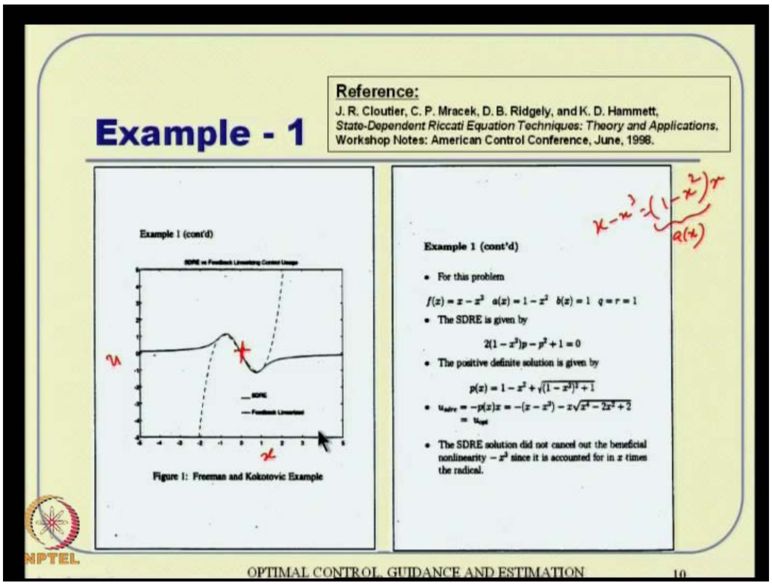
(Refer Slide Time: 17:12)



This X dot equal to, I think where is this, one second, X dot equal to X minus X cube plus U, that is the problem and you want to operate with X dot equal to minus X. This would be the closed loop. This is the system dynamic actually. So, if you equate the two, then what happens here is, you get this X minus X cube plus U is equal to minus X because we know that this is this is a stabilizing solution actually.

So, this is this is stable. We know know for sure actually. So, this all is for the control, actually after that. So, it will turn out to be minus 2 X plus X cube. Basically, it is as simple like that. It is very very easy way. I mean in in one line two lines you can derive it. The control solution which will make the system operates, something like X dot equal to minus X actually which is stable anyway. That is, what is done in this is, this is your feedback linear zing controller which is X cube minus 2 X, and then it operates like this actually. Well, well you can also argue that this also does this job because it also takes X to 0 anyway. That is what basic introduce table solution means. So, this kind of solution is probably too much complex and probably not needed basically, but I will quickly see that there are some nice properties of these compared to this feedback linearizing control actually.

(Refer Slide Time: 18:45)



That is what you can see. When you when you plot it, the solution and thing like that, it all happens that you can actually, first of all you can put it it into this. This f of x is x minus x cube. So, you take it there and you just write it as a of x equal to one minus x square into x x minus x cube. This is something like this, x minus x cube is equal to x into or sorry this is something like (( )) 1 minus x square into x. So, this is your a of x actually.

That is, that is that is what is done here, once you have that, you can put it back into the Riccati equation and the Riccati equation will throw some equation like this. Hence, you will get a solution for p and then, you get a controller or which is minus p times x because p is 1 here. So, it will turn out to be exactly the same is what you have seen here actually.

Remember, this comes from hjb equation, Hamilton Jacobi bellman theory. This one, this can be derived that way, but if you just substitute this and then put in this Riccati equation. And then, try to solve a controller, it happens to be exactly same actually. Anyway, so, they are the two solutions available. One is the optimal control solution and the other one is the feedback linearizing solution or dynamic inversion solution. So, if you plot it it in a in a state space sort of thing, I mean if this (( )) what happen this is actually control versus x actually.

So, what you really want is is some some regulation about 0 0 point, this point actually. So, what happens here is, you can you can see this, I mean both are fairly close to each other around the point, but as thus deviation starts to become more and more, then the feedback linearizing solution quickly diverges to infinity; that means, you really need a very high amount of control compared to the optimal control.

In fact, optimal control just stabilizes to a finite value close to 0 and in fact, it goes to 0 actually when the derivation becomes large. Why does it happen is because if you can see this this this system dynamics, x dot equal to x minus x cube, the minus x cube is actually a stabilizing term. Because if you if you take x x dot equal to some minus x to the power k and k is odd, then it can very easily be solved that v of x, you take this half x square (()) function, and then v dot you can say x times x dot and this is nothing but minus x to the power k plus 1.

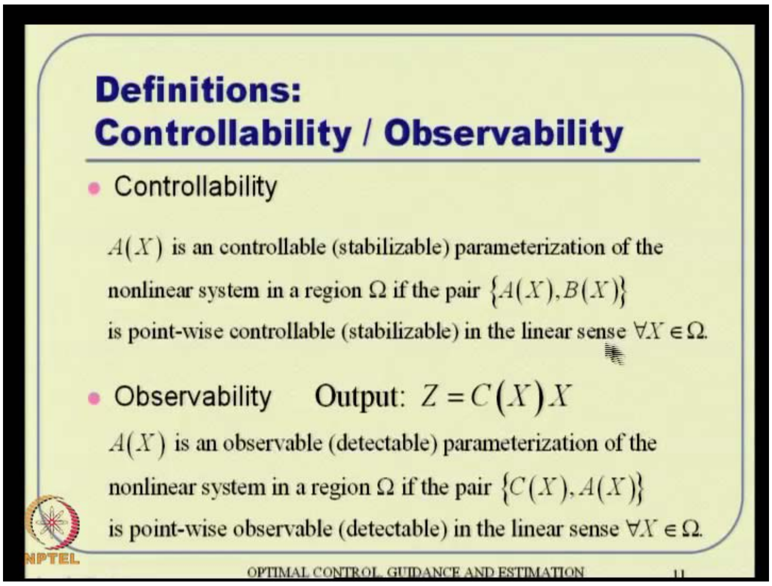
So, this is nothing but minus x to the power k plus one basically. So, where k plus 1 happens to be even and hence, to always negative definite and hence it is stabilizing. So, in other words, what you can see here is, this x to the power q minus x to the power q is a stabilizing term; however, when x goes to the magnitude of x is less than one, then then x cube is very less compared to x and wherever you remember, this is a plus power here; that means, x dot equal to x that is the stabilizing term actually.

So, when the deviation is large and minus x cube is in the stabilizing, we do not need any control. When the deviation is small it is unstable system. So, we need a control actually for that. And that entire good feature is, there in optimal controller whereas feedback linearization or dynamic inversion space control, what it does is, it blindly applies this control. In other words, every time it tries to enforce this actually, this x dot equal to minus x solution.

In the process, the control that is required also goes to infinity because even more the x plus x cube term there, when x goes more than one; that means, it becomes larger and larger. Then, it is a direct function of x cube actually. So, the control magnitude goes to infinity actually. So, that is the right thing to have. So, when you when you have optimal control solution there are certain nice things, nice properties around that actually.

So, there is, the authors try to rigorously point it out through this example that, we do not conceal beneficial nonlinearity by using this optimal control theory based approaches. So, this is called beneficial nonlinearity when you have minus x u term. Its power demolishes when when it will comes down below one actually; however, when the derivation is lost, its power is good, and hence it you should retain this this feature as much as possible.

(Refer Slide Time: 23:35)



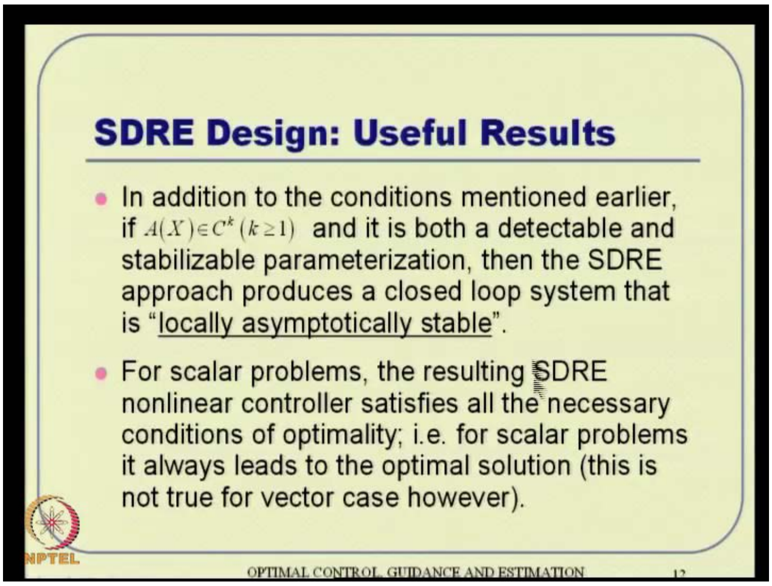
So, the claim here is, optimal control solution does retain that kind of here actually. Now, let us how go through some little bit theoretical things as to why is it popular and why it organize linearization and all that actually. First is some definition and first thing is what is called is Controllability and Observability. Lot of you know, but what you really need is just point wise controllability. In other words, if you have this numbers for X actually, then A of X and B of X is as good as some sort of a lti system a and b sort of thing linear time invariant where constant A B are use actually.

So, if that happens for every such case that A of X and B of X, you substitute some number and each, the pair A B is stabilizable every point, then it is called point wise controllable or point wise stablilizable actually for for all X in the in the domain of interest omega basically.

Similarly, A of X is an observable or detectable parameterization of the non-linear system, if this pair C of X and A of X. In this case, you have to talk about an output also and then, output has to be written in this form. So, once you write it in this form, then it turns out if the pair C of X and A of X point wise observable or detectable. Then, it is called an observable parameterization actually.

So, if this happens to be A and B, happens to be like point wise controllable, then A of X is called controllable parameterization. If C of X and A of X pair is detectable or observable, then A of X is called observable parameterization actually. This concept is used for observable and filter design. You may not need it here (( )), since our focus is to design control actually.

(Refer Slide Time: 25:14)



So, the result tells something like this. It turns out that, all these things we have mentioned right, this this I mean five conditions you mentioned here, and we also mentioned that this has to belong to class C K actually, where K at least 1.

Now, this first theorem tells you, first theorem tell us, that in addition to those conditions, if A of X happens to be class C K function also, and is also both detectable and stabilizable parameterization, then SDRE approach always leads to a closed loop system, that is locally asymptotically stable.

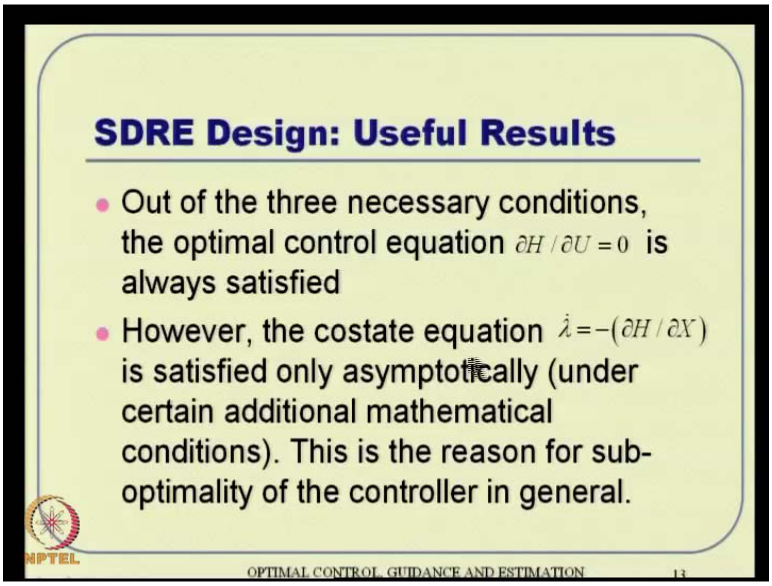
So, even more this choice choice of doing this (()), this F of X writing, this A of X into X is is not unique. So, different different diametrical pops up and things like that and hence, you have this point wise stabilizablity detectability concepts, no matter what you get, whatever way of X, if it satisfies both the both the conditions; that means, it has to be of class C K and also it has to be detectable and stabilizable parameterization. Then, the nice theorem tells us that it, I mean it leads to a closed loop system that is always asymptotically stable locally of course, actually.

So, that local asymptotically stability guarantee is there basically. That enhances the confidence that things will long go back no matter whatever A of X isolate, I just I just have to verify that A of X is is detectable, I mean and stabilizable as well as it has smoothness properties actually.

The second theorem tells us that for scalar problem, like example that we discussed here, this is actually a single state problem. That is the scalar problem. For scalar problem, the resulting SDRE non-linear controller satisfies all the necessary conditions of optimality. That means, for scalar problem, it always leads to the optimal solution. It does not lead to sub optimality in scalar signs actually.

And you can, you have demonstrated that also here, because this example clearly shows that if you solve it through this approach, then you sincerely land up with the same controller, where no matter whether you solve from that equation or you approach this through a Riccati equation. (( )) In other words, fair Riccati equation approach and take this A of X equal to 1 minus X square and thing like that and both lead to the same solution actually. Well, it is for scalar problems we do here with optimal solution actually.

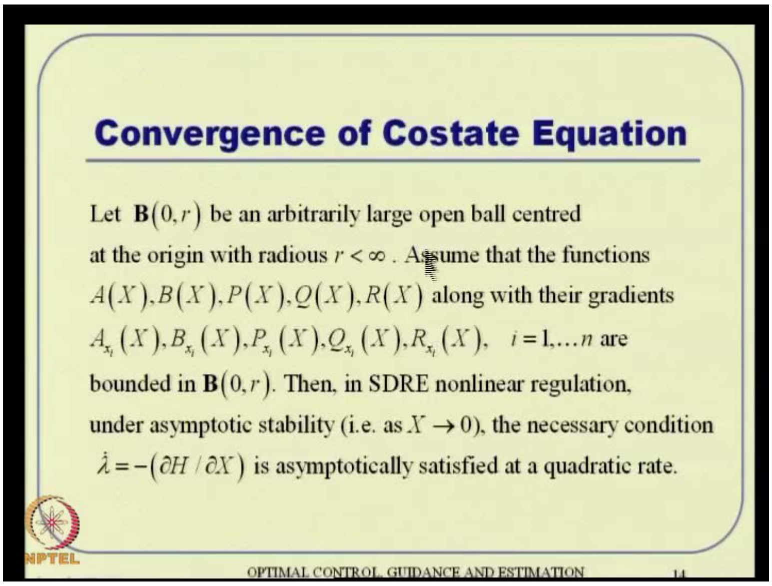
(Refer Slide Time: 27:51)



However, what happens to the vector problem. Now, the condition tells us that in in general in general, the vector for the vector problems, out of the three necessary conditions, the optimal control equation; that means, del H by del U is always satisfied, numeral 3 equations satisfied, state to state costate and optimal control equations.

State equation is anyway satisfied, because that is all the system dynamics will evolve and then the remaining, out of the remaining two, one equation is always satisfied, no matter whatever parameter is you do actually. The only problem is the costate equation does not satisfy exactly actually. However, the costate equation, which is lambda dot is minus del H by del X is satisfied asymptotically, and that happens only under certain additional mathematical conditions as well actually. This is the reason for sub optimality of the controller in general basically.

(Refer Slide Time: 28:47)



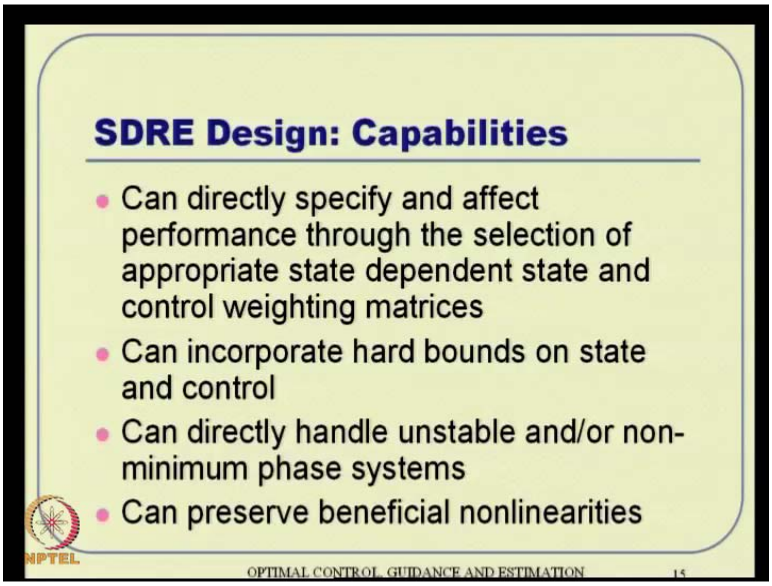
Now, the question is what those additional mathematical conditions and all. This tells us something like that actually. So, let B of 0 R be an arbitrary large open ball centered at the origin with radius, some finite radius actually, radius is less than infinity.

We have assumed that the functions, all all these functions A of X, B of X, P of X, Q of X, R of X, A and their gradients as well; that means, you take all these first order derivatives and all, they are all bounded within this ball, understand. It talks about some sort of an open ball centered at the origin with finite radius, and it also demands that all these functions along with there, I mean partial derivations, they are bounded actually in that ball.

Then, it tells you that SDRE non-linear regulation of, I mean in SDRE non-linear regulation under the asymptotic stability. That means, when X tends to 0, the necessary conditions lamda dot minus del H by del X is asymptotically satisfied and not only that, it satisfied at a quadratic rate as well actually. So to summarize in a simplistic way, then what happens is, all that you have to make sure that the all these state dependent functions as well as the partial derivative, I mean should remain bounded in that wall actually.

Then, this lambda dot equal to minus of del H by del X, that is, the costate equation is satisfied asymptotically. It is satisfied asymptotically at a quadratic rate, which is nice. It will also very quickly converse to this one, this whatever lambda dot or whatever lambda pops up, that will quickly satisfy this, as we start applying this controller and as X starts developing towards 0 actually.

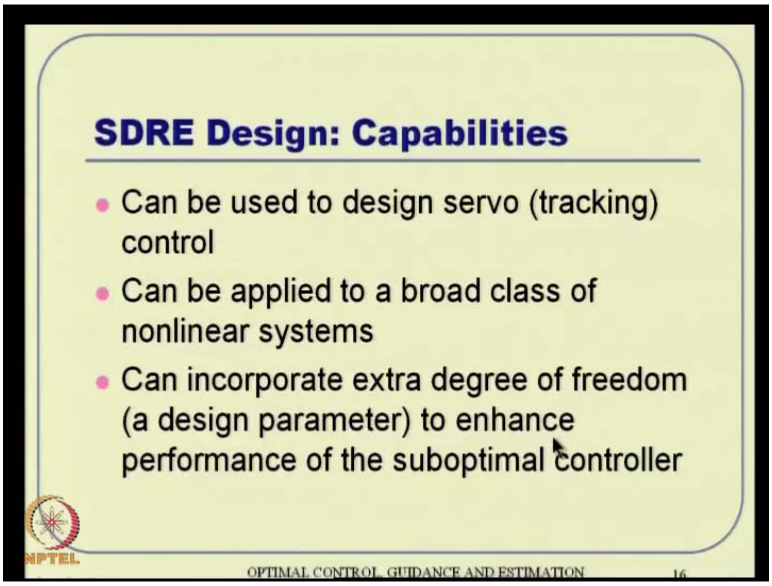
(Refer Slide Time: 30:37)



Now, coming to different things like what are the capabilities of SDRE and all this can be summarized something like this. You can directly specify and affect performance through the selection of appropriate state dependent state and control weighting matrices that Q of X and R of X matrices actually. It can also, there are extensions to tell that it can also incorporate hard bounds on state and control. That can also be counted for actually. It can directly handle directly handle unstable and or on non minimum phase systems also actually. That is the generic feature for for any optimal control base designs. So, it also retains that kind of behavior actually.

And, as we demonstrated in that scalar example, it can preserve beneficial nonlinearities as well actually. So, whenever the nonlinearities are good, it keeps it and whenever is not good, it tries to a kind of discard it actually.

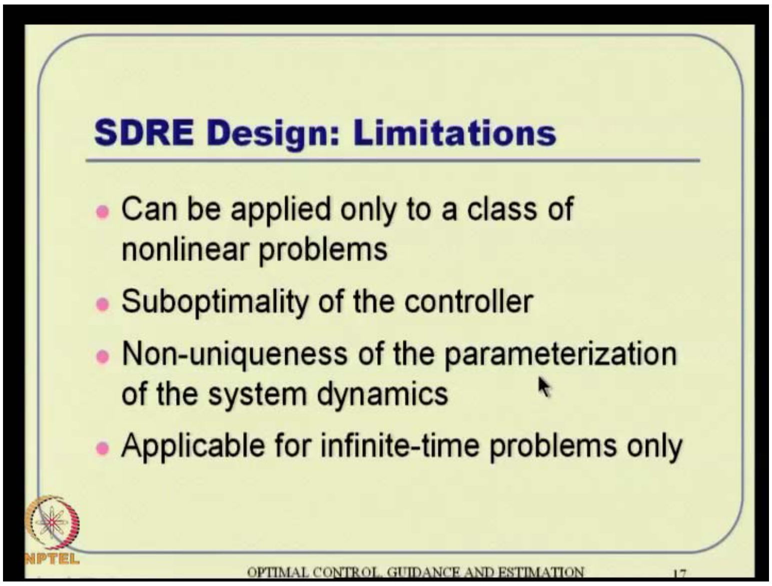
(Refer Slide Time: 31:39)



So, it can also be used for tracking control applications, thus that you can see that in a couple of slides later. It can be applied to a broad class of non-linear system that is, what we are talking is, this control fn system and stc parameterization, where a of x has certain certain nice properties that is not a very restrictive class of non-linear system, and many problems including aero space problems, as well as a robotic something like that, it can be actually model it that way.

So, the model naturally satisfies those kinds of things actually. So, this is not a restrictive class of systems and hence, it has been applied to many different class of problems, and hence it has gained its popularity actually. And interestingly, in late, it also turns out that it can incorporate an extra degree of freedom in the design parameter to enhance the performance of the suboptimal controller. We will see that in an example actually.

(Refer Slide Time: 32:37)



What are the limitations? The very first limitation is that it can be applied only to this class of non-linear problem that we are talking about, which is in general, true. Still, you should remember that, it is not a very universal non-linear control design that we can widely take and apply to any any non-linear problems actually.

The second is; obviously, we have to leave with sub optimality in general and only one nearby scalar problem we have optimality. But in general, we have to just get happy with the sub optimality of the controller. But the point here is, if you have this one, it can, it can incorporate some extra degree of freedom and thing like that. We will we will outline that idea very soon, but with that, it turns out that the sub optimality of the controller that we are talking about is not really very bad. It it can happen to be very closed optimal actually.

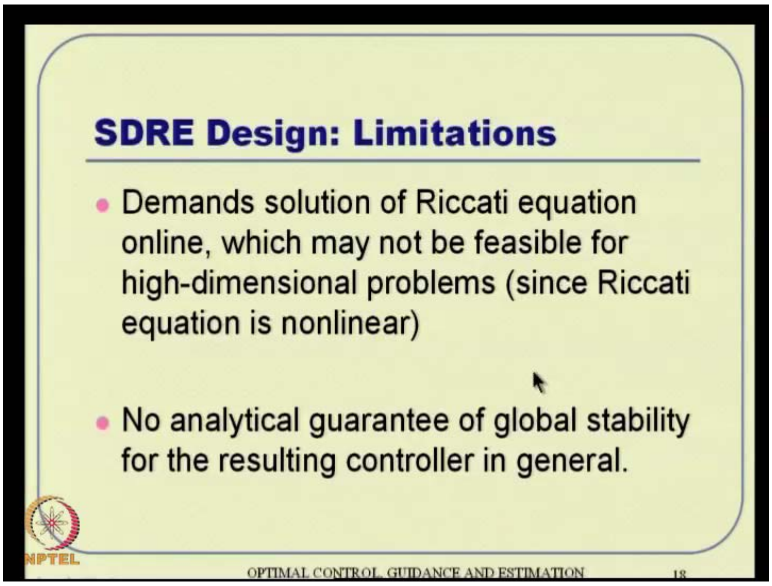
And another limitation is that, it is this very very major limitation rather in my view, is it is this non uniqueness of the parameterization of the system dynamics. That means, that the stc form that we are talking about can be written in various ways, and that happens to be a major bottle neck actually. In other words it works, but it will not work very very good way, because it all depends on the on the choice or the experience or the insight of the problem problem that the designer has actually.

So, this happens to be kind of a drawback in this design. But no the nice thing is, no matter whatever a of x as we select, there is local asymptotic stability guaranteed actually provided that a of x, and this way some some mild some mild conditions actually.

The other other limitation is that it is applicable to infinite time problems only. But even I will not insist on that because off late in another I mean in 2010 to 11, and all that whereas in papers where the concepts like this are getting used for finite time regard equations also. So, it is not really a major bottle neck, because the ideas are start appearing for finite time problems as well actually. Where you remember for finite time problems, we need this differential Riccati equation are not algebraic Riccati equations actually.

So, anyway those of you are interested, can see some some literatures around that line, but those theories are not as well double after. They are not as straight forward as infinite time problems actually. So, there are additional things to be done and all that actually. So I encourage all of you who are interested to see, they can see some of this development in the literature actually.

(Refer Slide Time: 35:14)

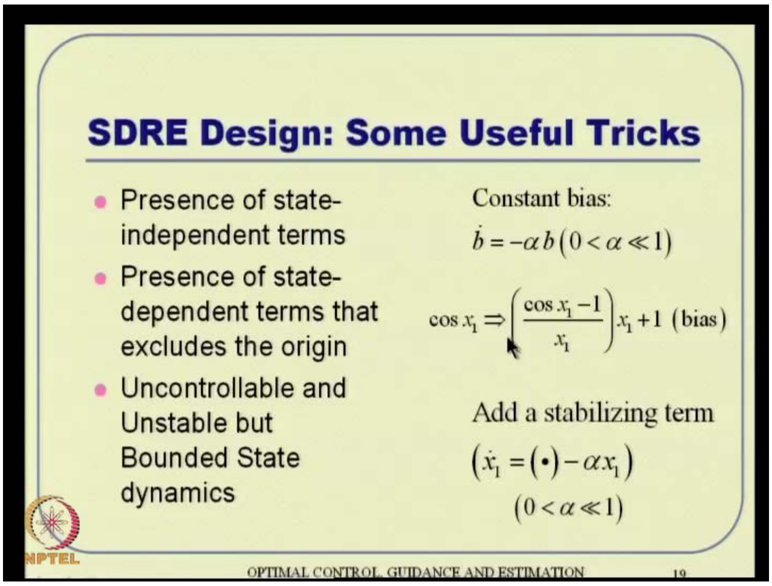


Another limitation is that it demands a solution of Riccati equation online, which may not be feasible for high dimensional problems. That means, if your Riccati equation, remember it is a it is a matrix non-linear matrix equation, and it has multiple solutions. We have to select a positive definite solution thing like that actually.

So, when the dimension of the matrix increases then, we have this feasible condition, if I mean, online solution becomes an issue actually. So, how big the problem can be like, if we have some two states very much, five states very much fine, ten states, twenty states, fifty states, thousand states and all. I mean fifty states, now, be in other words ten-twenty states. About fifty states, you start worrying about it, all depends on the time constraint of the process. But if you have a high dimensional problem, something like 1000 states, 2000 states, 50000 states and all that then obviously, it is not possible actually.

So, for for flexible system dynamics and all or infinite dimensional system, when truncated to some some, let us say thousand states and all that is not possible to use the SDRE technique online actually. Also remember, there is no analytical guarantee of global stability. So, we also talked about stability, but if the stability is local, it cannot be global actually. So, be careful about in that aspect as well.

(Refer Slide Time: 36:34)



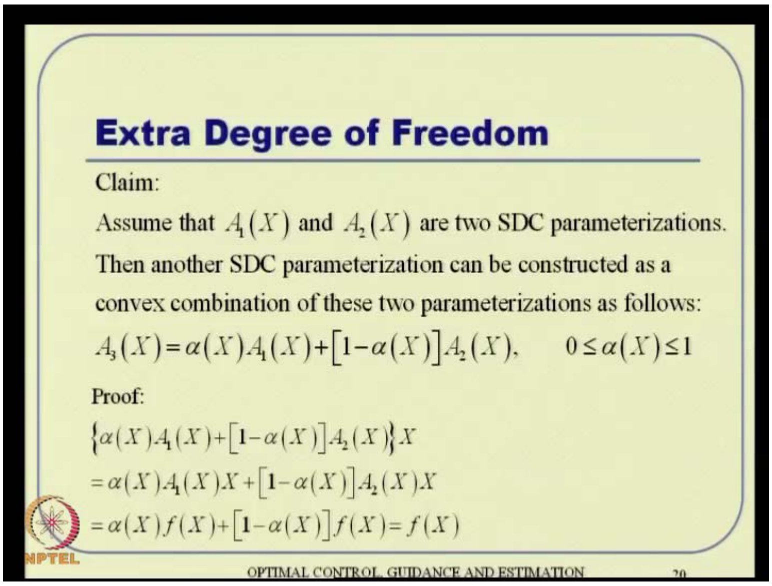
So, there are some useful tricks suggested by by Clautier (( )) and his group. So, first thing first, what (( )) if there is a constant bias term sort of thing; that means, remember what we really require this. All these things will be evolved around this this requirement that f of 0 has to be 0. If f of 0 is non zero, then what do you do? Now if it it is non 0, probably, if you have x dot equal to f of x plus b of x plus some b vector bias vector let us say, then this x dot will not go to 0, x dot will become b basically. Then what happens there actually. So, like that how many of these things will evolve around that actually.

So, first thing is, if it is a bias term, then one thing to handle handle, it is to construct an artificial state something like this, where alpha is very close to 0 positive number which is very close to 0. Then, then what happens is, it is like slowly decaying quantity sort of thing. But the decaying is such a slow rate that, I mean for all practical purpose, it is like a constant variable actually.

Now, still in this ideas a little further, we have this, suppose we have this cosine x form in the system dynamics and know that 1 x, 1 goes to 0, then cos 0 is is 1 which is not 0. So, what you do about that? You can do this manipulation, this cos x 1 minus 1 divided by x 1 into x 1. If you see that, then next 1 goes to 0, then this 1 1 minus 1 which is 0 and we have this 0 by 0 form. Then, we can talk equations of finite numbers sort of things is not very bad that way. Then, we have this this plus 1 and this this plus 1 can be interpreted as a bias term and hence, it can be handled this way actually.

Then what happens for uncontrollable and unstable, but bounded state dynamics. So then, those situations again, you can introduce some some term like this; minus x dot x 1 dot. Whatever existing things is there, we just put minus alpha time x 1, some term where alpha again is a small number. Then what happens? It tries to kind of nullify that. When 1 x 1 goes larger and larger even though alpha is more, x 1 dot is largely dictated by this term let us say. Then, it tries to kind of prevent that from really becoming very large and all that actually. So, these are some of the tricks suggested in the literature to handle handle it within that SDRE frame work actually.

(Refer Slide Time: 39:07)



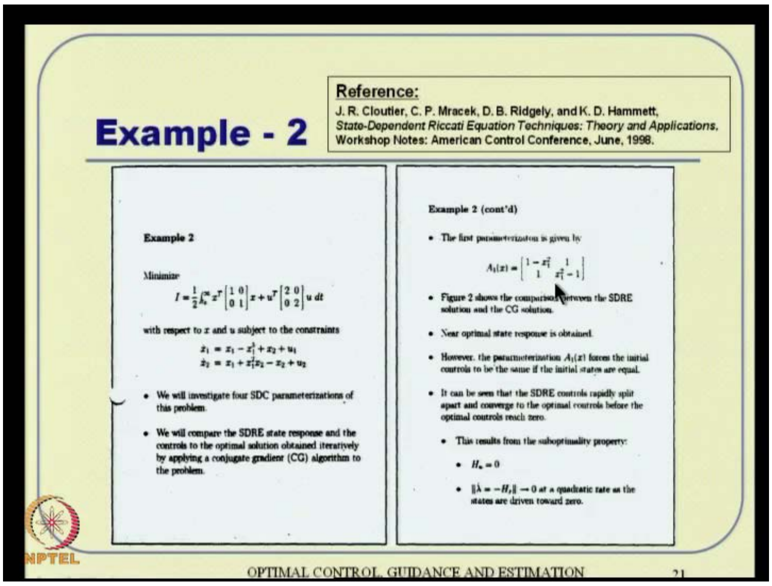
Now, coming to that concept that I was talking that something called extra degree of freedom and all that. So, what happens here is, suppose, because even more SDC formulation or SDC parameterization is not a unique parameterization actually. That means, somebody can come up with A 1 and some can come up with A 2 parameterization. Now the question here is, can I not formulate an A 3 because I already know A 1 and A 2 something like this, this. This is something called convex combination of A 1 and A 2 basically.

So, I construct an A 3 of X, which is a convex combination of A 1 and A 2; that means, alpha times A 1 plus 1 minus alpha times A 2 where alpha itself can be a function of X also. Then, what happens that A 3 will also work. How does it work? Now, let us let us analyze this. If you if you substitute A 3; that is, A 3 times X, and A 3 times X is nothing but this one, because A 3 is this one right. So, substitute A 3 A 3 times X is nothing but alpha X times A 1 X of into X. This X goes comes here and X comes here also. Then then you can also see that A 1 of X is a parameterization. That means this is f of X. A 2 of X is also a parameterization. That is also f of X; that means, the two terms cancelled out and you have the f of X only.

So, what we are talking is, A 3 can be a valid parameterization for the same f of X. In other words, if you know A 1 and A 2, then we can always construct an A 3 something like this. What happens here is alpha becomes concern additional tuning point quantity.

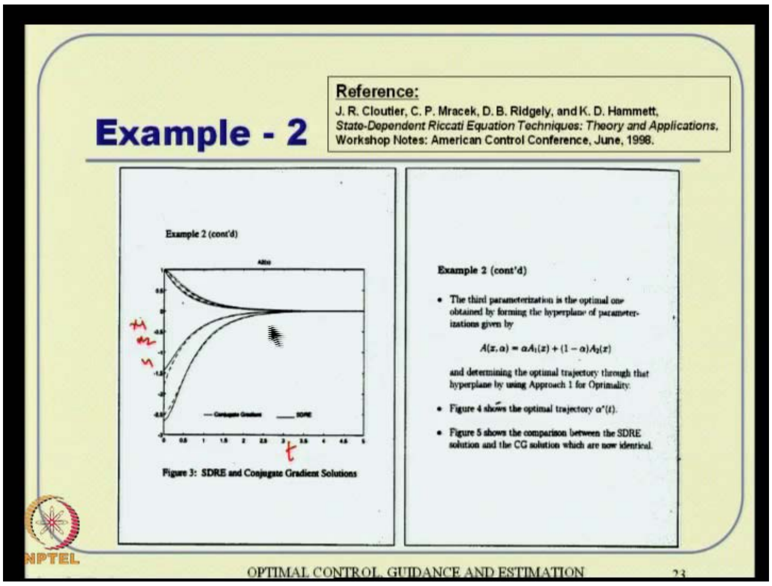
So, it gives us some sort of additional freedom actually. Then this freedom, this alpha can be tuned at some sort of an optimization procedure offline, because this alpha is not fixed online. You can just do it somewhat offline, and then construct an A 3 out of A 1 and A 2. Then operate your SDRE solution based on A 3 parameterization rather actually.

(Refer Slide Time: 41:08)



So, that is the whole idea of extra degree of freedom and all that actually. Now coming, before going on we will have another example, actually two-dimensional example, and all we have this x 1 dot and x 2 dot something something like this. Remember, someone can construct A 1, this way and someone can construct an A 2 that way.

(Refer Slide Time: 41:25)

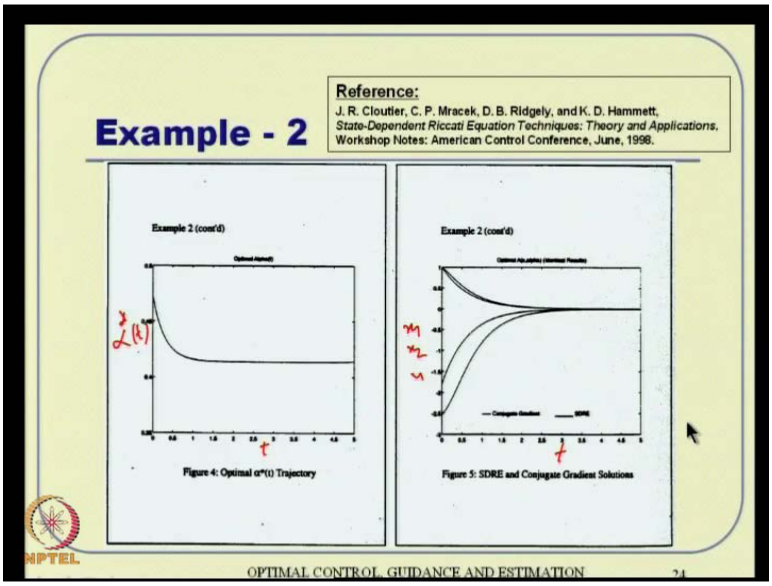


If you construct an A 1 somewhat like this, out of these these non-linear equations and all, then the plots turns out to be something something like this. You can notice they are close, but in the initial phase we have noticed here, little bit initial phase they are not close actually. They become very close to each other as the h time evolves actually. This is this is nothing but the time actually. This is the time part, and this is your both state and control.

So, we have here x 1, we have here x 2 and we have U actually. Anyway, so these are the control quantities. The point here is, even though if to eye it looks very close, but if you look at little bit carefully then initially these are not closed. But with time they become almost a kind of overlapping actually. Similar thing happens for A 2 also. Remember, A 2 is a different parameterization for the same problem actually.

So, again if you do the same exercise and again, if you try to plot it, similar things this is again t and this is both, x 1, x 2 and U, then you can again see the similar wave here. In other words initially, they are not same. Initially, there is some degree some amount of error and only later they become quite same actually, so what about selecting this way. You construct an A out of A 1 and A 2 this like an A 3 sort of thing, but we can consider A and then you operate it based on A.

(Refer Slide Time: 42:55)



Then you see, that it is exactly same everywhere. Even this, what you see is approximately same as time goes to thing and all is just we cannot estimate anything merely from beginning to end actually. That means, it actually gives us some sort of a solution which is very closed optimal, because these are all t and all that actually.

And you can also plot this this. Now, what what has been done here of course, the alpha is considered as a function of time. In other words, alpha is fixed as as we go a long actually. So, that is what you see that is some sort of a alpha trajectory or some sort of optimal alpha trajectory here actually and here is regular x 1, x 2 and and u actually.

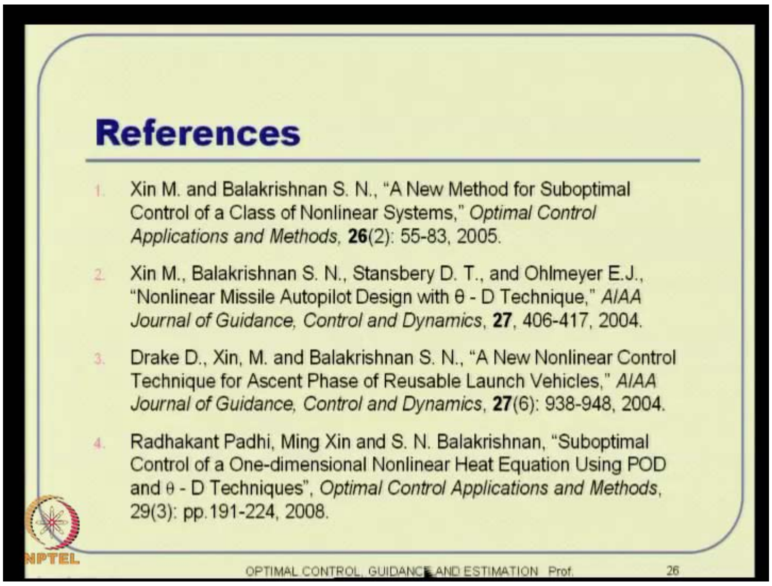
So listen, if if you do this, I mean if the procedure does allow that, in other words, this this tuning, alpha star tuning is not very computationally expensive. In other words, you can do it in closed form and something like that then why not? Actually, we can you can do that online also, and you can also see that very quickly alpha star converges to some some constant value and all that actually.

So, you operate and keep on computing alpha star and then, you compute an A which is a function of the alpha, and then apply this A which is the newly constructed matrix, and I mean apply SDRE on A basically using A. Then this is what the result is, which is very neutral actually that demonstrate that this method or this design approach is a additional degree of freedom using which we can actually go very close to optimality actually.

So, that is all about SDRE control design and more on that I will suggest that you read from references that I pointed out in the beginning. There are many many concepts, nice things both on controller as well as observe a filter and all that actually. Now very quickly, I will give an overview of what is called as theta d's control design which is again an extension of SDRE control design, and some time have this. This this part of it is actually double of y is largely with Ming Xin and and Balakrishnan actually. This Ming was his student, Balakrishnan student and all actually. So, what happens here is? The whole concept of the theta d control design tries to address one issue of the theory is that online solution of Riccati equation is the problematic actually. Otherwise, if the dimensional goes bigger and bigger online solution of Riccati equation is is a concern. We cannot really do that online actually.

So, how do you find circum in that? How do you kind of propose something? Some different technique which will address roughly the same thing same problem or similar problems, but we will not have that difficulty actually.

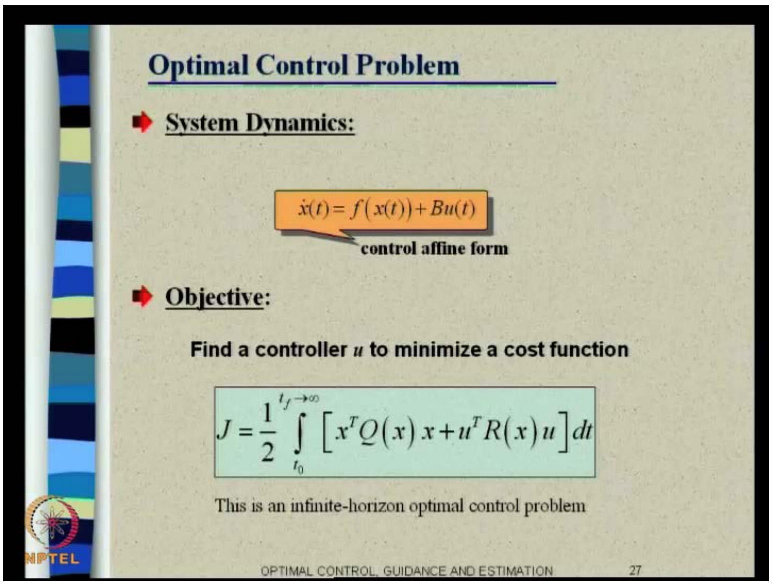
(Refer Slide Time: 45:52)



So, they were some of the references largely Ming Xin and Balakrishnan and all and actually I mean Balakrishnan happens to be my Phd (( )) grade also. So, we had common paper as well with the with the extension of this approach to this distributed parameter system and all that. So, that is a different concept all together, but for the regular method and for for knowing these details of the method, I will recommend this one. This has a lot of all this theoretical resolves, demonstrative examples and things like that which will clear clarify your ideas very well. This has been applied to various various practical problems not necessary only academic problems.

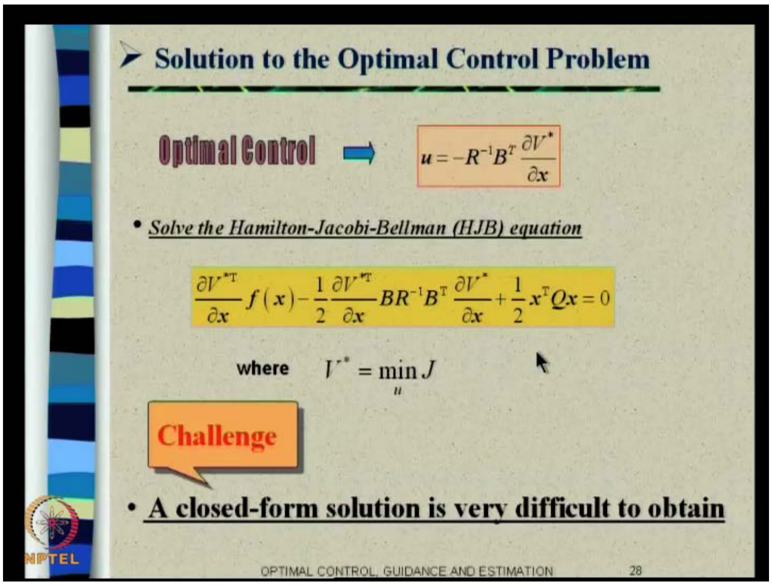
So, you can see some some missile (( )) auto pilot design, and some some reusable launch vehicle design I mean, and then we also have this this extension into distributor parameter is systems and thing like that. It is purely generic and it can be applied to (()) last class of problems actually.

(Refer Slide Time: 46:50)



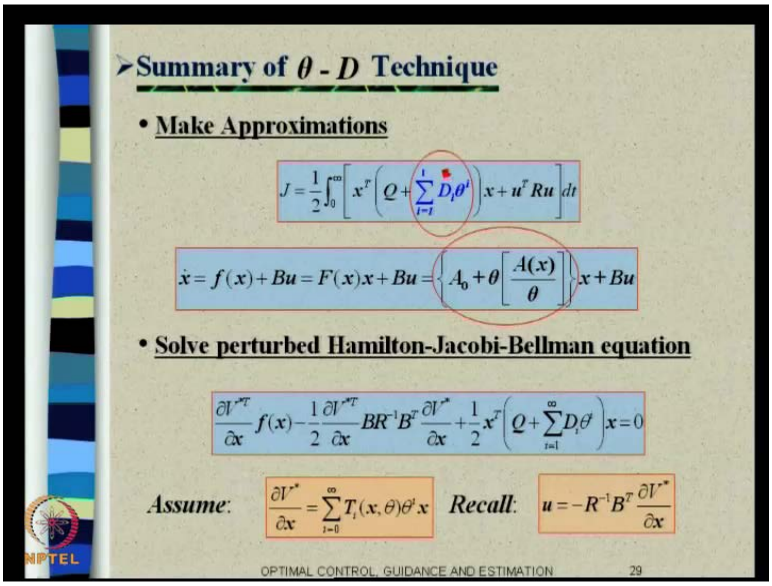
So, let us see what it talks about first. First of all, it talks about fairly sense system dynamics; that means, you have this control affine system. X t is f of x plus b times u, and then you have to find their objective which is again the same, where this infinite time controller and we want to minimize this quadratic cost function - quadratic looping cost function and all that.

(Refer Slide Time: 47:13)



And you know from HJB theory, and we will see that theory little later, that the an optimal control can be represented as u equal to minus R inverse B transpose lambda. But the optimal lambda can be represented as del V by del x also where v star is nothing but optimal cost actually. And this optimal cost satisfies the HJB equation and putting that HJB equation, we can see that this V V star what we are talking about has to satisfy this. Where, V star is the minimum cost actually. That means, if you work along the minimum controller control, that minimize that this is cost, and all that and that is the v star. That will be the result actually. So, solving this equation in a close form is length. In general, it is not feasible. People have to see what alternative things can be done actually.

(Refer Slide Time: 48:01)



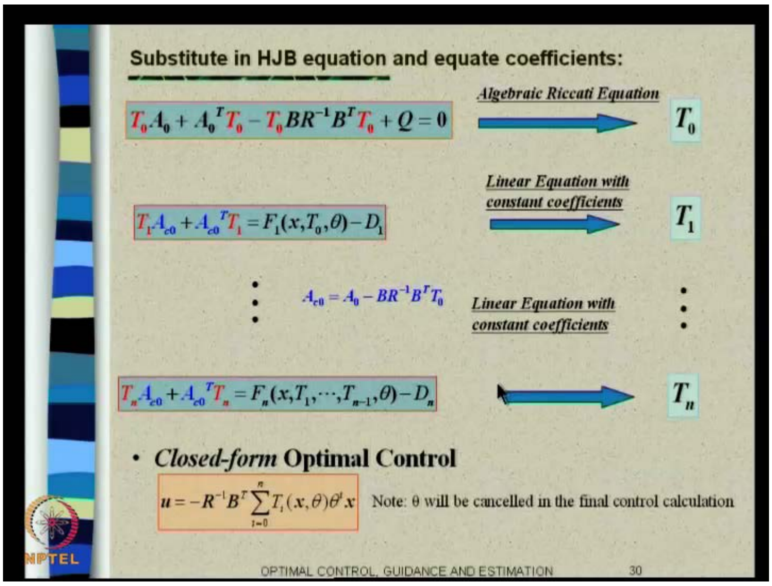
Well again again as I told, I will not give a very detail elaborate explanation, but in a quick overview sense, first of all what you need to do is is write this cost, I mean cost function as Q times some (( )) term as well as the steady equation is something like f of x into x. What you have here is something like a constant matrix a not plus this this expansion actually.

So…So, I think this a small error. This is not equal to (()). What you are doing is protecting the cost function little bit, and and modifying this system dynamics little bit as well actually. Then it tells out that if you if you take it, if you take the other equation and substitute that instead of Q A to substitute to this expansion, and this actually whatever q plus this expansion and this this tends to be like that.

Now, here is the trick. What what is done here is del V by del x which is nothing but lambda actually. So, the del V del x is is expanded into in terms of a power series on this is on some some additional variable theta. So, T I has a reference to be a matrix where the theta happens to be which comes under a scalar quantity and all that actually which is, I mean we like taking the l taking the motivation from the some of the powers (()) at least some of Jacobi equation all were each actually. Will give an example little bit later as we go along, but the, but the difference is long. Sometime people take the least or powers series expand and all. Instead of that, what he what, he done with is lambda or del V by del x means which is vector that is expanded as some sort of matrix time, some scalar time states actually.

So, once you write this and then if you can substitute back here, this expression, you can substitute back here and then, collect various powers of theta. That will you give series and in terms of theta and theta and powers are theta theta square theta q like that, this also give for power power this actually.

(Refer Slide Time: 50:20)

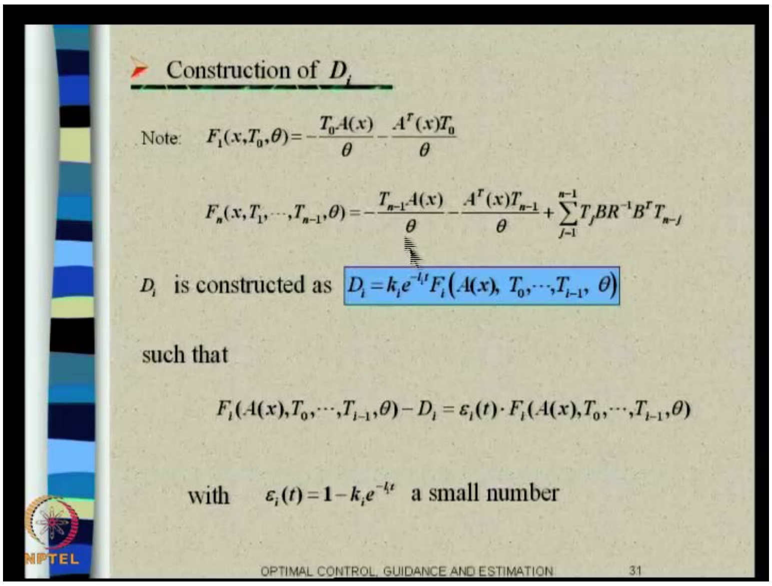


So, you do that and then you collect various coefficients, and coefficients of various powers of theta, and then… So, it happens that, ultimately it will resulting some times of Riccati equation, and this form and it will result in bunch of (( )) equation rather, instead of Riccati equations, in terms of this series response actually.

As a good thing is, this T 0 is a function of A 0 only and even more A 0 is, A is a constant parameter relation. That means, this this Riccati equation what we are talking can be solve of (( )). So, we are talking about one Riccati equation, which can be solve off line and a bunch of (( )) equation that is needed to be solved online and this series, even though theoretically it is infinite series and all that, but it see here and lot of vertical application all that it is needed is about three to four terms in that actually. Very quickly, it is it kind of converse is actually. So, what it it mind sound is one Riccati equation off line solution, and about four to five (( )) equation which requires online solution actually.

And then, this D 1s are also appearing here then all that way. So, what it tells what is the solution procedure tells is something like that. You solve one solution from here and get your T T 0 ready.

(Refer Slide Time: 51:50)



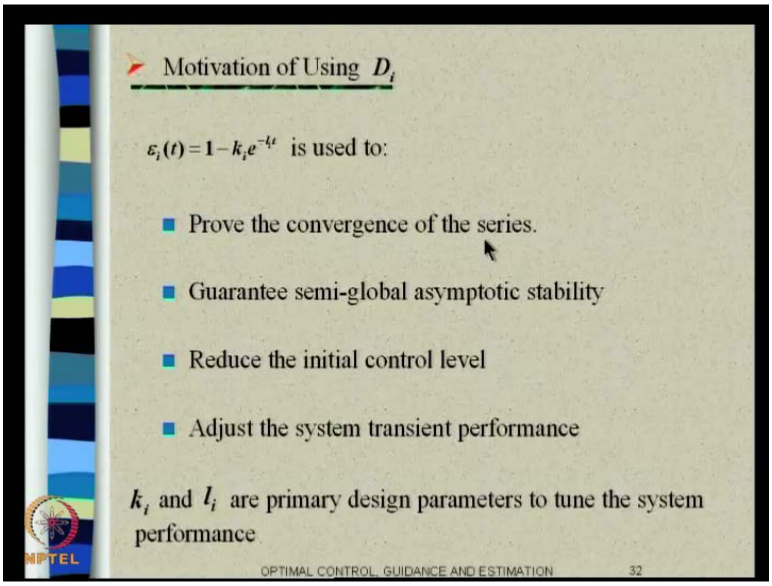
Now, you solve that this one, but you needed D 1 radii also and D 1 can be constructed something like this actually, T i ultimately something like that. So, first you get it T 0 then, D D 1. Once you have D 1, we get T 1 and from there, you got to compute D 2 and once you get D 2, you can solve for T 2 like that, each these approx process like that.

Finally, once you have the solution ready, then then this u can be represented as something like this, because you remember u is equal nothing but R is minus R inverse B transpose lambda which is, these this is nothing but these series actually. So, you can substitute that and get the optimal control solution in close form actually.

So, now construction of D I and you can see and refer for the details what the D I can be solved something like this and even more this does not come naturally, but you are the investigators has put some ki and li here, which is claimed something like one additional tuning parameter actually. So, it it happens that the Q T satisfies some these kind of relationship and where if sudden I, if you take A this way and this equation will be a satisfied and well (( )) I in this form actually.

The claim here is, if you do not do that, then it requires the problems of large amount of control actually and this A, you have to be careful to tune this ki li in such way that control requirement will be solved. However, obviously, there are, whenever there is additional tuning (( )) there are also like a scope for improvement on all all. So, if you somebody is interested to see further investigate and then come up of with systematic procedure of exhausting this ki and li that will be nice actually.

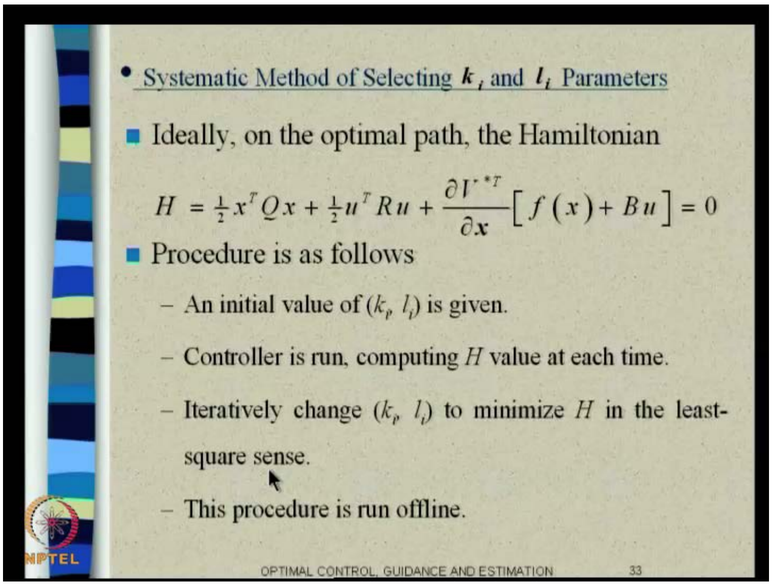
(Refer Slide Time: 53:34)



This is what I told. This this solve is used to prove the convergence of the series. It also it guarantee some semi global asymptotic stability and all. It reduces the initial control level that is what I told you already, and it adjusts the system transient performance is has been actually.

However, remember this ki and li did not come from this series expansion and all. It was just thought about it later. And once you heard this additional tuning parameters, these are the advantages. But also remember that, right now, as per as my knowledge is concerned, this reference is just tuning parameters. However, somebody has to select these values in trial and error. If any of you who can come up of some sort of a systematic procedure, just in these ki li, that will be nice actually.

(Refer Slide Time: 54:21)

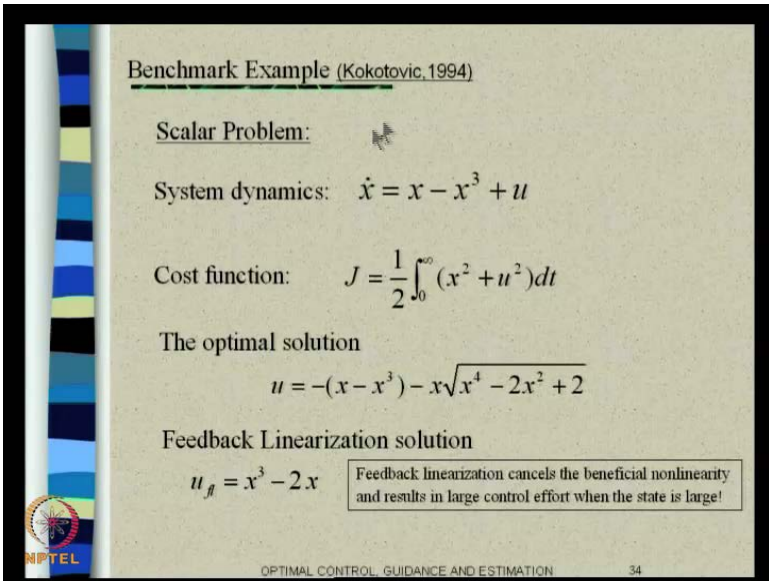


Now, well, there are, I mean, I am not telling that the systematic procedures are not available, they have investigated themselves, actually explored that this, but again there is scope of improvement on top of that actually. Now, one way that (( )) propose is something like that, ideally on the optimal path, Hamiltonian has to be 0. We know that. On optimal path, remember one of our basic earlier lectures, we told that in general, an optimal path, Hamiltonian is only a function. Hamiltonian is constant actually that what it is. So, it is not an exclusive function of time and in addition to that, in a regulative prolong if this is a constant and finally, it goes to 0 then, it is 0 everywhere.

Because at x goes to infinity, then all straight line control goes go to 0 and in Hamiltonian which which which will in respond, will all go to 0 actually. So, and I will see, I mean this is a kind of effect actually, in other words the entire optimal path Hamiltonian also v 1 0. So, you try fixed ki li which is (( )) happens actually.

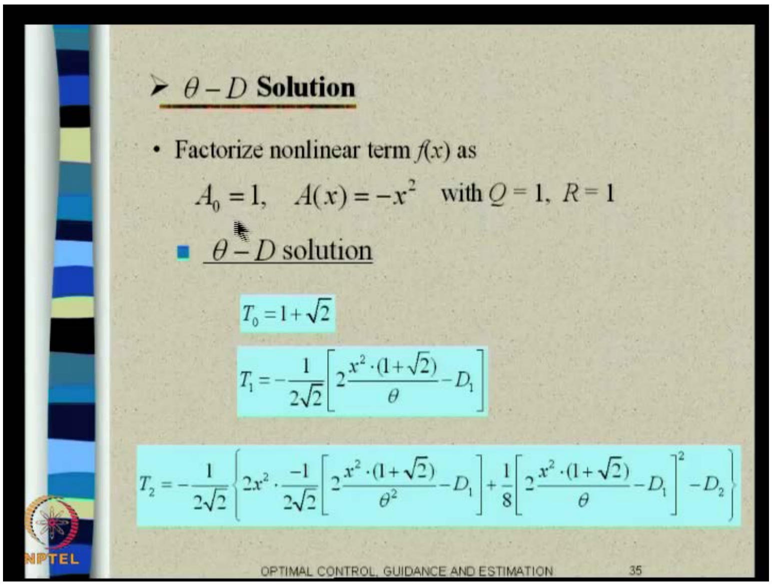
So, what it tells is, is selecting the initial valve of ki li, something like guess value basically. Once you start running in a control, then we can compute your H and iteratively, take near to change this ki li to minimize h in the least co efficient actually. So, this procedure can be done off line to select a process, select the proper value of a ki li, but again when you talk about off line value, remember, actually it depends upon the initial condition of the state also. So, we start with yesterday, do this and again start this differentiate may not converse therein. So, your talk about (( )) minimize sort of things in other words take lot of initial condition in try to kind of run it run low trajectory and then, try to do some other iterative change on all other actually. So, anyway of that approach which already there. But again I told it somebody same thing about better approach, it will be given good actually.

(Refer Slide Time: 56:54)



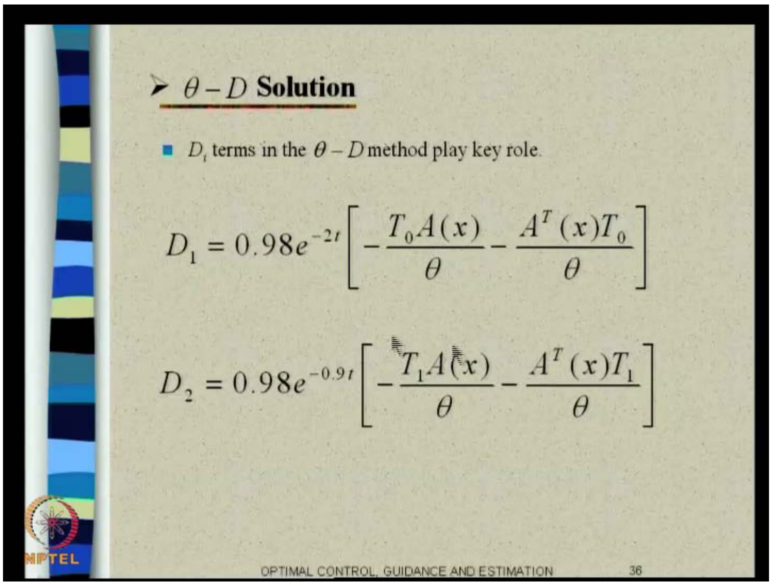
And again before stopping, this is the standard same bench mark example problem where, x minus x cube plus u. We just discussed in SDRE frame work also. Cost function is again similar, and we also know that is close form solution for that. We discussed about that and one of the feedback solution feedback linearization solution can happen to be like this and this all. Ultimately, at least it comes is the beneficial non-linearity, and it leads to these last control effort when the state is large actually where does the linearity have been… Now, the question is, does theta will also be shown the some sort of result actually.

(Refer Slide Time: 56:56)

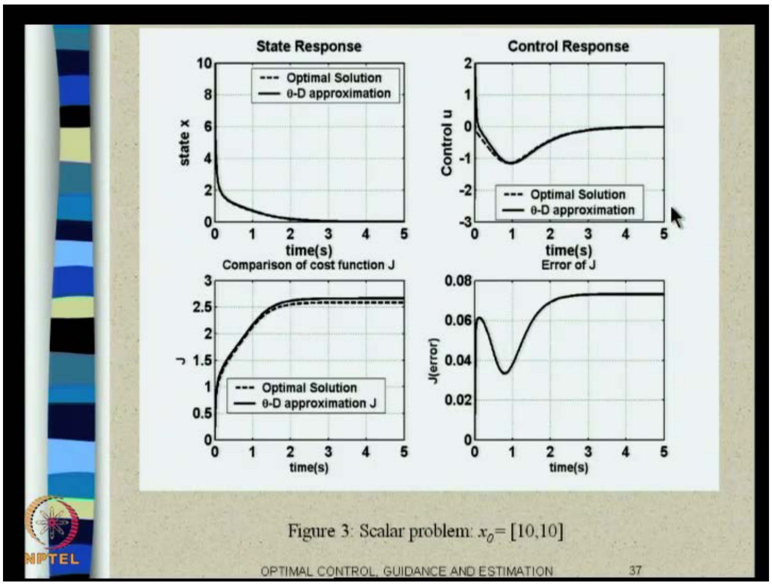


It happens to be yes. We can decompose it very easily, you can talk about A 0 is 1 is remember that the A metrics in SDR was 1 minus x square actually that gives as A 0 equal to 1, and then well I can put it that A 1 actually of A 1 is minus x square, and then think like that actually and with Q equal to 1 and R equal to 1 actually. So, theta D solution turns out to be something like this, t does not also have to be like this. T 1 turns to be like that, T 2 turns to be like that and D 1 and D 2, we can complete something like this.

(Refer Slide Time: 57:34)

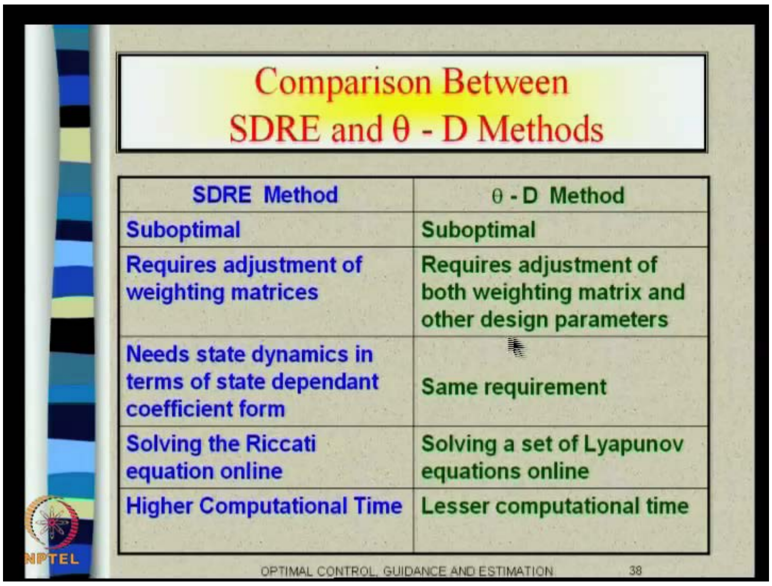


(Refer Slide Time: 57:38)



And then finally, we can see the results are very close to each other actually; that means optimal controls solution in close form, and theta is the solution is not very badly actually in sense.

(Refer Slide Time: 57:47)



So, in summary there is a small comparison thing that you can think about. On one hand, we have A SDRE method and on the other hand, we have theta D method. Both are applicable for control affine systems only and do not forgot actually, and both lead to sub optimal solution. This require adjustment of weighting matrix and Q of x r of x actually, but this this method it requires the adjustment of the weighted matrixes, and other design parameters as well.

Here, once you adjust the waiting matrix is all Riccati equation solution. We do not have control on that actually. It is just a procedure that you to follow, but you have an addition to selecting that key of key of (( )) we have to select additional design parameters. That means, how do you decompose into that series a not a 1 and all that actually, when the ki li and all that way.

So, it gives us lot of design parameters. One way it is good because you can tune the design parameter. But in other words, it also where because too many design tuning parameters means, design gets confused and it will lead more optimality and things like that actually. So, it is not really very good if you have too many dimensions actually occurs. So, in SDRE method, it it means the state dynamics in terms of state dependent coefficient form and as I told both these also requires that.

So, in that sense, this not advantage actually. But this SDRE method demands that you solve this Riccati equations online, whereas theta D method does a major advantage here, where it demands that you solve a set of Lyapunov equations online. Remember, Lyapunov equations are linear equations actually. They are non non-linear equations actually. It can be solved very quickly.

So, in general it is it requires a little bit higher computational time whereas; theta D method can done in less computational time actually. So, this is, thing as a summary chart and probably those appear interested more, you can read more literature and then, get familiarize yourself pictorially. So, with that I think will stop this lecture and thanks a lot.