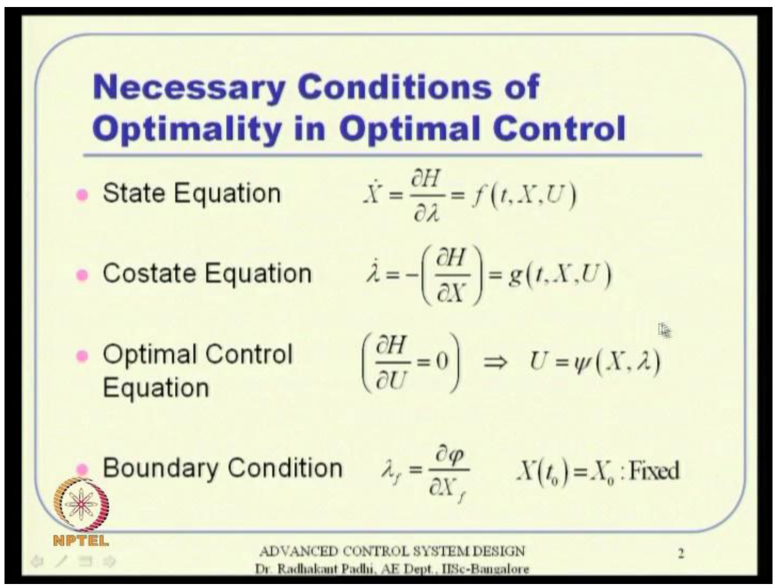
**Advanced Control System Design**
**Prof. Radhakant Padhi**
**Department of Aerospace Engineering**
**Indian Institute of Science, Bangalore**

**Lecture No. # 26**
**Classical Numerical Methods for Optimal Control**

Hello, everyone will continue with our lecture number twenty-sixth this time. So last lectures we have been we have reviewed (( )) of various and concepts, as well as. We have seen how to use it for optimal control formulations and that we discussed about some examples and all and we also, discussed why those problems are typical to solve in general and then, there is a necessity for solving those in a computers norm actually.

(Refer Slide Time: 00:53)

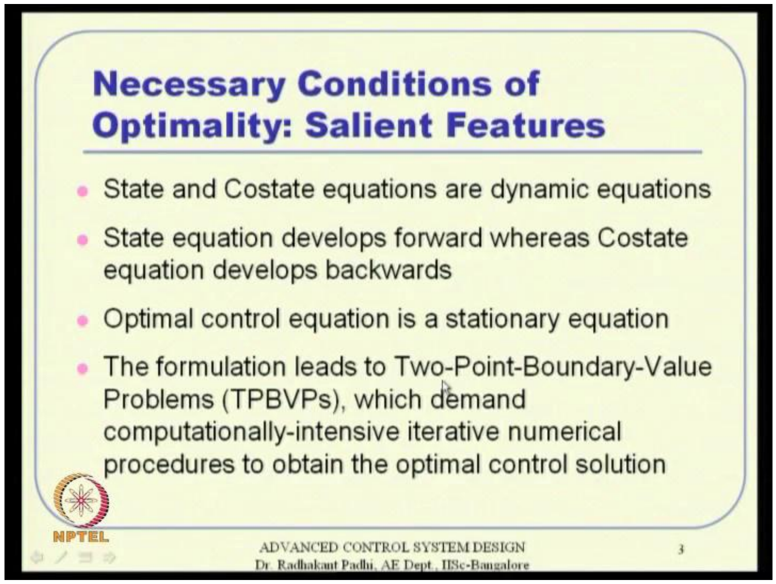



So, we will see those couple of those traditional classical computers new methods in this lecture. The summary is mean this summary of this necessary conditions of optimal control is something like that we all derive it last time. First thing it shows some necessary conditions optimality is three major equations. One is the state equation Costate equation and the optimal control equation and we discussed last time that the state and Costate equations are dynamic equations you see a dot there, this is n dimensional differential equation this is also, n dimensional differential equation.

So, these are the differential equations, where as this optimal control equation is normally a stationary equation not a static equation. That means, (( )) you can solve u as a function of x n lambda from here. Either this can be solved explicitly that means, the nature of this equation allows you to solve view explicitly as the function of x n lambda or it may so happen that you may also, need some numerical method to like Newton option something to solve this equation in a computers (( )) in a numerical way also. So, they, but the main problem here in this entire difficulty comes from the fact that this state equation boundary condition is given at initial time, say t 0 is available to us where is the boundary condition for the Costate is available only at final time.

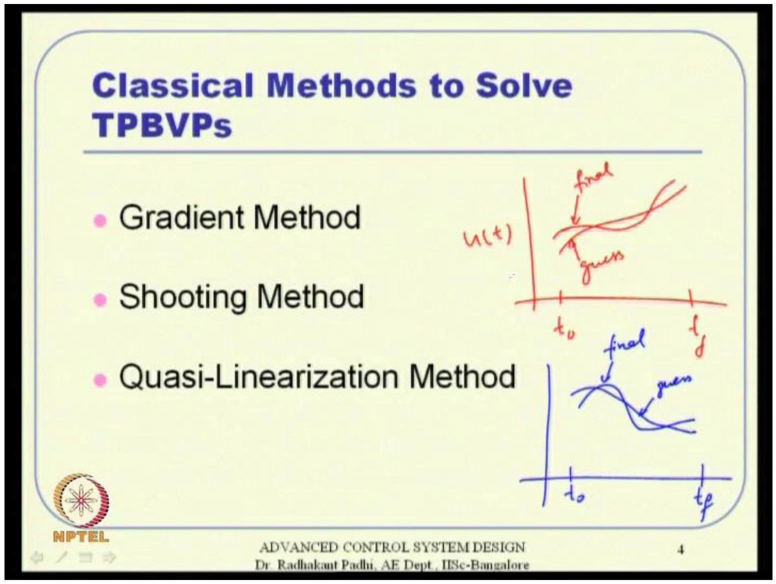
That means n boundary conditions are available at initial time where as n boundary conditions are available at final time and you have at 2 n dimensional differential equation basically. So, that means we are left out with the with a condition where, we have to discuss these two point boundary value problem. If these differential equations are two n n conditions are available at the initial time and n conditions are available at the final time that is the major problem. Suppose, we knew every boundary condition at one point of time either initial or final or somewhere in between think like that, then you could have simply integrate to the differential equations either forward or backward. But since we do not know that how far the things are here and how far the things are there, we settle in it is numerical intensive procedures to solve these. So, let us see how do we do that.

(Refer Slide Time: 03:07)




So, these are some of the things, I just discussed the salient points of these necessary conditions optimality that the state and Costate equations are dynamic equations. State equation develops forward where as Costate equation develops backward, the optimal equation is a stationary equation you may or may not able to solve it in a in an explicit manner. If you cannot then you can again use numerical procedures to solve that and the core of the point is this entire formulae on least two this so called two point one boundary value problems, which demand computationally intensive iterative numerical procedures, to obtain the optimal control solution so, we will see what are those.

(Refer Slide Time: 03:50)




So, in this lecture we will study three classical methods, these are very heavily used in a many industry especially, in aerospace industry. And these three methods either own advantage drawbacks something like that, I will not talk too much on those aspects we will see how to use these thing actually, how can you use these methods in practical problems like that. What are the methods, what are the procedures and all that? So, first we will discuss about what is called as gradient method, second we will talk about shooting method and third do we will talk about something called quasi linearization method. So, probably they are of like increasing order of complexity what then once you understand that and try to implement then, probably the more complex (( )) is that means, the convergence properties are better you think like that so, just keep that in mind.

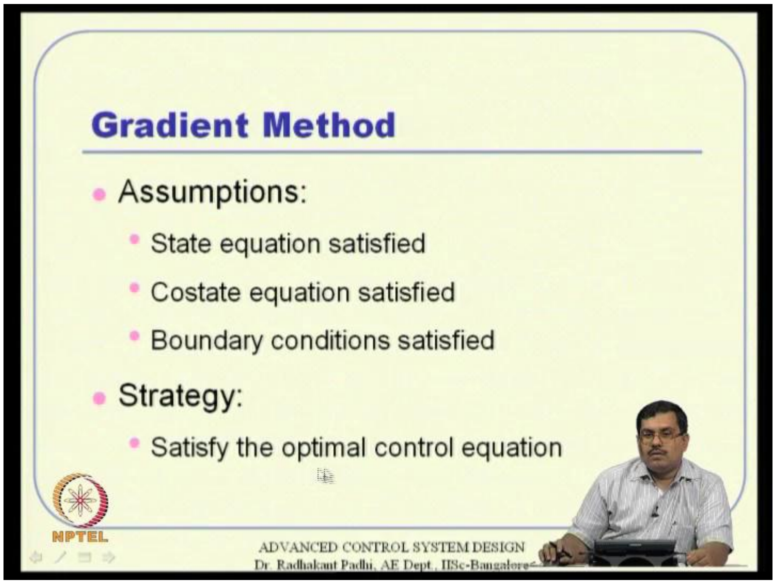
So, gradient method before I move into any method by the way all these methods will assume that we have to guess a missile condition for the control that means we really need to guess a control history from t 0 to t f. Then we need to update that control history as that (( )) process actually that is the that is the core of those of in any method. So, you need to really guess some control, control is still that may that is in the typically not trivial by the way that means, if you really want to guess a good control history. Then you should understand the physics of the problem first and based on that some intuitive design or some I mean some heroistic design you can do to get a good control history.

And also remember these are necessary conditions and then you to be slightly careful about that in other words whatever, is your guess history that you are talking from t 0 to t a, around that the control may stabilize so, let us try to understand that consider a little. Suppose, you have this time t 0 to let us say t 0 to t f and you have this u of t let (( )) a bit scalar so, this is let us I have guess this history basically this control history. Now, what will happen here is, I if I guess this control history most likely my final control is also going to stabilize around that basically. So, this is let us say my guess and this is my final now suppose, I guess some other thing for the same problem let us say suppose, is seek guess some other thing so the same problem t 0 to t f.

Let us say this time I guess something like this then my solution convergence may happen something like this. That means, this is let us say my guess, guess history of the control this is my final. So, just be careful that what while you are guessing you are your guess control also realize some certain (( )) certain science or at least stabilizing control or some other control agent that is not optimal think like that. And then whatever, your guess history you are feeding to the control design most likely you are going to get a solution around that. So, guessing that the procedure may guessing procedure itself is a is a irestic science by itself. So, be careful about understanding the physics of the problem, while applying some of these methods try to utilize some guess in the beginning and then you update on that.

For example, missile gradient problem in aerospace typically the gradients problem, I mean very a classical gradient law is proportional (( )). So, if you really want to find out an optimal guidance and non-linear optimal control base theory based guidance law, then probably you would like to guess a latex history using n guidance and then using that latex as a guess value you can try to update your things it is update your latex history. They are subjective to the problems actually just keep that in mind, let us let us proceed further we talk about gradient method first.

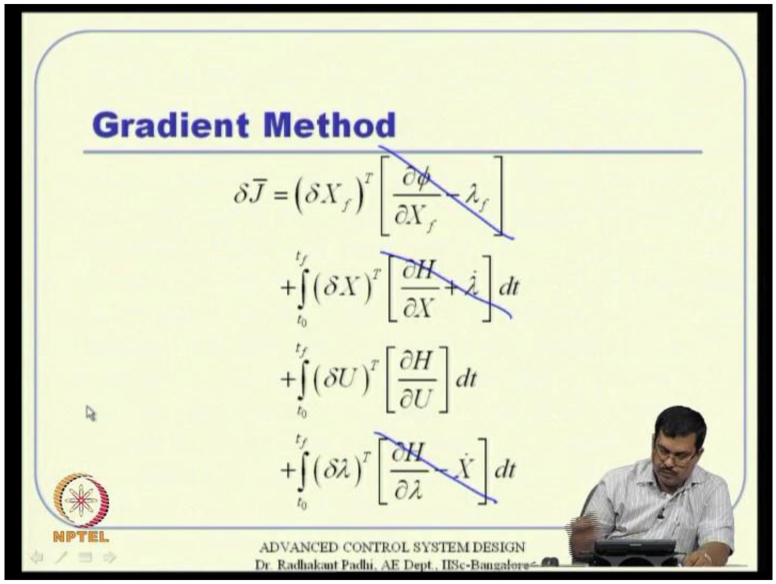
(Refer Slide Time: 07:48)




In the gradient method out of these three equations that you are having includes so, what you are to what you are assuming is that state equation Costate equation optimal control and think like that are available. So, what you are assuming is that out of all the conditions and remember, if all if any of the condition in not satisfy then it is not optimal control. So, out of all these conditions we will assume that state equation is satisfied Costate is satisfied, as well as boundary condition is satisfied, what is not satisfied is this one with respect to the guess solution actually.

So, we have guessed a control history and with respect to thus that guess that is certainly not optimal we want to make it optimal, but with respect to that guess we are assuming that state equation Costate equation and boundary conditions all are satisfied. And then what is not satisfied it is only this condition and hence, that is a non optimal control solution. We want to make it optimal by successive enforcement of this condition. So, that is what our strategy so, let us see how do we do that? So, that is what I told state equation is satisfied Costate is satisfied boundary condition is also satisfied. So, what is the strategy? Strategy is successive do some successive iterations so that or optimal control equation will also be satisfied actually thus to thus the strategy there.

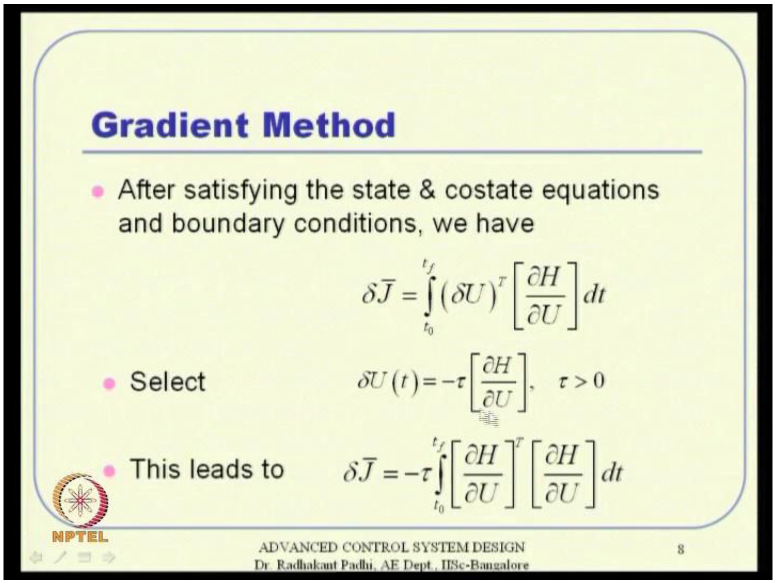
(Refer Slide Time: 09:09)




So, now obey to the derivation part of it what we discussed last class so, we just we derived this expression del J bar J bar is the augmental (( )) that and we wanted to make sure that this is equal to 0 and hence, this all these coefficients needs to be 0 and that is how we derive this necessary conditions (( )) that.

So, now what you are telling here is because Costate equation is satisfied let us say because our assumptions one by one let us say. So, the boundary condition is satisfied so this is zero this is gone state equation is satisfied so this is gone this is x dot equal to of t x basically. And Costate equation satisfies that is also gone so, what is left out is only that part that is that is left out, guess only this part that is left out actually so, let us see how do you makes (( )) that.

(Refer Slide Time: 09:58)



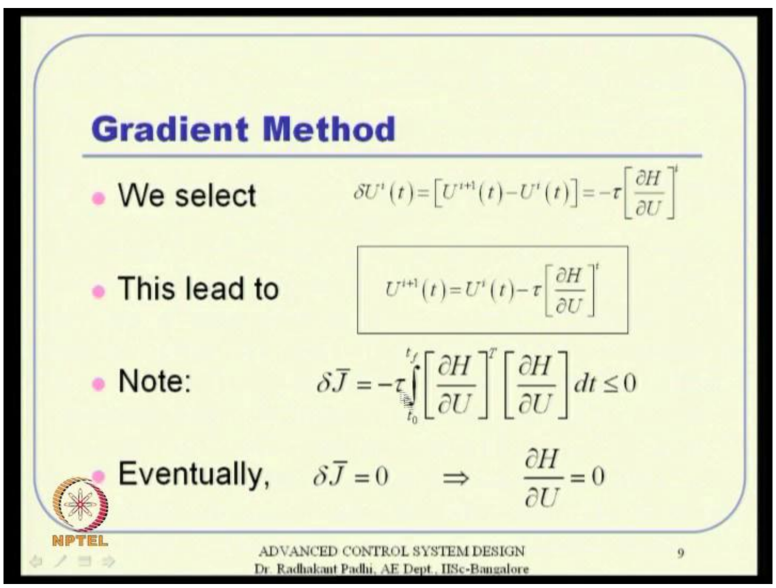

So, what you what this gradient method tells is once this, I mean once these conditions are satisfied we are left out with del J bar is that and you want to make sure that del J bar decreases in successive iterations density. And remember J bar is nothing but a positive quantity actually in this J bar itself is a positive quantity so, you want to make sure that the del J bar is negative quantity so, that successive iterations decreases and ultimately stabilizes somewhere actually. So, J bar turns out del J bar the first variation of J bar you are having only this term now, let us select that del u is something that we need to select that is the one that we need to update you remember that.

So, u already we have some u basically this is what is your is del, I mean by definition this is what is del u basically this difference actually what you see here. So, this difference whatever, see whatever thing you are looking this difference is something that is that needs to be found out actually, that is nothing but del u bar I mean del u basically. So, that del u is something that we were interested to find out and now, let us we let us say propose that the del u will selected that way. So, negative of del s by del u with some learning factor tou which is greater than 0 basically.

Then what happens? Then if you plug this one by care you are left out with some term like that which is certainly a positive quantity del s by del u is a vector so, it is something like extranspoze sort of thing so, it is a certainly a quadratic term inside. So, this quadratic term inside is certainly positive, because every term is square square addition of squares. And then, if you have a negative sign as you (( )) is that how you is your first quantity any way so del J bar becomes negative. So, you have a positive quantity for which the del J I mean J bar is a positive quantity your del J del J bar is enforced to be negative. So, that is how we mix your that the J bar decreases actually so, we are talking about a minimization problem here so, if it is a maximization you do the reverse thing.

So, for a minimization problem this is the variation that you need to select actually that is that any point of time t k let us say then I will evaluate del s by del u at that one that time t k and that is my del u k. So, I will update my control based on this del u k basically that is how it happens there.

(Refer Slide Time: 12:31)

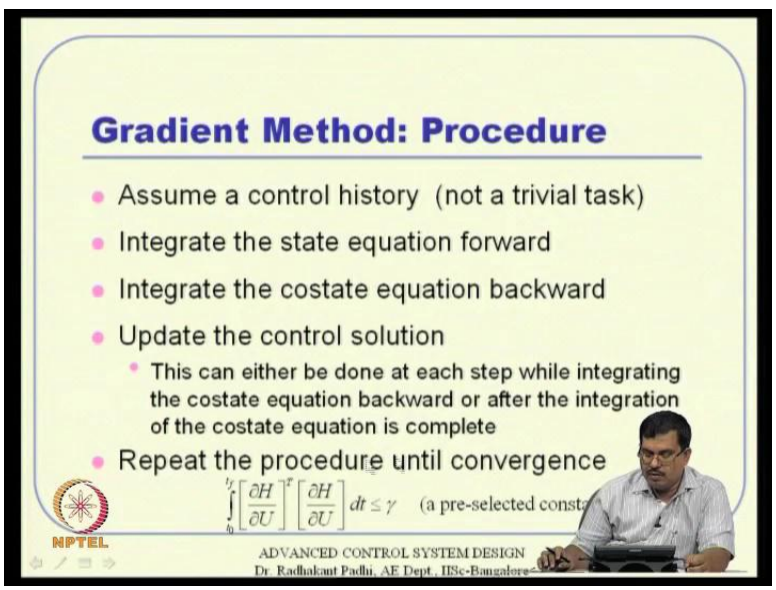



So, that is what we are doing here that if you observe this expression well del u y let us say that is what it ith iterations sort of thing. It is actually like by definition it is del u I nothing but u i plus 1 minus u i that iterations this is nothing but that minus tou times del s by del u iterations of course. And then these lead to this condition that u I plus 1 is nothing but u I minus this thing actually. So, and this is also remove this is negative gradient direction actually this is also like in compatible with what we know in static optimal and numerical methods actually steepest decent method and then also like that is also called gradient method.

So, this using this gradient, gradient is what gradient of hemaltanian with respect to control so, that is what you are using the variation change in control that you are looking for is nothing but negative of some learning factor multiplied by gradient of H hemaltanian with respect to U and this is why his gradient method comes. That is what you need to do and ultimately del J bar turns out to be like that which is guaranteed to be less than equal to 0 and eventually del J bar, because it is a success successively decreases and J bar is a positive quantity so, ultimately del J bar will stabilize at 0 and that is what we want for optimal control. Remember once you how to derive this ultimately we want this del J bar to be equal to 0 that is what that is what we ultimately want.

That is why because these are already these are already 0 we are receiving that these are 0 anyway and that is what we are assuming the we are ensuring that is also 0 and hence, this everything become 0 so, that is what we are doing this.

(Refer Slide Time: 14:25)




So, what is the procedure now? I think in gradient method we need to assume a control history again it is not a trivial test we need to have a good understanding of the physics of the problem to come up with this good control history. Here to start with and then once we have this history you in n place then, what do you do to integrate the state equation forward, because remember initial condition for the state equation is known to you. Now, control is also known to you so, as if you go back to the state equation that we started once we know your full control is to U F t and you know a initial condition for X 0 this equation can be integrated forward.

So, you can keep on integrating this from t 0 to t f because every time you have a control value known to you basically. So, the at t equal to t f you will find some X of value and based on that X f value you can evaluate this fellow lambda is equal to del pi by del X f. Because pi is a function of t f X f only so del pi by del X f is available to you and t f is at anyway available and by integrating this equation forward you also know x f value. So, this equation can be evaluated now. Now, what happens we have, because you know the final condition final boundary condition for lambda you have this differential equation anyway so, you can integrate this equation by (( )) actually.

So, once you have this equation can be integrated forward and this equation can be integrated by (( )) actually. So, then while doing this backward integration you have you have two choices you can either keep on updating your control based on this whatever, we were proposing here based on this update what you are doing. And you can you can update this control first and then integrate one step backward and then keep the process go in that way or sometimes people update the control history fully, from t f to t 0 you can integrate the entire lambda history first and then update the control history everywhere.

So, that is the two choices with you actually, let me summarize again the we use to start with a control history guess so you have this X 0 mount to you so, then integrate this state equation forward then you have X f available so because based on that you can evaluate lambda f. Now, once lambda f is available you can in principle evaluate integrate this equation backward remember this control history is available to you X history is also available to you by now. And then this is also lambda history also is getting available, because you are integrating backward by the way there is a small print mistake this is also a function of lambda engine (( )).

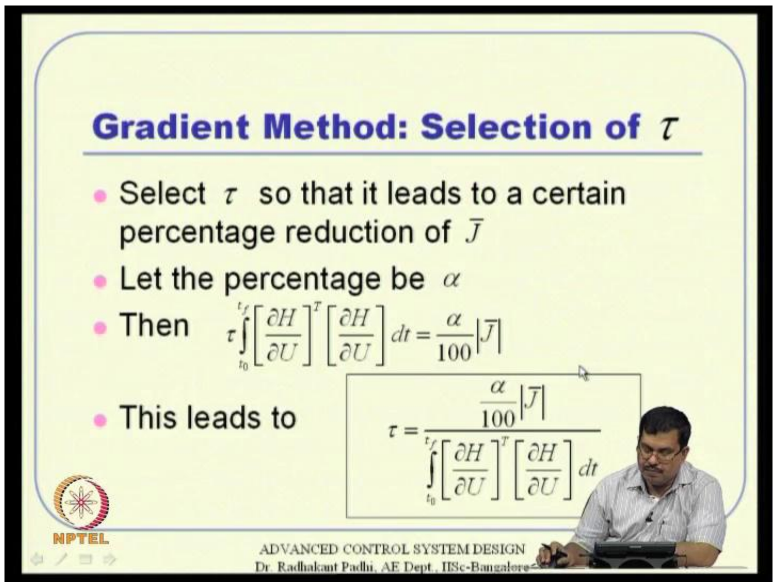
This is a it is a function of t x u lambda everything actually so this part. So, that is what so you can a once you know lambda f then this equation can be integrated backward and you can integrate. The entire lambda is to backward from t f to t 0 and then, think about into your updating your control history everywhere, from t 0 t f or we can do it one at a time in other words you will you update the control once and then integrate the Costate equation one delta d, update the control history again at that particular point of time then integrate it one more delta t like that.

So, the choice the is based on the (( )) whatever, you want to do you can do actually so, this is what you are doing actually. Now, this is gradient method so this is all that procedure that I am writing here or written here and integrate the state equation forward and then integrate the Costate equation backward and update the control solution. That means, is a again this can either be done at each step while think integrating the Costate equation backward or after the integration of the Costate equation is complete so, either the choice is yours. And you have to repeat this procedure until convergence until convergence obviously, you cannot expect that del H is by del U is exactly equal to 0 you have to stop somewhere in a very small tolerance in the sense.

So, this is one of the can we did del H by del U transpose time del H by del U is a quadratic term integrated from the t 0 t f is that is entire scalar quantity should be less than equal to say times pre selected constant small constant value gamma basically. So, that is where you stopped and then you (( )) optimal control. Now, this is there is a small (( )) here in this small m b b t how do you select this value tau actually now, the idea is tau can be selected either pre selected in other words if you this needs to be typically 0 and one it is never greater than one also. And in sometimes you (( )) that very small value greater than one or initially, you can start with a larger value and then while you go towards optimal value this tau needs to be reduced.

So, that means, it if the convergence properties are fast something (( )) that actually now, one can also think I will fix a small value of tau and from the beginning to end then what happens is you have a slow convergence process convergence is not fast. And if you have a larger tau value then, it jumps it oscillates around the optimal (( )) actually it never converges in a very good way. So, one procedure tells us that that tau I can select initially is slightly higher let us say point nine or something then gradually I decrease yes my iterations proceed. And then finally, I will see that that condition is met this tolerance condition is met (( )) each other that is the procedure that is the strategy.

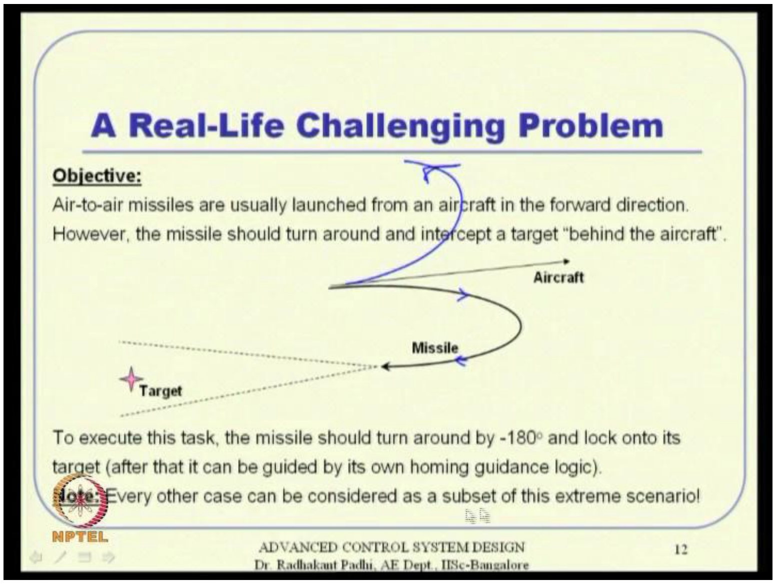
(Refer Slide Time: 20:03)




Other approach other idea is something like these you can select a tou so, that it leads to a certain percentage reduction of J bar at every iterations. That means let us see let us select a tau or you can aim to select a tau that, every iteration when I do then this J bar value is going to reduce by certain percentage let us say 10 percent 15 percent like that. So, that led the percentage we have number of alpha then what it tells you these are the del J bar what is left out here, that has to be equal to certain percentage of J bar that you already have. Because that means alpha y entering to that actually so, that way you can compute a tau based on this equation you can solve for tau from here and that is your tau value.

So, sometime this procedure helps you and we are all subject to the non numerical procedures will also contain some, some degree of art which are problem depended. So, as you work on this any of these problems you will get more and more experience then you will (( )) feel how to select a (( )) value of alpha and tau and think like that.

(Refer Slide Time: 21:13)




So, that is the procedure now, let us go through a little real life challenging problem out here, remember last time we have seen some classic class rooms sort of text book examples and all that. Now, this is a problem that I have taken from literature so, problem is something like this. Now, what happens is normally in air to air combact we are like you know in space problems sort of thing you have a career aircraft typically you launch a missile for a target that is in front. So, that means, if your target is somewhere here normally you will you launch this missile somewhere here so, it is it is going to go in a very far away it is not very close anyway. Then what you do is the aircraft takes a turn and goes away actually these does not go to the target, because target can (( )) otherwise.

So, that is normally what is done what you are interested here is not like that, let us say the target itself is a is in the backward hemisphere it is not in the forward thing. So, how do I what I can career with sensor in my aircraft which look both forward as well as backward I can set up a set of sensors actually, which can identify that there is a target in a backside also, let us say an (( )) target an aircraft or something like that. Then the idea here is if I take let us say like I can take half of the missile on my wing the aircraft has only limited space so, you can take half of the missile which look forward which can be launched forward and half of the missile which can be launched backward.

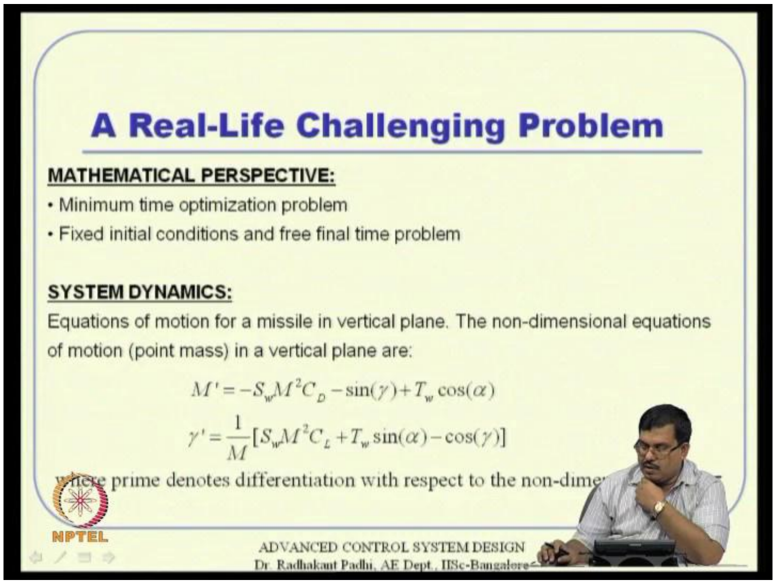
That is the choice, but once you have that choice to the comeback capability reduces, because you never know how many targets are in front and back. And if is a backward can attack only backward and forward can attack only forward then certainly your capability is limited. So, what you nearly need to do is you will take the all the missiles in the frontward direction only in the wing and then depending on the situation we are going to launch those missiles in forward direction only so, launching direction will be say at only forward.

Now, if the target is backward then this air launch vehicle is suppose to turn back an attack actually so, once it turns back then only it is own seeker and it is own detection mechanism can look at the target, until that it will not be able to look on to the target. So, the first primary task is to mix air that this one rotates by 180 after launch missile in the aircraft after launch the aircraft will go that way let it go that way and the missile will come that way and then try to at attack the target. So, that is that is the that is the problem actually so the first thing what we want to do is after launch the missile should turn by 180 degree that is what our main aim and usually, when we talk about missile guidance problem we are more interested in turning of the velocity vector not the turning of the attitude itself.

But if once we assume that there is a small angle of attack and all that then velocity vector and attitude are almost parallel to each other. So, they are same directions same so learn to discuss too much on that what we are telling here is we will launch this velocity vector in one direction, these are all about velocity vector only it is nothing do with missile orientation (( )) and attitude is not considered here. So, we launch in forward direction and then this is suppose, to turn back and see the target after it reaches the target problem is over anywhere. So, our problem is to turn the vehicle from 0 degree to 180 degree after that how it engages and all is a separate problem it is a problem by itself.

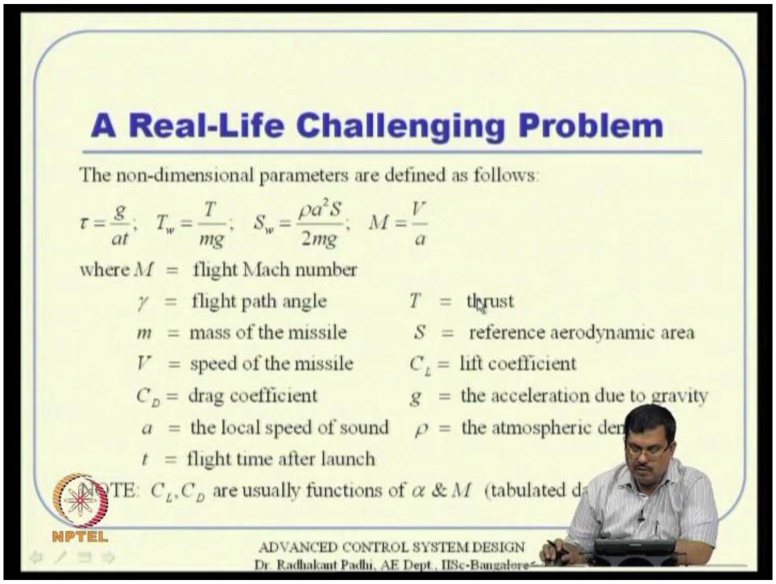
It was able to do that anyway test primary test is at 180 turn it by 180 each there typically want to turn it by minus 180, but we do not want to because this air launch missiles are typically carried under the wing actually it is not carried over the wing. Under the wing then the way aircraft was that way then certainly the it is a safer that the missile turns at the downward direction, otherwise if there is a chance that it may hit your own aircraft. So, obviously you want to (( )) that so it has to turn and turn it by 1 minus 180 degree and then lock on to the target actually after that it can be guided by it is own homing guidance logic actually. We can also say that every other case can be considered as a subset of this extend cases if any what is (( )).

(Refer Slide Time: 25:46)



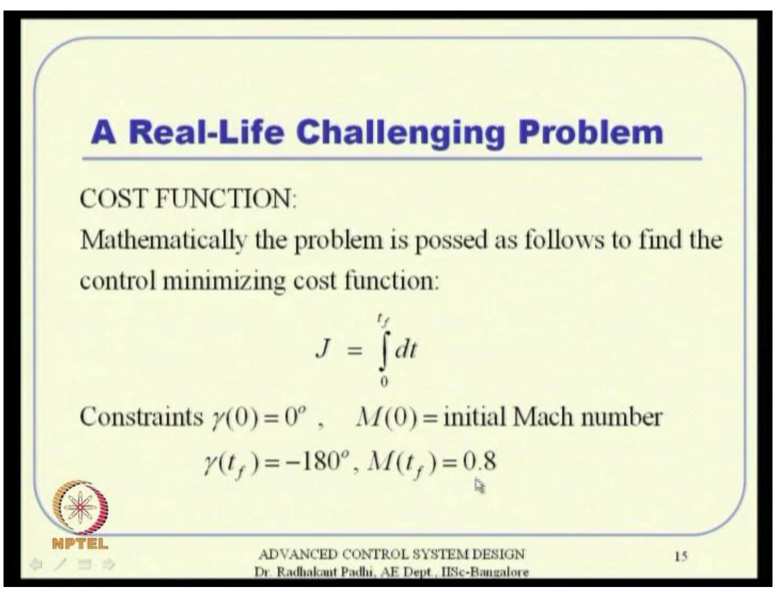

So, this problems are I mean given in detail here so, what you are telling is if primarily interested in mach number and gamma, gamma is the flight path angle mach number is velocity of vehicle divided by speed of sound in that medium is equal to (( )). So, we have this typical loss here and also we know that this is this questions are given in terms of non dimensional quantity even including time, time is also non dimensional zed so non dimensional parameters are like that.

(Refer Slide Time: 26:13)



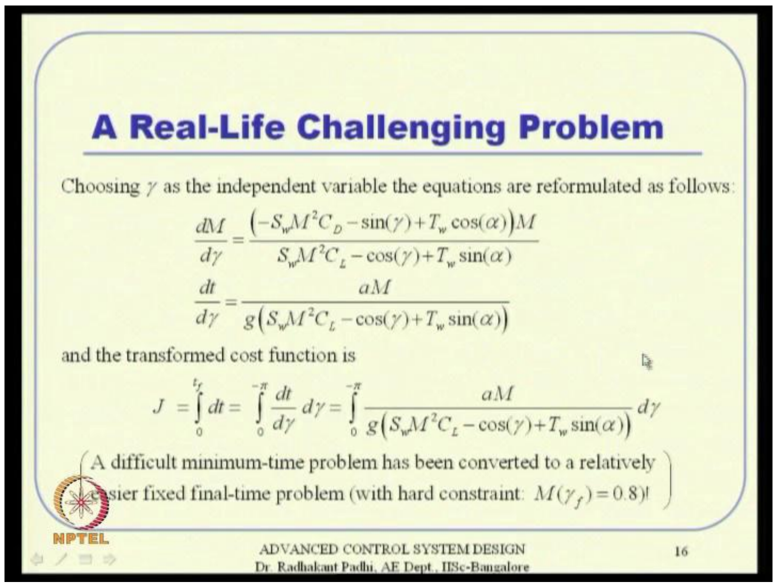

Typically this non dimensional things helps us to write generic formulations actually that this is not a very specific to missiles to anymore that particular missile anymore. All the variables that you see here are all generic non dimensional zed quantities. So, what is the idea here now you cannot waste your time by twile turning so, these turnings should happen as quick as possible so certainly it is a minimum time turning problem.

(Refer Slide Time: 26:46)



So, that means my cost from sun is like this so my t f minus t 0 in t f minus 0 or whatever, is to be minimized. And the constraints that acts on the vehicle initial flight path angle is 0 final flight path angle needs to minus 180 initial mach number is known to us then let us say the final mach number we want it is 0.8. So, we do not want too much of velocity drop either while this turning happens, the velocity drop should not be very high actually so, we want specific final value of the mach number that 0.8 actually let us say.

(Refer Slide Time: 27:21)



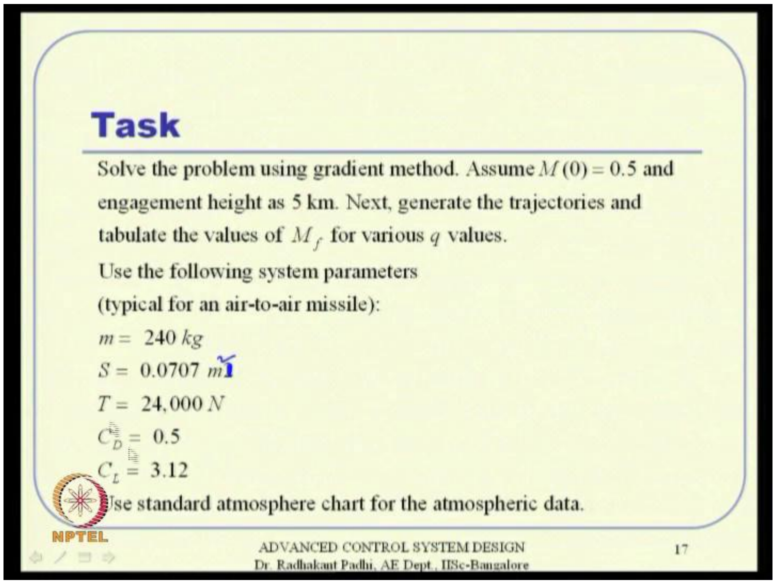

So, this is what we are talking we are all non dimensional quantity M dash is d M by dega. Now, what is going on here is M dash and gamma dash is with respect to tau which is non dimensional type. Now, conveniently what is done here this innovative way sort of thing the minimum time problems are typically harder problems to solve, you can see the del equal to one here and then you take a h equal to that one plus lambda transpose (( )) will happen. And then you want to take del s by del u you may not get anything to solve for control. So, just to be done in a different way and there are procedures to do that this is typically goes through this (( )) control and all that actually so, you will not discuss too much on that.

But compare to that what is done here in this in this particular literature very interestingly that this change of very (( )) say proposal. In other words I do not want to visualize this equation in terms of tau or non dimensional time, but I want to visualize as the functional of or differential equation in terms of free variable as gamma, gamma is the flight path angle. So, if it is free variable is gamma then d t by d gamma I can write one of our gamma to (( )) actually so, that is one of our gamma to at it is expression that means, gamma acts as a free variable here where as t h is a state variable here, they say little different way of looking at the problem.

So, if I J is something like 0 to t f d t then it is equivalently it is 0 to minus pi d t by d gamma and d t by d gamma expression is available here so, that is put here. So, you have a cost function, which is no more one inside here now and the limits of integration has taken care of this 0 to minus pi these are all on limits per pi that you want to actually four gamma that is what you want for four gamma basically. So, that is taken the limits now and here what you are putting is d t by d gamma into d gamma so, d t by d gamma into d gamma is nothing but d t.

So, this change of variable you are not compromising the goal you are still solving a minimum time problem anyway, but while changing that the problem appears in terms of gamma which is 0 two minus pi and think like that. Now, subject to only one differential equation here d m by d gamma this part has gone inside the state equation. So, what was initially a two dimensional problem two dimensional states and all it is now, converted to a one dimensional constraint problem, with these hard constraints in place through this limits of integration and this is what you are talking about a minimum time optimization problem.

(Refer Slide Time: 29:58)




Now, you can use a generic theory and then you can actually try to solve these problems. And this is a task that I am giving you if you are interested to solve this the task is you try to use this gradient method and try to solve this problem for initial condition of mach number is point five and the engagement height at 0.5 kilo 0.5 kilometer think like that. And you can finally, take this some of these parameter values which are again representative numbers and just to have a feeling of what is going on that only not with respect to any particular vehicle for say.

This is not really m two this is actually m square obviously, in the meter square basically say so, this is what you what you need to do so my task that I am giving you is you try to experiment the gradient method based on these numbers. And you on the way you may need some atmosphere data and all, because that square unit to compute dynamic ratio and we can use some standard atmosphere. There are international standard atmosphere u s standard atmosphere Indian standard atmosphere like that (( )) level. So, you can take any standard atmosphere to compute air density as, a function of 8 and then you can you are ready to work on this problem.

This is your cost functions what you want to minimize these are the bounds limits with respect to gamma and the state equation that is coming along with this cause from semi only this one. So, you derive the necessary conditions and then you talk about solving this problem for with respect to these parameters and place it. So, that is this is state representative of the some aircraft to air missile sort of thing what is certainly not with respect to any particular vehicle. I just given these numbers for experimentation and then having a feel of what goes on it is here. Anyway so, that is what it is now let us go to the shooting method the next one. So, what you are doing here is again the say revisiting the same problem so that is what we want to have a different method here.

The problem is exactly same state Costate equation dynamic optimal control stationary you have split boundary condition half an initial time half in the final time so, let us see how do you do that. Now, this particular method assumes that you whatever, control is to that you guessed is like shooting is on to the target. So, suppose you want to suit suitable to the target then initially weighing something it obviously, does not fall on the whole size it does not fall on the target directly, but after it falls somewhere you know how much it is travelled and think like that.

So, that means there is the picture here suppose this is somewhere, here I try to fire a target something I some stones or thing like that it has gone fall somewhere inside. So, this is my actual position where I left over I landed this is where I should be basically so there is obviously, this error is available to me. So, using this error I will update my control history in such a way let us say in this particular case this initial launch single that is what we are telling. Now, using this error information I have to revise my launch single in such a way that next time I will do something like that closer to that.

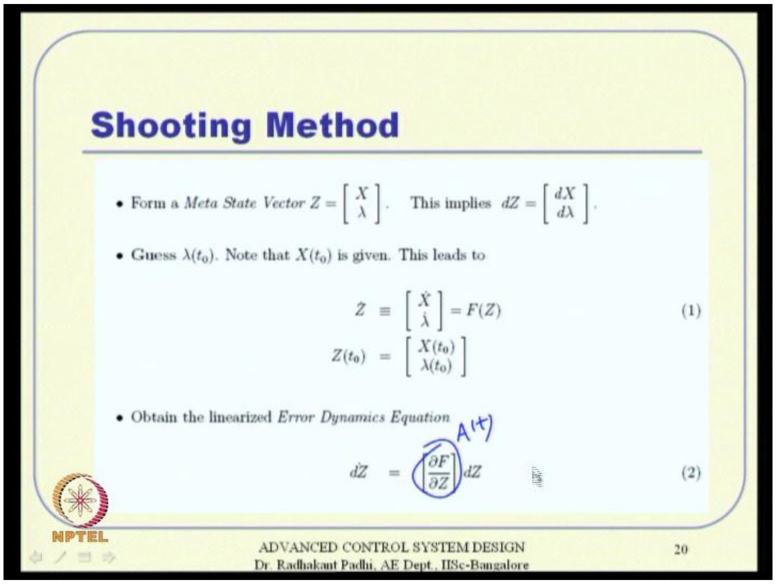
Now, next time I may be I may ever shoot actually so that is fine so, then you can I know that there is a solution in between so, every time I use this error information that is where I should be where I am falling at two equal to t f. And then using that information I will come back and try to update my control history. So, that is like shooting we are initially shooting and which falls somewhere then again you go back and shoot in some other direction fall somewhere, some again falls somewhere, very closer to target. But not on the target and then you try updating this control or guess whatever, you are telling initial guess then you proceed further.

Now, in this procedure we are assuming that initial guess for lambda is known to us remember that, if all conditions are known to us and then one you n n t 0 or any particular point of time then you are you can integrate the equation together. But because that is not happening so what you are telling is let us hold our breath for this one for a second will not consider this boundary condition for a second. But will rather assume that lambda of t 0 is available to us so that is available. So, t x of t 0 is available lambda of t 0 is available then together it is available so I can set certainly integrate this state and Costate equation together both in forward direction only.

Now, once I update then ultimately I will get an x f and then I can evaluate with respect to that x f my I can evaluate my lambda f and that lambda f may not be kind of same, because this update will throw me some lambda f whereas, my correct lambda f should come from this equation. So, that will throw me some error in lambda f and using this error in lambda f, I should be able to correct my initial guess of lambda what I am talking. So, that is the procedure so I demands that I start with an initial guess value of lambda vector.

And also remember that lambda is a Costate variable does not have any physical meaning so, guess in that again requires that you guess a control history from t 0 to t f and then integrate, once forward and once backward and then whatever, Costate equation integrate backward it will give a correct value of lambda t 0 guess.

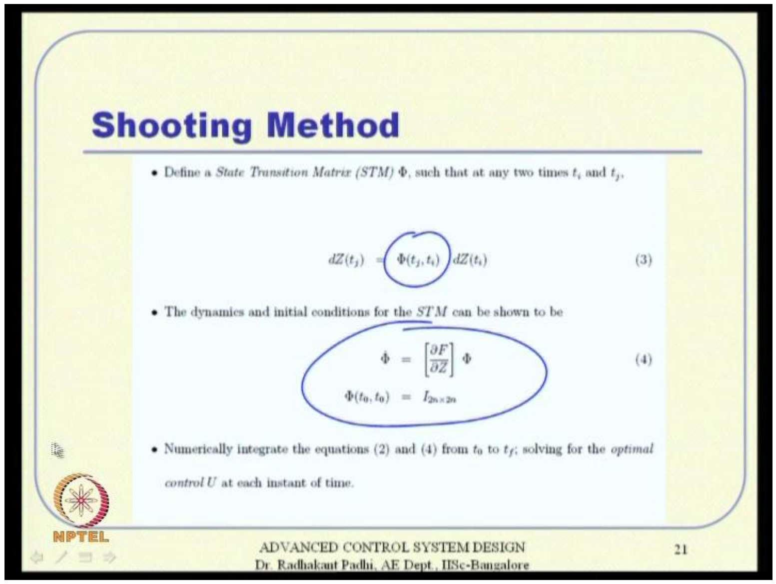
(Refer Slide Time: 35:27)




So, still you require a control history guess basically so that you cannot get rid of that. Now, shooting method is something like that I have guess this lambda 0 so, I form my meta state vector x and lambda, then this implies that there is d z that error in this meta state vector you also d x d lambda. So, I guess lambda t 0 so x to 0 is given anyway so, this leads to this equation that lambda dot x dot and lambda dot together, if I see that is z dot is nothing but f of z. And z of t 0 is now available so, I can integrate the equation forward together basically. And simultaneously I also have even error integration error equation d z is equal to d x dot d lambda dot and z dot is that so, if I consider this linearization and all that, then d z dot is nothing but del f by del z into a z you know del f by del z is nothing but something like a matrix that we know.

So, this is something like a, but unfortunately this is actually function of time that is it is actually it becomes a something like a time bearing a matrix actually. So, that is where these is the error equation is a linear equation that we are interpreting, but it is a time bearing linear equation.

(Refer Slide Time: 36:42)

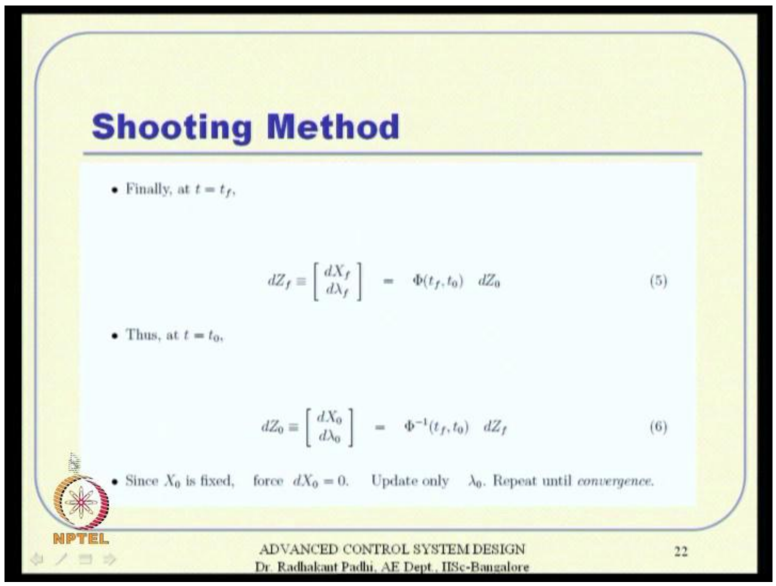



So, to solve that we need this say concept to state transition matrix actually so, that has state transition matrix for d z this equation is given something like this that is the state transition matrix. It is state transition matrix and this state transition matrix we know that it is satisfies the same differential equation so, we can I mean these in a matrix since actually now. So, this from this equation this the this state transition matrix equation is also known to us pi dot is equal to del pi bar del z into pi. What the good part of it is pi f t 0 t 0 is identity that means, this equation that you are having here for pi this is independent of my problem description that I am talking.

Whatever, problem whatever initial condition it is independent of that so that means this integration of this pi equation can be done independently. That is why the state transition matrix concept is actually useful so, I increase that, but also remember that the dimension of this pi matrix is two n two n by two n. So, even if you are talking about let us say symmetric matrix like that which may not happen then you talk about a still you are having some n to enclose one by two elements free basically. So, that means a differential equations you have to integrate and think like that.

So, in general this is a last dimensional matrix and hence, this is the last dimensional differential equations you know you need to integrate actually, that is where it becomes computationally (( )) mean intensive basically. So, what you are doing here is you integrate this equation as well as the equation and then you are ready with, this optimal control U to you need to keep on solving on the way basically. So, that is this equation still needs to be solved so that that is what you are doing here. So, you integrate numerical integrate equation number two and equation number four while solving this optimal control u at each instant of time. So, that is this optimal control equation needs to be solved at each instant of time.

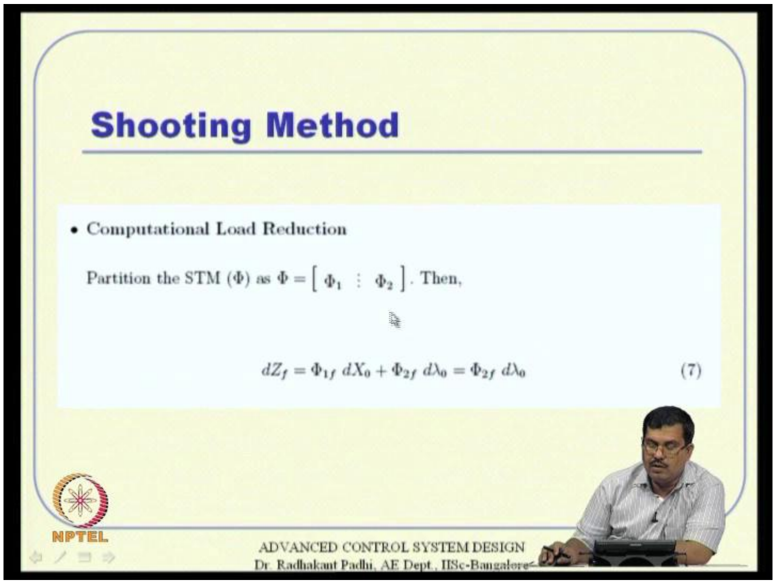
(Refer Slide Time: 38:45)




So, that is what you are doing here and finally, what you are getting at t equal to t f we are getting d z f something like that that is for ultimate time. Also remember that these state transition matrices are never singular so, I should be able to invert this and then try to get a value of d Z 0.

That is where I got my d lambda 0 that is the correction that I need for lambda with lambda 0 basically.

(Refer Slide Time: 39:08)

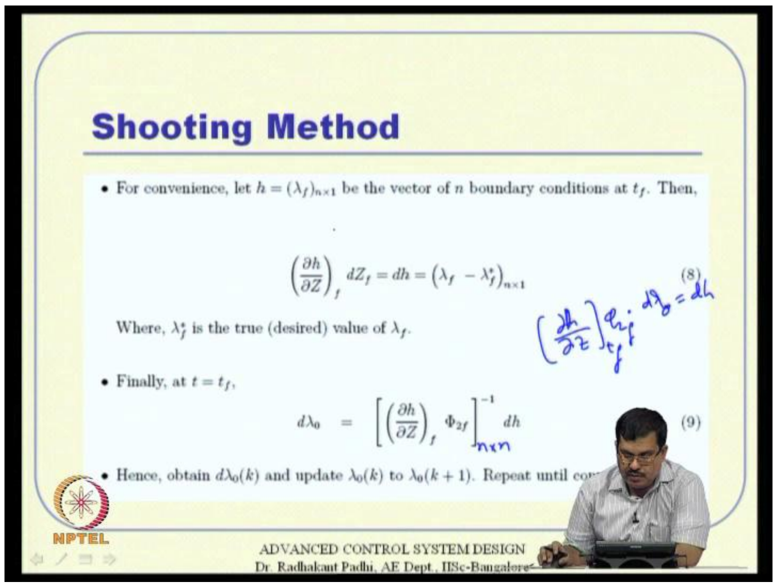



So, that is my ultimate aim I was I started with some lambda 0 guess and then I want to update this lambda 0 actually so, this is my lambda 0 that I am looking for so d lambda 0. And then obviously, you require half of this matrix only if b f 0 is something that we do not want to utilize, if f 0 is fixed any way so d x 0 we forcefully put it a 0 even though the computation compute some something we do not want to buy that we just want to assume that this d x 0 is 0. Because x 0 is fixed we do not want to change the problem basically so, we need to repeat this procedure until convergence actually so, that is that is the thing that you are talking.

Now, also remember because you do not want to compute this I mean do not want to make use of this d x 0, then the idea is why to compute this d x 0 you need not compute it that way. That means whatever, matrix inverse that you are talking here that is two dimensional this matrix inversion that you are talking here, actually two n by two n one matrix dimension. And instead of doing a two n way two n way matrix inversion, if you just do half of this and obviously, you need to do n by n matrix inversion which is this can be computationally I mean simpler.

So, you can save some computations once you do that then you can this try to split this pi vector into two sub matrices pi one pi two like that vertically. And then you can interpret that this d z f that I am talking about can be interpreted that way and I will take only lambda two f times d lambda 0 part of it. But remember lambda two f is not a square matrix so, I cannot talk about inverse so that I have to have some matrix, which will this is I mean some matrix which is related in a square matrix so, I can take inverse of that.

(Refer Slide Time: 40:45)



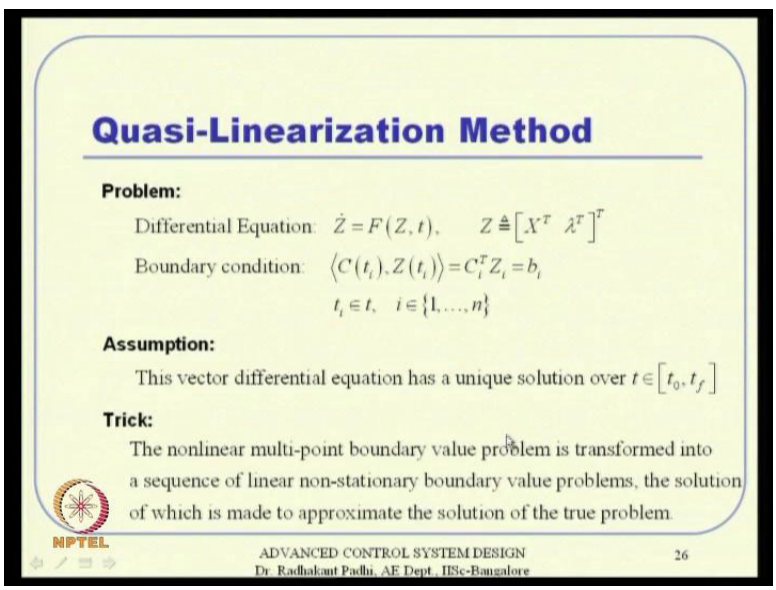

So, for that you can for convenience you can write this h which is lambda f so that is coming from this boundary condition that is lambda f equal to the del f by del x if also I think like that. And hence, I can interpret that this d has that I am talking here is given as something like that d h, one side you can see that it directly coming from errors in z f so, that is what it is and other side is lambda f minus lambda x star lambda x star is the ideal values for lambda f that the boundary condition. So, that is the true or desired value for lambda f so that is we need to do that and hence, this equation what you see here now, if you go back actually you can interpret only d lambda f only that d h.

So, what you it that will happen that d h by d I mean what you have what will happen is some something like del h by del z evaluated at f into pi to f is equal to d h actually, that is what you will do. Now, if that happens now there is a small mistake here probably into these into this into d lambda 0 is equal to d h, then obviously you can solve for lambda 0 d lambda 0 that way actually you can do this algebra yourself. Like make use of this equation and then go back to this equation and then try to extract what is d h and then that is equation will be (( )) and hence, you can do about this and this matrix dimension that you are talking is actually (( )).

So, instead of doing two n by two n matrix yourself you will be able to do n by n matrix yourself. So, that is that is the procedure that you can repeat until convergence actually so, that you got you do in shooting method. So, you started with a lambda 0 and then integrated it forward and then you saw that there is an error for that for the lambda f thing so, using the error in lambda f you want to correct your something in lambda 0. And that is like shooting unshooting actually that is why it is called shooting method.

So, is there need to repeat this procedure until convergence that is the shooting method. Now, let us go to the last method which is quasi linearization method as a in this particular one I will be more interested in talking in two point boundary value problem in general and multi point boundary value conditions in general. So, what is certainly useful and optimal control solution per same, but will not go back to that same equation that we discussed, but rather will discuss in a more generic way so, let us see what is the problem?

(Refer Slide Time: 43:31)

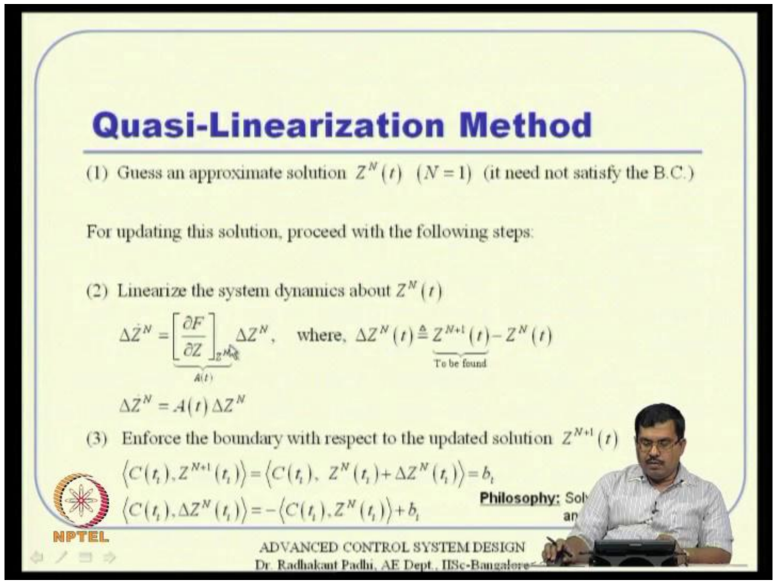


Problem is we have a set of differential equation z dot is a f of z, z can contain x n lambda. And then there are boundary condition which are given in this form that means heat transpose z I whatever, you are looking at here is equal to d I where I t these I t I know what you are looking at belongs to t and I stands for n to n x want to n. So, it can happen something like what you are telling here is boundary conditions can given at multiple time actually, it need not be given only at initial final, but it can be given at distinct points of what time. And need not be given as specified value for say it can be given as say some sort of a linear combination of those values those components of z in general basically.

So, it can contain something like linear equations from of both x and lambdas together basically so, it is a more generic formulation and let us say how do you solve this actually. And we also assume that this vector differential equation has a unique solution over this t 0 to t f otherwise there is a problem for solving this actually. So, what is the trick? Trick here is the this non-linear multiple boundary value problem is first transform into a sequence of linear stationary boundary value problems and the solution of those problems is made to approximate the solution of the two problem.

That means, you break a kind of dividend value actually so, you this entire non-linear multi point boundary value problem, you first divide it into a sequence of linear boundary value problems not stationary of course. And then the solution you find out for those problem and then you think that the solution of this problem has to be made approximate, I mean has to be made equal to the solution of the two problem then approximate same basically that is what we want to do.

(Refer Slide Time: 45:24)

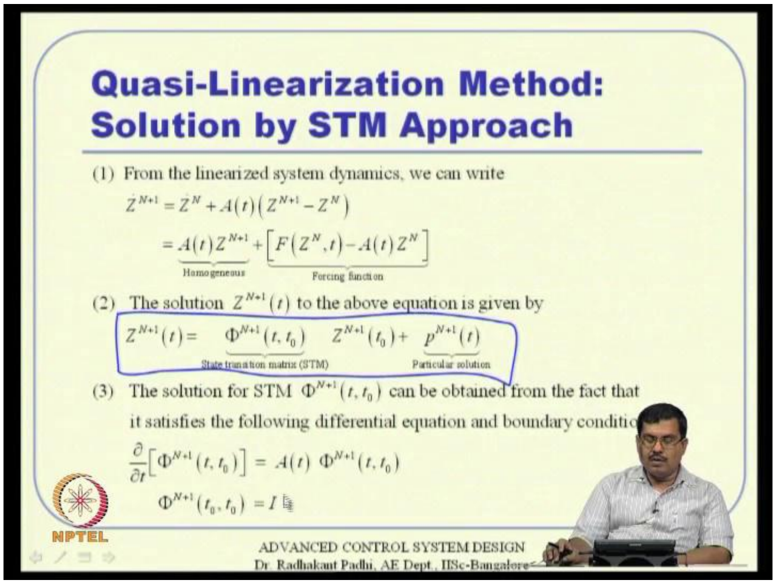



Now, how do you do that obviously, we start with a guess approximation that is let us say z of n is available to us that is the guess value. And it need not satisfy the boundary condition it may start with something which exactly may or may not satisfy all the boundary conditions. So, obviously you want to repeat this so that it satisfies the boundary condition essentially basically (( )) our idea there. So, for updating the solution you need to proceed this way let us say we Linearize this equation z dot is f of z t and then that is what you get, again the similar way what we discussed del f by del z terms out to be a f t and this a f t and this z n z delta z n by definition is z n plus 1 minus z n basically.

So, what you are looking for is you have nth iteration value known to you this capital n is a iteration number sort of thing and then n plus 1 is something you want to find out. So, t for that the error equation is like these which is a time bearing linear equation again that is what you are talking. Now, in the boundary conditions say this is the boundary condition so, let us plug it back whatever, so the boundary condition is not satisfied at n, but we want to make it satisfied at n plus 1 at least. So, if I substitute this condition as n plus 1 solution now, at least I should be able to satisfy the boundary condition and that is what I aiming for actually so, c f t I n dot n dot product of z of n plus 1 t I that also satisfy equal to I, but you know this z n plus 1 t I is nothing but this one actually z n plus del z n.

So, if you look at if you want to see this you are in this linear inner product is a linear operator and you expand you expand it into two terms and you say c, I mean c f t I into delta z n of t I is nothing but negative of that plus v i. So, why I can interpret this equation I mean this condition from this equation as well actually so, that is what I do here. Now, from the linear system dynamics what you want to do? I mean if you go back to this linear system dynamics again you substitute this definition.

(Refer Slide Time: 47:34)



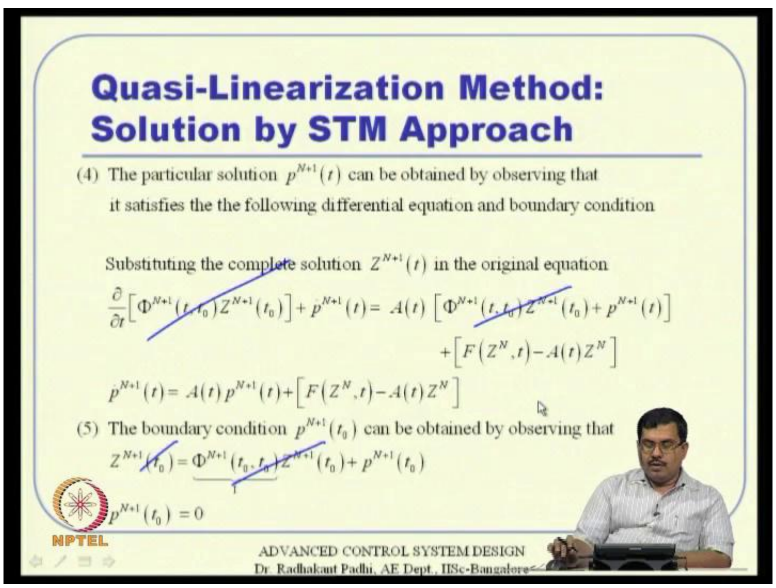

Then you can sell z n plus 1 dot is nothing but this (( )) this (( )) because this one minus this one is that by that is what we have here, making use of the definition we are telling z n put z n plus 1 dot is nothing but this one plus this one. So, obviously it gives you some sort of a homogeneous part and then there is a forcing function. So, if there is a homogeneous part there is a still a linear equation by the way this n plus 1 sense so, this linear equation you have a solution which is given as like a homogeneous solution and a particular integral actually in a particular solution. So, this z n plus 1 is the direct solution that I am looking for remember I started with z n, but I am looking for a solution in terms of z n plus 1. And hence, I am of course, is interpreting all these equations that are available to me basically.

So, I have this z n plus 1 dot which is something like this part of that homogeneous expression homogeneous equation and part of that is actually forcing function basically.

Now, the solution of the above equation is certainly this one through a state transition matrix again basically. And again this the state transition matrix satisfy these conditions now, what is the thing here? If you observe this equation we have the form ready now, this equation what you have here so, you have the form of the solution ready basically. Now, what you what you really need is how to build on this actually like how do you find out this, how do you find out this and how do you find out that? And that is what once you that then the solution is ready anyway basically.

So, here you are telling how to find out this and that is coming from this, because this is the static I mean state transition matrix which is satisfy the same differential equation and boundary condition is independent of any other condition. So, this can be integrated separately to get this basically.

(Refer Slide Time: 49:24)

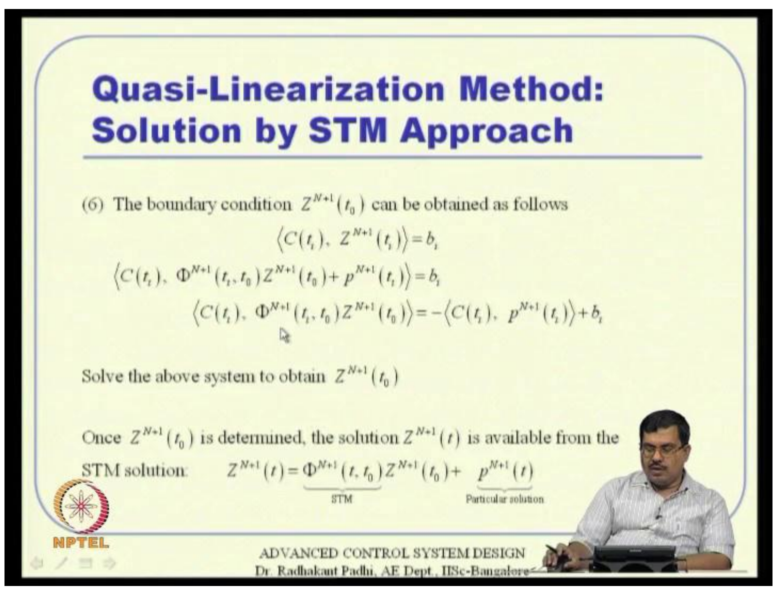



Now, what about these two? Now, this particular solution p n plus 1 we are talking about this one now can we obtain by observing that it satisfy the same differential equation this is any particular solutions satisfies the same differential equation. So, then you put it back and then you see in the same differential equation sense what happens, that is my solution anyway this entire solution. This entire solution I will put it back in my differential equation solution what I have there and then I tell all these equations are far known to me, because this satisfies some what is happening here, is this equation as I know will cancel out this you form will cancel out from this one.

So, that is the condition that will come up from come from here basically, this condition what you see here that condition will help me in cancelling out that first two term and you are left out with that one. So, here have differential equations for this p this p n plus 1 t and you also need a boundary condition for that this is a differential equation. So, the boundary condition for that comes from this condition what you have her n plus 1 is nothing but this by definition and this identity of course. This is identity then what happens, this is this cancels out again this one goes through this one so, you are left out with the condition that p n plus 1 p n plus 1 t 0 is actually zero.

So, you have a two differential equation for p n plus 1 and you have a corresponding boundary condition also and that is how you solve this particular thing basically.

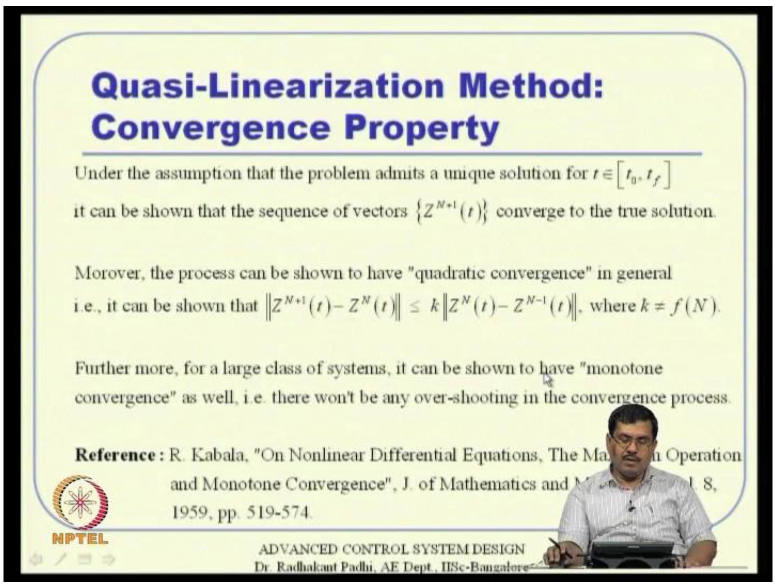
(Refer Slide Time: 51:08)




Now, what about the rest one? This is I mean if I still need to have this condition to grow to have a solution of that and that condition I will go back and tell the boundary condition of these, I know this is my boundary condition that I that I want to satisfy. So, I put it back my solution form this is my solution form that one I will put it back in this boundary condition that is given to me. And then again this inner product is a linear operator on that, I can split this equation in two parts and then I can tell this is this inner product in the left hand side is nothing but equal to the, this inner product negative sign with this plus v I term basically.

So, that means this condition on this n plus 1 is available to is not (( )) available to you yet, but these are the set of once of linear equations and hence you can solve this linear set of linear equations to get these z n plus 1 at t 0. And hence, the recomponent of that is ready so we have an idea of how to compute this through this equation; we have an idea of how to compute that through this set of equations this equation and this equation and now, we have a idea of how to compute this z n plus 1 t zero. So, if you this equation if you look at it we have all the building blocks ready now so we have we have everything that is in place basically, we can compute this z n plus 1 the updated control basically so, that is how this procedure operates basically.

(Refer Slide Time: 52:19)



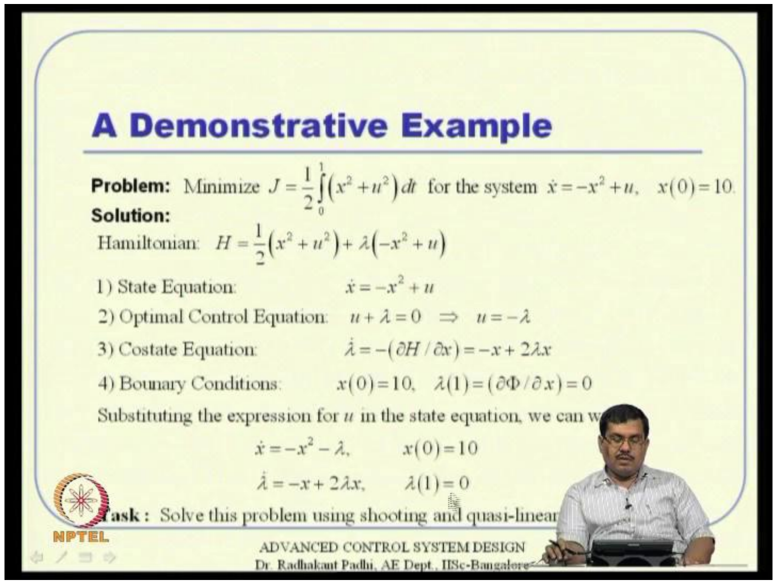

Now, it is a very need convergent property of this quasi linearization method which tells you actually tells that so this theorem reads something like that. So, under the solution on the assumption that the problem that means, the unique solution that is what we wanted in the beginning aim of that, yes one of the assumption was the v ten to t 0 f this problem has a unique solution. So, if you under that assumption it can be shown that the sequence of vectors that that is what you are computing in an iterative sense that the sequence of vector will actually converge to that true solution that is the first guarantee, it will converge to the true solution.

Once that guarantee is in place then, want to ask the questions at what rate actually and that guarantee is also there it tells that the process can be shown to have quadratic convergence in general that means, in a normal sense it has to decrease that way. Which is extremely (( )) to also remember Newton loss of technique is also, some of these quadratic convergence properties and all that. So, what it tells is the convergence can be very fast actually so, every successive iterations norm of that will be less than equal to some positive constraint and the difference of the previous the two previous values.

I want an update like let us say z n is my current value actually, I want an update value so I have an error with respect to my previous solution and I have an error with respect to my updated solution. Now, with the error with respect to the updated solution is less is certainly guaranteed to be less than equal to some constraint time the error in the with respect to the previous solution. So, that is what it tells actually so that way if the convergence can be fast actually quadratic convergence is tells you like that. And further more for a large class of systems it can also be shown to have monotone convergence as well which is extremely good in addition to quadratic convergence.

Monotonic convergence means, it will converges from one side only if you start with a positive number it will decrease, decrease and go to 0 it will never averse shoot and then oscillate around 0 and then converge, that is that is monotonic convergence. So, you can see that there are multiple levels of guarantee here one is it will converge to the true solution first and not only it will converge it will converge at a quadratic rate. And it will also, converge in the monotonic sense basically in this it is extremely powerful method in my view and that sense basically so, you can try some of that and then have a feeling yourself.

(Refer Slide Time: 54:51)




Now, before I conclude I will also suggest that you solve this particular problem using shooting method as well as quasi linearization, then you will have a some sort of a feeling of what goes and it is actually, a very small problem while certainly give you some feeling. So, this is a quadratic cause function x square x by del t, but this not a linear state equation it say minus x square term is there certainly is a non-linear differential equation. We have initial condition like that and you want to minimize that with this cost function, I mean all the necessary conditions optimality you can derive very easily basically because the cost function something you know.

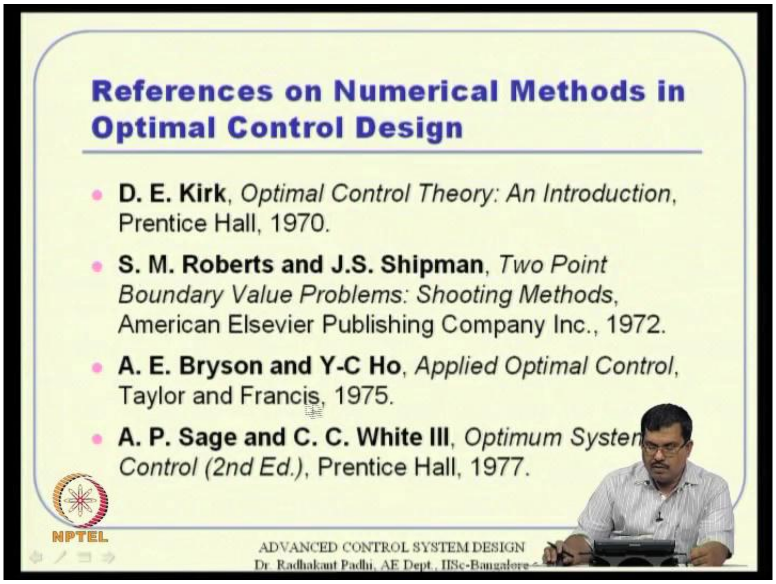
So, simultaneously l plus lambda f, because lambda is one scalar again because the state is also scalar so, you have this half of this x square plus lambda times this 1 minus x square plus u so that is what it is. Now, state equation is already known to you x dot equal to del x by del lambda well lambda is del x by del lambda is like that which is again the same thing so that is state equation. Then the optimal control equation as del x by del u equal to 0 so del x by del u means, one term is u so that one more term is lambda so lambda plus u 0 so u equal to minus lambda.

Costate equation is lambda dot equal minus l is by del x so, if you take differentiation of h with respect to x, then this is x one x term coming here minus so that there is a minus sign here and then there is a minus two x lambda which will become plus 2 x lambda basically, because of this minus sign. So, you have x dot is that u equal to that and then lambda dot is that boundary conditions sense x of 0 known to you. And lambda of one t f equal to one here so, that is what you have taken and lambda for a del pi by del x evaluated at one would pi 0 anyway you know pi terms so that is 0 (( )).

So, if you have this condition we tells you that x dot is that and x dot is minus x square minus lambda, because u s minus lambda so this particular plus you and substituting by minus lambda already. So, that way the one that the optimal control equation is already use actually so, you have this state equation that way and you have a Costate equation you have a two dimensional problem now. So, that means that z that we discussed here in this problem z is nothing but one x and one lambda two dimensional vector. And you have the corresponding boundary conditions known to you given at two different time so, you can try to solve these problem using shooting method as well as quasi linearization quasi linearization method.

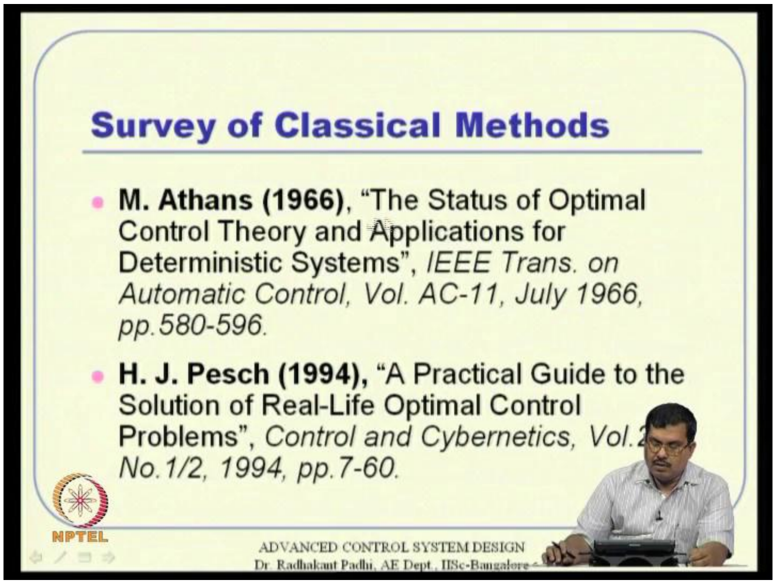
That is what I will suggest you to do that by yourselves using a small writing your small program files in whatever, language you want to do. And then have a feeling by plotting these, results and think like that so that is what I will suggest.

(Refer Slide Time: 57:34)



Now, references of the numerical methods sense some of these things I have taken from Kirk book especially the gradient method as well as the quasi linearization method I have taken from there. And shooting method is I do not know will I have taken from this one and can be available in many other books also with (( )) method and then the Bryson who is the standard optimal control (( )) there are many numerical methods as well. And then this is also a good book in my view, even though it was written in almost parallel to Bryson and y c ho thing so one is 75 one is 77 so that it.

(Refer Slide Time: 58:14)




So, they are the some of the references that you can see for more details on these methods and these are some of the two survey papers very classical survey paper, one is very old which is it talks about 66 and all that it what is a very nice paper. Talks about many status of optimal control theory and applications at that point of time 36 it gives us a very nice feeling of what happen initially. On 94 there is a very neat survey paper again which talks about real life control problem how you will make you how will you use some of these, practical solutions using many practical kind of numerical methods and things like that. So, I suggest that you read some of these for two for a clever bit of feeling of that. So, I will probably stop here thanks for that thanks a lot for the attention.