Optimal Control, Guidance and Estimation

Prof. Radhakant Padhi

Department Of Aerospace Engineering

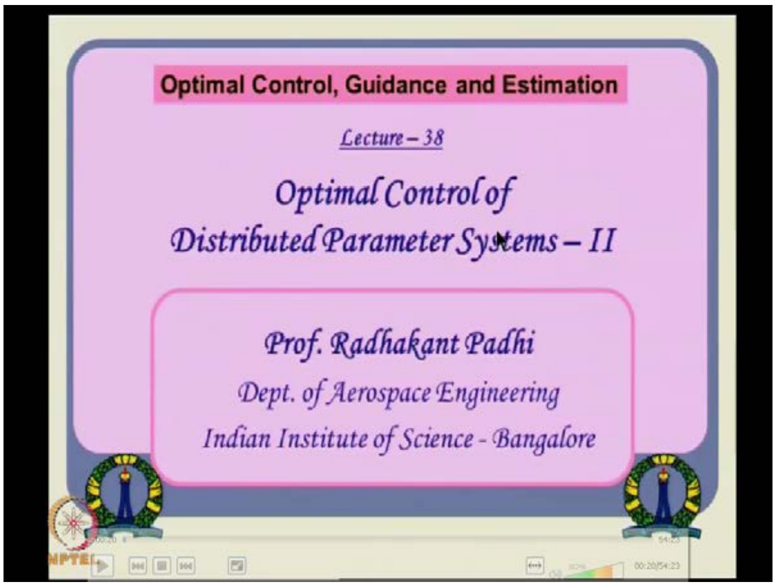
Indian Institute of Science, Bangalore

Module No. # 16

Lecture No. # 38

Optimal Control Of Distributed Parameter System – II

(Refer Slide Time: 00:20)



Optimal Control, Guidance and Estimation

Lecture – 38

Optimal Control of Distributed Parameter Systems – II

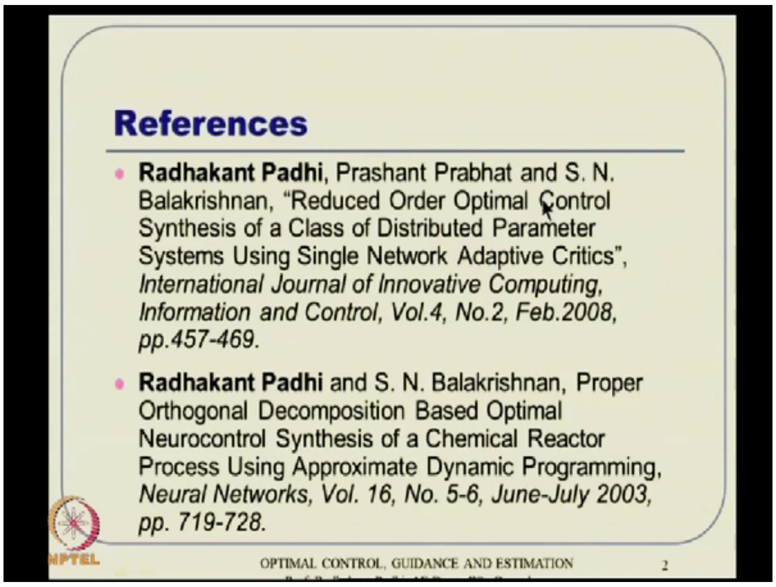
Prof. Radhakant Padhi

Dept. of Aerospace Engineering

Indian Institute of Science - Bangalore

Hello everybody, we will continue our lectures on optimal control of distributed parameter system. As I told in the last class, we will see some essential ideas getting embedded into optimal control of DPS, distributed parameter system. This is what we will largely talk here, both in the frame work of without model reduction and with model reduction.

(Refer Slide Time: 00:39)



References

- **Radhakant Padhi**, Prashant Prabhat and S. N. Balakrishnan, "Reduced Order Optimal Control Synthesis of a Class of Distributed Parameter Systems Using Single Network Adaptive Critics", *International Journal of Innovative Computing, Information and Control, Vol.4, No.2, Feb.2008, pp.457-469.*
- **Radhakant Padhi** and S. N. Balakrishnan, Proper Orthogonal Decomposition Based Optimal Neurocontrol Synthesis of a Chemical Reactor Process Using Approximate Dynamic Programming, *Neural Networks, Vol. 16, No. 5-6, June-July 2003, pp. 719-728.*

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 2

Largely the topics that I will carry for model reduction, we will talk about these two publications for some time, made it happen 2003 and 2008 like that. Then there are other ideas also in 2001 and were thinking like that way. Anyway you can find more details in some of these references.

(Refer Slide Time: 01:00)

Brief Review of Optimal Control through Single Network Adaptive Critic (SNAC) Design

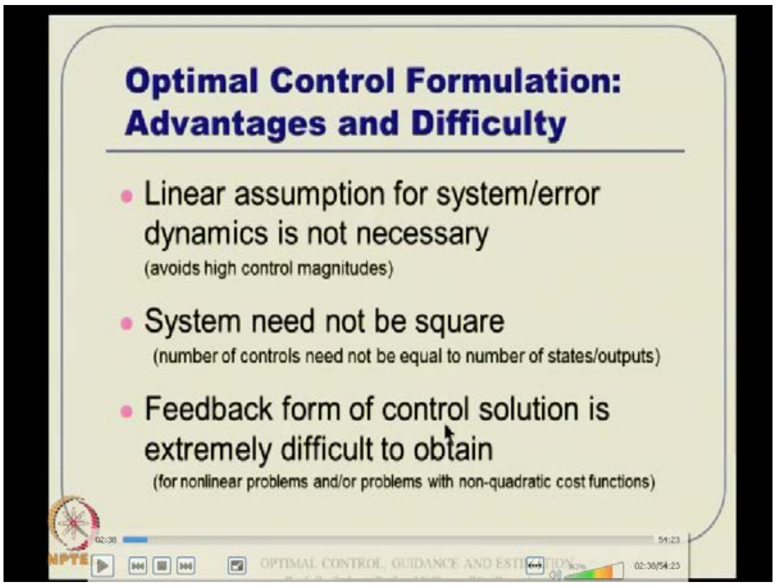
Prof. Radhakant Padhi

Dept. of Aerospace Engineering

Indian Institute of Science - Bangalore

So, a little bit very brief review of optimal control through single network adaptive critic; I mean we have taken lecture there, one full lecture is there for adaptive critic where you can see details on that. Just is a little bit of recap on what it is.

(Refer Slide Time: 01:17)



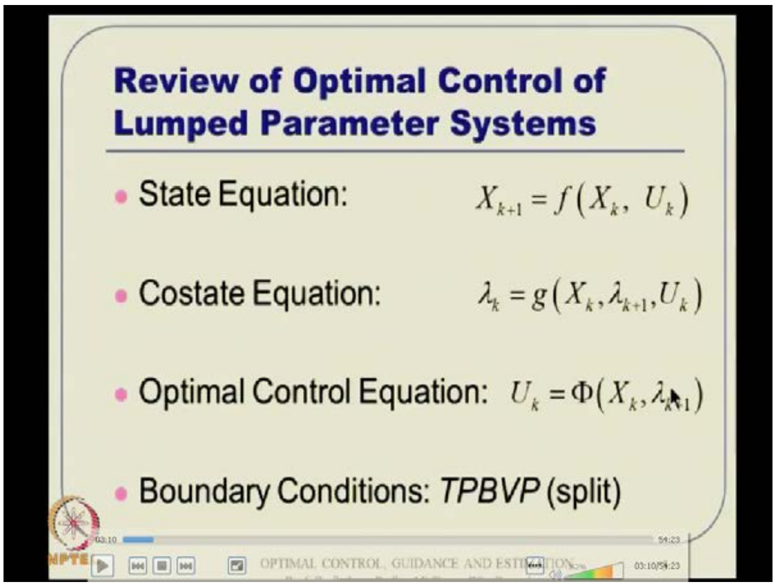
Optimal Control Formulation: Advantages and Difficulty

- Linear assumption for system/error dynamics is not necessary (avoids high control magnitudes)
- System need not be square (number of controls need not be equal to number of states/outputs)
- Feedback form of control solution is extremely difficult to obtain (for nonlinear problems and/or problems with non-quadratic cost functions)

So, first thing is optimal control formulation in general has several advantages like linear assumption for system dynamics or error dynamics is not necessary either. So, it avoids typically high control magnitudes normally. The system dynamics is accounted for planning the trajectory and hence your control normal does not become very high as some point of time. Then system need not… Now, coming back to this topic, what happens to be typically if the system error is large, if you go for stability sort of control theory then the control magnitude will become are large like PID control or dynamic universal like that. But, if you have optimal control in and it will tell there is time. It is not talking about moment sort of behavior, it talks about a behavior over period of time and all that. That is philosophical reason why it happens to be better distribution of control magnitude over a period of time accounting for the system dynamics.

So, that is one advantage of optimal control theory. The other one is system need not be square; that means, you do not have to have number of controls need not be equal to number of states or output. The third one is feedback form of control solution is very difficult these are good things, but this is very bad thing because feedback form of control solution typically happens to be quite difficult.

(Refer Slide Time: 02:39)



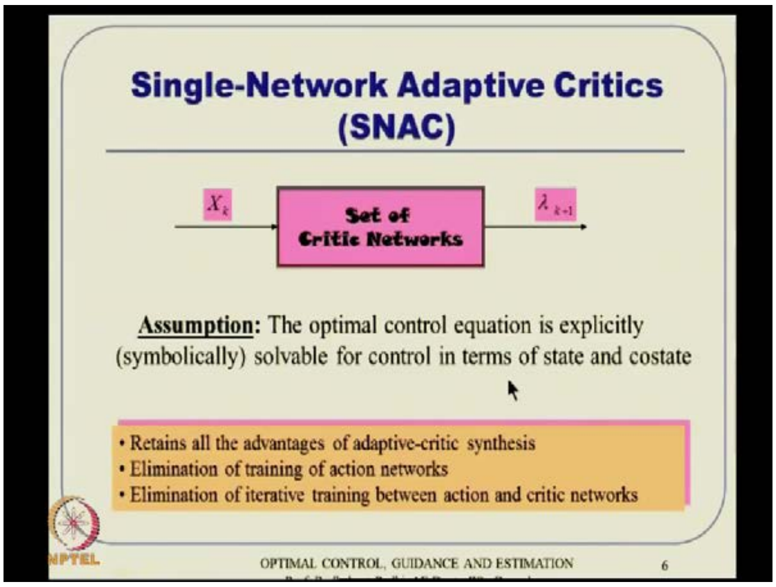
Review of Optimal Control of Lumped Parameter Systems

- State Equation: $X_{k+1} = f(X_k, U_k)$
- Costate Equation: $\lambda_k = g(X_k, \lambda_{k+1}, U_k)$
- Optimal Control Equation: $U_k = \Phi(X_k, \lambda_{k+1})$
- Boundary Conditions: *TPBVP* (split)

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION

So, now coming to the review of little bit distinct time domain optimal control theory, especially coming through this approximate dynamic programming set of ideas, this is finally the thing. We have State equations, we have Costate Equation and we have Optimal Control Equations. A typical list closed to a state equation double of forward costate equations double was backward and these are two dynamic equations. This is static equations. So, all this things we know, and remember U k is a function of X k. But, lambda k plus 1, is the discrete time optimal control theory.

(Refer Slide Time: 03:14)



Single-Network Adaptive Critics (SNAC)

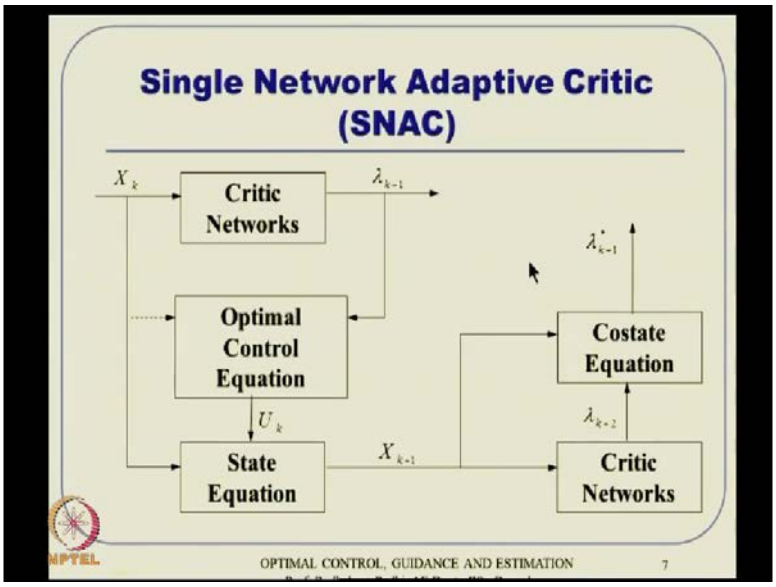
Assumption: The optimal control equation is explicitly (symbolically) solvable for control in terms of state and costate

- Retains all the advantages of adaptive-critic synthesis
- Elimination of training of action networks
- Elimination of iterative training between action and critic networks

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 6

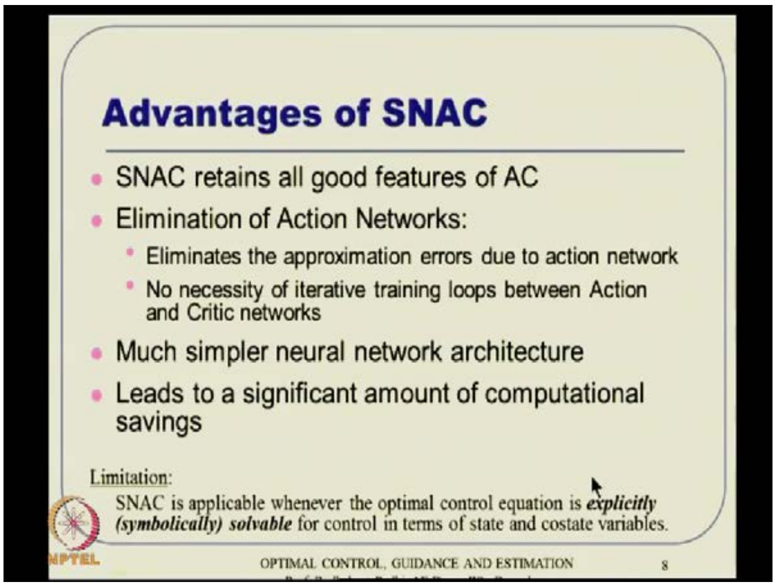
Now, coming to this single network adaptive critic idea, we have to synthesize a set of new networks in a such way that given an X k information, we should be able to find what is a our lambda k plus 1 information, so that is what the single network adapter is a critic synthesize is all about. An assumptions here is optimal control equation is explicitly or symbolically solvable for control in terms of state and costate. The advantage is obviously it retains all the advantages of adaptive critic and it eliminates the action network from the training process and also eliminates the iterative training between action and critic. This is what it is; there is a little bit comparison with respect to adaptive critic versus single network adaptive critic. We have talked in detail in one of the previous lectures.

(Refer Slide Time: 04:08).



Then the whole idea here is single network adaptive critic is something like this. We have this X k; we have this critic networks, here is to get lambda k plus 1. Obviously, to begin with this is non optimal. So, you have to get something like optimal lambda k plus 1 with respect to this X k, so that knowing this X k and lambda k plus 1, you can use that in optimal control equation and get U k. So, initially this is non optimal, but still you can use these two and get your U k. When your X k and U k are known, hence you get an X k plus 1. Again, you use same critic network get lambda k plus 2 then using these two we get lambda star k plus 1. Taking this has come to an input and that is desired output, you trained this network and stop when this two are kind of close to each other and that is what we have discussed little before.

(Refer Slide Time: 04:57)



Advantages of SNAC

- SNAC retains all good features of AC
- Elimination of Action Networks:
  - Eliminates the approximation errors due to action network
  - No necessity of iterative training loops between Action and Critic networks
- Much simpler neural network architecture
- Leads to a significant amount of computational savings

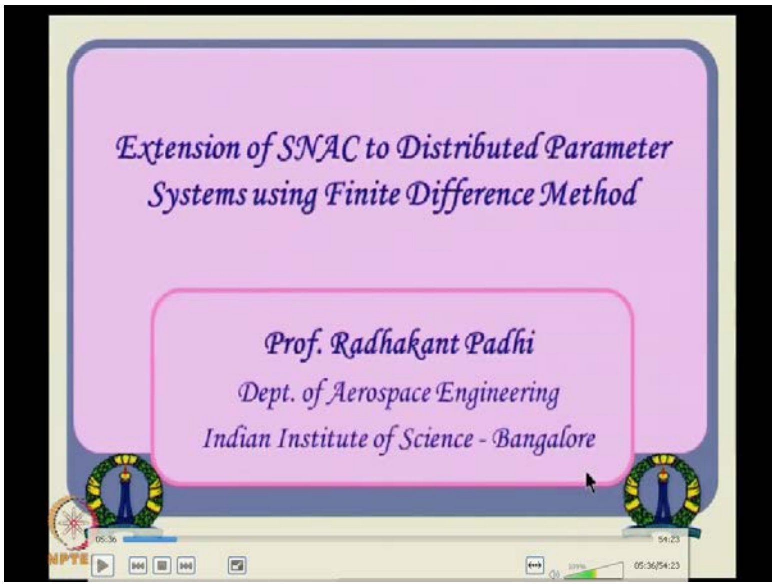
Limitation:

SNAC is applicable whenever the optimal control equation is ***explicitly (symbolically) solvable*** for control in terms of state and costate variables.

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 8

Now, the advantage of the single network adaptive critic is it retains all good feature of adaptive critic and eliminates the action network. Hence, it eliminates the approximation errors due to action network. No necessity of iterative training also between action and critic networks. It turns out to be much simpler than the neural network architecture, much simpler than the neural network architecture for synthesis of optimal control for this class of problems. So, it leads to some sort of significant amount of computational savings, especially when we talk about distributed parameter system synthesis. But, limitation again is it is applicable only when the optimal control equation is explicitly solvable in terms of other two variables.

(Refer Slide Time: 05:36)

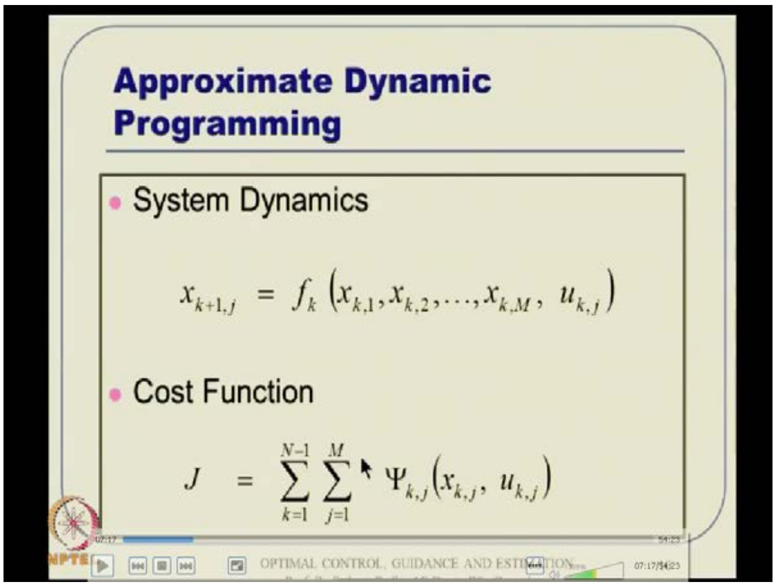


Extension of SNAC to Distributed Parameter Systems using Finite Difference Method

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Now, how do extend these ideas to distributed parameter systems? Here, this finite difference method comes very handy. We are not bothered about model deductions and things like that. We will just blindly put the grid points and try to get some solution sort of it.

(Refer Slide Time: 05:52)



Approximate Dynamic Programming

- System Dynamics

$$x_{k+1,j} = f_k\left(x_{k,1}, x_{k,2}, \ldots, x_{k,M},\ u_{k,j}\right)$$

- Cost Function

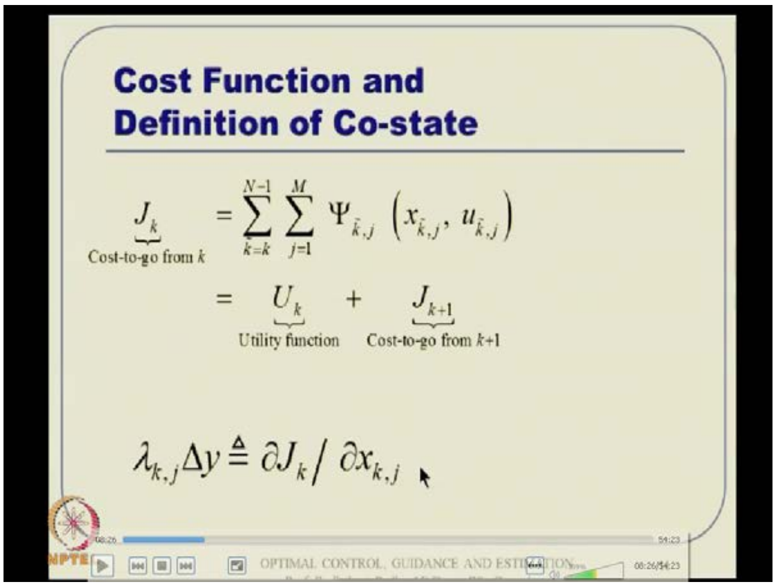
$$J = \sum_{k=1}^{N-1} \sum_{j=1}^{M} \Psi_{k,j}\left(x_{k,j},\ u_{k,j}\right)$$

So, let us see what you (( )) the program of which we were talking about. Having a system dynamics of this form x k plus 1 is something like this. We are all discretized at that node points; k, represents evolution of time was j represents with grid points in space. So, here by any grid point location j, this system dynamics is a function of all other node points' state values and the control applied at that particular point.

This is again whether it is all node points or a limited to one or two node points; it again depends on the order of the system dynamics and special derivative. If you have the second derivative like the example that we discuss last time then probably it is a function of neighboring nodes. If you have an fourth order derivative, suppose del square x by del sorry del forth x y del y forth like that, then it may for reporculate I mean the reporculation can go to little more grid points and both sides, so like that depending the system dynamics.

In general, you can think that at any node point the system dynamics behavior is the function of states at all other node points and control applied at that particular node point at that particular time also. The cost function to optimize is given in this form and everything is in discrete domain. So, first thing is something like a summation over squares then next thing is a summation over time; double integral sort of thing what we saw in the continuous time domain.

(Refer Slide Time: 07:21)



Cost Function and Definition of Co-state

$$\underbrace{J_k}_{\text{Cost-to-go from } k} = \sum_{\tilde{k}=k}^{N-1} \sum_{j=1}^{M} \Psi_{\tilde{k},j} \left(x_{\tilde{k},j}, u_{\tilde{k},j} \right)$$

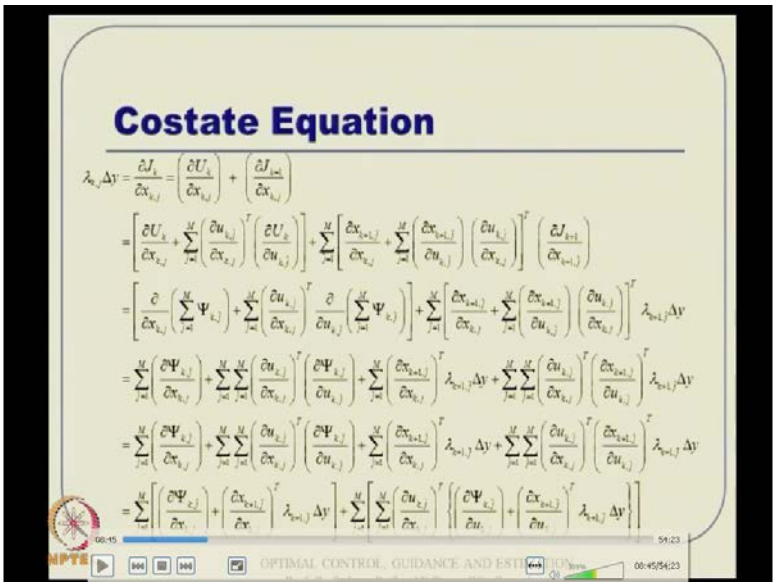
$$= \underbrace{U_k}_{\text{Utility function}} + \underbrace{J_{k+1}}_{\text{Cost-to-go from } k+1}$$

$$\lambda_{k,j} \Delta y \triangleq \partial J_k / \partial x_{k,j}$$

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION

Now, going to the same idea of approximated dynamic programming sort of thing, you can whatever is J k, k starts from here; k tilde starts k to N minus 1 that is cost to go from k. We can split this into two parts; first it is utility function it times k and then J k plus 1 that is cost to go from k plus 1 is like dynamic programming sort of ideas. Then there is a definition lambda and the definition turns out to be like this. We define lambda at grid points J at time instant k and that is lambda k j and in this form; lambda k j times delta Y happens to be like these. This is a kind of compatible definition and the distributed parameter settings basically. Remember, in continuous times lambda happens to be lambda Del J by Del x that is we know that, but blindly you cannot put it here. We have to be slightly more carefully here, to tell that lambda k j is this one divide by delta Y or this time delta Y is equal to that. That is a definition of lambda here.

(Refer Slide Time: 08:27)



Costate Equation

$$\lambda_{k,j}\Delta y = \frac{\partial J_k}{\partial x_{k,j}} = \left(\frac{\partial U_k}{\partial x_{k,j}}\right) + \left(\frac{\partial J_{k+1}}{\partial x_{k,j}}\right)$$

$$= \left[\frac{\partial U_k}{\partial x_{k,j}} + \sum_{j=1}^{M}\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)^T\left(\frac{\partial U_k}{\partial u_{k,j}}\right)\right] + \sum_{j=1}^{M}\left[\frac{\partial x_{k+1,j}}{\partial x_{k,j}} + \sum_{j=1}^{M}\left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)\right]^T \left(\frac{\partial J_{k+1}}{\partial x_{k+1,j}}\right)$$

$$= \left[\frac{\partial}{\partial x_{k,j}}\left(\sum_{j=1}^{M}\Psi_{k,j}\right) + \sum_{j=1}^{M}\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)^T \frac{\partial}{\partial u_{k,j}}\left(\sum_{j=1}^{M}\Psi_{k,j}\right)\right] + \sum_{j=1}^{M}\left[\frac{\partial x_{k+1,j}}{\partial x_{k,j}} + \sum_{j=1}^{M}\left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)\right]^T \lambda_{k+1,j}\Delta y$$

$$= \sum_{j=1}^{M}\left(\frac{\partial \Psi_{k,j}}{\partial x_{k,j}}\right) + \sum_{j=1}^{M}\sum_{j=1}^{M}\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)^T\left(\frac{\partial \Psi_{k,j}}{\partial u_{k,j}}\right) + \sum_{j=1}^{M}\left(\frac{\partial x_{k+1,j}}{\partial x_{k,j}}\right)^T \lambda_{k+1,j}\Delta y + \sum_{j=1}^{M}\sum_{j=1}^{M}\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)^T\left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)^T \lambda_{k+1,j}\Delta y$$

$$= \sum_{j=1}^{M}\left(\frac{\partial \Psi_{k,j}}{\partial x_{k,j}}\right) + \sum_{j=1}^{M}\sum_{j=1}^{M}\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)^T\left(\frac{\partial \Psi_{k,j}}{\partial u_{k,j}}\right) + \sum_{j=1}^{M}\left(\frac{\partial x_{k+1,j}}{\partial x_{k,j}}\right)^T \lambda_{k+1,j}\Delta y + \sum_{j=1}^{M}\sum_{j=1}^{M}\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)^T\left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)^T \lambda_{k+1,j}\Delta y$$

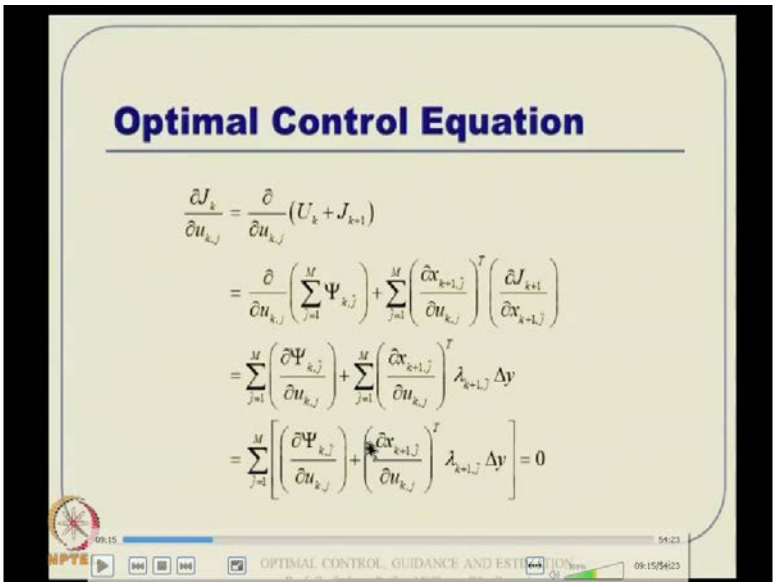
$$= \sum_{j=1}^{M}\left[\left(\frac{\partial \Psi_{k,j}}{\partial x_{k,j}}\right) + \left(\frac{\partial x_{k+1,j}}{\partial x_{k,j}}\right)^T \lambda_{k+1,j}\Delta y\right] + \sum_{j=1}^{M}\left[\sum_{j=1}^{M}\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)^T\left\{\left(\frac{\partial \Psi_{k,j}}{\partial u_{k,j}}\right) + \left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)^T \lambda_{k+1,j}\Delta y\right\}\right]$$

08:45 54:23

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 08:45/54:23

Then costate equation is convoluted process. We have to do really a lot of book keeping in getting there. But it can be done, you do not get afraid to see this algebra and all that. The term by term we can do, especially when you are really understood the derivation with respect to the lump parameter system from some one of our previous lectures. We can start similar thing, utility function, then cost to go, and then see what terms is what, what is the dependence on other things. Do not forget this summation over special domain and things like that. Carry out the algebra again term by term, I will not explain too much here, but it is there in paper and is there in this slide also here.

(Refer Slide Time: 09:09)



Optimal Control Equation

$$\frac{\partial J_k}{\partial u_{k,j}} = \frac{\partial}{\partial u_{k,j}}\left(U_k + J_{k+1}\right)$$

$$= \frac{\partial}{\partial u_{k,j}}\left(\sum_{j=1}^{M}\Psi_{k,j}\right) + \sum_{j=1}^{M}\left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)^T\left(\frac{\partial J_{k+1}}{\partial x_{k+1,j}}\right)$$

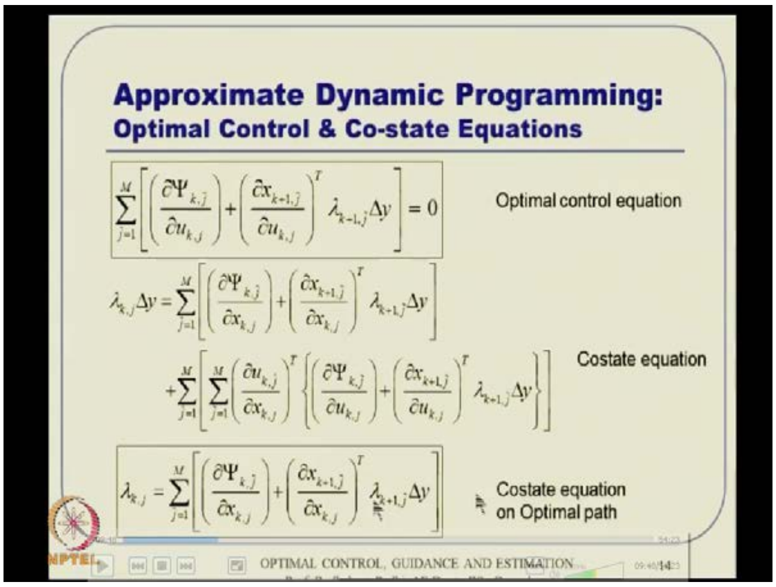
$$= \sum_{j=1}^{M}\left(\frac{\partial \Psi_{k,j}}{\partial u_{k,j}}\right) + \sum_{j=1}^{M}\left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)^T \lambda_{k+1,j}\,\Delta y$$

$$= \sum_{j=1}^{M}\left[\left(\frac{\partial \Psi_{k,j}}{\partial u_{k,j}}\right) + \left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)^T \lambda_{k+1,j}\,\Delta y\right] = 0$$

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION

So, carry out the algebra you will finally end up these some expression like these. Then optimal control equation also you can do some algebra and then will be equal to zero. Then go back to that and tell if this expression happens to be zero, and then what is the simplification that I can get here, so obviously, this term what you see here happens to be 0, the last term. So, you land up with only these terms.

(Refer Slide Time: 09:31)



Approximate Dynamic Programming: Optimal Control & Co-state Equations

$$\boxed{\sum_{j=1}^{M}\left[\left(\frac{\partial \Psi_{k,j}}{\partial u_{k,j}}\right) + \left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)^T \lambda_{k+1,j}\Delta y\right] = 0}$$

Optimal control equation

$$\lambda_{k,j}\Delta y = \sum_{j=1}^{M}\left[\left(\frac{\partial \Psi_{k,j}}{\partial x_{k,j}}\right) + \left(\frac{\partial x_{k+1,j}}{\partial x_{k,j}}\right)^T \lambda_{k+1,j}\Delta y\right]$$

$$+\sum_{j=1}^{M}\left[\sum_{j=1}^{M}\left(\frac{\partial u_{k,j}}{\partial x_{k,j}}\right)^T\left\{\left(\frac{\partial \Psi_{k,j}}{\partial u_{k,j}}\right) + \left(\frac{\partial x_{k+1,j}}{\partial u_{k,j}}\right)^T \lambda_{k+1,j}\Delta y\right\}\right]$$

Costate equation

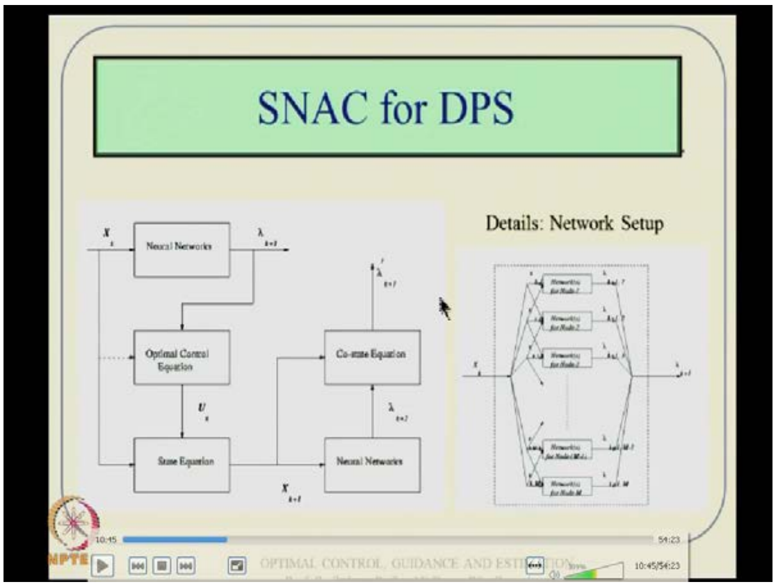
$$\boxed{\lambda_{k,j} = \sum_{j=1}^{M}\left[\left(\frac{\partial \Psi_{k,j}}{\partial x_{k,j}}\right) + \left(\frac{\partial x_{k+1,j}}{\partial x_{k,j}}\right)^T \lambda_{k+1,j}\Delta y\right]}$$

Costate equation on Optimal path

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION

So, this summarizes this behavior that you have optimal control equation and in general costate equation is given like this where as on the optimal path that this costate equation can be simplified to something like this. So, you essential deal with this equation as optimal control equation and this equation is something like costate term of optimal path.

(Refer Slide Time: 09:50)



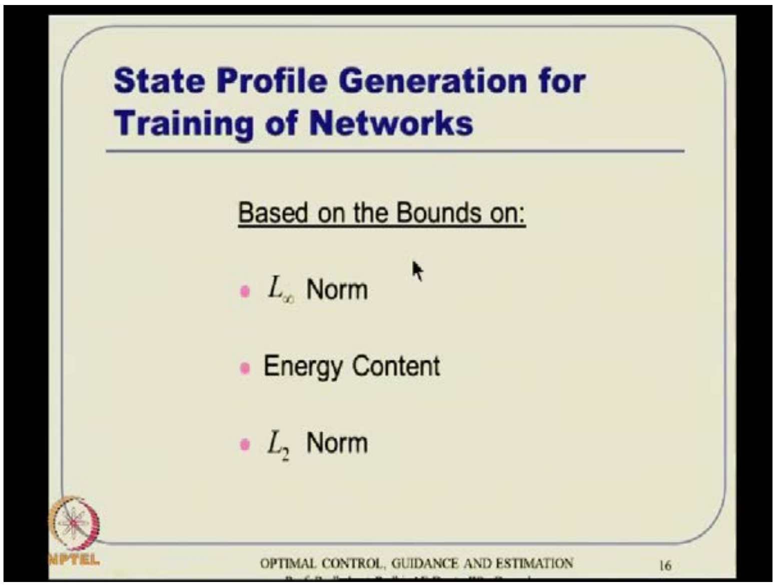
SNAC for DPS

Details: Network Setup

And this single network adaptive critic for DPS, distributed parameter system, roughly remains the same; very similar structure is what you have seen before. The only difference is at the neural network structure is different and that structure, this one turns out to something like this. It is at neural network at node one, node two, node three and all that you see, they are not inputs are not only X k; X k is the value of state at time instant k, but they also take this. See, for example, this network at node two will take input from node one and node three as well.

So, the neighboring node grid point values also turns out to be an input to the same network, otherwise the network do not converge . So, this is the proposition that we first proposed in the around 2000. In 99 or 2000 like that and then resulted in the 2001 paper. Anyway, the reference I will give you. So, this is very similar structure. The neural network details, what you see as neural network here, we have to really worry about synthesizing in this sense basically. That is the difference here.

(Refer Slide Time: 10:54)



State Profile Generation for Training of Networks

Based on the Bounds on:

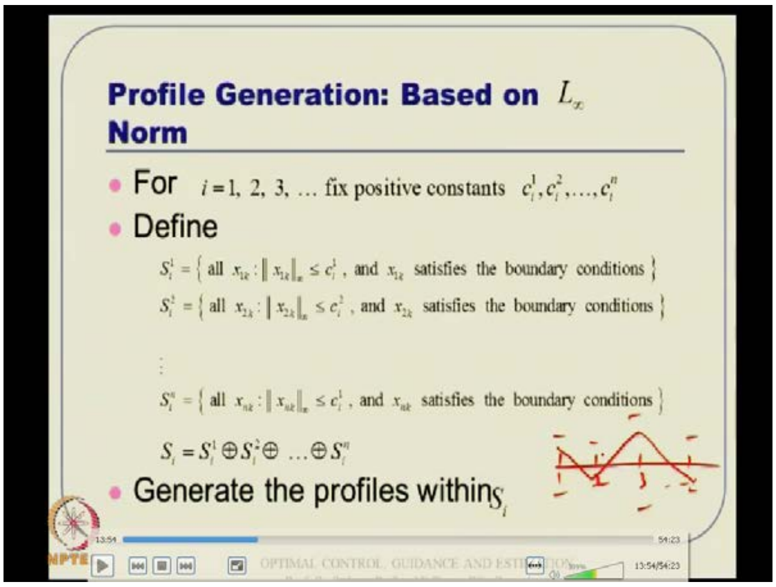
- L_∞ Norm
- Energy Content
- L_2 Norm

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 16

Then how do you generate, let us say a profile for neural network training. That can be done in various ways, which is again necessary for this other approach where model reduction, sort of POD approach, we also need all that, because if you talk about lump parameter system, you know a bound from where the system dynamics state will deviate and all that. So, from there you collect or randomly keep on collecting a bunch of point values that means, you can randomly select large number of state values for training purpose. There is no problem.

But if you talk about distributed parameter system that lectures is not there. So, what we do is we have to be slightly more careful and then tell now initial conditions are profiles over spatial domain, remember that. So, because of that you we have to bringing some concept like L infinity norm are probably depending on the problem we can have some sort energy content ideas, or in general you extend the energy contain ideas and then propose something like L 2 norm an all that. Remember, for the state values the state profiles are needed and state profiles cannot be discontinuous. State value, state profile has to be continuous over this spatial dimension. So, that is what restricts us to choose that and it is also a good thing, because of infinite number profiles. The more and more you become realistic then that training becomes better and better.

(Refer Slide Time: 12:22).



Profile Generation: Based on L_∞ Norm

- For $i = 1, 2, 3, \ldots$ fix positive constants $c_i^1, c_i^2, \ldots, c_i^n$
- Define

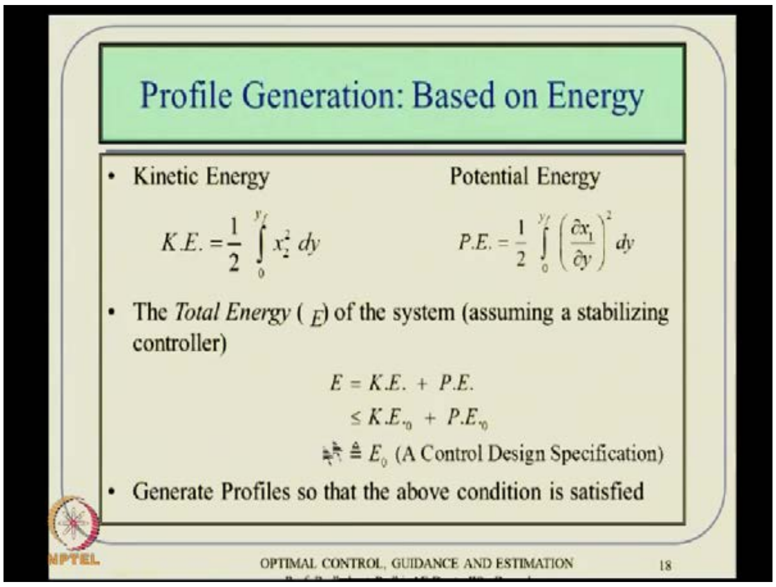
$$S_i^1 = \{ \text{all } x_{1k} : \| x_{1k} \|_\infty \le c_i^1 \text{, and } x_{1k} \text{ satisfies the boundary conditions} \}$$
$$S_i^2 = \{ \text{all } x_{2k} : \| x_{2k} \|_\infty \le c_i^2 \text{, and } x_{2k} \text{ satisfies the boundary conditions} \}$$
$$\vdots$$
$$S_i^n = \{ \text{all } x_{nk} : \| x_{nk} \|_\infty \le c_i^1 \text{, and } x_{nk} \text{ satisfies the boundary conditions} \}$$
$$S_i = S_i^1 \oplus S_i^2 \oplus \ldots \oplus S_i^n$$

- Generate the profiles within S_i

So, the very first thing if you really want to worry in L infinity norm sort of thing which is not really very good idea mind you, but still you can do that. You can see, you can decompose this state values to various component that x 1 x 2 x 3 x 4 are the components of the state vector, remember, in distributed parameter sense only. Then you can see this, then you can tell my values at all these, this x 1 k infinity norm has to be less than some value and this x 2 k infinity norm has to be less than equal to some value like that. And then you finally, tell my S i which is the domain of interest for me and is direct sum of these states.

So, essentially you generate the profiles within this S i. So, the idea here is well I mean pictorially speaking, something like this. You tell something like degree point have these of some 1, 2, 3 something like are there. Now, here tell like this value is bounded here, this value is bounded here, that value bounded this much and this is that much. Independently, I will select these points within that bounce and then I will join these points. This is pictorially ideas of that. But, this is not really a very good idea in my view because see 1 2 3 4 talk about this (( )) point things and all that. So, we need better things than that because this piecewise signing is also not a very nice idea.

(Refer Slide Time: 13:58)



Profile Generation: Based on Energy

- Kinetic Energy

$$K.E. = \frac{1}{2}\int_0^{y_f} x_2^2\, dy$$

Potential Energy

$$P.E. = \frac{1}{2}\int_0^{y_f} \left(\frac{\partial x_1}{\partial y}\right)^2 dy$$

- The *Total Energy* (E) of the system (assuming a stabilizing controller)

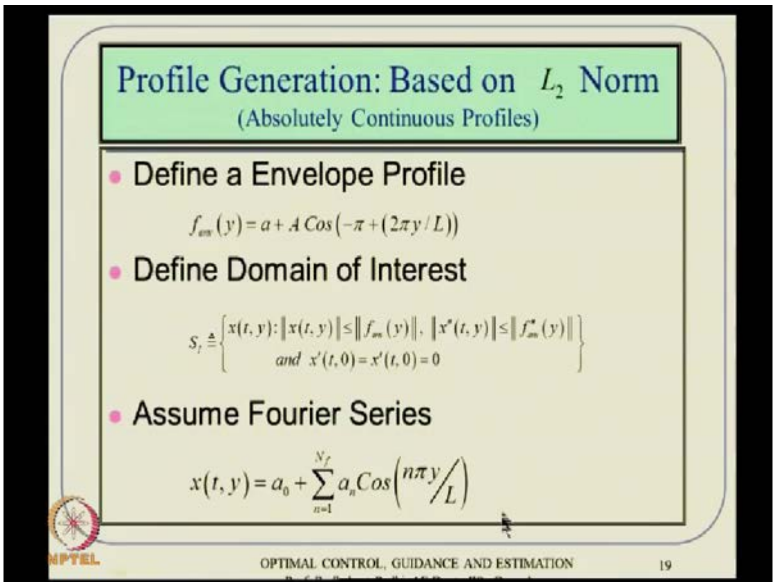
$$E = K.E. + P.E.$$
$$\le K.E._{t_0} + P.E._{t_0}$$
$$\triangleq E_0 \text{ (A Control Design Specification)}$$

- Generate Profiles so that the above condition is satisfied

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 18

The next idea is something like these, something like a energy based and the let say you defined kinetic energy is something like integral of 0 to y f x 2 square and potential energy is this one, del x 1 over del y square d y sort of all thing. This is very naturally it will come from distributed parameter system analysis. So, if it is like that, the problem definition is like that and the implication is like that, then you tell at the maximum that total energy something, which is the nothing, but kinetic energy plus potential energy can deviate by only so much. In another words, my entire energy content of the system is bounded by that value E naught. That has to be specified as a control design specification sort of thing. if that is the case then satisfying this condition and this definition you can generate this smooth profiles in a good way.

(Refer Slide Time: 14:53)



Profile Generation: Based on L_2 Norm

(Absolutely Continuous Profiles)

- Define a Envelope Profile

$$f_{env}(y) = a + A\,Cos\left(-\pi + (2\pi y / L)\right)$$

- Define Domain of Interest

$$S_t \triangleq \left\{ \begin{array}{c} x(t,y) : \left\| x(t,y) \right\| \le \left\| f_{env}(y) \right\|,\ \left\| x''(t,y) \right\| \le \left\| f''_{env}(y) \right\| \\ and\ \ x'(t,0) = x'(t,0) = 0 \end{array} \right\}$$

- Assume Fourier Series

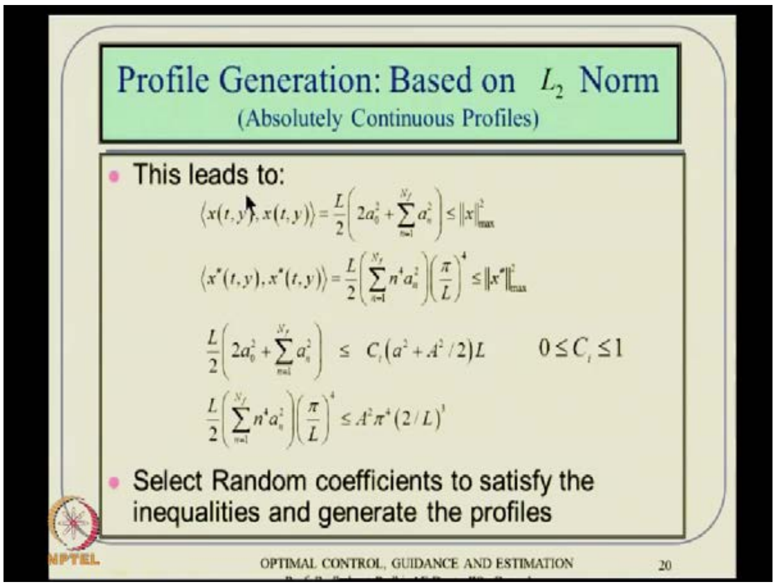
$$x(t,y) = a_0 + \sum_{n=1}^{N_f} a_n Cos\left({n\pi y}/{L} \right)$$

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 19

How do you do that? Let us talk about something like a function. You first define like an envelope profile. It does not mean anything. All that it means is depending on this envelope profile whatever (( )) these all, whatever E naught, I will calculate using this formula, K not plus P not that is going to give me E naught value. So, thus the meaning of this envelope profile. and this was as I just selected for a different reason in a different problem. So, you can vary according to your wish or you can simply fix number also for that. This does not have to.

Next, you define domain of interest something like these x of t y is such that x of t y norm has to be less than equal to some f envelope of f y. This is where the utility comes even more and more. Once you know this one then this numbers can be selected in a good way. In this particular example, you take position thing that has to be bounded by that kind of thing, as well as double special derivative is bounded by this double spatial derivative, which is come out to that and this conditions also has to be 0 basically.

(Refer Slide Time: 16:08)



Profile Generation: Based on L_2 Norm

(Absolutely Continuous Profiles)

- This leads to:

$$\langle x(t,y), x(t,y)\rangle = \frac{L}{2}\left(2a_0^2 + \sum_{n=1}^{N_f} a_n^2\right) \le \|x\|_{\max}^2$$

$$\langle x''(t,y), x''(t,y)\rangle = \frac{L}{2}\left(\sum_{n=1}^{N_f} n^4 a_n^2\right)\left(\frac{\pi}{L}\right)^4 \le \|x''\|_{\max}^2$$

$$\frac{L}{2}\left(2a_0^2 + \sum_{n=1}^{N_f} a_n^2\right) \le C_i(a^2 + A^2/2)L \qquad 0 \le C_i \le 1$$

$$\frac{L}{2}\left(\sum_{n=1}^{N_f} n^4 a_n^2\right)\left(\frac{\pi}{L}\right)^4 \le A^2\pi^4(2/L)^3$$

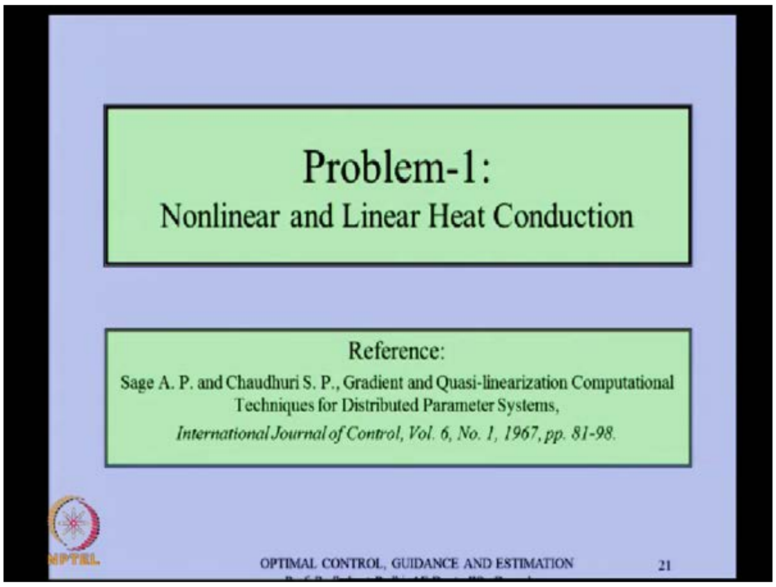
- Select Random coefficients to satisfy the inequalities and generate the profiles

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 20

Then here is the trick and the trick is we assume fourier series in this form and then this fourier series, this leads to this, I mean if I compute this inner product, it turns out to be this expression and by definition of our initial conditions profile generation domain, this value this inner product out is less than certain maximum value, which is fixed number. That means this constraint equation has to be satisfied. Similarly, if you compute this inner product, this is the expression and this has to be less than a fixed number which is also coming out of that. The numbers have to be compatible, remember that. You cannot put arbitrary numbers just for fun in all that way and that is where this envelope profiles help us to gain gets a compatible number set and getting that.

What is the idea here? Now, in this particular case, we have to generate this, a naught to a n, from a naught, a 1, a 2, to a n, in such a way thus these two equalities have to be satisfied. Now, this represent, remember this value, x max square turns out to be like this and x dash norm x square turns out to be like these. So, satisfying these inequalities we have to generate numbers. Now, once you generate number satisfying the some conditions like this; finally remember this way to put it back in this formula, which will give us a smooth profile. So, that becomes much more meaningful and the algorithm sense, let me not took talk too much. It is available, I mean, if somebody interested can think a little bit and plug the algorithmic there.

(Refer Slide Time: 17:32)



Problem-1:
Nonlinear and Linear Heat Conduction

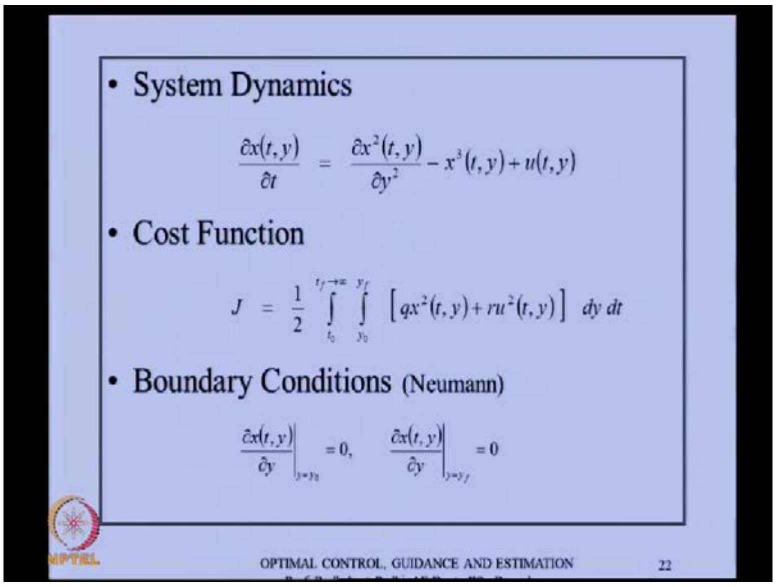
Reference:
Sage A. P. and Chaudhuri S. P., Gradient and Quasi-linearization Computational Techniques for Distributed Parameter Systems, *International Journal of Control, Vol. 6, No. 1, 1967, pp. 81-98.*

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 21

Now, coming to the problem one, which is of non-linear and linear heat conduction and the reason for taking this problem is like these. This is initial stage of these developments of the methodology and things like that. We wanted to have confidence on the problem and you also wanted to have some sort of comparison study with respect to a known result sort of thing.

This known results that are typically available from operational theory, I mean infinite dimensional operation operator theory, but they are available only for linear system. So, when we went to this one, I mean this international journal of control and things like that and I mean you are getting some sort of inspiration from there and we want to experiment on that. Also there are infinite dimensional distributed parameter system and control theory books available and from that book some results can be obtained from there for at least for comparisons.

(Refer Slide Time: 18:26)



- System Dynamics

$$\frac{\partial x(t,y)}{\partial t} = \frac{\partial x^2(t,y)}{\partial y^2} - x^3(t,y) + u(t,y)$$

- Cost Function

$$J = \frac{1}{2}\int_{t_0}^{t_f\to\infty}\int_{y_0}^{y_f}\left[qx^2(t,y) + ru^2(t,y)\right]\,dy\,dt$$

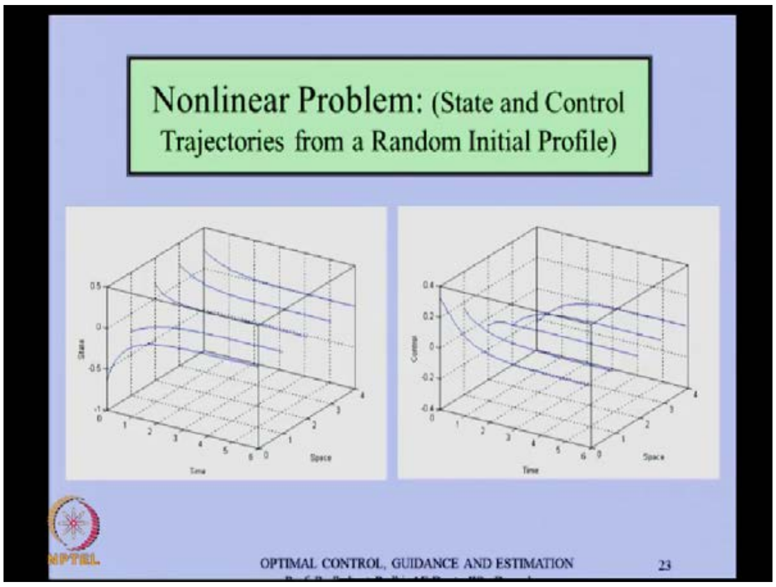
- Boundary Conditions (Neumann)

$$\left.\frac{\partial x(t,y)}{\partial y}\right|_{y=y_0} = 0, \qquad \left.\frac{\partial x(t,y)}{\partial y}\right|_{y=y_f} = 0$$

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 22

Now, the system dynamics is something like this; very close to what we seen before, but the fact that if you ignore this minus x cube term, it turns out to be the same problem, what we started with this in the previous lecture in the very beginning slide if you see that. So, this particular minus x cube introducing in this term makes this system dynamic non-linear. Cost function is very similar to what we started again, the quadratic cost function. Boundary conditions are also similar, now both Neumann boundary condition. So, if you ignore this term, we have some way of comparing this thing with respect to the previous results that you already have using LQR theory essentially. If you ignore this term, we know how to kind of formulate an LQR theory in grid point. So, somewhere the results should lead as to same value sort of thing.

(Refer Slide Time: 19:27)

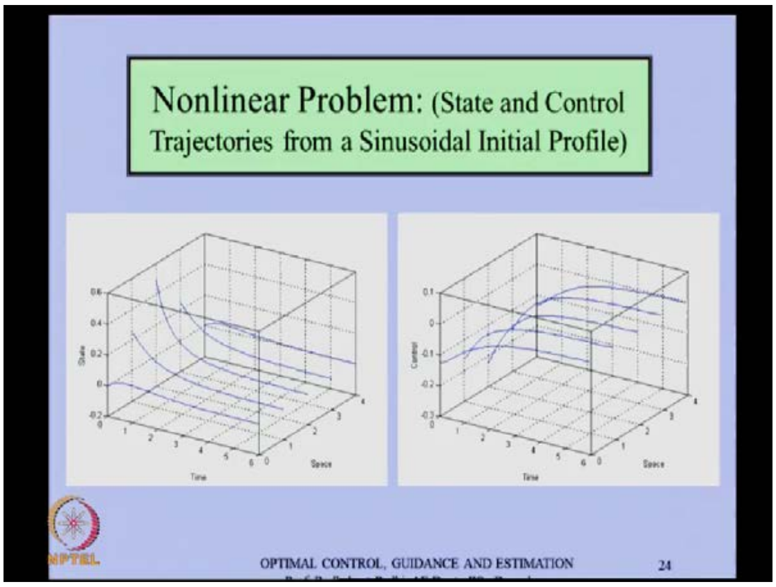


Nonlinear Problem: (State and Control Trajectories from a Random Initial Profile)

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 23

So, going back to that and this is what it is, the result tends out to be like that. You start with initial random numbers and all; finally it will bulge on to kind of same value everywhere. Essentially, this is a deviation problem, so everywhere thing has to go to 0. Remember, this line represents the 0 line. So, every profile has to boil down to this particular line which is 0. That is what happens and that is what happening and associated control values also you can see depending on the deviation it takes about the kind of an opposite value of control and thing like that. They are all with respect to the non-linear problem; remember that so we cannot have a real comparison sort of thing basically.

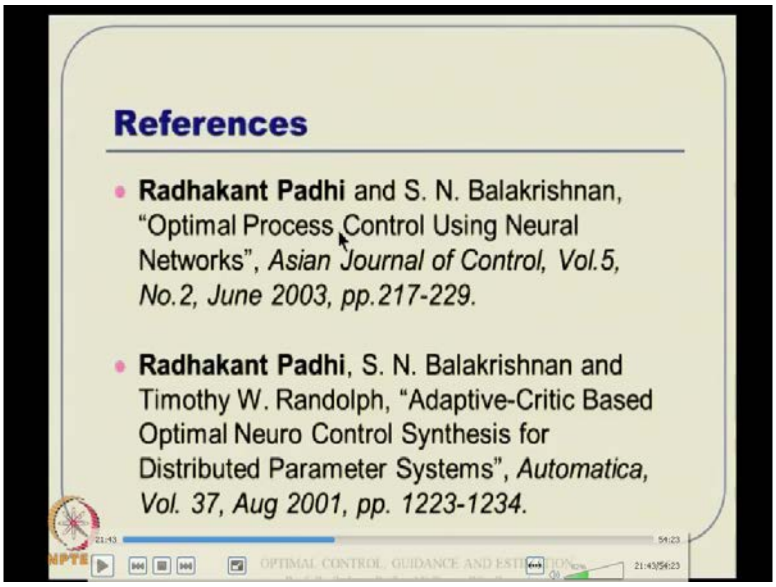
(Refer Slide Time: 20:08)



Nonlinear Problem: (State and Control Trajectories from a Sinusoidal Initial Profile)

So, again little more validation sort of things from a sinusoidal profile and even if it is continuous profile, we are doing experiment with respect to the grid points values only. So, we do not have any idea about what is happening in between it. No matter, if even if its initial condition which is sinusoidal finally, everything goes down to I mean everything decreases and it merge down to be 0. Now, somebody can argue that is a dissipating system anyway. In this system dynamics, this is a stable term is stabilizing term and this is also a dissipater term; that means, this is stable dynamics anyway. So, what is the big deal? Now, if you really stimulate this part of the system dynamics well it will take you to 0, but it will take you in a different way, I mean it takes some more time to take there.

(Refer Slide Time: 21:01)

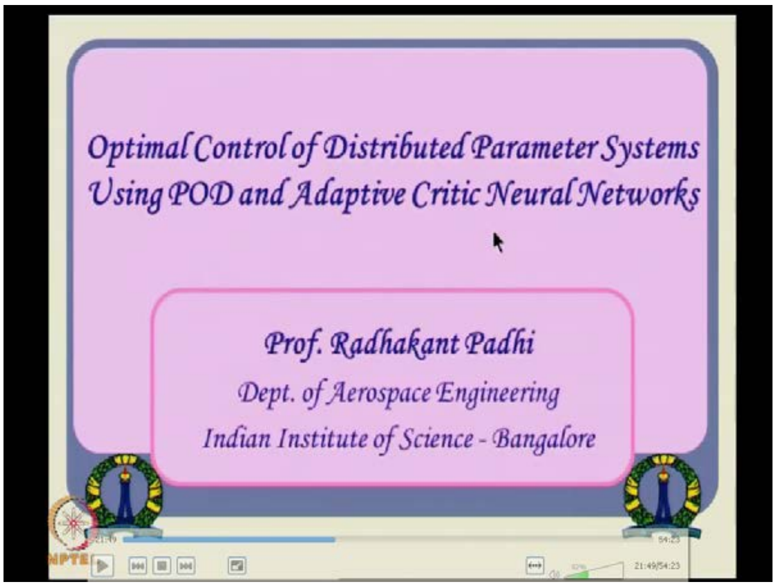


References

- **Radhakant Padhi** and S. N. Balakrishnan, "Optimal Process Control Using Neural Networks", *Asian Journal of Control, Vol.5, No.2, June 2003, pp.217-229.*
- **Radhakant Padhi**, S. N. Balakrishnan and Timothy W. Randolph, "Adaptive-Critic Based Optimal Neuro Control Synthesis for Distributed Parameter Systems", *Automatica, Vol. 37, Aug 2001, pp. 1223-1234.*

More on that I think this particular idea was not pursued too much heavily because there is no modal reduction concept inside it. Hence, it is not very elegant really because infinite dimensional theory has to as some sort of modal reduction to make the system dynamics or synthesis procedure more elegant. But, in fact, somebody is interested in looking more in that you can read some of these journal papers. As I told the very first paper happen to be into 2012 in Automatica. The next one is in 2003 in Asian Journal of Control. So, if you want to see some of these details you can see that. This is a conceptual paper and this is little bit paper realistic paper with some process control problem.

(Refer Slide Time: 21:49)


Optimal Control of Distributed Parameter Systems Using POD and Adaptive Critic Neural Networks

Prof. Radhakant Padhi

Dept. of Aerospace Engineering

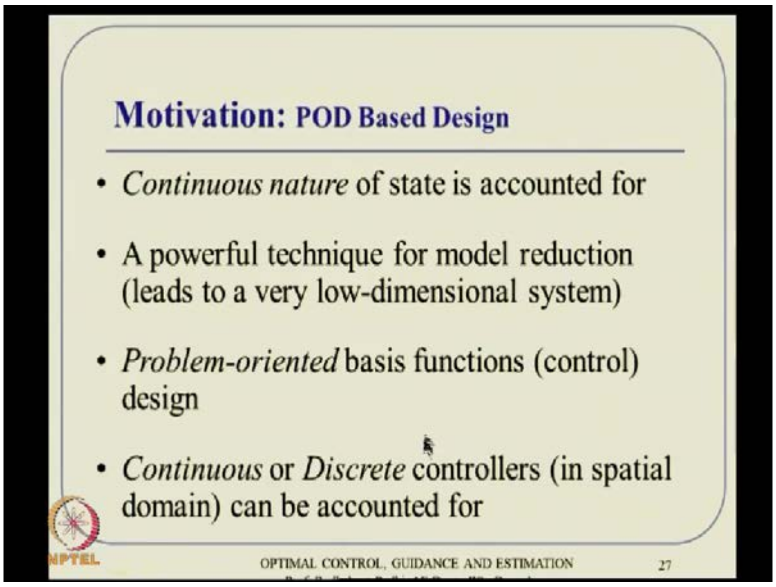
Indian Institute of Science - Bangalore

Now, coming to the next idea, this is where the modal reduction concept is coming awake and here it is what you called as proper orthogonal decomposition and that will nothing to do with control design. This will essentially gives us some sort of basis function, which on using should be able to do a better job. If you do not use that you can still use a Galerkin projection method and then use this generic basic function, but that will again result in large number of states variables, which we do not typically want it. But, on other hand, this POD is not a miracle technique either. Now, if you ask me this POD, the proper orthogonal decomposition ideas appear heavily in homogenous system dynamic solution and that is very much valid, very much justified, because that solution nature is not changing anything.

I mean there is no control input. So, there is nothing like the solution behavior depends on the control input, so there is no feed back there; that means, every time you solve it from different initial conditions, the solution is resulted is unique and depends on the system dynamics. Those problems are ok. But, when you talk about control in the loop then we all know that the open loop behavior and close loop behavior are two different ball game altogether. That means, openly behavior can be stable and close loop behavior can be unstable. Sorry, open loop behavior can be unstable, close loop behavior can be stable and much more beyond that.

So, utilizing the… how do you generate the POD bases function? Because, essentially you need some sort of a random representative solution profile to begin with. But, solution profile itself serves as a function of control and hence it is not available, before synthesizing the control. And that becomes a kind of loop and hence people have kind of proposed something called Re-POD and then online adaptive POD; all sort of things. Anyway, will not talk too much on that, we will assume that the close loop system dynamics behavior can somewhat be captured in the representative solutions set and hence utilizing the solution set, we can generate POD basis function set also.

(Refer Slide Time: 23:59)



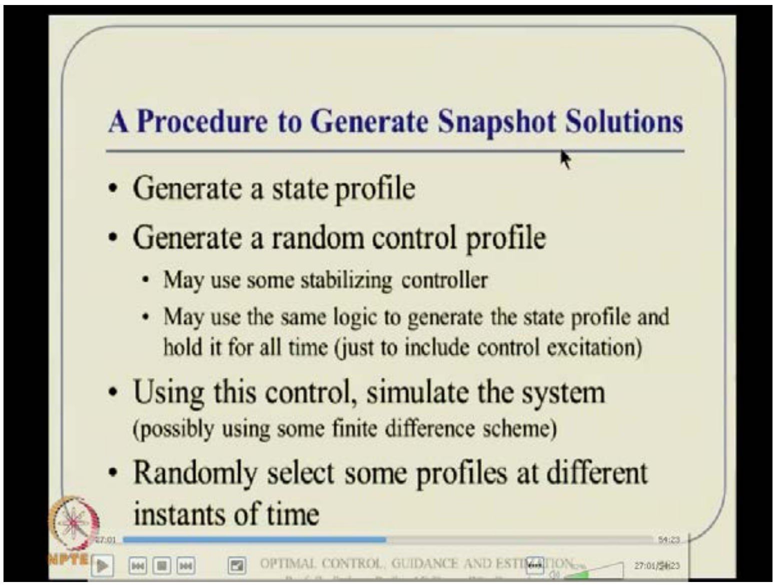
Motivation: POD Based Design

- *Continuous nature* of state is accounted for
- A powerful technique for model reduction (leads to a very low-dimensional system)
- *Problem-oriented* basis functions (control) design
- *Continuous* or *Discrete* controllers (in spatial domain) can be accounted for

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 27

Let us continue and get started with that. The motivation for that is first of all continuous nature of the state is accounted for, unlike the earlier design where only we bothered about discrete point behavior. The second one is it is a powerful model reduction technique; it leads to some sort of very low dimensional system. Why this become low dimension is because we do not deal with generic basis function, but we rather deal with problem oriented basic function for control design. That is the key reason why it becomes, I mean why it result low dimensional representation really. Then it also is experimented with both continuous actuators as well as discrete actuators. So, we will not talk too much on discrete actuators in this class, but continuous actuators sense we will talk about and then more and that you can read out to find out.

(Refer Slide Time: 24:50)


A Procedure to Generate Snapshot Solutions

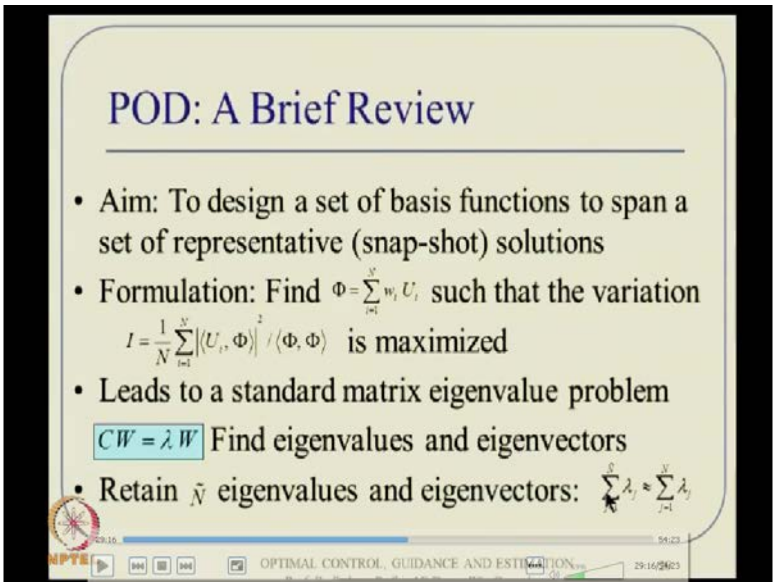
- Generate a state profile
- Generate a random control profile
  - May use some stabilizing controller
  - May use the same logic to generate the state profile and hold it for all time (just to include control excitation)
- Using this control, simulate the system (possibly using some finite difference scheme)
- Randomly select some profiles at different instants of time

The very first thing as I told is to starting with some representative solution, which is called as snapshot solutions. So, this is fundamental, I mean key things to design a good set of basis vector and once you come up with this basis vector set properly, everything else turns out to be very good. If that itself is not good then everything else will turn out to be not so good. So, be carefully about that, when we talk about generating snapshot solutions. So, how do you do that? First you generate the state profile and then generate a random control profile. Now, when you talk about profiles here, remember this are distributed above space basically. So, generate some sort of a state profile as an initial condition and then generates some sort of random control profile also as some sort of a control history. But, remember, you cannot generate very randomly.

Because, if you really put random control then the system dynamic behavior can be very bad; it can go to instability behavior and we do not want that. With application of control, whatever dynamic it results in the close loop, we want to mimic that, in some sense. So, for that reason may be use some sort of stabilizing controller or we can also use some sort of logic to generate the state profile and just hold it for some time. You do not allow it go to or keep on implementing for a long time and all that. That means just to include a little bit of control excitation, you want to see that, even if it is unstable little bit, it may be still ok in some times.

Then utilizing this control similar to the system dynamics, possibly using some finite difference or finite element, whatever it is, and then you can randomly sample it out of that. That means, this behavior, which evolves with time that also is a lot of profiles. So, you do not want on keep on collecting everything, but randomly you select some profile at different instants of time. Then you will have some sort of snapshot solutions and using which you can designs this basis function.

(Refer Slide Time: 27:04)



POD: A Brief Review

- Aim: To design a set of basis functions to span a set of representative (snap-shot) solutions
- Formulation: Find $\Phi = \sum_{i=1}^{N} w_i U_i$ such that the variation $I = \frac{1}{N}\sum_{i=1}^{N} \left|\langle U_i, \Phi\rangle\right|^2 / \langle \Phi, \Phi\rangle$ is maximized
- Leads to a standard matrix eigenvalue problem $\boxed{CW = \lambda W}$ Find eigenvalues and eigenvectors
- Retain $\tilde{N}$ eigenvalues and eigenvectors: $\sum_{j=1}^{\tilde{N}} \lambda_j \approx \sum_{j=1}^{N} \lambda_j$

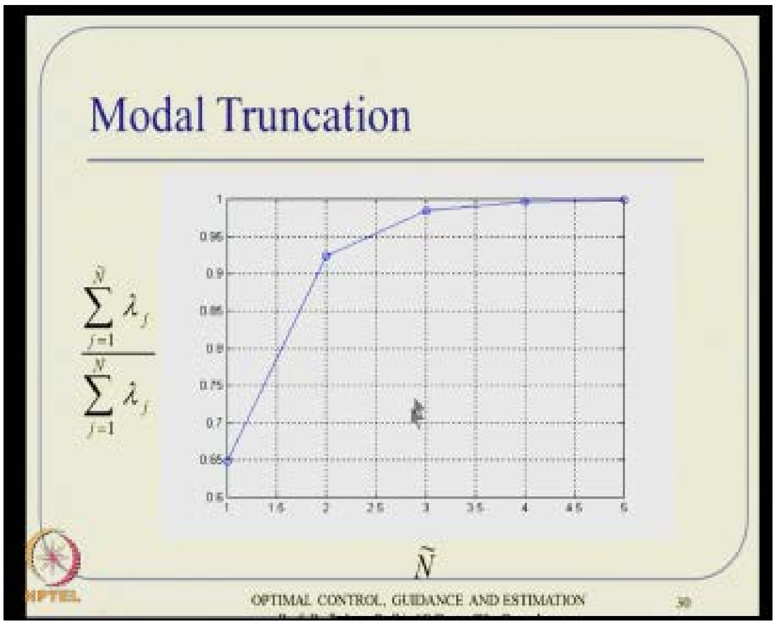
Now, a small outline, again the math details I will not touch it, but the small outline is like this. Once, you have a set of snapshot solutions then you come here and ask these question. Can I find the basis vector, sorry basis function phi in this way, where U i is nothing but, the snapshot solution. Can I find phi as a linear communication of these snapshot solution, such that this function what you see here, the integral value that means, this summation over this U i inner product phi and then absolute value square and then normalized that.

Now, that represents the spread of this basis function really. So, if I talk about, I got the basis function then how well it can represent my solution set and that is captured by maximizing this cross function; again maximizing this, not minimizing it. So, if you go through the algebra, which is readily available in number of test paper and it turns out that it leads to some sort of standard matrix Eigen value problem and this Eigen value problem is nothing but the CW equal to lambda W.

If you have that kind of thing; obviously, this is N number of Eigen values and all that, so find Eigen values and Eigen vectors accordingly. Then that N will turn out to be number of snapshot solutions; this N, the same N, but obviously, you do not want that. If you want to retain everything then there is no model reduction. So, it turns out that the very first Eigen values after ordering and all, I mean, if you finding the Eigen values and then order them, because this happens to be a symmetric matrix, it will all happened to be kind of real Eigen values and all that.

So, you can order them and it turns out that very quickly these Eigen value set will converge in this sense; that means, if you start summing of lambda 1 plus lambda 2 lambda 3 lambda 4; then probably even if you some for up to 100, then it does not matter, it will be still approximately the same and it is very quickly will converge to that summation. In other words, their total spread of the basis function set contains within some first couple of basis function.

(Refer Slide Time: 29:19)



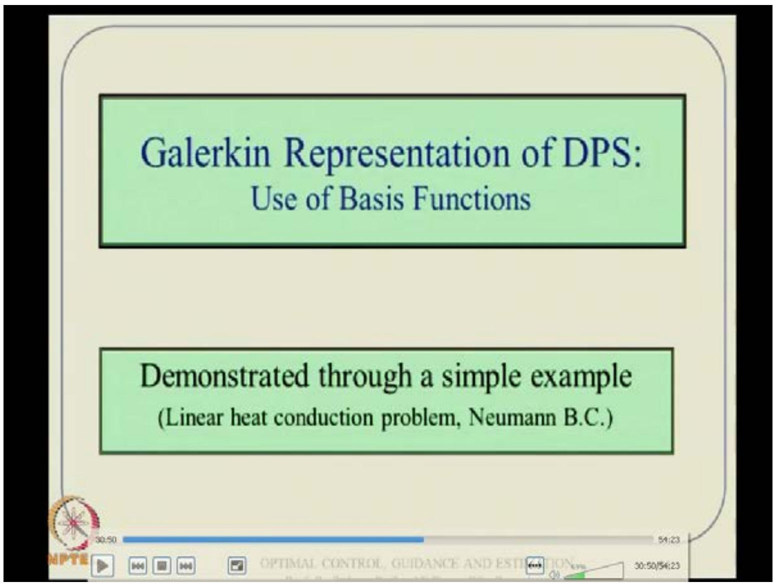
Modal Truncation

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 30

And pictorially you can see that way in this particular problem, if you take just five basis functions, the first one will contain 65 percent. First two will contain about 92 percent, first three will contain about 98 percent like that; 97 percent 97.5 like that and then it is just takes about five basis function, we are done. Even if you collect some 200 basis 200 snapshot solutions to begin with, we will ultimately land up with these five basis function vector and hence five dimensional representations of long parameter systems. It is that powerful. It again depends on the solution, I mean, depends on the representation of the solution through snapshot and that is again the key. What it tells you is capturing the behavior that is captured in the snapshot solution.

Now, the question is whether the snapshot solution really behave, I mean close to the close loop behavior of the system dynamics or not and that is up to us to decide or guess of things like that. If it is close to that, this truncation will happen even very quickly and that means we are done. But, this picture tells everything; even though we started with 200 snapshot solution ultimately five basis function it results into, because of this behavior; this lambda convergence behavior. Hence, the number of s is, I mean the dimension of the lump parameter system happens to be just five.

(Refer Slide Time: 30:45)

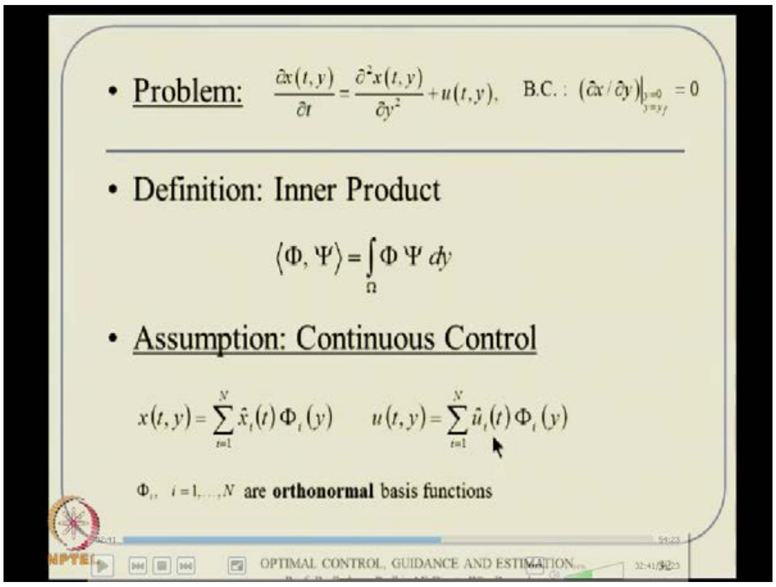


Galerkin Representation of DPS:
Use of Basis Functions

Demonstrated through a simple example
(Linear heat conduction problem, Neumann B.C.)

Then after you get the basis function, finally, you have to go through them for some order reduction procedure. In other words, from PDE to ODE, you have to get it using this basis function and all that. So, generication it is becomes mathematically complex anal I will write or I mean I will kind of demonstrate that using a simple a example.

(Refer Slide Time: 31:04)



- **Problem:** $\frac{\partial x(t,y)}{\partial t} = \frac{\partial^2 x(t,y)}{\partial y^2} + u(t,y)$, B.C.: $(\partial x/\partial y)\big|_{y=0, y=y_f} = 0$
- **Definition: Inner Product**

$$\langle \Phi, \Psi \rangle = \int_{\Omega} \Phi \Psi \, dy$$

- **Assumption: Continuous Control**

$$x(t,y) = \sum_{i=1}^{N} \hat{x}_i(t)\Phi_i(y) \qquad u(t,y) = \sum_{i=1}^{N} \hat{u}_i(t)\Phi_i(y)$$

Φ_i, $i = 1, \ldots, N$ are **orthonormal** basis functions

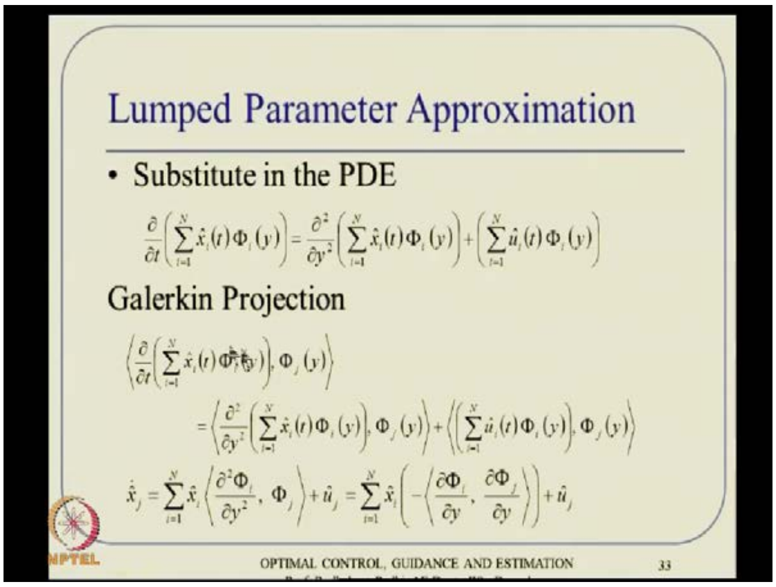
OPTIMAL CONTROL, GUIDANCE AND ESTIMATION

We will go back to the same equation with linear heat equation with Neumann boundary condition. We started with this first example problem of the previous lecture. So, this is where the fundamental understanding happens. So, we start with this symbol linear PDE with Neumann boundary condition. And when we talk about inner product, the inner product definition in everything, what we talk here is in continuous time, I mean in continuous space it is something like this. When inner product of phi tends to psi, just by and sign multiplication integrates over domain and that is the definition of inner product.

Then here we assume continuous controls; that mean, this state and this control we expand in this form. This is a model for representation. Now, this phi’s are basis function that is available through this analysis and all that, whatever this solution set that will give you something there. Anyway, we will get to that. Once, you have the set of basis functions, this phi is from 1 to N and this case it is one to phi. So, x is the state vector or state vector is expanded as purely a function of time, times purely a function of space. This is separation of variable sort of ideas.

Similarly, control is a purely function of time and purely function of space. Now, the function of space is already computed, because of this orthogonal basis functions and all. What remains to be computed is something line u hat of and x hat; that means, x given and x of t y, we should be able to find out what is x i hat of t at every i. Then based on that we should to be able compute this u i hat and based on that we have to get continuous controller. That is the overall idea there.

(Refer Slide Time: 32:48)



Lumped Parameter Approximation

- Substitute in the PDE

$$\frac{\partial}{\partial t}\left(\sum_{i=1}^{N}\hat{x}_i(t)\Phi_i(y)\right)=\frac{\partial^2}{\partial y^2}\left(\sum_{i=1}^{N}\hat{x}_i(t)\Phi_i(y)\right)+\left(\sum_{i=1}^{N}\hat{u}_i(t)\Phi_i(y)\right)$$

Galerkin Projection

$$\left\langle\frac{\partial}{\partial t}\left(\sum_{i=1}^{N}\hat{x}_i(t)\Phi_i(y)\right),\Phi_j(y)\right\rangle$$

$$=\left\langle\frac{\partial^2}{\partial y^2}\left(\sum_{i=1}^{N}\hat{x}_i(t)\Phi_i(y)\right),\Phi_j(y)\right\rangle+\left\langle\left(\sum_{i=1}^{N}\hat{u}_i(t)\Phi_i(y)\right),\Phi_j(y)\right\rangle$$

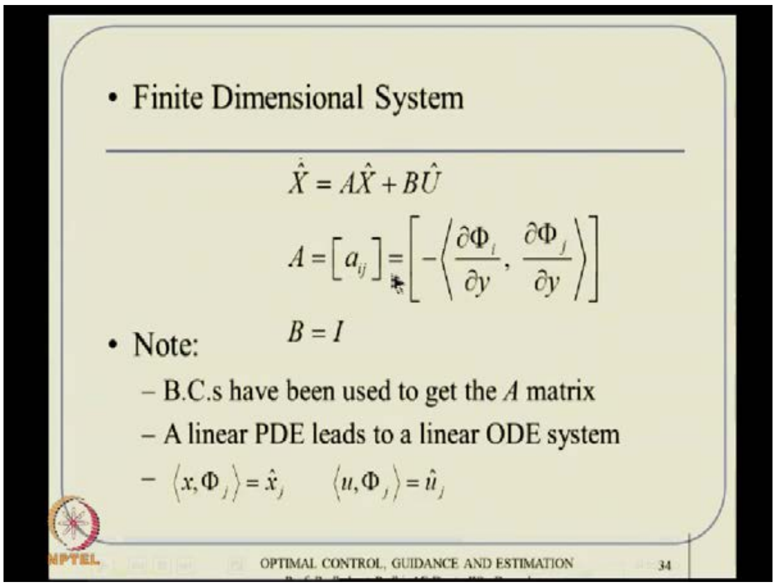
$$\dot{\hat{x}}_j=\sum_{i=1}^{N}\hat{x}_i\left\langle\frac{\partial^2\Phi_i}{\partial y^2},\Phi_j\right\rangle+\hat{u}_j=\sum_{i=1}^{N}\hat{x}_i\left(-\left\langle\frac{\partial\Phi_i}{\partial y},\frac{\partial\Phi_j}{\partial y}\right\rangle\right)+\hat{u}_j$$

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 33

How do we do that? First of all, we have this system dynamics. In this expansion, this expansion you can put back here and then this will result in that sort of representation. Now, we talk about inner product with respect to phi j. So, if we do inner product with respect to phi j, remember these are all orthogonal basis function; that means, unless i equal to j, the inner product is 0. If i equal to j, I mean this is normalized sort of thing, but typically this is sometimes one if it is normalized are already or other otherwise particular value. So, this represents inner product. Let say it is, let me talk about that. These are orthonormal basis functions, not necessarily orthogonal, so that the orthonormal means if inner product phi i and phi j happens to be 0 unless i equal to j and it i equal to j this is one.

So, utilizing that behavior here and noting that summation, this partial derivative everything happen, is an inner product also. These are all linear operator, you can carry out the algebra and ultimately it results in some sort of equations like this. And these are numbers, remember that. This is not, see thee basis functions themselves are orthogonal, but once you take special derivatives and then take inner product, they are not orthogonal. So, you have to really compute it numerically using trapezoidal rule and thing like that. So, this cannot be represented through delta function and all that. So, this has to be evaluated explicitly. Similarly, this also you have to be evaluated explicitly. But, none the less, if you think about doing that, this one either these are that, it can be actually evaluated explicitly. That is not much of a big deal.

(Refer Slide Time: 34:42



- Finite Dimensional System

$$\dot{\hat{X}} = A\hat{X} + B\hat{U}$$

$$A = \left[a_{ij}\right] = \left[-\left\langle \frac{\partial \Phi_i}{\partial y}, \frac{\partial \Phi_j}{\partial y} \right\rangle\right]$$

$$B = I$$

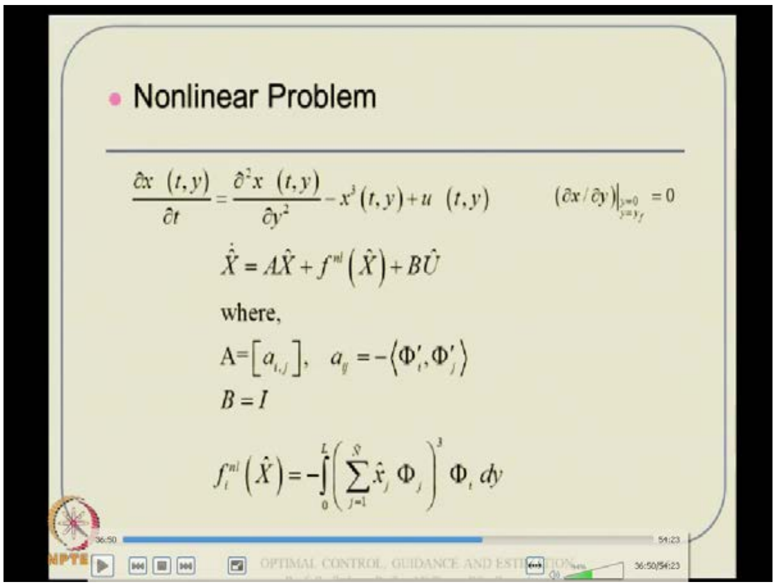
- Note:
  - B.C.s have been used to get the A matrix
  - A linear PDE leads to a linear ODE system
  - $\langle x, \Phi_j \rangle = \hat{x}_j \qquad \langle u, \Phi_j \rangle = \hat{u}_j$

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 34

Now, it will ultimately result in a finite dimensional system and then final dimensional system will tell us that this is the form where X hat of dot is AX hat plus BU hat; where A matrix, the component of a i j can be computed that way. B happens to be identity. So, ultimately we got representations in the finite dimensional space, lump parameter representation really. But, this X hat does not represent any grid point solution. This components of X hat which is x 1 hat to x 2 hat and things like that, they will go and sit here.

And once everything is known, once you put it back in this formula, then u will get x of t y, similarly, for the control. But, individually it represents the magnitude that will go along with that basis vector, basis function really. That is the meaning of this time varying numbers. This is the meaning and if you really want to extract these x i hat, which is a function of time then again utilizing this ortho-normality behavior and all, it turns out that inner product of this and that is nothing but the inner product of u and phi j which is u hat j basically. So, given an x, you first compute through this inner product, this follow. Now, our job is to somehow connect this two. These are time domain behavior. Once you know this, we have to find out this and once you know that we go back and tell that I got my control throughout. That is the whole idea there.

(Refer Slide Time: 36:20)



- Nonlinear Problem

$$\frac{\partial x}{\partial t}(t,y) = \frac{\partial^2 x}{\partial y^2}(t,y) - x^3(t,y) + u(t,y) \qquad (\partial x/\partial y)\big|_{\substack{y=0\\ y=y_f}} = 0$$

$$\dot{\hat{X}} = A\hat{X} + f^{nl}(\hat{X}) + B\hat{U}$$

where,

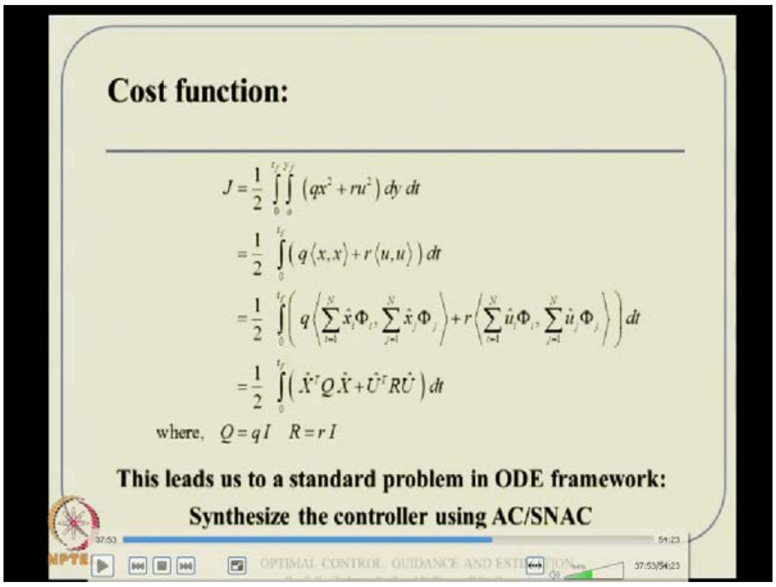
$$A = [a_{i,j}], \quad a_{ij} = -\langle \Phi'_i, \Phi'_j \rangle$$

$$B = I$$

$$f_i^{nl}(\hat{X}) = -\int_0^L \left(\sum_{j=1}^{N} \hat{x}_j \Phi_j \right)^3 \Phi_i \, dy$$

So, given a non-linear problem, again it will result to a similar sort of things, but there will be an extract term here. This extract term will be something like this, because of this minus x cube will have an additional non-linear term and that non-linear term turns out be something like these. By the way, this problem has also been solved using this theta d method and all that. Those of you interested, you can see that in one of our publications in optimal control applications and methods. So, that is a different approach all together, but here we are talking about solving it using a SNAC; single network adaptive critic.

(Refer Slide Time: 36:55)



Cost function:

$$J = \frac{1}{2} \int_0^{t_f} \int_0^{y_f} \left(qx^2 + ru^2 \right) dy\, dt$$

$$= \frac{1}{2} \int_0^{t_f} \left(q \langle x, x \rangle + r \langle u, u \rangle \right) dt$$

$$= \frac{1}{2} \int_0^{t_f} \left(q \left\langle \sum_{i=1}^{N} \hat{x}_i \Phi_i, \sum_{j=1}^{N} \hat{x}_j \Phi_j \right\rangle + r \left\langle \sum_{i=1}^{N} \hat{u}_i \Phi_i, \sum_{j=1}^{N} \hat{u}_j \Phi_j \right\rangle \right) dt$$

$$= \frac{1}{2} \int_0^{t_f} \left(\hat{X}^T Q \hat{X} + \hat{U}^T R \hat{U} \right) dt$$

where, $Q = qI \quad R = rI$

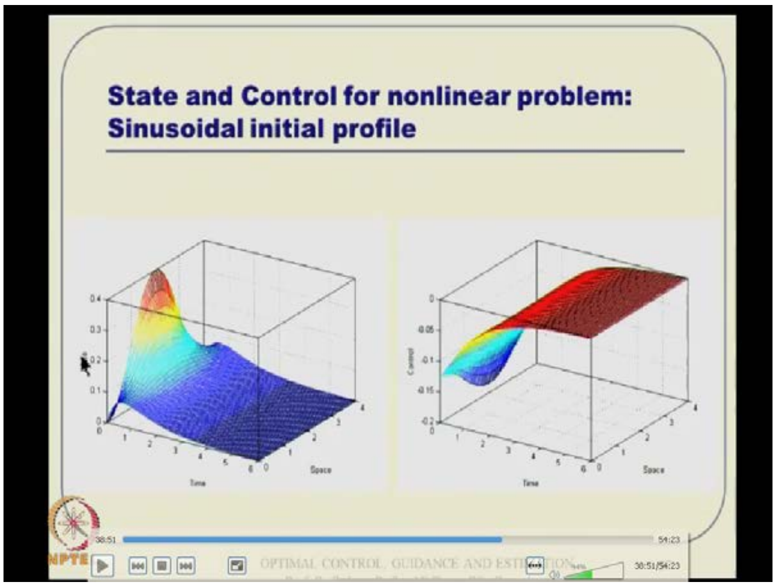
This leads us to a standard problem in ODE framework:

Synthesize the controller using AC/SNAC

For the cost function, again it is like this. But, we have to have a representative cost function in the time domain. So, again this one is nothing but the inner product of x and x, inner product of u and u. Again, substitute the expansion; x is nothing but the expanded way like these and again same thing. Then u will be something like that. You expand it; this summation and that summation and put it together, (( )) again and it turns out that it is not difficult at all; j turns out to be something like this, where Q is nothing q times I and R is r times I.

So, this J what we see here and this system dynamics what we see here, they represent some sort of a regulator problem, but it is a non-linear regulator problem. Also, because this is a control affine to begin with, the PDE also results in some sort of a control affine non-linear system dynamics. Obviously, when we have a control affine non-linear system dynamics and then we have quadratic cost function. You can exceed this idea of adaptive critic or signal network adaptive critic like that and get a better solution.

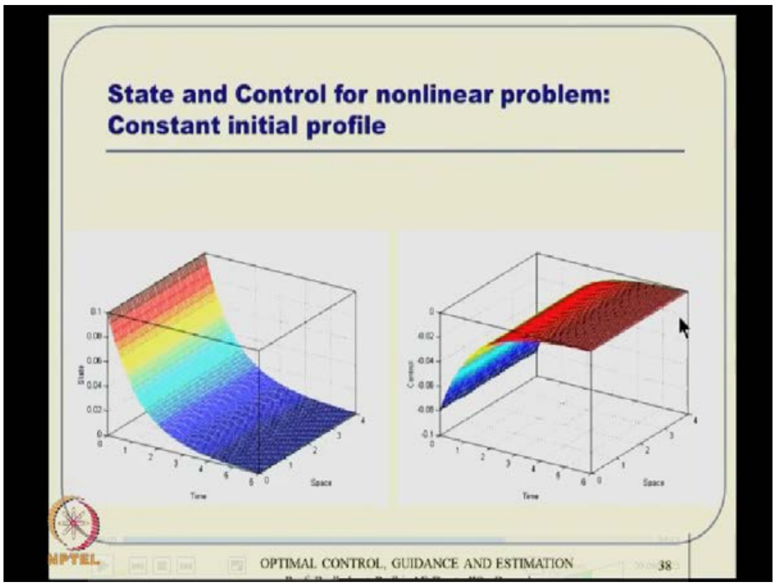
(Refer Slide Time: 37:59)



State and Control for nonlinear problem: Sinusoidal initial profile

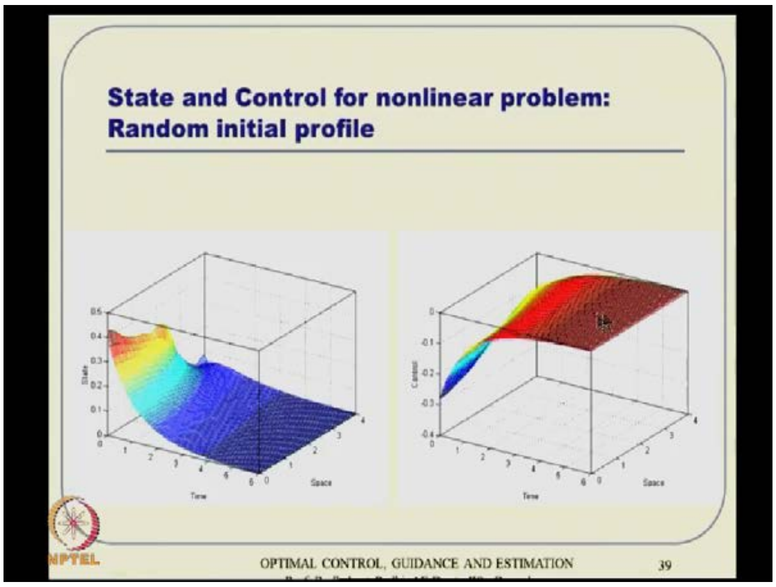
Now, how do you get this sort of results like these? How to the results look like or something like these. You have sinusoidal initial conditions. I am not talking too much details of after this, because we know this compatible set of cost function in lump parameter and this is a system dynamics that goes along with that. We have also talked about how do you generate random initial conditions and all that, little before using two three different ideas. After you generate, how do you come up with this coefficient values, time varying coefficient values that also we discuss. So, (( )) components you discussed already and (( )) we know how to synthesis it. So, I am not talking that particular architecture in detail again. How does the result look like? The results are something like this. You have a sinusoidal initial profile, everything turns out to be 0 very quickly and control also goes to 0 simultaneously.

(Refer Slide Time: 39:02)


State and Control for nonlinear problem: Constant initial profile

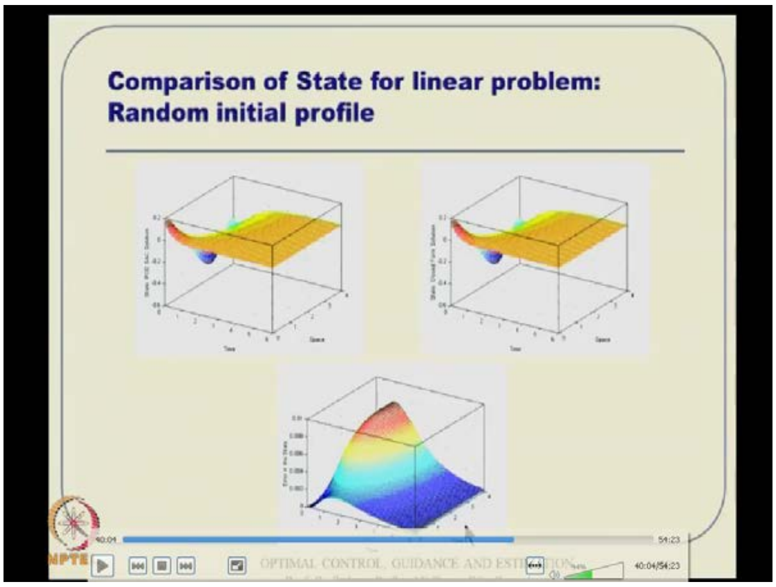
Now, what about this state and control of non-linear problem? From different initial condition including constant it again goes to 0 and here control also goes 0.

(Refer Slide Time: 39:10)


State and Control for nonlinear problem: Random initial profile

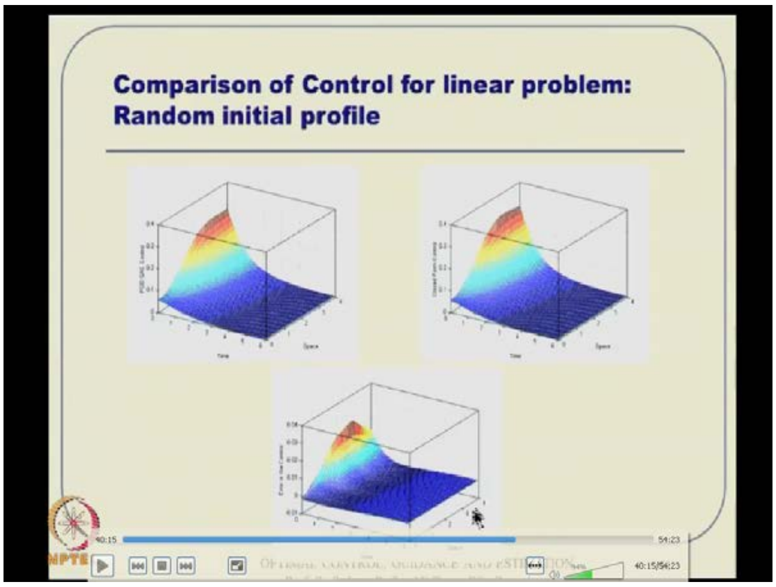
What about this? Some sort of a random profile and you tested for various number of, many number of times. Initial profile is very arbitrary here, but what smooth. So, that one if it is there then something like these, which all happens nicely.

(Refer Slide Time: 39:25)


Comparison of State for linear problem: Random initial profile

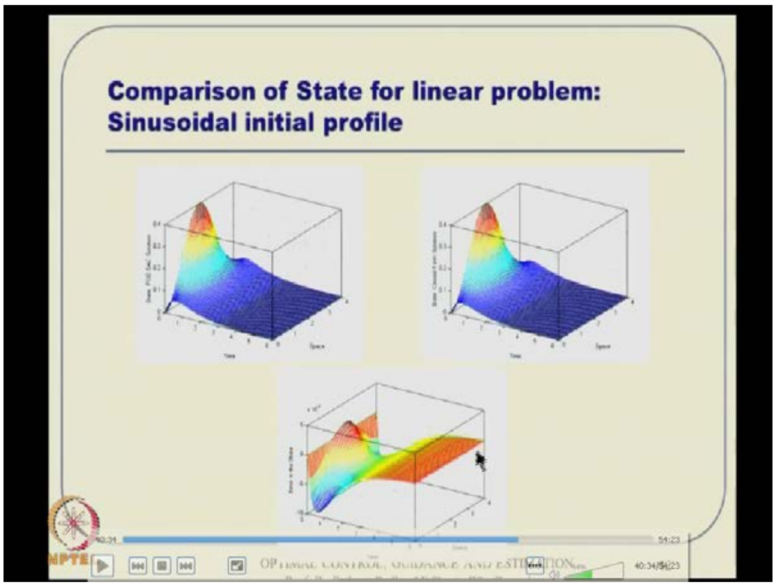
Now, again we come back to the linear problem and then you want to have some sort of comparison with results. These results are taken from a text book of infinite dimensional operator theory based quadratic regulator design and all that. So, this problem has been solved there and using that solution, whatever is proposed there in the particular book we talk about utilizing this. So, this is the (( )) solution and this is the close form solution. Visually you can see very close, but error also you can plot and see error values are very small and it ultimately goes to 0.

(Refer Slide Time: 40:07)


Comparison of Control for linear problem: Random initial profile

Similarly, you can see this from a different initial profile. Again, visually they are same. Error sense also they go to same, very small values, to 0 ultimately.

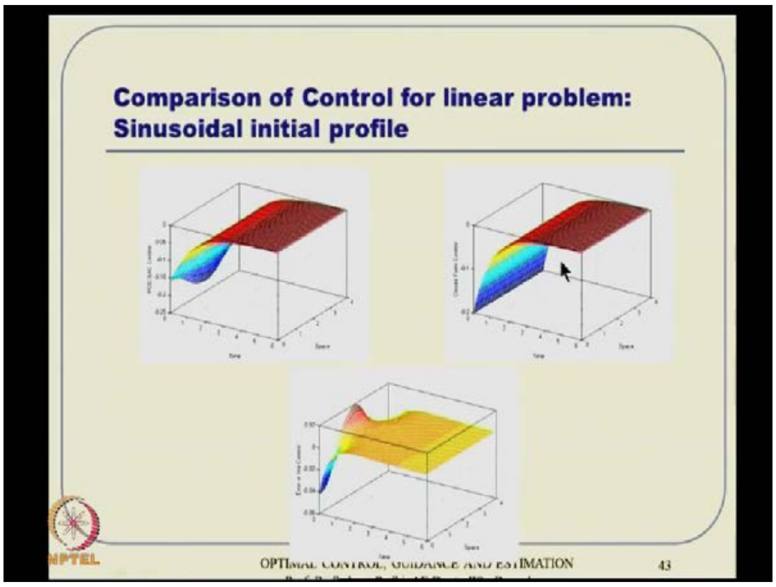
(Refer Slide Time: 40:21)



Comparison of State for linear problem: Sinusoidal initial profile

These are comparison results, which will guarantee that things are happening in a good way. Again, from a different initial conditions, sinusoidal profile where it looks very similar, error also 10 to the power minus 3 here, remember that. (( )) and all that. In that sense, but finally, it is very close to 0. This line is 0 line; you can see that it almost closing that.

(Refer Slide Time: 40:47)

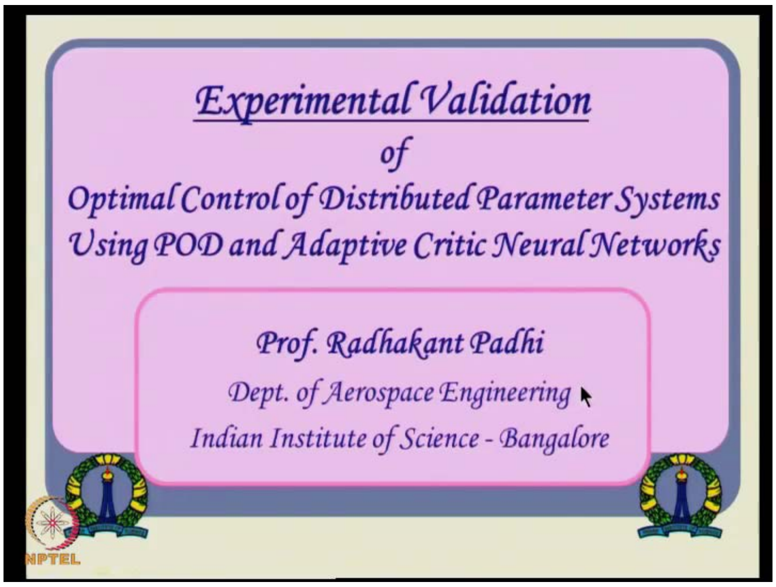


Comparison of Control for linear problem: Sinusoidal initial profile

These are confidence building exercises. Then another one, comparison of control linear problems, sinusoidal initial profile, you have a control comparison here and then even though, it because of this control; this control is not same as that control. Obviously, it not expected to be. This control is very good, because the (( )) see some sort of a constant profile, the close form formula that will automatically do lot of good things there, because it does not approximate anything to begin with.

But, here we always work with an approximate system dynamics. So, there will be some small derivation initially. Ultimately, an asymptotic seems very quickly, they all go nicely and then that way it is not bad. Even if there is a variation, the variation is not really very bad. This variation what you see is 0.2 here, is always 0.2 with a little bit plus, I mean more than little bit lesser than minus 0.2. Sometimes, it will be smaller things like that. But on an around minus 0.2, the average value will turn out to be very close to minus 0.2 only. Error sense, this is what it is. Error things are happening, but very quickly around that time, very quickly the errors are converging to 0. Above two seconds of operation; it turns out to be two or three seconds of operation, errors are gone.

(Refer Slide Time: 42:11)

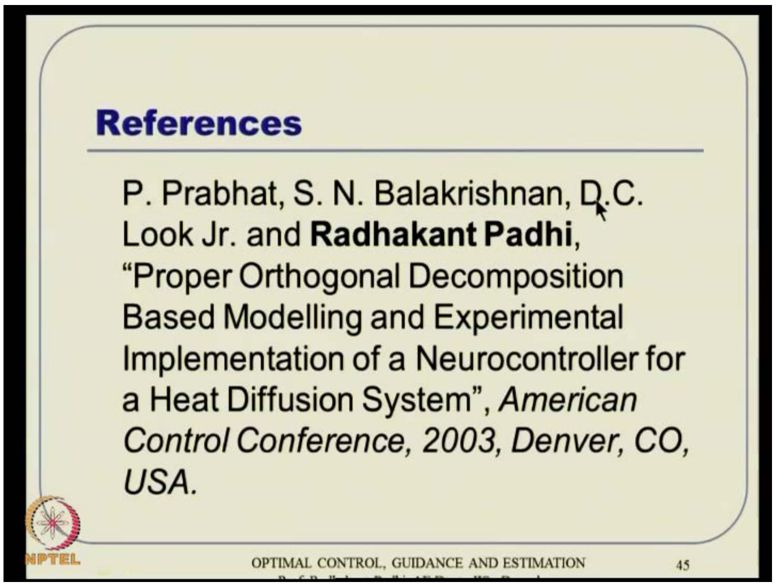


Experimental Validation
of
Optimal Control of Distributed Parameter Systems Using POD and Adaptive Critic Neural Networks

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Now, this idea, both these ideas have been also experimentally validated, not just only with math actually. So, this experimental validation settings also gives us a little more confidence of the ideas that have been proposed.

(Refer Slide Time: 42:26)



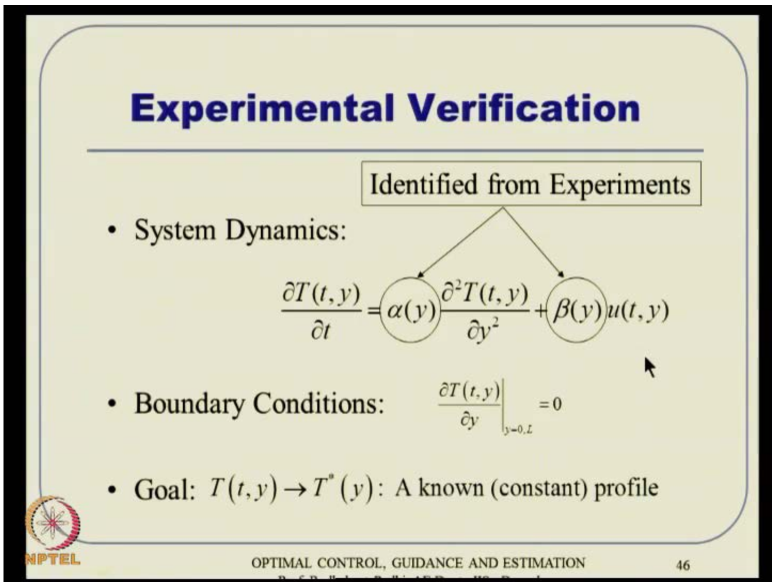
References

P. Prabhat, S. N. Balakrishnan, D.C. Look Jr. and **Radhakant Padhi**, "Proper Orthogonal Decomposition Based Modelling and Experimental Implementation of a Neurocontroller for a Heat Diffusion System", *American Control Conference, 2003, Denver, CO, USA.*

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 45

So, for little details on that you can look at this conference paper if you are interested in.

(Refer Slide Time: 42:33)



Experimental Verification

Identified from Experiments

- System Dynamics:

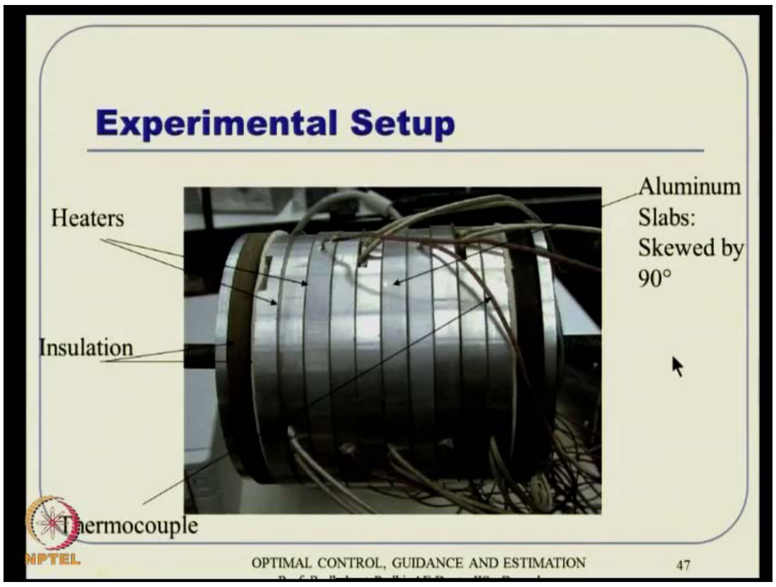
$$\frac{\partial T(t,y)}{\partial t} = \alpha(y)\frac{\partial^2 T(t,y)}{\partial y^2} + \beta(y)u(t,y)$$

- Boundary Conditions: $\left.\frac{\partial T(t,y)}{\partial y}\right|_{y=0,L} = 0$
- Goal: $T(t,y) \to T^*(y)$: A known (constant) profile

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 46

I mean the first problem in any experimental validation turns out to be the modeling accuracy. What we conveniently assumed is alpha equal to 1, beta equal to 1; in our all math analysis is not true. In general, it turns out that alpha is not alpha, beta is not even constant; they are function of y. So, this system identification is first thing to do and that itself is a procedure. We have to do several experiments and things like that. One of my colleague Prasanth Prabhat was a master student. He did lot of these experiments and sense and then made this collaboration. This alpha and beta, those have to be identified from this experiment first and that was carried out first thing.

(Refer Slide Time: 43:22)



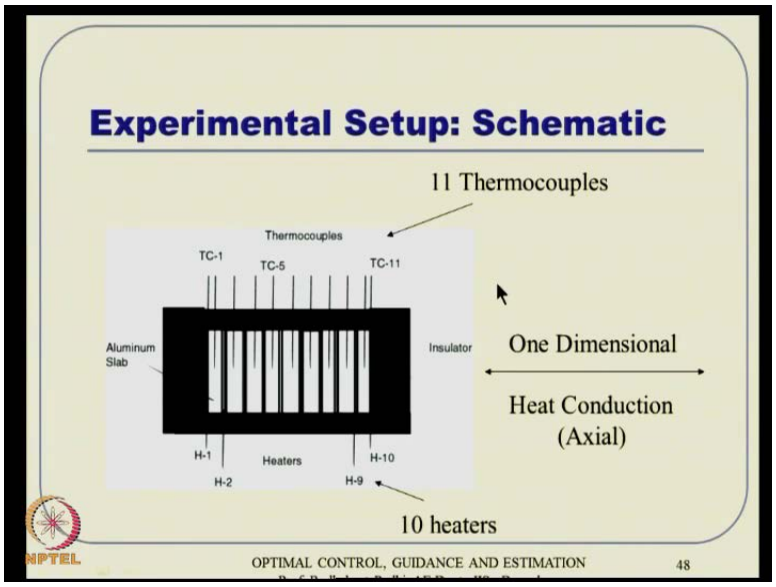
Experimental Setup

Heaters
Insulation
Thermocouple
Aluminum Slabs: Skewed by 90°
NPTEL
OPTIMAL CONTROL, GUIDANCE AND ESTIMATION
47

That I will not talk too much detail on that, of course, and this was the experimental setup. We have this because experimentally we cannot talk about continuous sensors and things like that. So, we have this layers of sensor actuators, sensor actuators two like that and layers of the aluminum slabs, which are, I mean which are setup like this. Also, remember our boundary condition tells that they have to be insulated at both the ends. So, both the ends have to be insulated in some degree of precaution was taken for that also at both ends.

And then if they have to be skewed away; that means, some sensor and actuator cannot be on the same rows sort of thing. Then it may peak of some wrong values and all that. Sometimes, you skew it away from that, so that is ok for some degree of reliability and things like that of the numbers. But, truly I mean this one dimensional representation that we are talking about does not hold good really. This is a cylindrical problem and hence there is a radial dissipation of energy also, not only axial. This is the whole problem actually. I mean, here you are assuming that all slots will attain a equal temperature throughout immediately, which is not really true, but except that the results are very encouraging. So, this is what you see there.

(Refer Slide Time: 44:42)

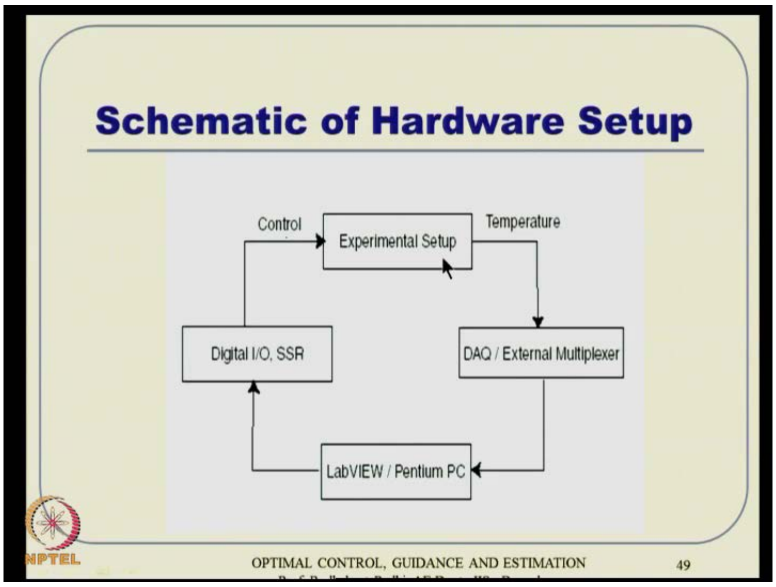


Experimental Setup: Schematic

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 48

This is schematic of this. You have this eleven thermocouples, you have the aluminum slab; you have taken one dimensional heat conduction modeling sort of thing and they were ten heaters like that.

(Refer Slide Time: 44:58)

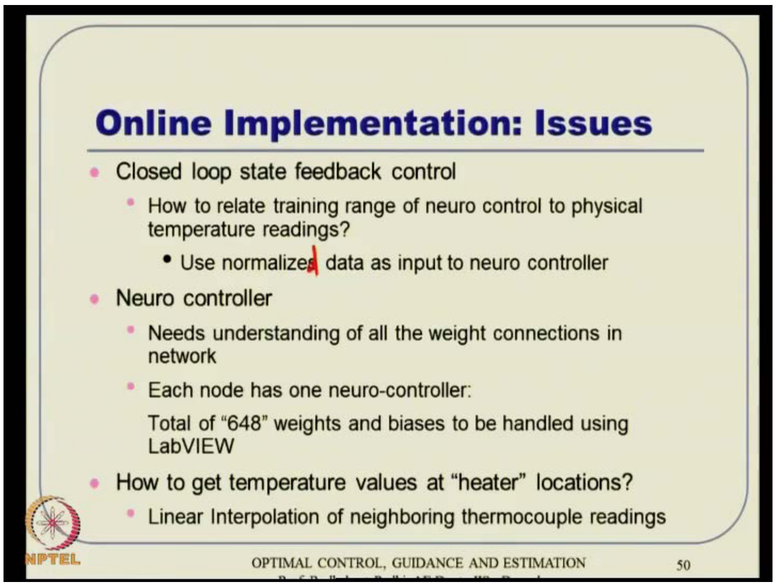


Schematic of Hardware Setup

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 49

There are some hardware settings and this conceptually is like this. You have kind of experimental setup where you have to measure the temperature and there has to be some sort of I mean external multiplex sort of thing. Then you are pass it through this implementation through LabVIEW and that time it was Pentium PC. Then some digital IO and the control input were given to the experimental setup.

(Refer Slide Time: 45:21)



Online Implementation: Issues

- Closed loop state feedback control
  - How to relate training range of neuro control to physical temperature readings?
    - Use normalized data as input to neuro controller
- Neuro controller
  - Needs understanding of all the weight connections in network
  - Each node has one neuro-controller:

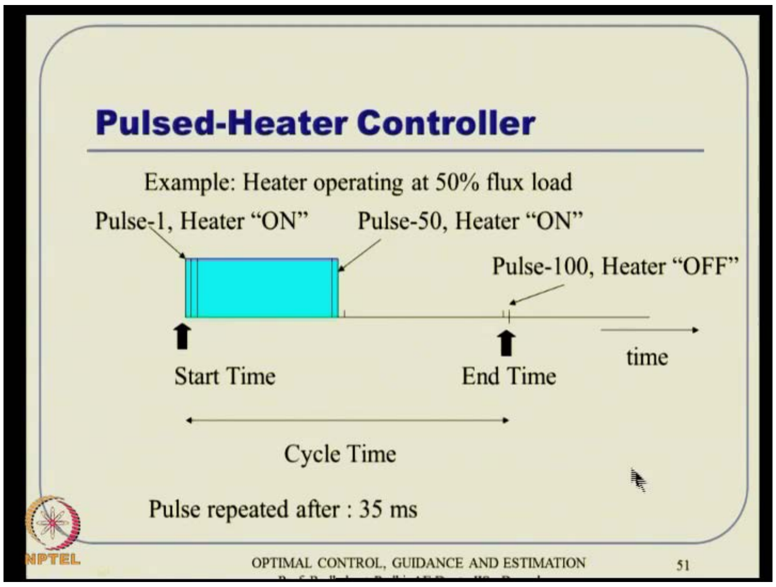
    Total of "648" weights and biases to be handled using LabVIEW
- How to get temperature values at "heater" locations?
  - Linear Interpolation of neighboring thermocouple readings

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 50

So, the implementation issues several things: first thing is closed loops state feedback control. So, how to relate training range of neuro control to physical temperatures readings? So, that was an issue. So, use normalized data as input to neuro controller and that was some sort of a solution that was proposed. Use normalized delta as input to neuro control and that solves it. The neuro control needs understanding of all the weight connections in a network. Each node has one controller. So, total something like 648 weights and biases has to be handled using LabVIEW and that time it was manual coding in LabVIEW.

So, LabVIEW take, you manually code this numbers, 648 numbers and each of the number will have some large number of digits after point also. If faithfully you have to code it manually basically at that point of time. Now, it may not be done, because there is some automatic procedure available. Now, the question is how to get temperature values at heater locations? Because, we are assuming also that something is called collocated or non-allocated. When you have collocated, I mean all throughout you have been assuming collocated exsum; that means, where you are applying control, exactly there itself your measuring the input, I mean measuring the output. Typically, that is not feasible. So, non collocated implementation is typically available in distributed parameter system because it is the exactly same point you cannot take measurement.

(Refer Slide Time: 47:01)



Pulsed-Heater Controller

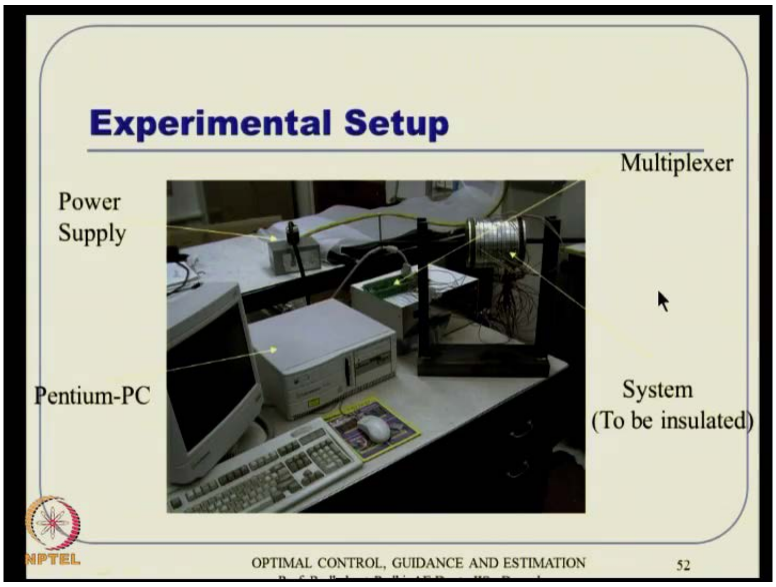
Example: Heater operating at 50% flux load

Pulse repeated after : 35 ms

So, then how to get the temperature values at heater locations? Those are like practical implementation issues. There are also issues in a control implementation sense, because you can always give heat input, you can heat it out, but when necessary, you cannot cool it down. There is no cooling mechanism and also there is something like, if you want 50 percent of the flux load. That mean total something of some maximum excitation is there. Let say you want 50 percent, then how do you implement? It may not be feasible also in that sense. So, in a way, in a different way it was done.

It was some cycle time is there and then if you really want 50 percent, then only up to 50 percent it was made on then rest of the time it was left free. Obviously, all this approximation that you do is part of the experiment also degrades the system dynamics behavior, the assumption that we rely on the theory. So, you do not except two much of heat results in that sense, but despite all that the results will be encouraging.

(Refer Slide Time: 47:54)

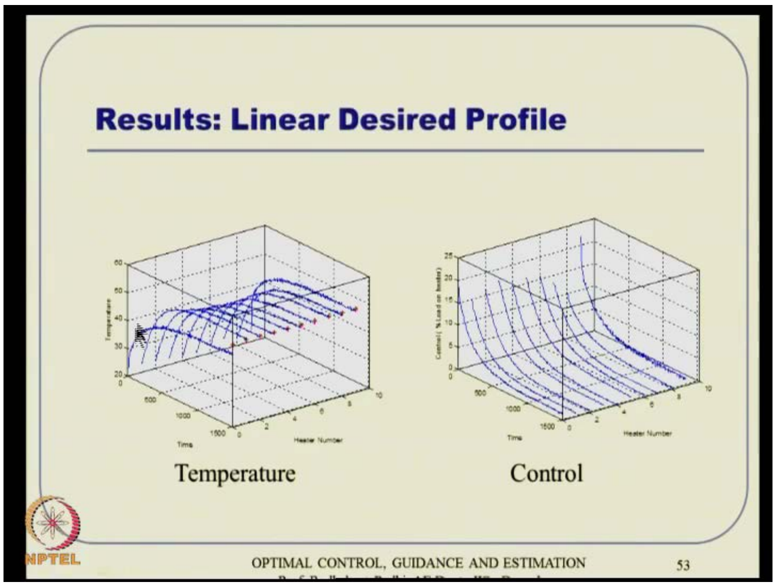


Experimental Setup

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 52

So, this is in summary, this is the experimental setup. There were some set of data communication and thing like that. Ultimately, there was a multiplexer through which this set, I mean this was connected to the PC like that and there is a power supply which is connected. I mean this is very standard thing. And then this is a system where ultimately the experiments still keep on going. Again, that was a Pentium PC through which commands were given and sense of value were read and all that.

(Refer Slide Time: 48:23)

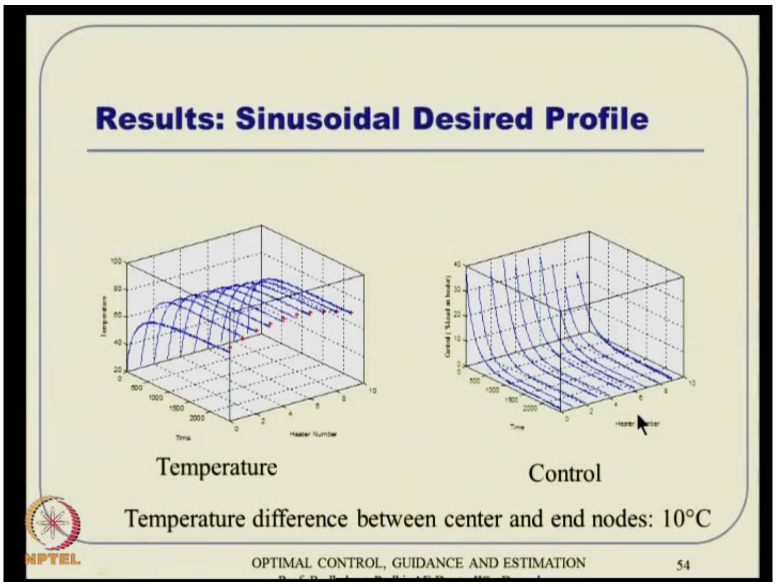


Results: Linear Desired Profile

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 53

This is final result. I mean, ultimately we want to uniform temperature and all that. So, very end, at the boundary, it was not able do a very good job; obviously, there is a insulation problem and implementation issue and all that. But, nonetheless, it was able to control in a very good way. It was coming to all constant values and irrespective of various starts, actually. So, if you really want a constant profile then roughly it is doing the job.

(Refer Slide Time: 48:47)



Results: Sinusoidal Desired Profile

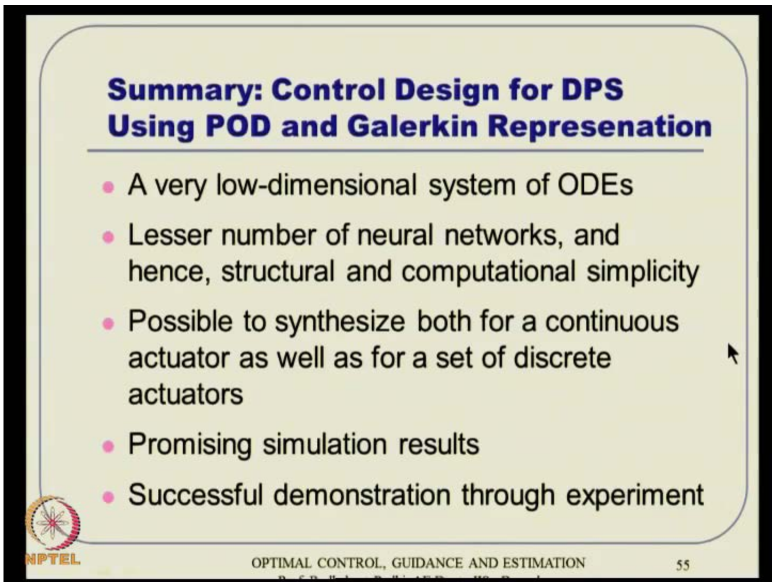
Temperature

Control

Temperature difference between center and end nodes: 10°C

If you really want a sinusoidal condition, then this is not constant thing. This is a slight part of, I mean slight sinusoidal behavior. So, look at the start things and try to connect. This is actually a kind of sinusoidal thing. Even if you want that it was happening. So, all that things, these are nice things, which gives us lot of confidence that experimental validation also it can happen.

(Refer Slide Time: 49:20)



Summary: Control Design for DPS Using POD and Galerkin Represenation

- A very low-dimensional system of ODEs
- Lesser number of neural networks, and hence, structural and computational simplicity
- Possible to synthesize both for a continuous actuator as well as for a set of discrete actuators
- Promising simulation results
- Successful demonstration through experiment

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION 55

So, summary of these entire control design approach using this POD and Galerkin representation thing like that. First of all, it results in a very low dimensional system of ODEs. It is again this entire way of mechanizing are handling this PDE system is somehow you have to have some representation of ODE and then excite this ODE control design philosophy is something like that. One way is to do not bother about model reduction and then simply go ahead and implement in finite deference. Also, let me put a comment here that even if you do that the finite difference once implemented, and then utilizing that last dimensional system, there are some model reductions techniques available also.

So, that way also you can reduce the model. First, you will generate the large dimensional representation. Then you excide this Eigen vector comes out all that and then finally, you will reduce system dynamics there itself. That is also a possibility to look at or to look into. Another representation is the people have also thought about why final difference all the time, I can talk about finite element, finite volume. So, if you incorporate those things, then not only have to, not only I am getting some grid points values, but I also get some numbers in between the grid points. So, those things are also possible, once you use this finite element, the finite volume approaches like that.

So, this opens of lot of different dimensions and depending on your comfort zone, you can explore what you want. Also these are very relevant to these chemical industry problems. So, process control people, if you are, I mean, happen through goes through these lecture may be find it very useful. Coming back, the lesser number neural networks it will result because these low dimensional representation and will also result in lesser number of neural networks. Hence, in a way, it will result is structural and computational simplicity. And it is possible to synthesize both for continuous actuator as well as for discrete actuators. Discrete actuators things, I have not talked about it here, but if somewhat interested you can read it one of our publications, when we got lot of promising results and then successful demonstration through experiment also.

Then various other problems are solved. For example, it has some ecology problems, some newer management issues in the jungle and all that you have done it. It is just for fun and then this is not one dimensional PDE, but it has two dimensional PDE; that means, two special variables y 1 and y 2. The animals will be spread in different direction in jungle. So, one direction, well it is not sufficient to handle that, but essentially the idea remains the same and the tricks and techniques remain same.

No matter, whether one dimensional ODE or their one dimensional PDE, here one dimensional means one special dimension, essentially. So, whether it is one special dimensional PDE or 2 or 3 or 4, it does not matter, because the basis function state the load and once you introduce this separation of variable concepts sort of thing, the time domain, the time evolution behavior is captured through the coefficient. So, because of that the generality is maintained here and hence you do many things there.

But, obviously, the math complexity and computational complexity, it tries to pile up, because in one sense you talk about single integration of two special variables for evaluating the inner product. For evaluating you will have double integral sort of things or two dimensional integral sort of thing, so then the math and computational complexity will creep in rarely. But, in general, in my view, POD happens to be good thing, a good approach; however, as I told in the beginning that POD all relies on the snapshot solution. So, that is bottom line.

Now, going back and thinking a little bit about snapshot, the solution is good and the procedure is good as long as the snapshot solution as good. Now, how do generate good set of snapshot solution? That again depends on intuition, depends on the prior experience, and depends on several procedures. Then as I told before the people propose different way of doing things something like adaptive POD and then dynamically re-optimization of POD and things like that; that means, we do not generate the basis functions once and keep on working on that. You can have a procedure where basis function themselves get refined and as you go along, because you will have better snap shot solutions as you go along. So, that kind of ideas are available in the literature, you are most welcome and encouraged to read all of that and then get well further. This is fascinating area of research in my view and lot of practical significance in general as well. So, with that I will stop this lecture thank you.