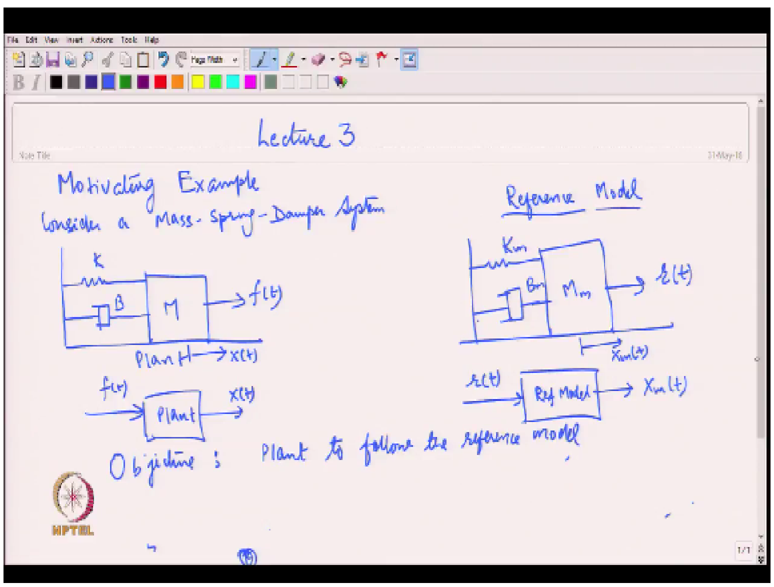
**Nonlinear & Adaptive Control**
**Dr. Shubhendu Bhasin**
**Department of Electrical Engineering**
**Indian Institute of Technology - Delhi**

**Lecture – 03**
**Model Reference Adaptive Control Part - 1**

Okay, good morning everyone. Welcome to lecture 3 on this nonlinear and adaptive control course. Last time, we had discussed the preliminaries required to understand this course. I encourage all of you to go back and learn about these preliminaries in detail. So today's, we will move on to our first adaptive control design. So let us start with a motivating example.

**(Refer Slide Time: 00:48)**



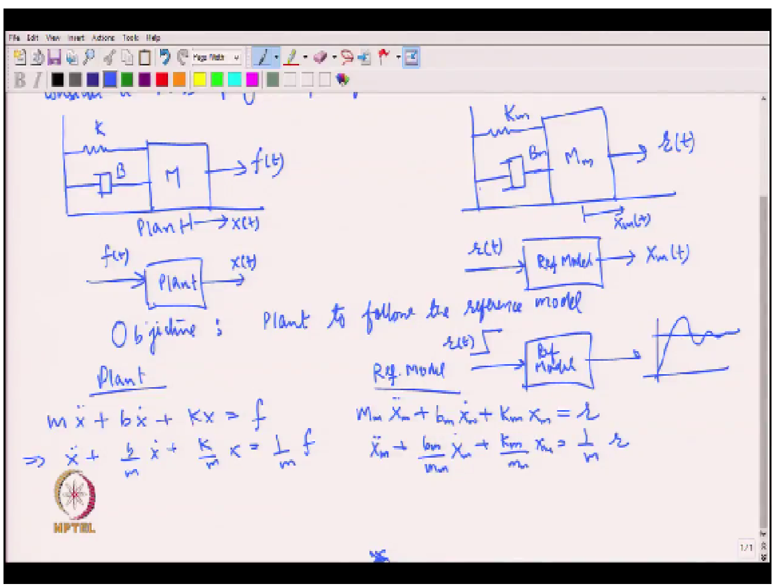

So let us consider a mass spring damper system. There is a mass M and there is a spring constant K and a damping constant B and there is an input force give by F of t, let us represent it by small f, okay. So now let us say that I am not very satisfied with the response of the system. It does not behave the way I wanted to behave. For example, if I give a step input to the system, I do not have, get the, the required transient response, the overshoot, the settling time.

I am not really happy with the response of the system. Right now also say that we have a reference model which you can call your favourite system. So, so let us say we have another system which has a desired response. So let us say we have, the spring constant is given by Km, the damping constant is given by Bm and let us say the mass is given by Mm. So, so let us say that this system has a desired response.

So let us call this as a plant, the first system as a plant. So the input to the plant is f of t and let us say that the position is X of t is the output which is the position of the, of the mass starting from 0. Let us say that here for the reference model also I have the position Xm of t. So for the reference model, the input is given, is, is denoted by R of t.

And let us say the output is Xm of t which is the position of this, of the mass, of the reference system. So since I am not happy with the way this, this plant behaves, I want that this plant somehow imitates the behaviour of this reference model. So how do we go about making sure that the plant follows the reference model? So the objective is for the plant to follow the reference model.

**(Refer Slide Time: 04:43)**



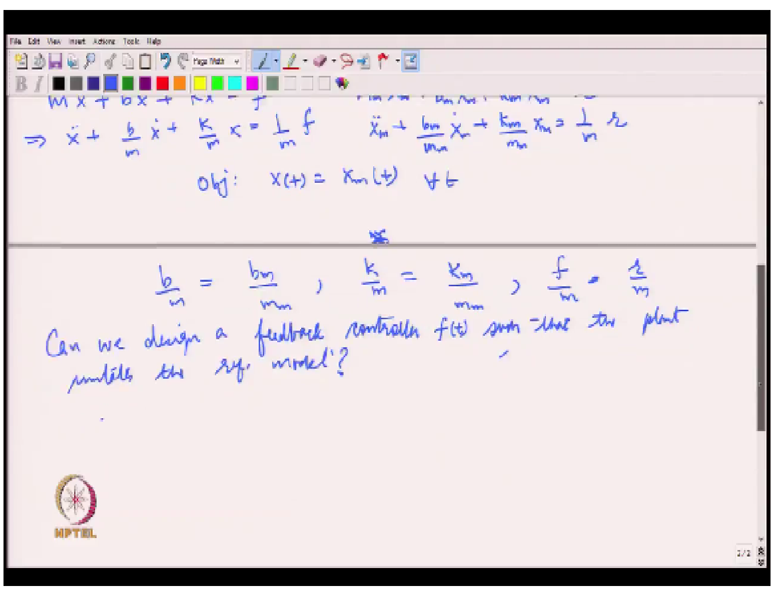

So by the way the mass spring damper system can be used to model many physical systems, example the suspension of the car. The main examples are you can find that where, whereever you have an inertia, a storage element, a dissipation element, these mass spring damper systems can be used to model such systems, okay. So it is, it is a practical example that we have considered and the task is for, for this plant to, to have a desired closed loop response.

So, so we can say that the reference model in a way captures the desired transient behaviour, right. So you can, maybe consider an example that so here for the reference model, let us say the input is a step input and the output is, is some behaviour which I really like and that is why I want this plant to behave similar to the reference model. So if I use the same input, then I should be able to get similar response or the same response as a reference model.

So one way to make sure that that the plant is, behaves exactly same as reference model is to make sure that the mass of the plant is same as that of the reference model. The spring constant of the plant is same as that of the reference model. The damping constant of the plant is same as that of the reference model. So if, if, if all these 3 components of the plant are equal to that of the reference model, then of course, the plant exactly same as that of the reference model.

So we can, we can look at it a little more closely by looking at their mathematical models. So for the, for the plant, we say that the mathematical model is given by Mx double dot+bx dot+Kx=f, right. And for the reference model, we have MmXm double dot+BmXm dot+KmXm=r, right. So, so let us divide the plant equation throughout by M. So what we get is X double dot+b/m x dot+K/mX=1/m f, right. Let us do the same thing with the reference model, we divide throughout by m m. So what we get is +Km/m Xm=1/m r, right.

**(Refer Slide Time: 08:42)**




Now of course the objective is for the plant to follow the reference model that is mathematically given as, so the objective here is for X of t=Xm of t, right for all time. So if that is possible that,

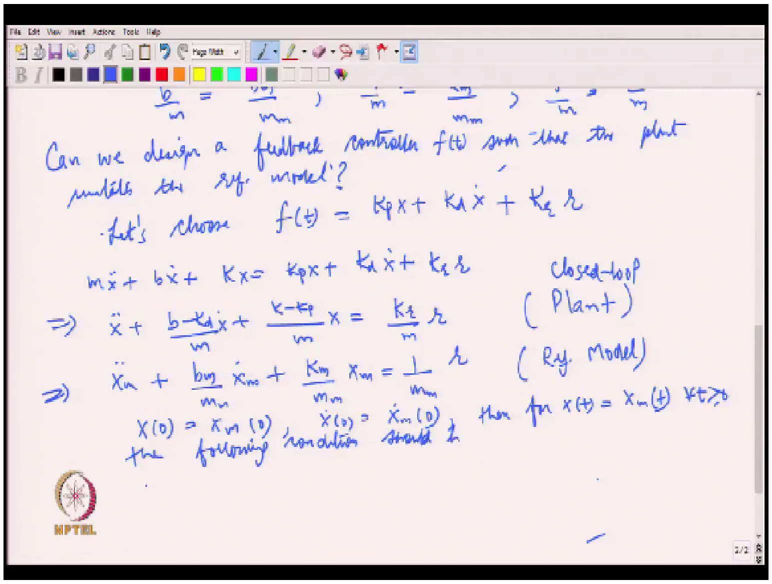
that is the best thing that can happen because we would like the plant to exactly follow the reference model. So if, the position of the, of the plant mass is exactly equal to the position of the mass of the reference model for all time, that will solve the problem, okay.

So how do we make sure that that happens? So of course as I mentioned one way to make sure is that b/m is same as bm/mm, K/m is same as Km/mm and f/m is same as r/m. So from here, what we say is, what we see is that we need to change our plant such that it, it follows the reference model. So which is not always a great thing. Because plant is something is given to you. It is a physical plant which you may not be able to change.

If you can change the plant, then of course you can do this. Otherwise, it is, it is very difficult to, to change the plant and it can also be very expensive to, to change the plant. So what is the next best thing that, that we can do? So the next best thing that, that we can do here is, is, is to design a feedback controller.

So the next thing is, can we design a feedback controller f of t such that the plant imitates the reference model. So since we cannot make the plant components same as that of the reference model, we would like that we design a control system around the plant and make sure that it behaves similar to that of the reference model. So that is the next question that how do we design a feedback controller f of t.

**(Refer Slide Time: 11:36)**




So for that, we can try with, with a state feedback controller. So let us choose f of t which is the input to the plant as a state feedback controller which is given by KpX+KdX dot. So in this case, there are 2 states, X and X dot to represent its position and velocity respectively. So we choose a state feedback controller KpX+KdX dot and since we are trying to track the reference model, we also have a tracking term Kr*r, okay.

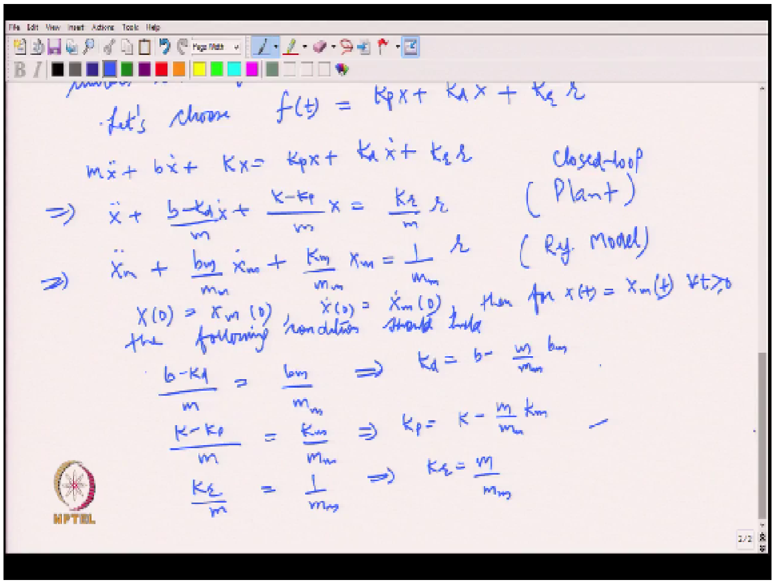
So the question now becomes, how do we choose Kp, Kd and Kr such that this control system is able to make sure that the plant tracks the reference model, okay. So for that we have to look at this more closely. So let us look at the, the mathematical equation for the plant and substitute for f of t. So if we look at this equation, we for the plant, we substitute for the controller f of t here and let us see what happens?

So mx double dot+bx dot+Kx=f which is given by KpX+KdX dot+Kr*r. So again we divide it by m and what we get is X dot+b-Kd/m+K-Kp/m, so this is X dot+this is X=Kr/m*r. So this is the plant and this is a closed loop plant equation because we have substituted for the controller, okay. Then let us have another look at the reference model. So that is given by Xm double dot+bm/mm Xm dot+Km/mm Xm=1/mm *r.

So this is the reference model, okay. What we have done here is we have, we have just chosen a controller f as a feedback controller, state feedback controller and now we are trying to see if we can make sure that that results in the plant following the reference model which is that X of t=Xm of t, right. So let us look at these 2 equations. What we see here is, is that we can compare the coefficients, right.

So, so if the initial conditions are the same for both the plant and the model, so the positions, initial positions and the initial velocities for both the plant and the reference model are the same, okay, then for, for X of t=Xm of t for all time, t greater than equal to 0, the following condition should hold.

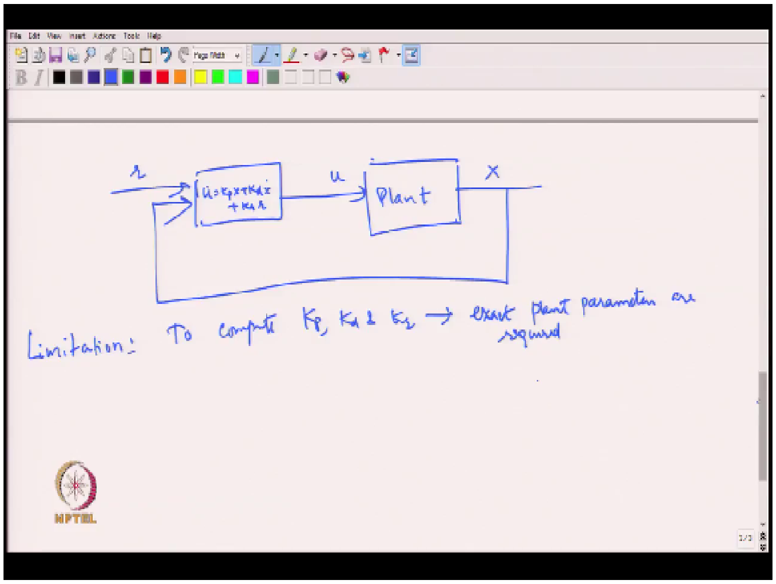
**(Refer Slide Time: 15:50)**



And what is that we just compared the coefficients and b-Kd/m is same as bm/mm, K-Kp/m=Km/mm and Kr/m=1/mm. So what this gives us is the values of the 3Ks as m/mm. This is, this one give Kp as K-m/m*Km and this gives Kd as b-/m/mm bm. So what we see here is that we can compare the, the coefficients in the 2 equations for the plant as well as the reference model.

And if the initial conditions for both the plant and the reference model are chosen to be the same, then these 2 essentially are the same differential equations and for all time, this illusion of both the, both the systems would be the same. So, so this is the simplistic approach that you could follow.

**(Refer Slide Time: 17:25)**



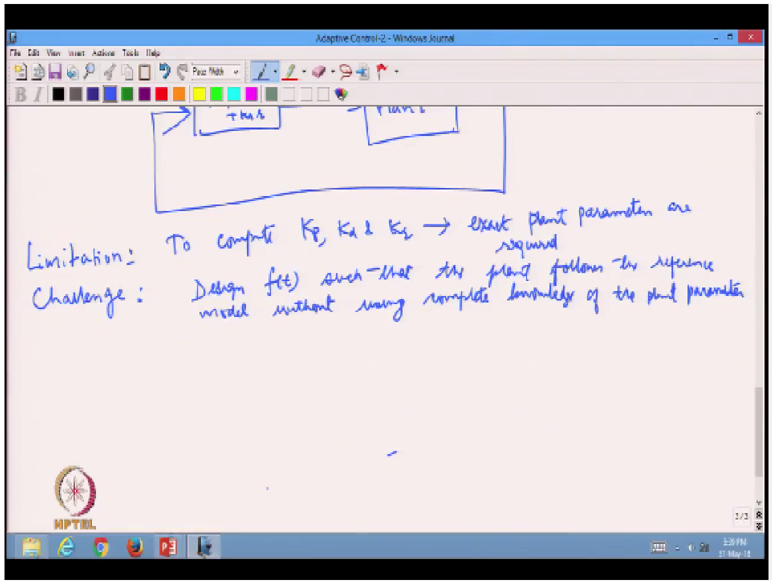

And so we, how we could reprint this as a, as a control system is we are trying to control this plant and it is, and the controller that we have used is simply Kp*X+Kd*X dot+Kr*r, right, that is the controller that we have used and the output of the plant is X, the input is u. So X is goes to the feedback and we also have the reference signal r. So this reference signal is the same signal that is the input to the reference model, okay.

So here the important thing to note here is that the, the gains Kp, Kd and Kr, they come from computation using these relations, right. So what is that, so this works in, in many cases. However, there was very serious limitation of this method and that is that to compute these gains Kp, Kd and Kr, you need to exactly know the, the plant parameters. So to compute Kp, Kd and Kr, you need to know the exact plant parameters are required to be known, right.

And that is a very restrictive assumption because in practice, it is very difficult to, to compute these parameters. You may be able to do an approximate job of finding out the mass, the damping constant and the spring constant but it is very difficult to exactly say that these are the plant parameters and X of t matches Xm of t only when these are exactly equal. So if there is a slight mismatch, then the responses can be very different, okay.

So just to recap, so here we have design a controller which we can call as a model following controller because this enables the plant to follow the reference model. The controller is chosen as a state feedback controller and the gains Kp, Kd and Kr of the controller are computed based on the comparing of the coefficients of the plant and the reference model and that needs exact knowledge of the plant parameters, okay. So that is the limitation of this method. So how do we overcome this limitation that is a challenge.

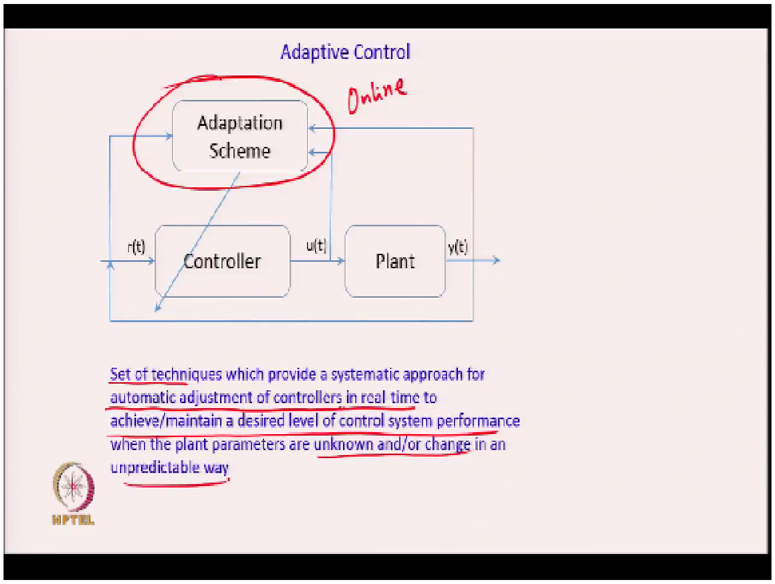
**(Refer Slide Time: 20:46)**




So I next write what the challenge is. So we want to design f of t such that the plant follows the reference model without using complete knowledge of the plant parameters. So in the case of mass spring damper system, the unknown parameters are the mass, spring constant and damping constant. So can we design a controller such that we either do not use any of these parameters at all or we only use some of these parameters and not, not all of them?

So the idea is not to use complete knowledge of the plant parameters. So this is where the power of an adaptive control comes in. As I, as I mentioned before, adaptive controllers are used in cases where you have plant parameter uncertainty, that is the parameters of the plant are not known or they change in an unpredictable way. So, so we want to take an adaptive control approach to this.

**(Refer Slide Time: 22:33)**

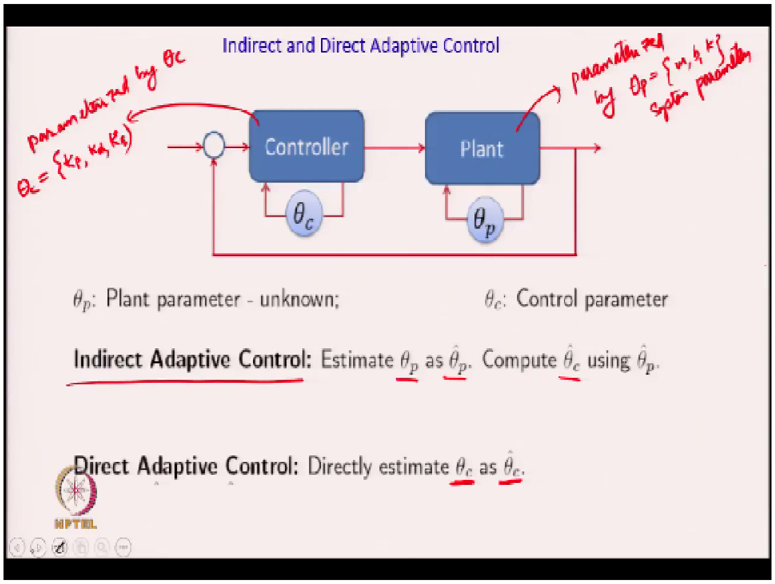



So what I mean to say is that here what we do is we construct a controller for this plant and then we have an adaptation scheme here. So this adaptation scheme has an inbuilt logic, an inbuilt strategy to, to adjust the parameters of the controller, okay. So the adaptation scheme's input are the input to the plant, the output of the plant, the desired trajectory or the desired reference signal and it has some way of adjusting the parameters of the controllers online.

So, so this is an online process, that is very important as compared to the offline approach that we had seen last, in the previous example where we had also considered an exact knowledge of the plant parameters. So adaptive control as I mentioned refers to a set of techniques which provide a systematic approach for automatic adjustment of the controllers in real time, that is very important.

So there is automatic adjustment of the controller parameters online, in real time to achieve or maintain a desired level of control system performance when the plant parameters are either unknown or they change in an unpredictable way. So this is the statement which explains the entire idea behind adaptive control, okay. So now I move on to the different flavours of adaptive control. So there are, adaptive control comes in 2 main flavours.

**(Refer Slide Time: 24:20)**



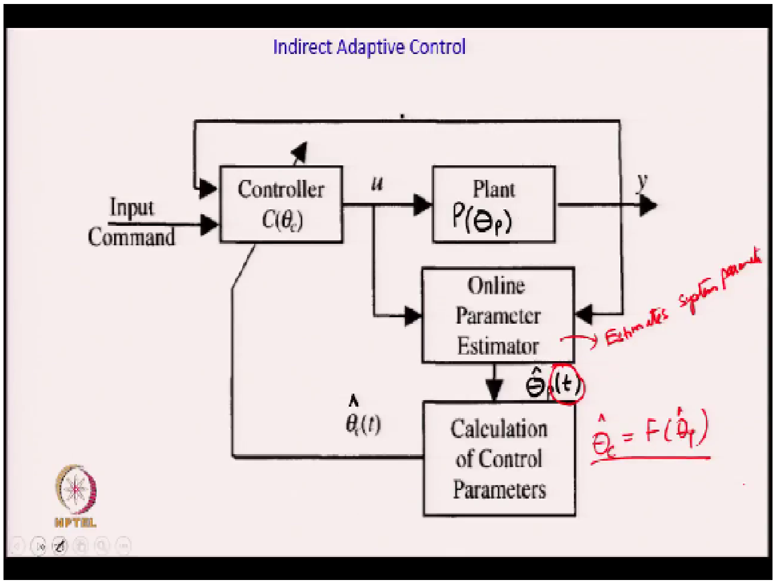

One is the indirect adaptive control and the other is the direct adaptive control. So I want to talk about them in brief. So let us again consider this plant which is parameterized by the parameters theta p. So for the mass being damper case for example, here you would have m, b and k as your parameters of the plant and let us denote all of them as theta p which we call as the system parameters.

So in a similar way, the controller is also parameterized. So in, by theta c and in the case of the mass being damper case where we had used the state feedback control law, the parameters of the controller are Kp, Kc and K, Kp, Kd and Kr, okay and let us say that the, the system parameters and the controller parameters are, are unknown. So in the indirect adaptive control case, the idea is first to estimate the parameters of the plant.

And then use that to compute the parameters of or estimate the parameters of the controller. So first we would like to estimate theta p. Let us call that estimate as theta p hat and using this estimate, we compute the, the estimate of the controller parameters which is given by theta c hat. So that is an indirect approach.

It is called the indirect adaptive controller and then the second flavour is a direct adaptive control approach where the controller parameter theta c is directly estimated as theta c hat. So the intermediate layer of first estimating the system parameters and then using them to estimate the controller parameters is not present. Here we directly estimate the controller parameters, okay. So these are 2 main flavours.

**(Refer Slide Time: 26:42)**

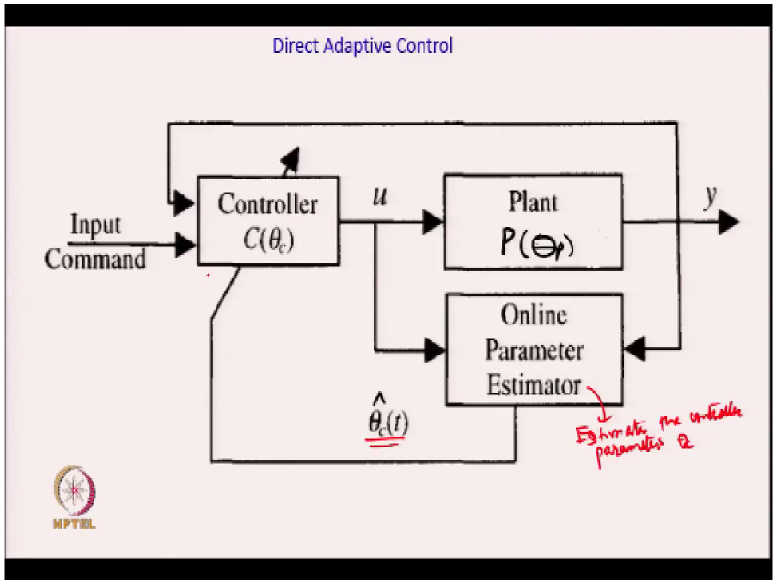



I have a block diagram of the 2 approaches. So let us just, in brief, go through them. So as I mentioned the plant here is parameterized by, by theta p, the controller is parameterized by theta c and in the indirect adaptive control approach, we use the input to the plant and the output of the plant as input to an online parameter estimator, okay which, which estimates, for the indirect case, it estimates the system parameters and so it online estimates the system parameters.

And we can denote it by theta p hat and the important thing to note is that these are updated as time goes on. So it is not just one value, theta p hat is a function of time. So as time goes on, we get newer newer and hopefully better and better estimates of the system parameters. And then there is another block which is used to calculate the controller parameters. So what this talk means is that somehow the, the 2 are related.

So the controller parameters would be some function of the plant parameters. So F is some function and if we obtain theta p hat from the online parameter estimator in, in the previous step, then just by using this set of algebraic equations, we could compute the controller parameters which are then used to update the, the controller and this is the indirect adaptive control approach.

**(Refer Slide Time: 28:37)**

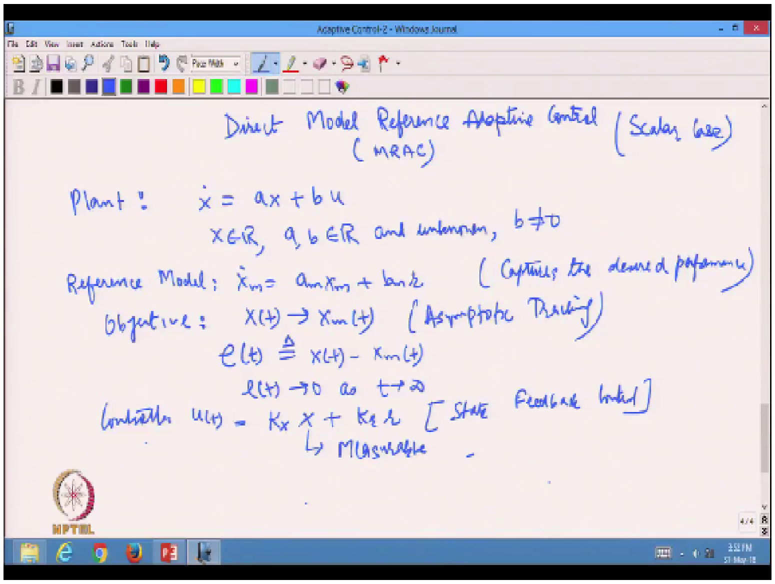



Let us move on to the direct adaptive control approach. So, so in this approach, the idea is very similar that an online parameter estimator has to be used. But here the online parameter estimator directly estimates the controller parameter. So this estimator estimates the controller parameters theta c. So let us call this, that estimate as theta c hat and again as you can see it is a function of time.

So this estimate keeps updating as time goes on. And that, so that is why we, we call this as an online estimator. So as theta c estimate changes, the controller is updated and, so this is very different from the fixed gain approaches that, that you might be aware of. For example, a BID controller, a state feedback controller, a lead-lag compensator or robust control approaches. So all these are examples of fixed gain approaches.

Whereas adaptive control takes a different view and it has an, so that the common thread between this direct and indirect approaches is that there is an online parameter estimator which at every instant of time gives you an estimate of either the system or the controller parameters which are then used to update the control. So that is the basic idea behind, behind using adaptive control.

**(Refer Slide Time: 30:20)**




The mass spring damper was just a motivating example to illustrate that, that adaptive controllers would be useful in situations where plant parameters are unknown. So now, now I will talk about the direct model reference adaptive control also called, call this is MRAC, but this is a direct MRAC case and just for simplicity, I will consider a scalar case, okay. So, so let us consider a plant which is given by X dot-ax+bu.

So here a, so X is a state which is a scalar, a and b are also scalar and unknown. Let us also assume that b is not equal to 0 and this is for controllability purpose, we would not want this to be equal to 0 because a can in fact be positive and so this can be an open loop unstable system. So if b=0, then, then you could never stabilize the system, okay. So this is a plant and then as we did for the model reference controller case.

For this case, we consider the reference model to be Xm dot=amXm+bmr. So this is a reference model which captures the desired performance and the objective is for the plant for, for, in the MRAC case, model reference adaptive control, the objective is always for the plant to track the reference model which mathematically just means that X of t tracks Xm of t. So if you recall the previous cases that we did, we had in fact got X of t exactly equal to Xm of t.

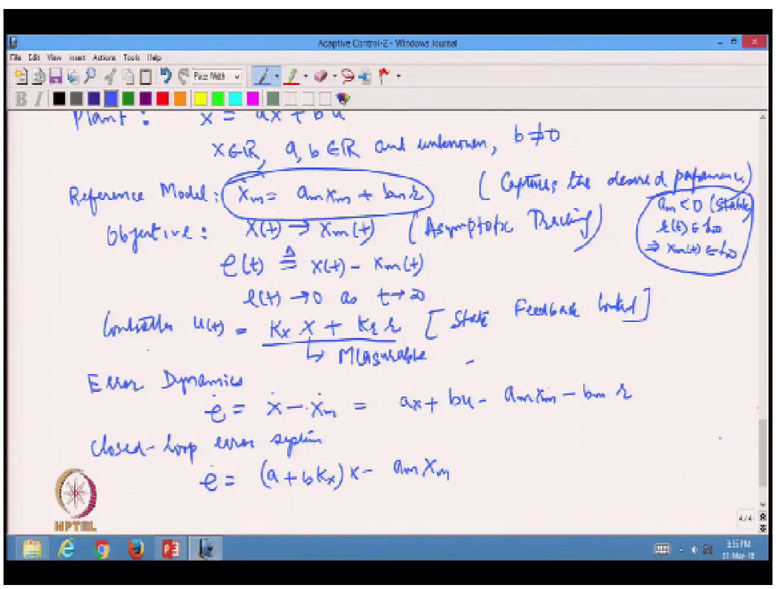
We will see in this is case it is not possible because we assume that, we consider that the parameters of the plant a and b are unknown and so it is not possible to get perfect tracking.

What we could hope for is asymptotic tracking at best, for exponential tracking, if you are lucky, okay. So I am hoping for asymptotic tracking, perfect tracking which means X of t is exactly equal to Xm of t for all time, cannot really happen here because a and b are unknown.

So we will see how we can, we can at least get asymptotic tracking. So the objective can be also written in terms of the error, so let us say e of t is defined as X of t-Xm of t and the objective is then for e of t to go to 0 as t goes to infinity, okay. So to achieve this objective, let us consider a controller which has a similar structure as before. So the controller u of t, let us say is given by Kx of X+Kr of r.

So here there is only one state given by X. So we, this is the state feedback controller. Kx of X+Kr of r. Okay, so the assumption here that we, that we are making is that the states are measurable, right. So X is in fact measurable. We have stencils to measure the state X, okay. So we should be clear what the objective is? We want that the error should go to 0 which means that the plant state tracks the state of the reference model. So let us look at the error dynamics.

**(Refer Slide Time: 35:21)**



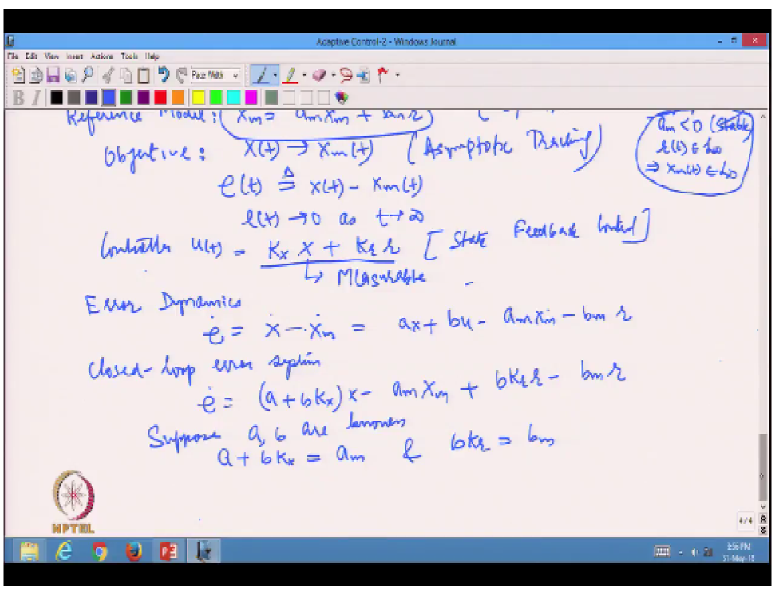

Which is e dot is given by X dot-Xm dot which is ax+bu-amXm-bmr. So let us substitute for the controller u which is given by, by this expression. So the closed loop error system is, is given by, so why we call this closed loop is because we substitute for the controller, we close the loop, this is the feedback control law. So e dot is then given by a+bkxX-amXm. So I missed something about the reference model.

So this reference model that we consider, of course the most important requirement of any reference model is that to be stable, right. Because we do not want the plant follow an unstable system. So for this to be stable, the requirement is that am has to be < 0 for a reference model to be stable, okay. And another requirement is for the reference signal r of t to be bounded.

So we do not want that the reference model should have inputs which are unbounded. They should have bounded inputs and since the reference model is, is also considered to be stable, so the output of this reference model Xm of t will also be bounded. So, these are some requirements for the reference model. So you have to be very careful when choosing the reference model. It should be a stable system which should capture the desired transient performance.

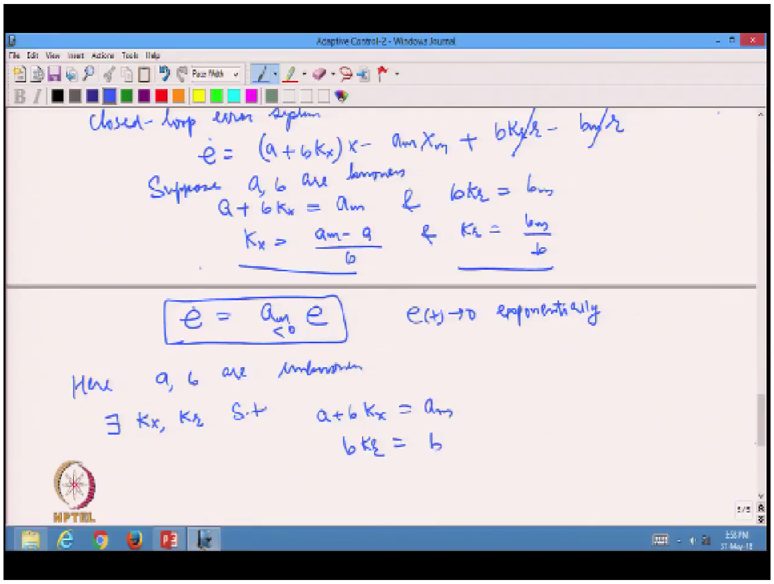
And the input to the reference model should be a bounded signal, okay. Okay so let us get back to our closed loop error system.

**(Refer Slide Time: 37:57)**



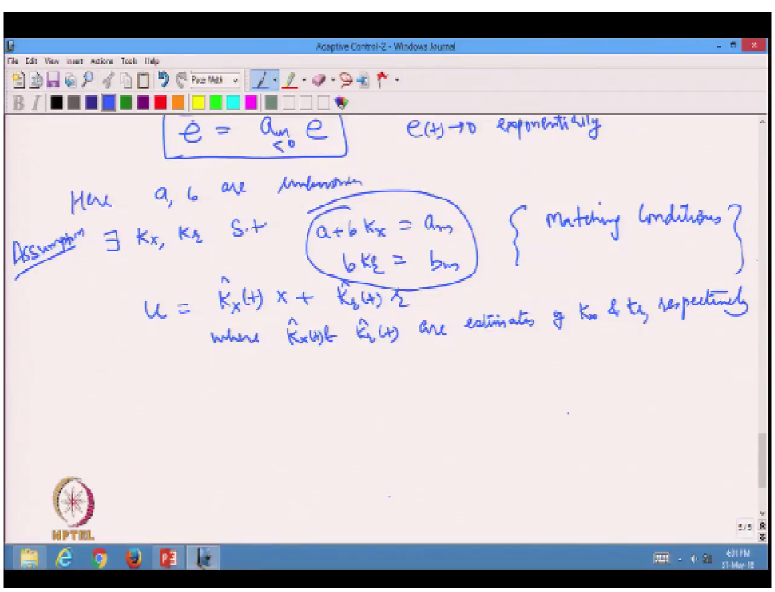

So here what we get is +bKrr-bmr, okay. So here I have, I say that the plant parameters a and b are unknown. So let us for a moment assume that we know a and b. So just for a moment let us assume that the a and b are known. So suppose a, b are known, okay. Just for a moment. So then what we can do such that this error goes to 0. So we see here that we want this error to go to 0, which means that if we make sure that a+bKx=am and bKr=bm, so if we make sure that these 2 conditions are true, then what can we say?

**(Refer Slide Time: 38:57)**




So if these 2 conditions are true, then e dot becomes equal to simply am*e, right. Because these 2 terms cancel off and this is what we get, okay. So e dot, right. So this is a stable linear system since am<0. So e will exponentially converge to 0, right. And here what is the condition? The condition is that we choose Kx=am-a/b and we choose Kr to be = bm/b. This is similar to the case that we had discussed when the parameters are known, okay for the mass spring damper case.

**(Refer Slide Time: 40:11)**



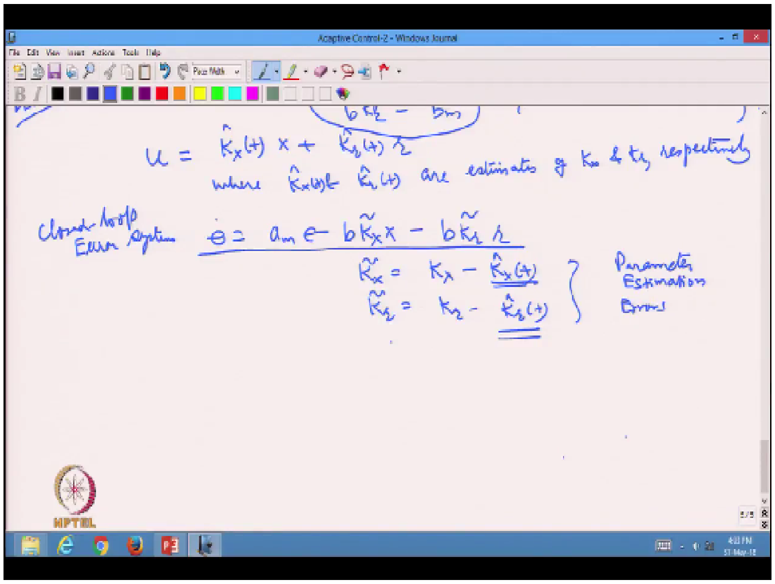
But the problem is when, when we say that a, b are unknown. So in this case both a and b are unknown. So how do we solve this problem for the scenario. So since a and b are unknown, we cannot find Kx and Kr, okay. But what we can say is that at least there exists these gains Kx and Kr. So what we can say is there exists some ideal gains Kx and Kr such that a+bKx=am and bKr=bm, okay.

So although we cannot find the exact values of Kx and kr, we assume. So this is an assumption that we make in the model reference adaptive control case and these are also called as a matching assumptions or the matching conditions. So the controller can be modified now to have Kx hat X+Kr hat r where Kx hat and Kr hat are estimates of Kx and Kr respectively. So since a and b are unknown, we cannot exactly find the, the value of Kx and Kr.

But what we assume here is that there exists these Kx and Kr, although we do not, we cannot compute their values but what we can do is, we can at least estimate, estimate these values. So we modify our controller to have the estimates of these gains. So u is Kx hat X+Kr hat r, okay. So these assumptions, if you look at these assumptions, how severe are these assumptions. So for our scalar case, they would always exists, Kx and Kr because we have assumed b to be non-zero.

So you could always find Kx to be am-a, so there would always exist some Kx and Kr even though we say that we cannot find it but this condition would always hold for the scalar case. However, when we get to the vector case or the multi-dimensional case, we see that these matching conditions may not always hold.

**(Refer Slide Time: 43:21)**

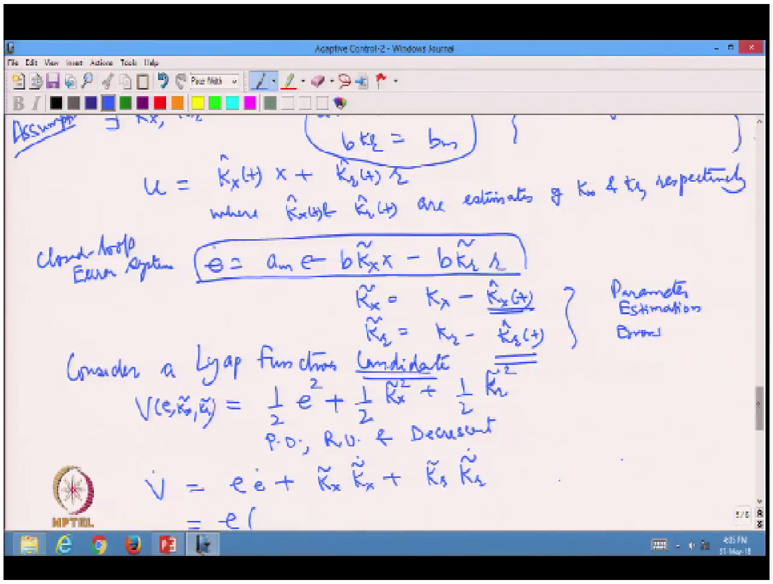



So that is why they are very important when looking at the model reference adaptive control case. The matching conditions must hold, so that is an assumption that we make, okay. So with this as a controller, the closed loop error system becomes, sorry, the closed, when we substitute for the controller in the dynamics, so the closed loop error system becomes e dot as ame-bKx~X-bkr~r.

So this Kx~ is simply Kx-Kx hat of t and Kr~ is Kr-Kr hat of t. So these are the parameter estimation errors. So in addition to the error e which is our primary objective, we also have the parameter estimation errors which ideally we would like to be to go to 0. So suppose if we look at this equation and we say that the parameter estimation errors are, are 0, then this equation becomes exactly same as what we had in the previous case which is e dot=ame.

But since Kx and, there is always, we cannot always, we cannot compute Kx and Kr exactly, so we get these estimation errors which we would like to minimize, okay. So, so how do we design for Kx hat and Kr hat because we have mentioned that these are time varying quantities but we still have not come up with the design for these. So as I mentioned before in my previous lecture that we use very rigorous approach through the Lyapunov stability analysis to not just analyse stability but also to design controllers.

**(Refer Slide Time: 45:40)**

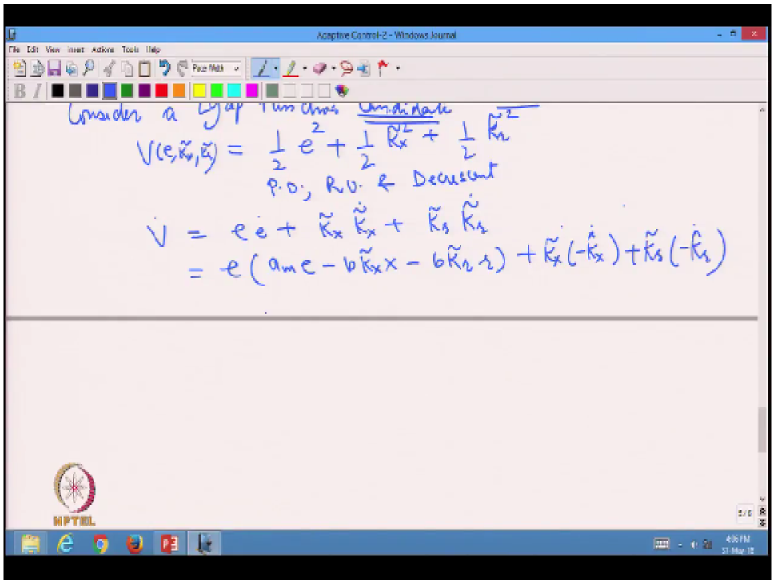



So let us look at, consider a Lyapunov function candidate. We want to see if this controller that we have designed is stable or not. So now as I mentioned, choosing a Lyapunov function candidate is a nontrivial task but we can start with some good guesses. So a good guess in this case would be 1/2e square, okay, where e is the error. So if you go through with this, you will find that it is not going shift forward to satisfy this but.

But we can further observe that here we have additional error states. So what are those error states, the parameter estimation errors. So let us try and include those also in our Lyapunov function candidate. So and the another thing is why do we call this as a candidate if because right now we can just choose a positive definite, radially unbounded, decrescent function of the error states.

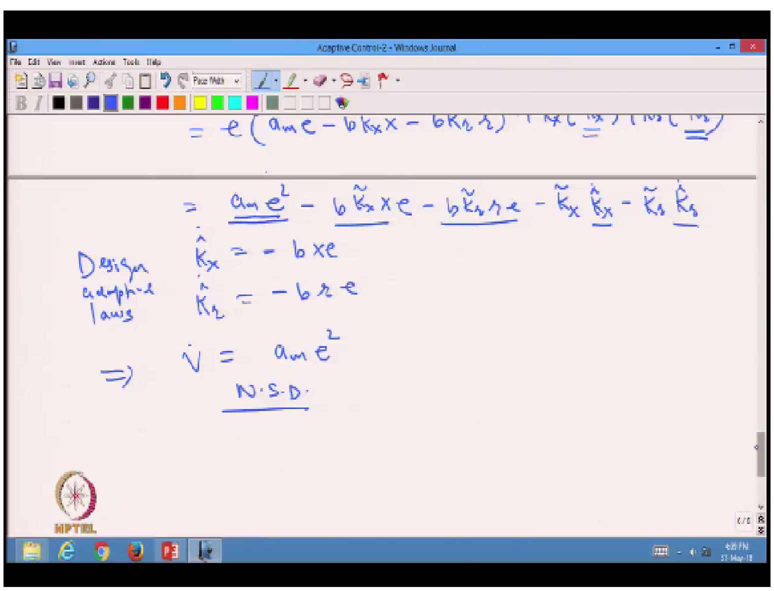
It becomes a Lyapunov function only when we could prove that V dot is at least negative semi-definite. So till then, till that time, this is just a Lyapunov function candidate. So it is, so V is a function of e, Kx~ and Kr~, right. So it is given by the sum of squares, 1/2e square+1/2Kx~ square+1/2Kr~ square, okay. So this function as I mentioned is positive definite, radially unbounded and decrescent. So it satisfies all the nice properties of a Lyapunov function candidate, okay.

**(Refer Slide Time: 47:30)**




So then how do we start analysing stability? The next step is to take the derivative of V. So V dot is e*e dot+Kx~*Kx~ dot+Kr~*Kr~ dot, right. And so what we can do is, we can start substituting for the dynamics. So e dot we have already found out to be this. So we substitute for that as ame-bKx~X-bKr~r, what else, that is it. And then we have, so Kx~ is given by Kx-Kx hat. So when we differentiate Kx~, this term is just will disappear because Kx is just a constant whereas Kx hat is time varying. So what we have is +Kx~-Kx hat dot+Kr~-Kr hat dot, alright.

**(Refer Slide Time: 48:58)**

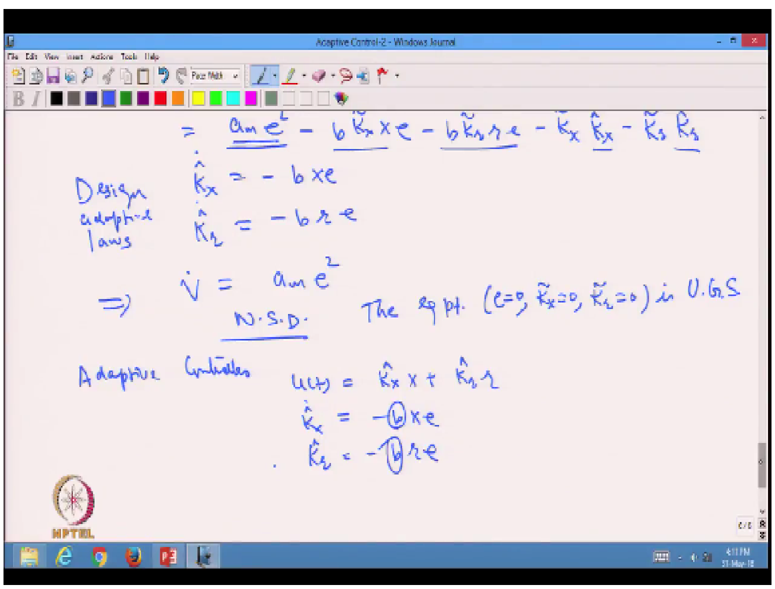



So why this is important is that now we can choose Kx hat dot and Kr hat dot in such a way that V dot becomes at least negative semi-definite and so that is, that is why this Lyapunov analysis in fact a design tool. So it helps us design the parameter estimators, okay. So, so this is some algebra, you could just, like this and then you have bKx~Xe-bKr~re-Kx~Kx hat dot-Kr~Kr hat dot, alright.

So the next thing that we have to do is to design Kx hat dot and Kr hat dot which are our, our estimators of the controller parameters Kx hat and Kr hat. So how do we design these such that V dot is negative semi-definite. So let us look at the expression of V dot again. So here we see that the first is, is negative because am<0. So this term is, is good for us. The other 2 terms, this term and this term, we do not really know the sign of these terms because Kx~, X, e, r, they are all time varying.

And we cannot say what sign they are? So we would ideally want to get rid of these sign indefinite terms by our design of Kx hat dot and Kr hat dot. The adaptive laws for Kx hat and Kr hat. So, so we have to design the adaptive laws. So Kx hat dot if we design as -bXe and Kr hat dot we design as -bre, then this will cancel these 2 terms and we end up with V dot as ame square, right. So which is negative semi-definite, okay.

**(Refer Slide Time: 51:53)**

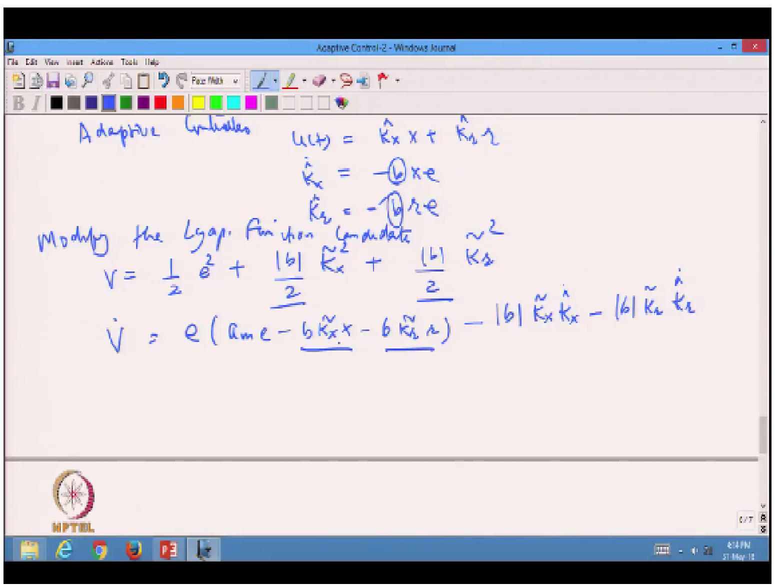



And since V is positive definite, radially unbounded and decrescent and V dot is negative semi-definite, what we can say is that the equilibrium point, so in this case, the equilibrium point is e=0, Kx~=0, Kr~=0, is uniformly globally stable, okay. So another point here is why is this negative semi-definite? Because we do not have negative expressions for Kx~ and Kr~. So, so V dot can in fact be equal to 0 for non-zero values of Kx~ and Kr~.

So since we are missing those terms, this is simply negative semi-definite and, and we can conclude that the, that the equilibrium point is uniformly globally stable. So just to recap the, the adaptive controller that we have designed, consists of the control law u of t which is given by Kx hat X+Kr hat r and the parameter estimate Kx hat dot=-bXe and Kr hat dot is given by -bre.

So, although we have been able to prove stability in, in this case, there is a slight problem, in fact, not a slight problem but there is a, there is a critical problem in this case, which is that the parameter estimation laws, they require knowledge of the, knowledge of b which we have considered to be unknown. So these differential equations cannot be computed, cannot be evaluated without, without knowing b. So this design clearly will not work.

**(Refer Slide Time: 54:16)**

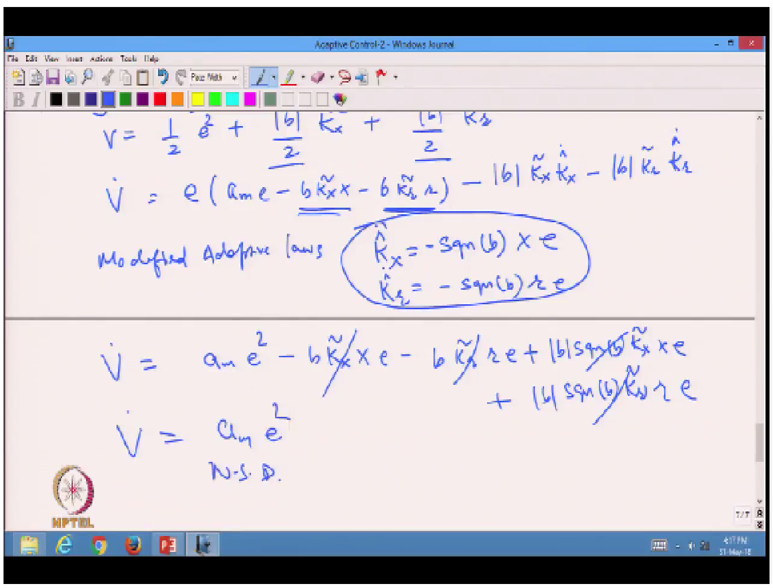



So to make sure that, that, that b is not present in the design, we will have to modify this design, okay. So how do we modify this? Well, let us choose a different Lyapunov function candidate. So let us modify the Lyapunov function candidate and let us see if we can do something with that. So let us choose V=1/2e square+b/2Kx~ square+b/2Kr~ square. So here, we have included in the Lyapunov function candidate an absolute value of b in, in these 2 terms and why can we include the term b here?

Well, you cannot include b in your controller, in your expression for u or your online parameter estimation laws because those are the equations that, that, that you have designed and that you are implementing. However, the, the Lyapunov function is, is merely for analysis. So even if b is unknown, you can still include unknown quantities in your Lyapunov function as long as these are positive.

So that is why we consider b, we do not consider just b, we consider the absolute value of b because b can be positive or negative. We have not really assumed the sign of b. So let us see what happens with this choice of the Lyapunov function candidate? So we find V dot=e*e dot which is ame-bKx~X-bKr~r-absolute value of bKx~Kx hat dot-absolute value of bKr~Kr hat dot.

**(Refer Slide Time: 56:47)**




Can we modify our adaptive update laws such that we cancel these 2 terms here which involve b. So that is why, I mean, we have chosen mod of b in the Lyapunov function, in the Lyapunov function candidate because this gives us an opportunity to modify the update law such that these 2 terms can be cancelled. So our modified adaptive laws can be given as Kx hat dot as sign of b- of sign of b and then we have Xe.

Similarly, Kr hat dot is given by -sign of b*r and e. So, so of course here we assume that the sign of b is known. So that is a less restrictive assumption then saying that we exactly know b. So if b is exactly known, then of course, you could use the, the update laws that we have designed for if b is exactly known. However, if you say that b is not known but I at least know the sign of b that it is either positive or negative, then, then, then these update laws result in V dot to be ame square-bKx~Xe-bKr~re-mod of b sign of bKx~Xe+mod of b sign of bKr~re.

So mod of b*sign of b gives you b. So this term exactly cancels with this term and this term exactly cancels with this term. So what we end up is V dot=ame square which is negative semi-definite and we can make the same conclusions about the equilibrium point which is that the equilibrium point is uniformly globally stable, okay. So we stop at this point. There are some more details which we still need to cover in this case and then we also do another case which is the indirect adaptive control, indirect model reference adaptive control case, okay. Thank you.