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

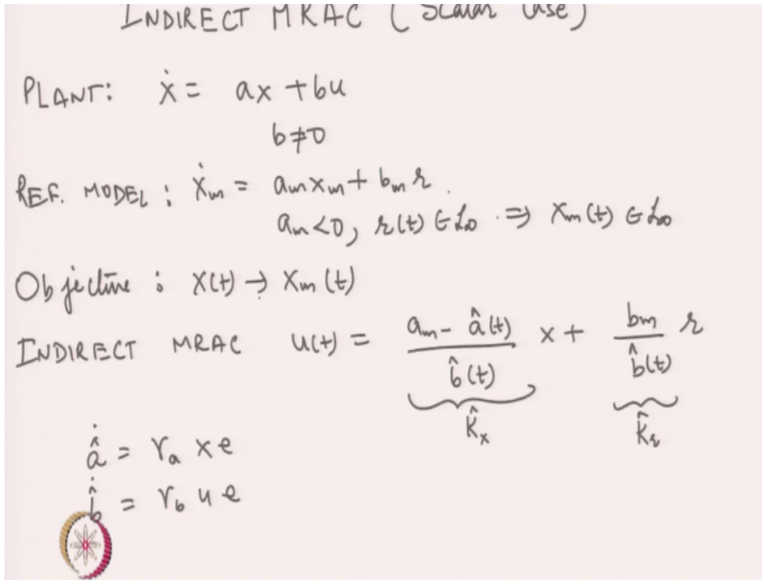
Lecture – 05
Model Reference Adaptive Control Part-3

Welcome everyone to lecture 5 of this course on nonlinear and adaptive control. So far we have then 2 designs direct model difference adaptive control and indirect model reference adaptive control. I also mentioned that salient feature of any adaptive controller is that there is an online parameter estimator which is of differential equation which is used to update the parameters of the system or the controller and that is how the controller is able to adjust its parameters online.

So the direct model reference adaptive control approach involved direct tuning of the controller parameters. So the online parameter estimator in that case directly adjusted directly adapted for the controller parameters. In the indirect case, the online parameter estimator estimates the system parameters and then the system parameters are used to compute the controller parameter using an algebraic relation.

So we almost finished the analysis for indirect MRAC in the last class, but there were some there is just the analysis, the stability analysis which was left unfinished after the modified the update laws. So in this class we will finish that. Before that I just want to recap what we had done last time on the indirect MRAC case.

(Refer Slide Time: 02:03)



So what we considered was a plant so where we are considering a scalar case and its LTI system so it is given by this expression. Here we considered that the parameter b can != 0 for system to maintain controllability. The reference model which the plant wants to follow is given by this expression and as I mentioned before this reference model encapsulate the desired behaviour that we want from the plant.

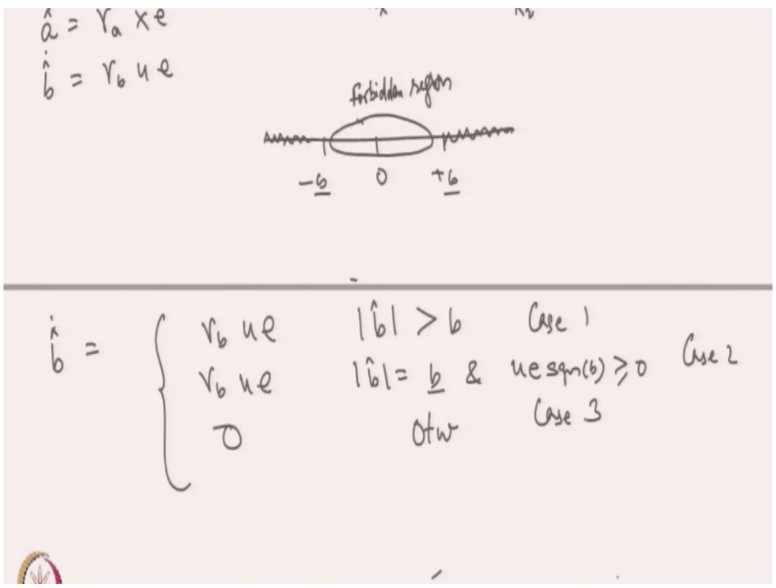
So somehow we have the reference model with us and we want this plant to follow the reference model. So the objective can be quantified by saying that we want that the plant state X(t) should track the reference model state Xm(t) asymptotically. So the reference model as I mentioned before is a stable one and the reference signal r(t) which drives your reference model is a bounded signal and as a result the reference model states are always bounded.

So these are required to set up the problem and so the indirect MRAC that we had calculated last time if you recall is given as am - a hat/b hat * x + bm/b hat * r and this coefficient we referred to as the controller parameter kx hat and this coefficient we refer to us kr hat. So notice that we are looking at a hat and b hat which are the actual system parameters. So we have the estimates a hat and b hat. We can then using this algebraic relation compute kx hat and kr hat.

So these update laws for a hat and b hat also we had computed last time and they are given as gamma a * xe b hat dot is given as gamma b * ue. This is how the system parameter estimates are updated. Here gamma a and gamma b are some positive adaptation gains x is a plant state, e is a tracking error, u is the input. So the problem that we have encountered with these update laws was that b hat could potentially go to 0.

So if that happens then that creates a problem with our control input in the sense that if b hat approaches 0 then the control input blows up which is not a desirable thing to have in your controller. So what we considered last time was a solution to this problem which was at if you somehow project the estimate of b hat to lie in a region which is away from the origin then maybe we can overcome this problem of the control going unbounded when b hat approaches 0.

(Refer Slide Time: 06:02)



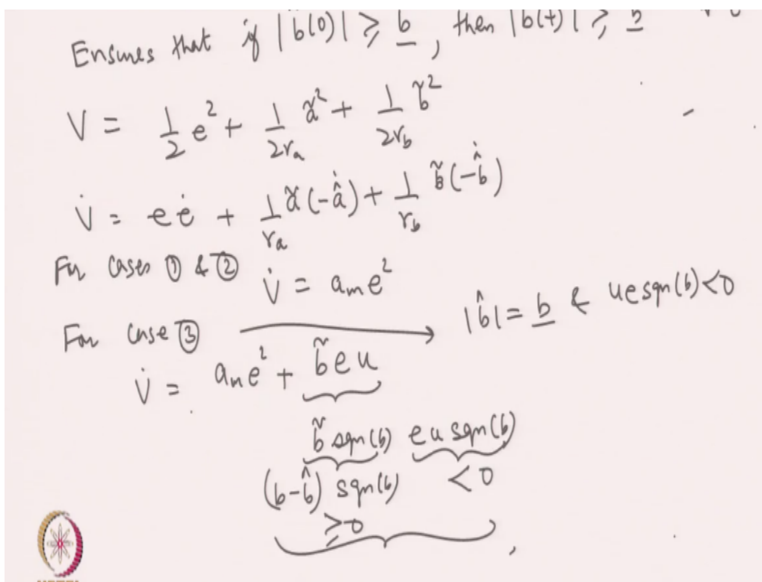
So the solution in a nut shell is represented by this so say so this is 0 and we know the lower bound on the actual parameter, system parameter b. So we know that the actual parameter lies in this region. The wiggly lines represent the region where the actual system parameter b lies and this is what we term as the forbidden region. So we do not want b hat to be anywhere inside the forbidden region. As long as it is on the wiggly lines we are fine with it.

So to make sure that there is this modification to the update law we use these modified expressions for b hat dot and all of this we had done last time so this is just a recap of that. So b hat dot is given by gamma b * ue for the case at the actual value of b hat is > b which is given by the same gamma b * ue. These update laws are dictated by the Lyapunov analysis. So it stays the same if b hat is on the boundary.

And the derivative of b hat at the boundary points in the region which is outside the forbidden region that means in the region where we have these wiggly lines so which is given by this expression. So this just represents that the derivative of b hat points in the direction outside away from the forbidden region. So that happens. We continue to follow this update law. Otherwise, we say that b hat dot = 0 which means that b hat retains the value that it had previously.

And we continue evaluating for these conditions and based on where b hat is we apply b hat dot cases. So this is case 1, this is case 2 and this is case 3. Case 1 and case 2 we basically apply the same update law that came out from the Lyapunov analysis and for case 3 when b hat lies on the boundary and the derivative is pointing inside the forbidden region that is when we stop the updation of b hat.

(Refer Slide Time: 09:25)



So this ensures that if we select the initial estimate to be > b lower bar then for all time we can say that b hat of t is >= b lower bar. This is for all time. So these update expression makes sure that if initially we select the estimate of b hat in the region where we have wiggly lines then for all future time it stays there and it does not come inside the forbidden region which is what we had desired. So for the indirect MRAC case we can say that we require the lower bound on b and the sign of b.

So this is slightly different from the direct case where we had only required but the sign of b was not in this case in addition to the sign of b you also require that the lower bound on the actual parameter b be also known. What we had not done last time was with this modification do we get a stable controller? So stabilizing control system so that is something that we will be doing in this lecture.

So let us consider the same Lyapunov function candidate that we had considered last time which is 1/2 * e square + 1/2 * gamma a * a tilde square + 1/2 gamma b * b tilde square. So this is positive definite readily unbounded decrescent Lyapunov function candidate. So we take the time derivative and then substitute for the closed loop system. Now if we substitute for the expression for a hat dot and b hat dot that I have just mentioned then what we get is 3 cases because for b hat dot we have 3 cases so for cases 1 and 2.

We see here that we use the same expression for b hat dot. So we get v dot to be = am * e square this is what we had gotten last time. So there is no change. So the change is in case 3. So for case 3 where we had actually modified the update law to b hat dot = 0 we get a different expression for v dot. So let us calculate what we get. So basically we have to substitute for b hat dot = 0 here what we end up with is am * e square + b tilde * eu.

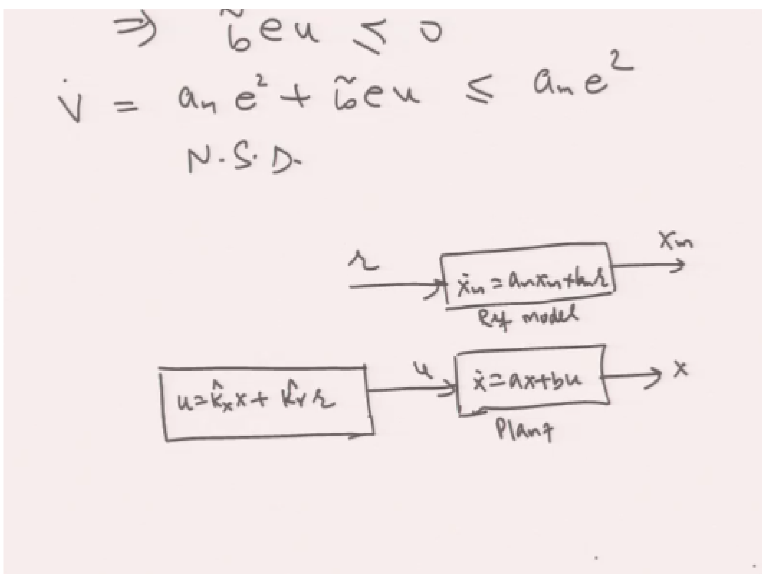
So this is what you will get when you do with the case 3 we get an extra term. So if you look at cases 1 and 2 we get v dot to be negative semi-definite and so the stability of the equilibrium point is assured. For case 3 we need to figure out if v dot is negative semi-definite. So am * e square is a negative term, but b tilde * eu is the term that we would like to know the sign of. So it is not apparent what the sign would be, but let us look at what case 3 is actually.

So what is case 3. So let us go back and look at the condition for case 3. So case 3 is the condition where let us just write it down. So case 3 is when b hat is on the boundary that means the absolute value of b hat = b lower bar and the derivative is pointing inside the forbidden region and that case when ue * signum of b < 0. So using this information and the fact that b hat dot = 0 let us see what we get.

So we get v dot to be = this and now the task is to figure out the sign for this term b tilde eu. So we can rewrite this term as b tilde * signum of b * e u * signum of b. So I have just multiplied this expression by signum of b square which is = 1. So it does not really affect anything, but it will help us figuring out the sign of this term. So we know that eu * signum of b is < 0 because that is case 3. What about this expression b tilde signum of b.

So b tilde signum of b is b - b hat * signum of b. So let us consider the 2 cases when the first case is when b is positive so then sign of b is > 0 and b - b hat is also positive. So this unless hat becomes = b so in that case it will be 0 so this expression will be >= 0 for the case when b is > 0. Let us consider the case when the parameter b is negative. So in that case signum of b will be - 1 and b - b hat would also be negative and so this entire expression would again be >= 0. So we can safely say that this >= 0 and these 2 combined would give you something which is <= 0.

(Refer Slide Time: 17:35)



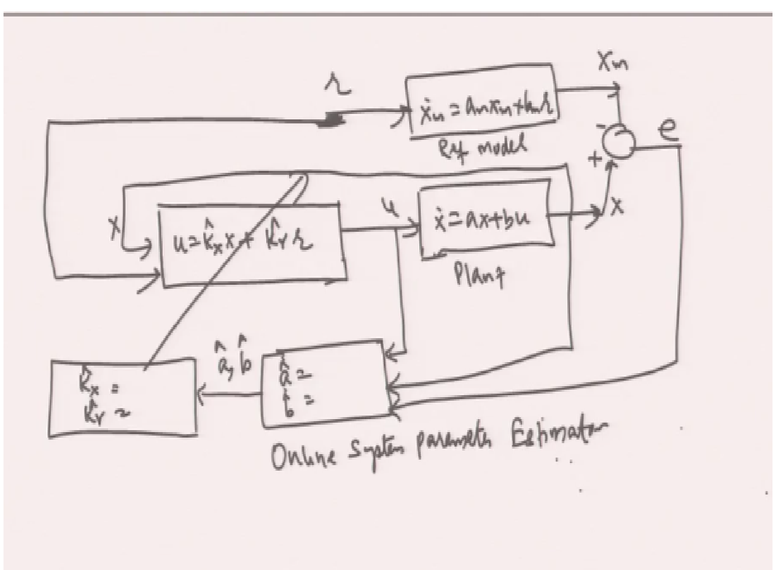
So what we can say from here is that b tilde * eu is always <= 0. So then let us look back again at v dot. So v dot is given by am * e square + b tilde * eu and we know that b tilde * eu is <= 0 and am * e square < 0 so we can say that v dot is <= am * e square which still makes this v dot as negative semi definite and so what we can conclude from here is that the same properties hold that we had for the case where we had used unmodified update laws for the case where we projected the estimates tool to lie in a region which is away from the origin.

What we see is that we again get v dot to be negative semi definite which means that the equilibrium point is Lyapunov stable and all the signals are bounded using Barbalat's lemma again we can prove that the tracking error converges to 0. So the stability properties are preserved even in the case of projection modification of (()) (18:58) laws so even that this modification we are able to prove that the system is Lyapunov's stable.

So in both the cases direct and indirect we have been able to show that we get stability. For the indirect case if we have to implement then we can look at a block diagram for that case. So we have a plant x dot = ax + bu with an input u and an output x. This is the reference model. This is the plant; this is the reference model which is the desired behaviour that we want from plant so this is given by Am * Xm + bmr.

This is given by the reference signal r and output is xm. So what controller have used here we have used a straight feedback controller with the dynamic gains so u is given by kx hat x + kr hat r and these expressions for kx hat and kr hat are obtained from the update laws of the system parameter estimates a hat and b hat.

(Refer Slide Time: 20:40)



So we have an online. So we have a hat dot = whatever we had found and b hat dot has the projection modified update laws. So these are the online system parameter estimator and the input to this block is the control input u the state x and the tracking error e which is just a difference between the plant state and the state of the reference model and what we get is a hat and b hat as the output which are the estimates of the system parameters.

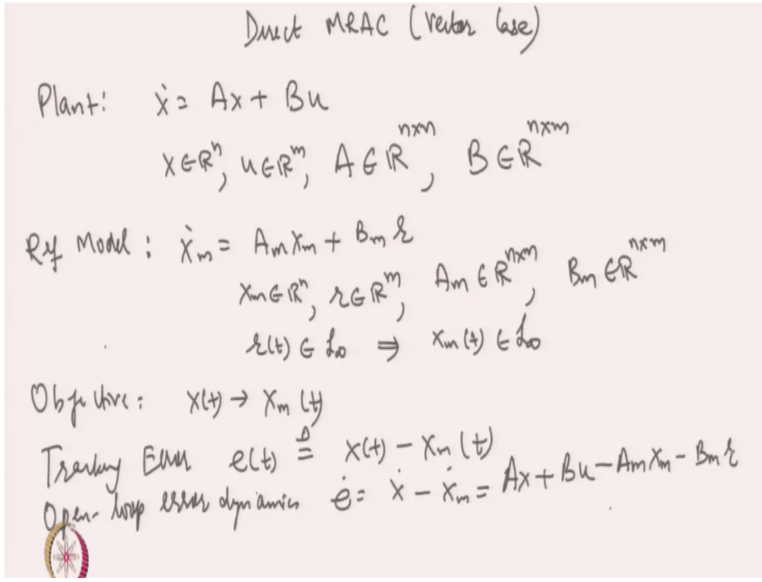
These are then used to compute the controller parameters kx hat and kr hat. So that expression we have mentioned to be this and this. So once we know a hat and b hat we can evaluate kx hat and kr hat and that is then used to update the controller. So the input to the controller is x and r. So this is in fact connected. So this looks fine. So we have seen the 2 cases direct MRAC and indirect MRAC.

And we have considered the plant to be unknown the system parameters A and B are unknown the plant is parametrically uncertain and we have been able to design an adaptive controller to track a reference model and we have actually shown that the equilibrium point is stable. Further we have also shown that the tracking error is convergent. So as for the difference between the direct and the indirect approach there is no general guideline as to which one is better.

It depends on what your application is in certain cases direct may be more suited for example for the indirect case we have an extra step where we compute the controller parameters using an algebraic relation so this step can get computationally expensive for large dimensional systems. So in that case the direct approach may be more beneficial because we directly compute the controller parameter at kx hat and kr hat.

However, in certain cases where you benefit from knowing the estimates of the system parameters a hat and b hat it might be better to use the indirect approach. So there is no general guideline as such. It varies from case to case. So now we move to the case where we have more than 1 state. So we consider a scalar example which illustrated the mechanics of how to design these adaptive controllers it becomes slightly more complicated if we consider the vector case.

(Refer Slide Time: 25:29)

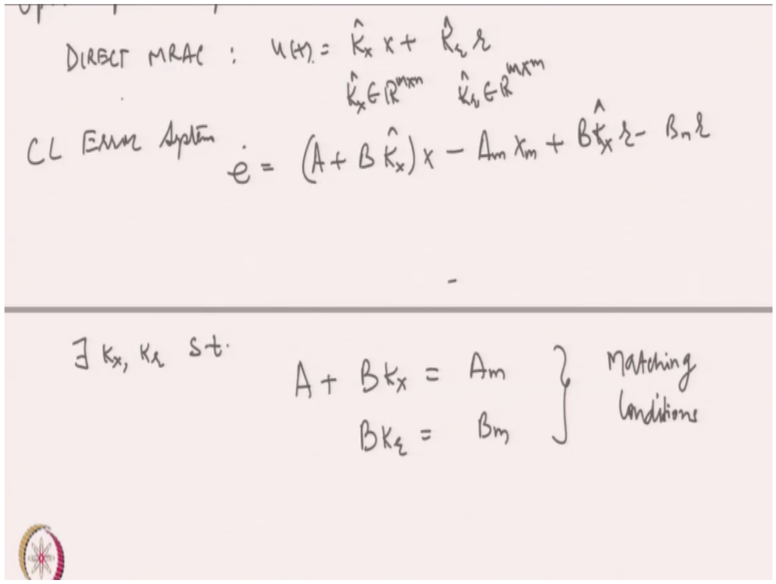


So I am going to do the direct MRAC for the vector case. So we considered the plant Ax + Bu so x is n dimensional and u is m dimensional which means that the matrix A is n x n and B is n x m. The reference model similarly is given by xm dot = Am * Xm + Bm * r. Again xm is n dimension, r is m dimensional and Am is n x m and Bm is n x m. Then again r(t) is a bounded signal which means that the reference model state xm(t) is also bounded.

So we have set up the problem. The objective again is to track the state of the reference model. So the control objective is stated x(t) tracking xm(t). So, although the mechanics here is similar to what we have done for a direct scalar case. There is some certain aspect which I want you to be exposed to and that is why I am doing this again for the vector case. Also notes that here in all the cases that we had done so far we assume that the states are available for measurement that means that there are sensors available to measure the entire state vector x.

The problem becomes more involved, more complicated when we say that we could only measure the output why which may not be the entire state vector. So let us continue with this problem. So the tracking error e of t is defined as x(t) - xm(t) and the open loop error dynamics is given by differentiating the tracking error.

(Refer Slide Time: 29:38)

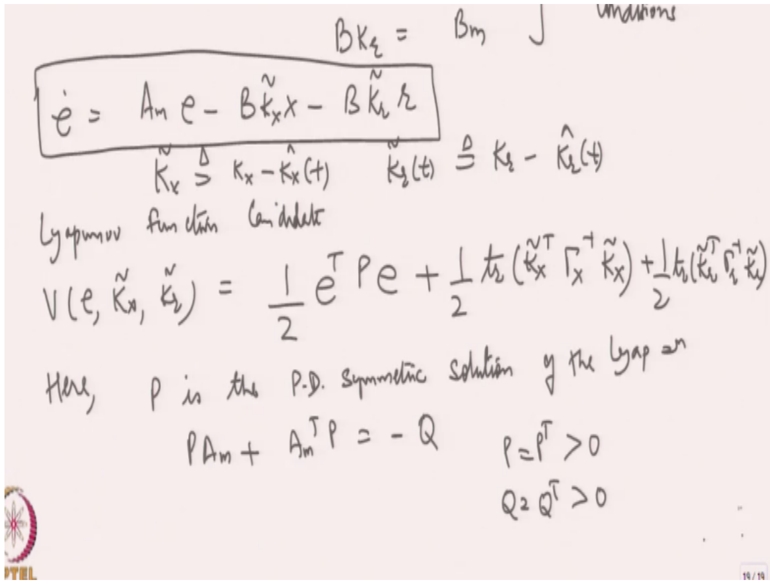


And the controller u similar to the scalar case is given by kx hat x + kr hat r. So this is a state feedback controller where kx hat and kr hat are time varying gains. So notice here that the dimension of kx hat is given by n x m matrix and kr hat is also a matrix in fact it is a square matrix and m x m. So here the control gains are in fact the control parameter are in fact matrices let us see how that changes the entire analysis.

So once you substitute this u in open loop error system we get the closed loop error system so the closed loop error system is given by substituting u in the expression for the open loop error system and what we get is (A + B * kx hat) x - Am * Xm + Bkx hat r - Bm r. So again we consider that there exists ideal gains kx and kr such that A + B * kx = Am and Bkr = Bm. So these are the matching conditions in the vector in the scalar case the matching conditions can be computed trivially.

And they exist for all values of A, B, Am, Bm, but in the vector case as you can see since A, B, Am, Bm are matrices it may not be possible to always find kx and kr which satisfies the matching condition. So in this case this matching condition assumption becomes very crucial and so once we assume that the matching conditions are true instead of using kx and kr we use kx hat and kr hat because we do not know the actual system matrices A and B so it is not possible to compute kx and kr before hand. So using these matching condition in the closed loop error system and doing some manipulation like we did for the scalar case.

(Refer Slide Time: 33:04)

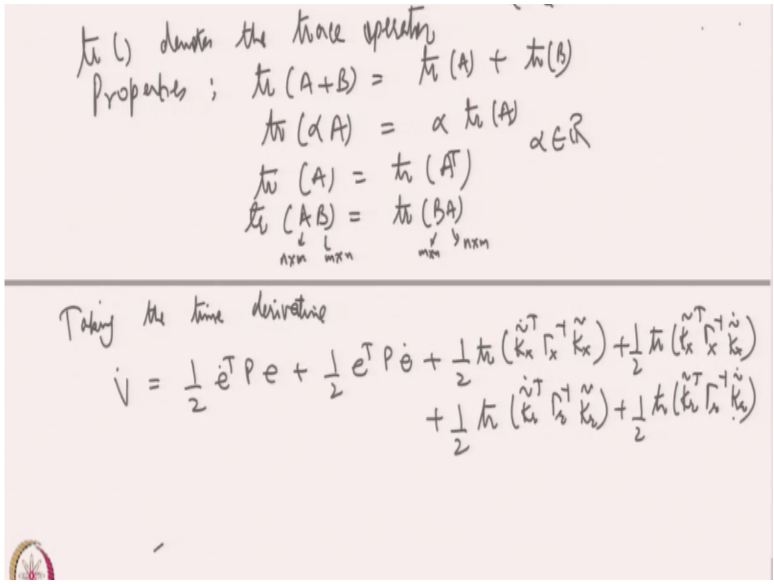


What we get is e dot = Am e - Bk x hat - so this is what we get where kx tilde is defined as kx - kx hat and kr tilde is defined as kr - kr hat. So these kx tilde and kr tilde are the parameter estimation errors and these case controller parameter estimation errors. So now we have got this closed loop error system and we need to analyze the stability of overall system consisting of the tracking error as well as the parameter estimation error.

So we consider Lyapunov function candidate similar to what we have done for the scalar case here there are some differences so v is the function of e and the other error states. So one way to choose the Lyapunov function candidate in this case is e transpose Pe + trace of kx tilde transpose gamma x inverse * kx tilde + transpose gamma r inverse kr tilde. So here p is the positive definite symmetric solution of the Lyapunov equation.

So what I forgot was that here Am is Hurwitz so it is a stable reference model. So p satisfies Lyapunov equation. P is positive definite and symmetric and Q is also positive definite.

(Refer Slide Time: 36:20)



And symmetric and tr represents the trace operator denotes to equates the operator which is the trace of a square matrix and which is given by the sum of the entries on the main diagonal. Why we have chosen the Lyapunov function candidate in this way so one reason that you could think of is that we want to chose the positive definite function and the first attempts that we want to make chose a sum of square so in this scalar case we had chosen the first term as 1/2 e square.

Since e is a vector here we choose it to be 1/2 e transpose pe and it just becomes more elegant if we use p as we will see later. The expression 2 and expression 3 are positive definite expressions so you can prove that this expression inside this trace expression involving kx tilde and kr tilde are in fact positive definite. So we want to choose some positive definite term involving the parameter estimation errors and since these are matrices and v the scalar.

We want to come off with the term which is positive definite, but at the same time it is a scalar term. So this trace operator is very useful in this scenario. In fact, it has very useful properties which will be exploiting so trace is actually a linear operator and some of the properties that we will be exploiting the trace of 2 square matrices A and B is given by so sum of these is given by the individual sums of the traces.

So trace of alpha * A is given by alpha trace of A where alpha is a scalar. Trace of A = trace of A transpose and a very important property that we will be using is the product of 2 matrices which are compatible dimensionally so A is n x m and b is m x n then trace of AB is given by trace of = trace of BA where B we know that is m x n and A is n x m. So the entire the product becomes m x m. So this is an important property that we will be utilizing.

So this Lyapunov function go back to the Lyapunov function. I just mention that this Lyapunov function is positive definite. It is also radially unbounded and decrescent. So why about decrescent here is because the closed looped system involving the tracking error e and the dynamics of kx tilde and kr tilde is in fact non-autonomous because of this reference signal r on the right hand side which is time varying. So if you want to use the Lyapunov theorems that we did in the previous lecture then we should talk about decrescent if are to conclude uniform stability.

So let us take the time derivative of v. so taking the time derivative v dot is given by e transpose * Pe + 1/ 2 e transpose * pe dot + 1/2 trace of (kx tilde dot transpose * gamma x inverse * kx tilde) + 1/2 trace (kx tilde transpose * gamma x transpose * kx tilde dot) + 1/2 trace of (kr tilde * gamma r transpose * kr tilde). So by now you should be comfortable with Lyapunov analysis and the fact that the analysis can actually help you construct the controller it is not just an analysis tool by it is also a design tool.

So then we go and substitute for e dot from the closed loop error equation.

(Refer Slide Time: 42:25)

$$
\begin{aligned}
&= \frac{1}{2}\left(e^T A_m^T - x^T \tilde{K}_x^T B^T - r^T \tilde{K}_r^T B^T\right) P e \\
&+ \frac{1}{2} e^T P \left(A_m e - B\tilde{K}_x x - B\tilde{K}_r r\right) \\
&+ tr\left(\tilde{K}_x^T \Gamma_x^{-1} (-\dot{\hat{K}}_x)\right) + tr\left(\tilde{K}_r^T \Gamma_r^{-1} (-\dot{\hat{K}}_r)\right) \\
&= \frac{1}{2} e^T \left(P A_m + A_m^T P\right) e + e^T P\left(-B\tilde{K}_x x - B\tilde{K}_r r\right) \\
&- tr\left(\tilde{K}_x^T \Gamma_x^{-1} \dot{\hat{K}}_x\right) - tr\left(\tilde{K}_r^T \Gamma_r^{-1} \dot{\hat{K}}_r\right)
\end{aligned}
$$

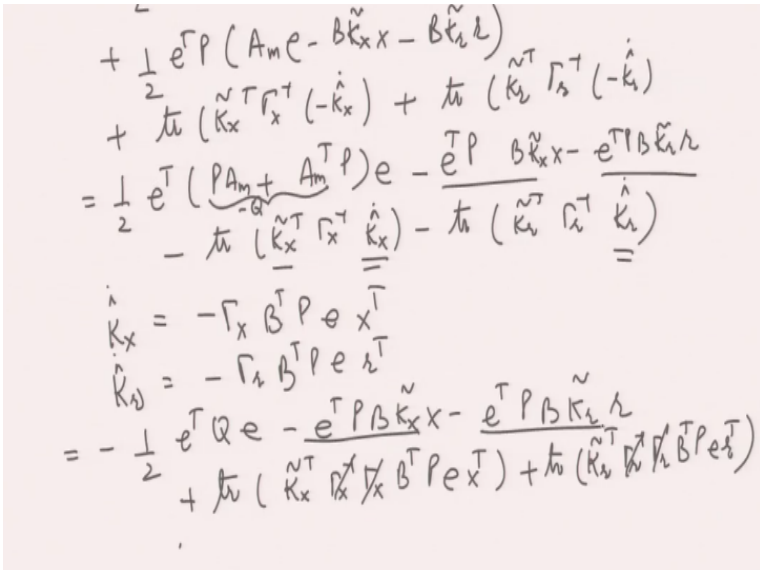
So this becomes e dot transpose so we have to take the transpose of that expression (r transpose * kr transpose * B transpose) + 1/2 e transpose P(Ame). So if you look at these 2 expressions and use a property that transpose of matrix = the trace of the matrix = trace of its transpose. We can expect that to get the expression as trace of kx tilde transpose gamma x inverse * kx hat dot + trace of kr tilde transpose gamma r inverse * kr hat dot.

So we have taken the derivative of kx tilde and kr tilde and this is what we get. So let us cancel common terms and after some manipulations what we get is 1/2 e transpose P Am + Am transpose P * e and then the rest of the terms. So the first term in the expression for v dot is this one. We know that P * Am + Am transpose P = - Q. So this is what we had assumed here that p is a solution of this Lyapunov equation and so we can replace this by - Q where Q is positive definite and so the first term in fact becomes negative.

Let us look at the other terms now we want to choose the update laws for kx hat and kr hat in such a way that we cancel the other terms in the expression for v dot because those terms are signed indeterminate. We do not know what the signs are. So it is better to cancel them using these update laws. The only challenge here was with the trace operator how do we come up with the update laws for kx hat and kr hat.

In fact, the trace operator will help us choose the update laws so if you look at this expression let us just simplify this expression to make our life easier so what we get here is - so this is - e transpose p and lets us get rid of the bracket here. Bx -. So now it becomes easier so we have kx tilde transpose here and we want to choose kx hat dot and kr hat dot and we have to cancel these 2 terms.

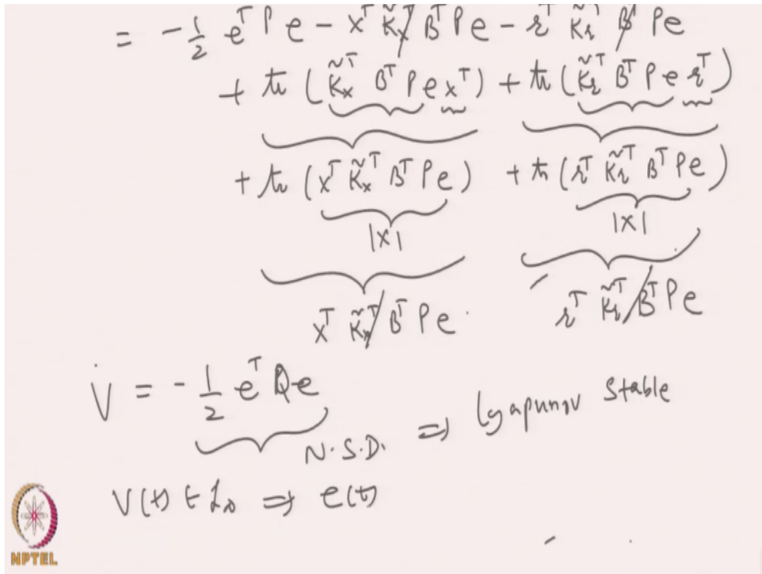
(Refer Slide Time: 48:40)



So if you look carefully and choose kx hat dot to be - gamma x * B transpose Pe * x transpose and kr hat dot to be B transpose * Pe * r transpose. So if you choose the update laws like that then you see how they are going to help us complete this analysis. The first term is e transpose Q e so we do not have a polym of this term because this is a negative term and then we have - e transpose * PB * kx tilde x - e transpose * PB * kr tilde r - trace of kx tilde transpose gamma x inverse kx hat dot

So which is what we get is a positive sign here gamma x B transpose Pe ex transpose. So it still looks very difficult to convert it in the form such that we can cancel these 2 terms and if you use a property of trace we see that it is doable. So one thing that we can do is to take the transpose of these terms that will make life easier for us.

(Refer Slide Time: 51:03)



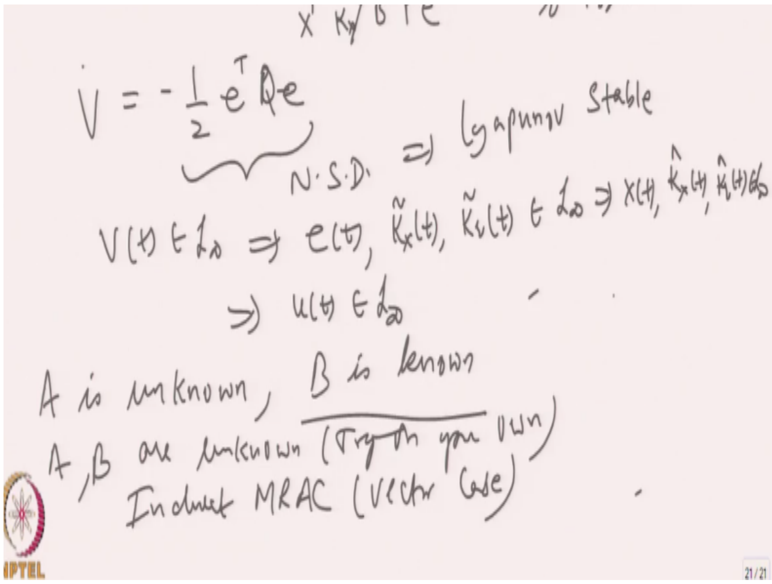
So let us do that so these Lyapunov analysis that typically run into pages we have to be patient with it r transpose kr tilde transpose * B transpose since P symmetric then we have + trace. So these update laws are strategically chosen to cancel these terms. So they are come out of (()) (51:53). So now it looks likes we can do something about it so if you consider this to be a matrix and this to be one similarly here if you consider this to be 1 equation this to be the other matrix.

Then we can use the product rule which is the trace of the product of 2 compatible matrixes = the trace of the matrixes taken in the opposite order so we can write this term as trace of x transpose kx tilde transpose * B transpose * Pe and this term we can write as r transpose * kr transpose * B transpose * Pe. So now it looks like at least the augment of the trace is similar to is in fact the same as the other terms but how do we cancel that because now we have a trace operator in these 2 terms.

So what we if you find out the dimension of the augment matrix. In both the cases these 2 are in fact scalar expressions. So these are scalars and trace of a scalar is simply the scalar itself. So this is further = x transpose * kx tilde transpose. So we remove the trace from this expression and this one also becomes r transpose * kr tilde transpose * B transpose * Pe. So now we can cancel this term and this term with this term. So finally we end up with V dot = - 1/2 e transpose Qe.

So this is negative semi definite and so the equilibrium point is Lyapunov stable. So we can further say that since v is positive definite and v dot is negative semi definite v dot is <= 0 we can say that v is bounded which means that the tracking error. So you have to look at the expression for v. So v being bounded means that e has to be bounded and kx tilde has to be bounded and kr tilde also has to be bounded. So (()) (55:16) very regress about it.

(Refer Slide Time: 55:26)



So e of t * kx tilde of t, kr tilde of t, r are all bounded and since xm which is the state of the reference model and kx and kr are all bounded quantities what we can further say is that x(t), kx hat (t), kr hat of t are also bounded so have we proved that all signals are bounded no we still need to prove that the control input is bounded so if we go back and look at what the control input is.

We find that it is kx hat which we approve to be bounded * x which we proved to be bounded. kr hat is also bounded and r is bounded because it is given to be bounded. So we can conclude from here that u(t) is bounded which is an important statement to make. So in this case we have considered that A is unknown however B is considered to be known. So how do we say that A is unknown and B is known because the controller does not require any information about A.

If you look at the controller u it is given by this expression and then let us look at how kx hat and kr hat are generated. So let us look at the update laws and we find that these update laws do not contain A so this design does not assume any information about the matrix A however you can see that this design still needs the matrix b. So that is something that we have to assume that we know the B matrix.

If you try and do the case where B is also unknown you might end up with a local result that is something that you could try on your own. Just say that global uniform global asymptotic it is not asymptomatic result but uniform global stability. The case where A and B are unknown is something that you can try on your own. You can also try the indirect MRAC in the vector case.

So in this lecture, we have covered the general problem of direct MRAC and we also finished the analysis for the indirect MRAC for the scalar case hopefully this has given you enough information about how to design model reference adaptive controllers. So, in the next class we will continue and so we will do the case where we have some uncertainty in the system dynamics.