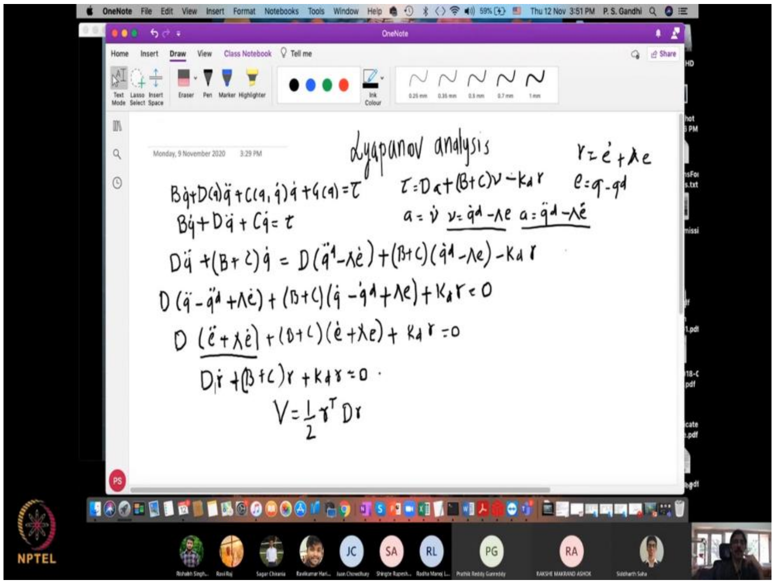
Design of Mechatronic Systems
Professor Prasanna S. Gandhi
Department of Mechanical Engineering
Indian Institute of Technology, Bombay
Lecture – 41
Trajectory Tracking Controller: Robotic System

(Refer Slide Time: 00:16)



So, this is a this is a form of the equation; now in addition will have this here B$\dot{q}$ term. So, this is a form of a equation that will have, so $\dot{D}_q$. So, we are ignoring gravity in this case. So, I am I am now not representing the dependence on the q and $\dot{q}$ of C and D; so this is our equation.

If you recall the controller τ is having this form Da plus; now I will combine this B plus C together here times ν, minus k_d. r. So, if you now see this a is given as $\dot{\nu}$, and this ν is given as q^dminus λe. So, remember these are all vectors here, and this is matrix here; typically, a diagonal matrix and positive definite matrix.

$$a = \dot{\nu}$$

$$\nu = q^d - \lambda e$$

So, so $a = \ddot{q}^d - \lambda$e. So, so we will use this form this a and this while substituting. So, here if we come now like your Bq, so let say let us start with D only here; so that here consistent. Now, with the so this is $D\dot{q}$ $\ddot{q}$, plus now again we combine here this B plus C terms $\dot{q}$, is equal to now if I substitute this here D times a. a is qd double dot minus lambda e dot, plus B plus C qd minus; this should be qd dot d dot here, minus lambda e minus k_d times r.

$$D\ddot{q} + (B + C)\dot{q} = D(\ddot{q}^d - \lambda\dot{e}) + (B + C)(\dot{q}^d - \lambda e) - k_d r$$

So, if I bring these terms on the other now, this becomes now D q double dot minus qd double dot plus lambda e dot. Now, B plus C we can take out and q dot minus qd dot plus lambda e. And then this Kd times r is equal to 0.

$$D(\ddot{q} - \ddot{q}^d + \lambda\dot{e}) + (B + C)\left(\dot{q} \dot{-} q^d + \lambda e\right) + k_d r = 0$$

Now, this I can write in terms of error again, e double dot; so, you remember e is defined here as e = $\dot{q} \dot{-} q^d$, D q minus qd. So, e double dot and then these derivatives will be follow accordingly; so, now this becomes here is e dot plus lambda e plus k_d times r. So, I can see that this is nothing but r dot; if you see r defined here is in the controller.

$$D(\ddot{e} + \lambda\dot{e}) + (B + C)\left(\dot{q} \dot{-} q^d + \lambda e\right) + k_d r = 0$$

You want to see that in controller here, so that for clarity. So, r is e dot plus lambda e; so I can just write here again, r is equal to e dot plus lambda e.

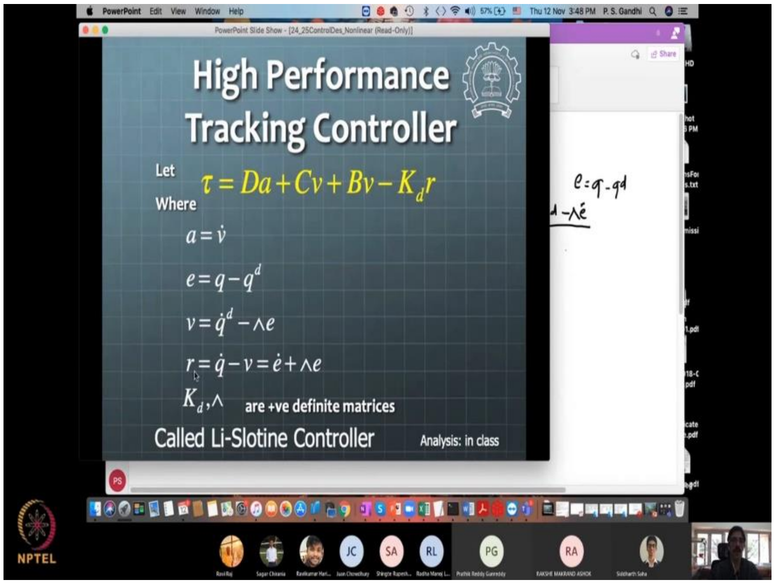
$$r = \dot{e} + \lambda e$$

So, this becomes D times r dot plus now B plus C times r plus k_d times r is equal to 0; and for this we need to propose Lyapanov function.

$$D\dot{r} + (B + C)r + k_d r = 0$$

So, can you think now for such a kind of system if you see what could be? It is not very intuitive; because we do not see the derivation of this from the energy perspective. Because we do not have this form that we started of with D q double dot plus all these kind of things. So, it does not have directly some energy based connection with do here right now.

$$V = \frac{1}{2} r^T D r$$



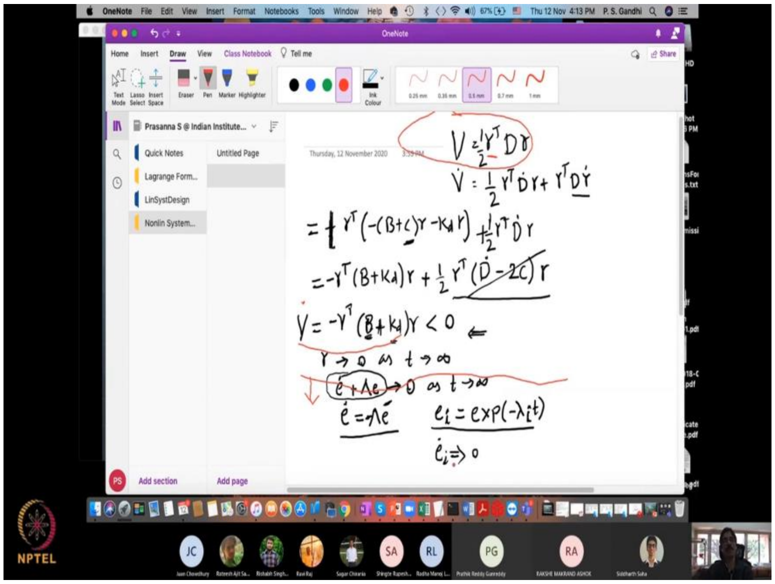
So, therefore we can directly kind of go on using like know this function candidate V as half. Now, I want see then I take a derivative; I will need this $\dot{D}r$ to be substituted, so I will. And D will denote the inertia matrix; so I can use that D as a inertia matrix here. But, instead of now error in terms e and e dot and other kind of a thing, we consider this in r terms and this is r transpose; is what I consider as energy function.

$$\dot{V} = \frac{1}{2} r^T \dot{D} r + r^T D \dot{r}$$

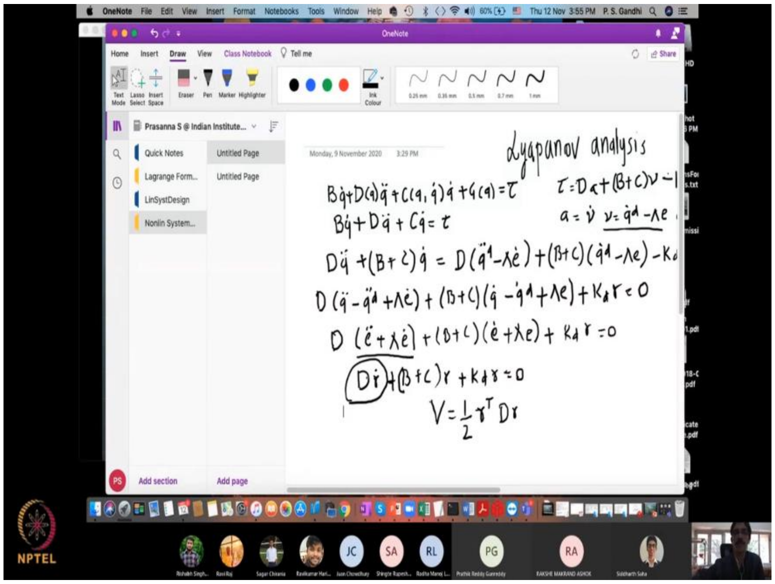
And this function candidate if you see in terms of r, if you expand it in terms of error; one can see that both for e and e dot considered as a, it will be a quadratic form in e and e dot; and that will be positive definite, because D is positive definite matrix. Or, if you say directly in terms of r, it is it is we can directly see that this is; because D is positive definite, this as a positive form, and r contains all my system matters. So, the system has only first order kind of a form here. So, I do not need now r dot to be coming in this expression at all. if this has been r double dot, then I would need r dot to be coming here for the positive definiteness.

Is that part clear that is what is a important kind of a concept here based on our understanding positive definite functions. And we want to propose now this function candidate which is positive definite; and so now you can work out further from here.

(Refer Slide Time: 09:00)

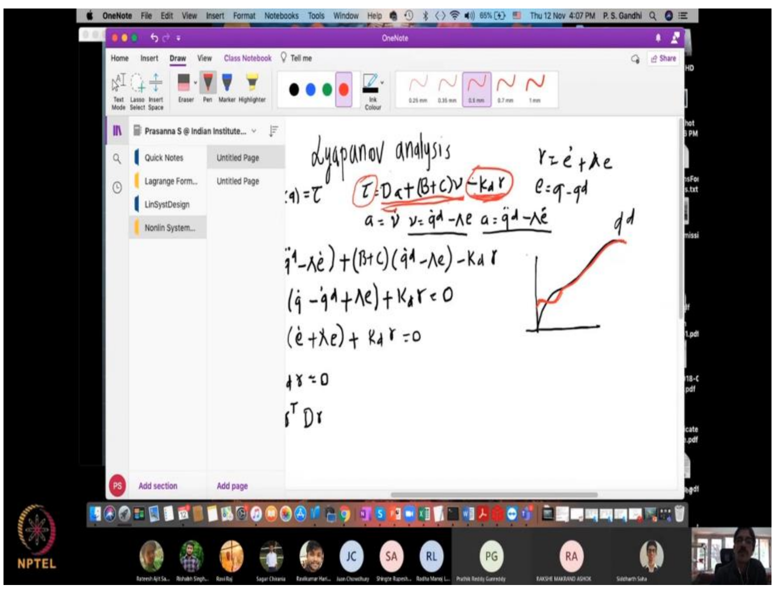


Suppose I take a derivative V dot, so we have now; you remember V is equal to r transpose D times r. And if I take V dot here factor of half rate, half rate of course here; sorry, so half. Now, I see again by using symmetry property I can combine two terms here; and I will do that further other part. I will first consider r transpose D dot r 2 plus; now 2 times, so I need not consider half here. Now, r transpose D r dot, this what I am considering here; because r dot transpose D r dot, r dot transpose D r is same as r transpose D r dot; because of the symmetry of that D matrix. So, that is what property I am using here



Suppose I take a derivative V dot, so we have now; you remember V is equal to r transpose D times r. And if I take V dot here factor of half rate, half rate of course here; sorry, so half. Now, I see again by using symmetry property I can combine two terms here; and I will do that further other part. I will first consider r transpose D dot r 2 plus; now 2 times, so I need not consider half here. Now, r transpose D r dot, this what I am considering here; because r dot transpose D r dot, r dot transpose D r is same as r transpose D r dot; because of the symmetry of that D matrix. So, that is what property I am using here.

So, with that I will get this D r dot kind of a form here, which was there in my previous equations if you see; D r dot is coming here, this is a form D r dot comes. This is what I will substitute now, as minus B plus C r minus k_d r. So, let me do that here maybe here, so that I get more space. So, half r transpose, now minus B plus C r minus k_d r; so let me erase this. So, plus my term r transpose, D dot r will be there; so actually this.



Student: Sir, interchange.

Professor: half is first is not here. So, with this now I can combine these terms the way we bond for D dot minus 2c to be coming up. So, r transpose now I use this c term that side, but here only like now this is my negative; B plus k_d and r, plus now this half r transpose. Now, this will be D dot minus 2c, this c is from the other side from this side here, this side and r. And I know that this is for D dot minus 2c is skew symmetric kind of a property, we get this term to be 0. So, this becomes only now minus r transpose, B plus C B plus k_d and r; so with B positive and r Kd to be chosen to be. B is actually our damping matrix, so it has to be positive; and then k_d some value you choose positive value; but the k_d matrix and then like you see that this has a property less than 0.

So, you will see r vector has both e plus lambda e dot, e dot plus lambda e; so both my e and e dot are going to kind of get represented in r. But, they are like some kind of a linear combination. So, if I now argue from here like this argument is important. So by Lyapanov theorem, I can say that with this I can drive this r go to 0, as t tends to infinity; asymptotic stability for r, because I have represented system in terms of r only so far. So, so given a system the r is equal to 0 equilibrium if I consider, that equilibrium will have a asymptotic stability based on this proof. So, you can see through the details of the definition of the, or the statement of the theorem; and you can see that this will follow. So, r will tend to 0 as t tends to infinity.

Now, what it means that r tends to 0? r has $\dot{e}$ plus λ e; this will tend to 0 as t tends to infinity. So, now if I represent this e as $\dot{e}$ so if this tends to 0; that means e = λ$\dot{e}$, sorry lambda e. So, $\dot{e}$ is equal to lambda e minus; and this will be having. So, if lambda is chosen to be a diagonal matrix, positive definite diagonal matrix; then $\dot{e}$ is equal to minus lambda e would lead all the $\dot{e}$ s, e1, e2, e3, e4 like that. Will have the same form e1 dot is equal to say lambda1 e, minus lambda1 e and thing be that. And if I now write the solution for each of the limits ei, will be then equal to e to the transpose of. Sorry, now this e is exponential e, let us e let us this exponential exponential of minus say, now this lambda 1 small lambda for each of this diagonal elements t;

$$e_i = \exp\left(-\lambda_{it}\right)$$

so that is a solution of a differential equation $\dot{e}$ is equal to minus lambda 1 e, ei dot is equal to lambda. So, so this is lambda let say lambda i here; so will be lambda i. So, all errors are then be given be 0 based on this; so these are the arguments one can use and then one can see that. So, error derivatives, so error i is given to 0, and error derivative also is to be will be then tend to 0; so this is also tend to 0. One can see from whatever ways like, directly from here also one can see it is will follow. Or, one can say that ok, if r tends to 0 and e is tending to 0; then e dot is also tend to 0 from this, this part definition of r.

So, these are kind of arguments one can use like one can establish this proof that, the error indeed asymptotically goes to 0. And this is happening for the case of a controller, which has tracking kind of a nature. This $\dot{q}_d$ is coming here, $\dot{q}_d$ is not 0; so q_d is in general function of time. So, when the when the desired trajectory's function of time, you can apply this controller; and you can see that the error. So, error which is difference between q and q_d this is a error q and q_d , actual trajectory and the desired time varying trajectory. It is not a constant part, this is like a time changing term; that error is driven to 0. So, what it means is that your system would, if I draw here just kind of represent.

So, if if this is my trajectory desired, so this is my q_d ; then my let me draw this. Let this actual trajectory is say originating say we started at the same point; but or maybe some other point, we do not know where this starting point for the trajectory would be. But, this will kind of go and settled and follow this trajectory exactly. So, this will this is kind of in the initial transient which will die down, and then it will very perfectly follow this trajectory. That is what it means that here error is driven to 0 for the tracking trajectory. So, any trajectory that you are given which is smooth function of time differentiable function of time; your controller is going to follow that trajectory in with a very small transients will try down, and then it will completely follow that trajectory very nicely.

And other part again I am we are treating that; we see that this controller as this, we are this these terms which are feed forward terms, this Da and B plus C times nu. These are feed forward terms that we are using, and then this is a feedback term; so, this k_d and r this is a feedback term. So, feed forward term and feedback term combination we are using here to derive, so that is. So, this term will take care off most of the control action to get to the this trajectory; and say these transients are initially driven. So, this initially will find with the transient this term is going to be contributing more to the control action.

And once the transient try down, these terms also will be go down; so you can afford to use very high gain for this k_d part here. When used high gain for k_d then your trajectory will maintain along the desired trajectory in a very very nice fashion. So, this is how because most of the terms that are there to take care off input needed for the system to be along the trajectory are already fed forward here. So, because of that the system will anyway going to try to kind of go to that along this trajectory. And whatever small errors coming because of disturbances and other uncertainties in the system, they will be taken care off by this derivative; I mean the feedback part of that.

And that is how you are driven nicely along the along this trajectory. And now this controller if you see this the applicability, it is applicable for all kind of a mechanical rigid body systems; also some kind of a flexible body systems, it is applicable to some certain kinds of conditions. So, this gives you very powerful tool for control implementation for any future mechatronic systems. That is why we will stop here; there are other things like say suppose I do not know this D very well. I have some uncertainty in D, so d is known as d plus delta d; or B and C are also not known completely. How do I handle that uncertainty part and make the controller robust? There are ways to do that basically the idea there is to keep track of what is my previous torque input.

And use that in some way to update things in such a way that I track this robustly do the tracking. Then one can say ok, look I want to adapt to the parameters, parameters are changing. Then again you can have some certain ways to deal with that kind of situation also. So, B, C parameters are changing in time slowly; of course the change is not as fast as your system man made, system man made control is much faster.

But, if that is changing, then also you can do the other controller's development that needs possible. But, the base of this all analysis is what we have covered right now. So, as we want to have these small changes of the robustness or adaptiveness, you will have some, you will find some addition terms defined; and some update law for the system parameters can come up in the process of controlling implementation.

So, of course all those things come as a additional computation cost. So, you see this controller computing will be much costlier then just having a feedback term computed, in terms of your microcontroller implementation. So, that is how we should think about implementation and theoretical development together in some sense. Or, you do the theoretical development and see what is it really viable for my implementation or not; those kind of things need to be thought about. So, that is how we start making (con), so here we have made this conclusion for r alone first. R tends to 0 as t tends to infinity; that is a thing we are getting by applying Lyapanov asymptotic stability theorem.

And now beyond that we are using some arguments of our own; these are not arguments from the Lyapanov. So, these all this from here whatever we are doing here, here onwards down is all our own kind of arguments. When r tends to infinity, then e dot plus lambda e tends to infinity; that means e dot will tend to minus lambda e. And based on this solution one can conclude that each of the components of error or entire error, considered as vector would proceed to 0 as t tends to infinity. So, that is a kind of idea that we have written here.