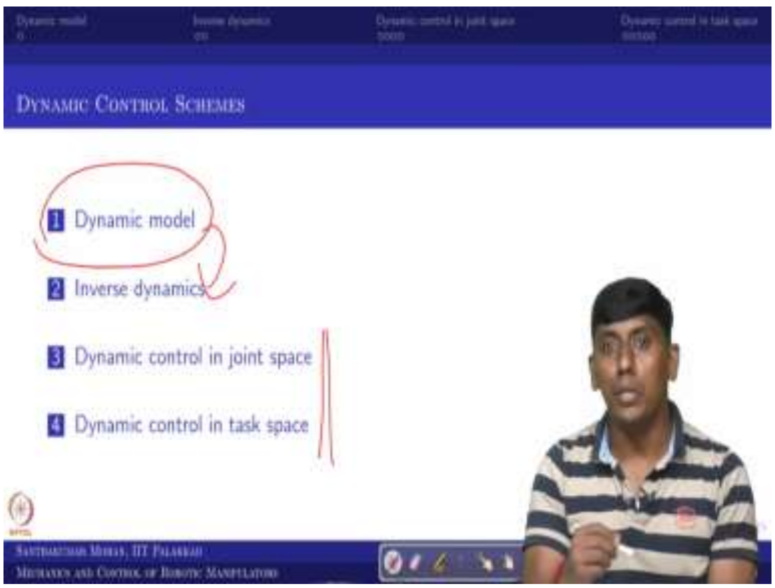
Mechanics and Control of Robotic Manipulators
Professor Santhakumar Mohan
Department of Mechanical Engineering
Indian Institute of Technology, Palakkad
Lecture – 40
Dynamic Control

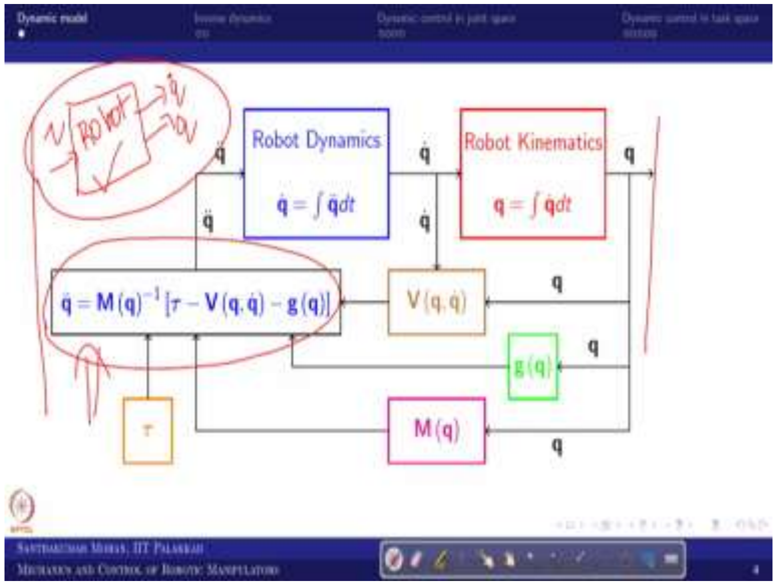
Welcome back to mechanics and control of robotic manipulator; this particular lecture we are going to see dynamic control. In the last lecture itself I told, so next part would be coming on dynamic control; so, will go straight away to the dynamic model first. Then we will see what is inverse dynamics and how we can do dynamic control in joint space and task space.

(Refer Slide Time: 00:34)



So, we are trying to see how to portrait the dynamic model in block diagram approach; and then we will move to the inverse dynamics. Then we will come to the dynamic control where the feedback elements also come into a picture.

(Refer Slide Time: 00:52)



So, then we will try to what exactly the robot kinematics which we called differential kinematic model has given. If you know the velocity, so we can see by integrating this you can find the position. So, what you can see by giving the q dot as an input, the kinematic block would give q as the output. Similarly, if you extend the dynamics blocks, so what would be the output?

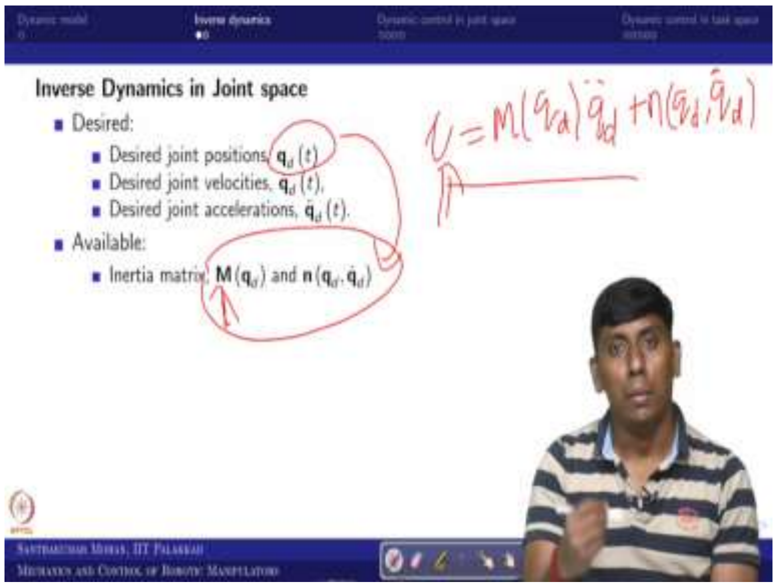
The q dot; but what would be the input, the q double dot. So, in that case so what one can see if I see my exact dynamic model, there is a gravity term is coming. So, I can find the gravity term by taking q as the element or you can say the state vector; then I can calculate. Similar way I can calculate the V of q comma q dot; so, which is a other effect vectors.

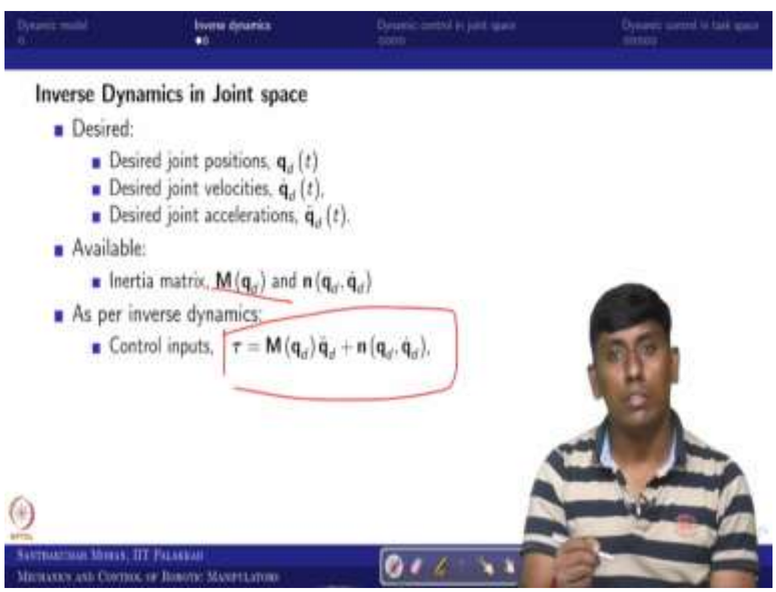
So, where q dot I can take and q, I can take it from here. So, similar way I can find the inertia matrix, because the inertia matrix is function of q; so, I can do it. So, now if I assume that the tau is known to me, which is the input; so, then what I can do? I can do the forward dynamics. Or, if I know the control of this tau, then I can do the control manner.

So, in the sense we can find this q double dot from this particular equation. So, this is the equation which we derived as an equation of motion, and that too like we have written in the acceleration one side; so, this is the equation we have derived. So, now this is what we call the robot dynamic model in block diagram approach. But, if I want to write it in a simple form, so what I can see the robotic system; so, I can call robot.

So, robot will give q dot, and q as the output by giving tau as the input. So, this is the robot dynamic model, simply I can say in this form. So, although this is the detail one; but this particular case I can make it as a simple this way. So, this is what we are going to use hereafter; so, for that only I have given this introduction. Let us move to the inverse dynamics.

(Refer Slide Time: 03:04)



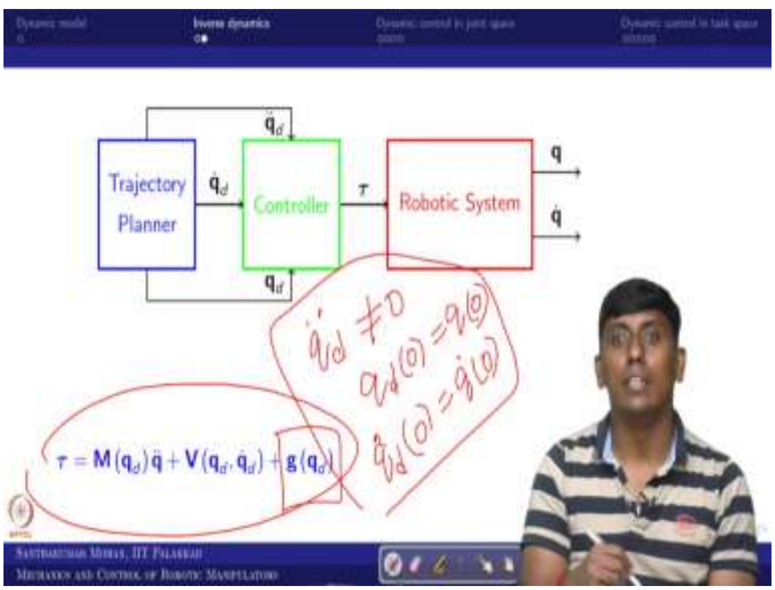


So, what would be the inverse dynamics, what would be given to us? So, the desired position, desired velocity, desired acceleration would be given to us. So, in fact you will go for even fifth order polynomial or third order polynomial we will be knowing this. So, then what would be available to us? So, would be available to us is so the inertia matrix based on the, the simple idea called the q dot desired and q desired is known.

Then we can find this n of comma, n of q desired minus q dot desired; and M of q desired is known. So, in that case what one can find? So, that tau we need to find. The tau I can write as the M of q desired into q double dot desired, plus so n of q desired comma q desired dot can be used. So, this is what the inverse dynamics model; so, this inverse dynamics model would be useful when there is no gravity involved.

Whenever, there is gravity involved, it is very difficult to use this as an open loop control. That I will show you in the simulation time, probably next two next lecture we will be seeing that in detail. But right now we can see in this case so for given this mu like M of q desired, and n of q desired comma q dot desired is known; so we can find this tau. So, this is what we call inverse dynamics.

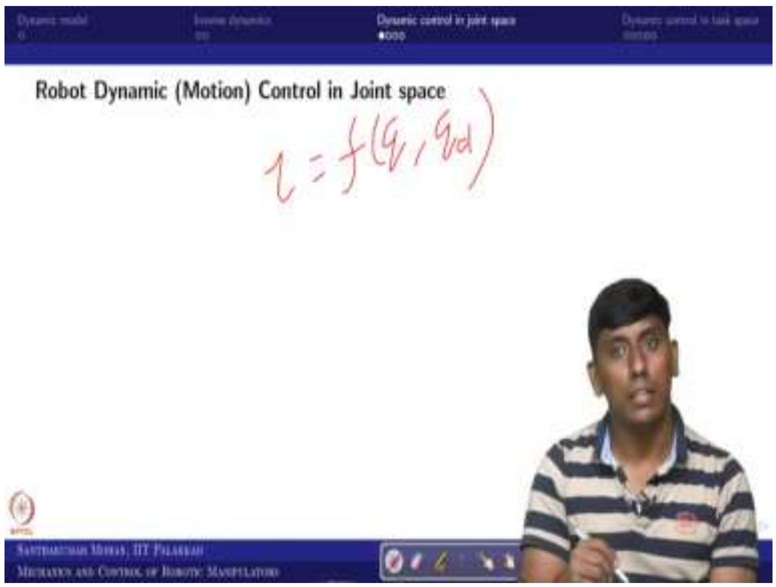
(Refer Slide Time: 04:40)

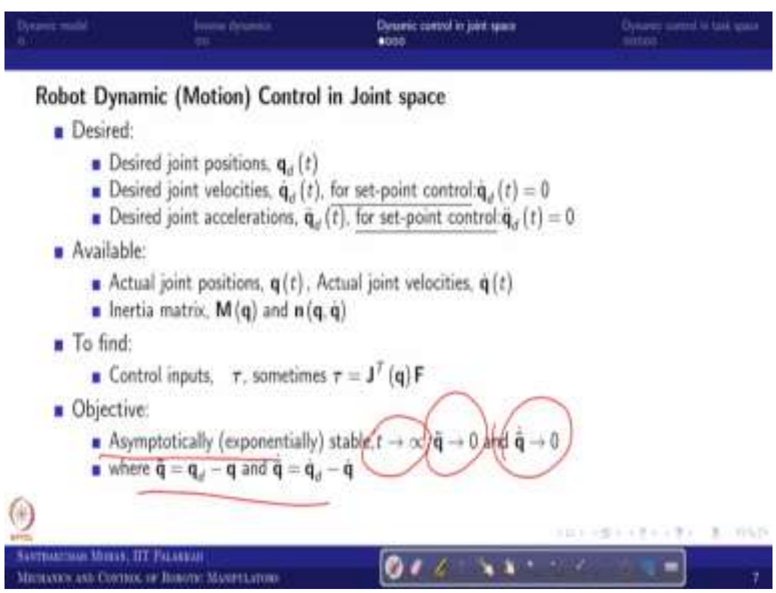


So, even that can go even further end. So, for example this is the robot dynamic model we can take it; so, in that case we can see the controller. So, in this case the controller is like open loop control; that too like feed forward control. So, if you have a block which is trajectory planner is known; so, which will give the q double dot desired, q dot desired, and q desired, then we can calculate tau in this form.

So, this is what we can see, so where I always separate the gravity term; so that we can use a gravity compensation later on. But, as you know so this would be open loop control, if if your q double dot desired is like non-zero. And so, q desired of 0 and q of 0 are same; then q desired dot equal to 0 and q dot also like 0. So, then only this would be work as a proper control. Otherwise, it is a simple inverse dynamics, but that will not follow that exact trajectory what you intended. So, then what would be the option? So, obviously the option would be dynamic control.

(Refer Slide Time: 05:57)





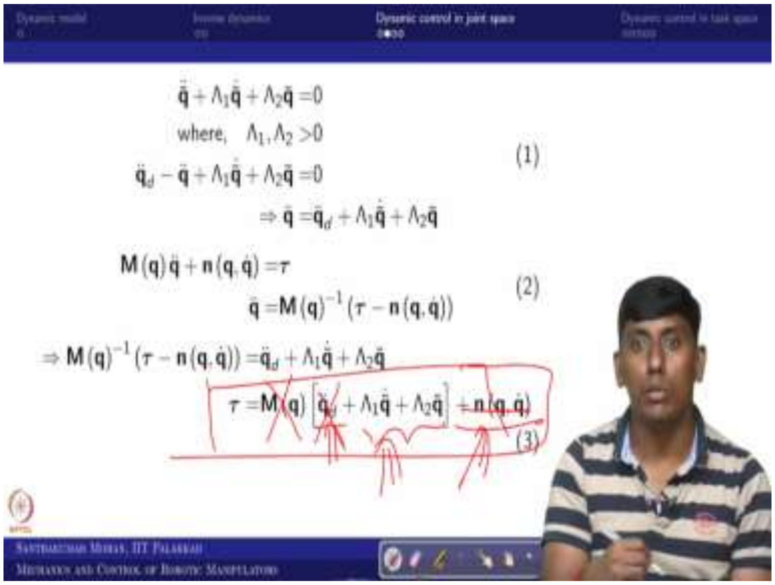
So, the dynamic control means what we additionally we expect. So, the actual position we will consider, take it as feedback, and put it in the loop and try to compensate your tau. So, tau would be function of so q and q desired; so that additional term also would be coming. So, but these are the cases which we are expecting; so, in that case what would be given, q desired.

So, q dot desired for example if it is a set-point, the q dot desired would be 0; so, similarly, q double dot desired also would be 0. But, for a tracking these all three would be given to us. So, what additionally available, we assumed that it is feedback; so then we assume that the actual joint position and actual joint velocity would be available.

Further, if we are doing a model-based control, so the M of q and n of q comma q dot also would be known. So, I already said we usually do a fused based control, so model-based combined with reactive paradigm that is what we are trying to use; model-based along with the feedback, we use it. So, that would be feed forward cum feedback control; so that is what one of the controls which we are going to see in upcoming slides. So, in the sense what we are trying to find out? tau.

For example, if you are trying to do it in a task space. So, then the task space would be F right; so, then J transpose of q equal to F, so, that also we can try to do it. So, in that case what one can expect? So, one can expect that the system is like a closed loop system is exponentially stable or asymptomatically stable; t tends to infinity. So, q tilde and q dot tilde tend to 0; so, what this? So, this is the way. So, the second order error dynamic supposed to be stable; so, we can take it second order error dynamics in this form.

(Refer Slide Time: 07:59)

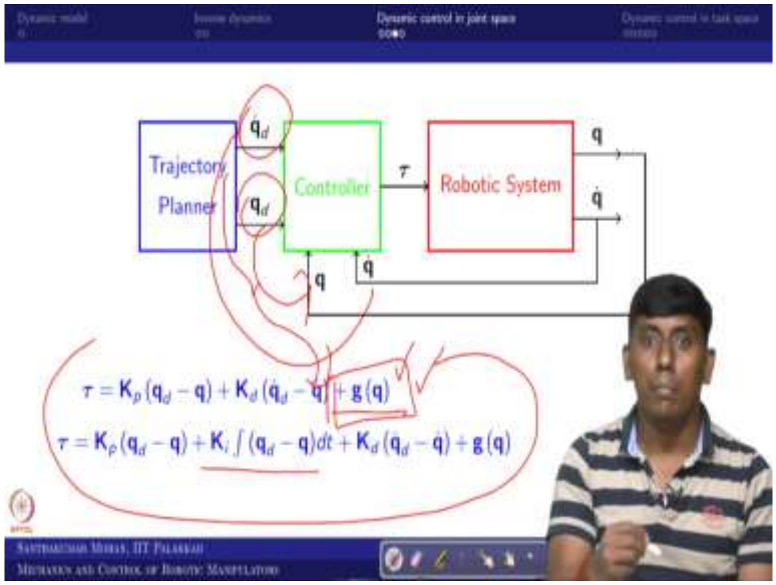


So, where lambda1 and lambda2 are a positive constant; in this case it is a matrix. So, we assume that it is a positive constant then we can see; so, this would be fulfilling what you wanted. So, we expand this equation, so we expand this q tilde double dot as q double dot desired minus q double dot. So, this q double dot we know from the equation of motion, and q double dot desired is available from the trajectory planner; so, this is what we can write it.

So, but this we know from the model or equations of motion; so, this would be known. So, we can equate these two, so what you will get? So, you will get this final form. So, here you can see that the tau is the input; so, we can rewrite and see what would be the tau, so, tau would be coming in this way. Sorry, this is you can see the feed forward term, this is a feed forward term, and this is feedback linearization, and this is the feedback term; here it is a simple PD control as such.

So, this is what we are seeing it as a computed torque control in general in robotic community people call; so, but we are trying to see that here. In addition to that it need not to be all the time like this; in similarly it need not to be this way. So, even the feedback linearization need not to be come; we can make it motion based control, simple PD control or PID control can be used.

(Refer Slide Time: 09:36)

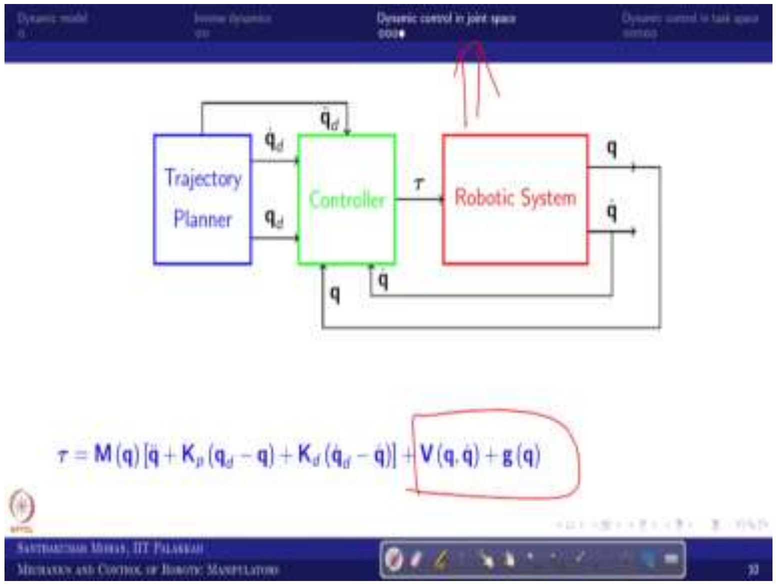


So, for that what we are taking it; so, before going to see that so we can see like the same thing in a block diagram form, so, this is the control. So, now we assume these two are fedback, so now that fedback and the trajectory planner is giving the q dot desired and q desired. So, then what we can see like this is the simple PD control along with gravity compensation. The same thing can be extended with PID, so then i term is coming again.

So, then this is a simple motion-based control; so, there is no feedback linearization or feed forward. So, we see what is your reference and trying to compare; and similarly we are trying to compare. This error so this error we are trying to use as a motion based. So, further if the system is in spatial, so the gravity also needs to be compensated; but most of the modern or even old manipulators all gravity balance system.

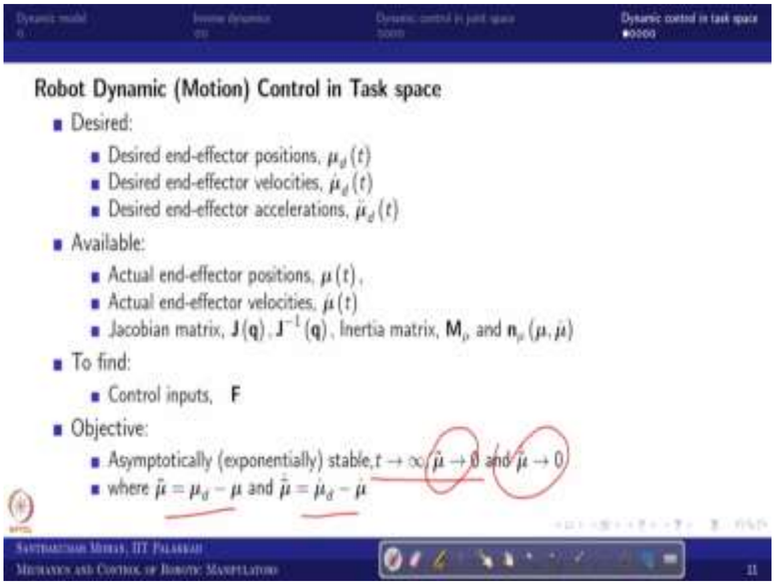
So, either swing-based or the cable driven-based, so or say some kind of electrical actuation based; so, gravity is already balanced; so, in that case simple PD and PID can be used. So, even you take in the industry ready manipulator, the PD control would be used; already we know why PD control is popular in robotic manipulator community. So, because the open loop system is like unstable; but this PD control makes it stable that is what the whole idea behind it.

(Refer Slide Time: 11:21)



So, we will go further; so, how the computed torque control will come here we can see. So, this is the controller, so now we are taking feedback; and in addition to that you are taking the q double dot desired, here taking a feed forward. In addition to that we are trying to compensate these terms; so, this is the computed torque control. And this is we have seen in the joint space; so now we will see the same thing in the task space.

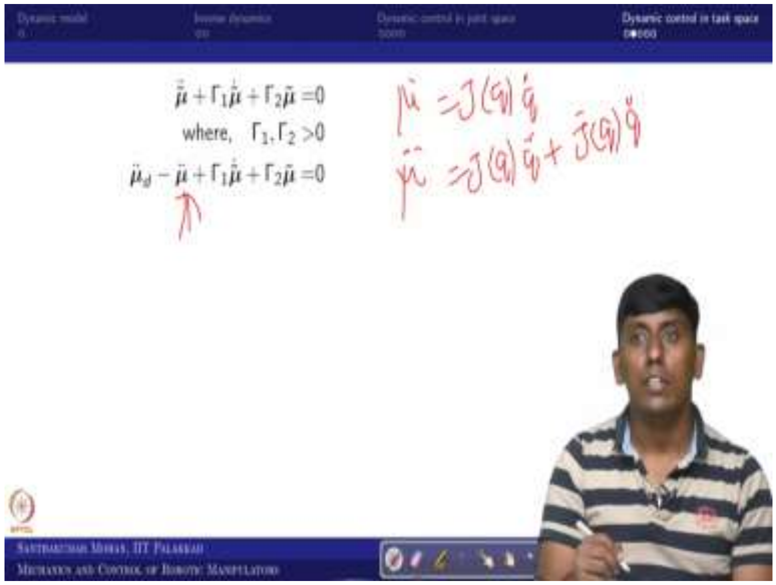
(Refer Slide Time: 11:50)

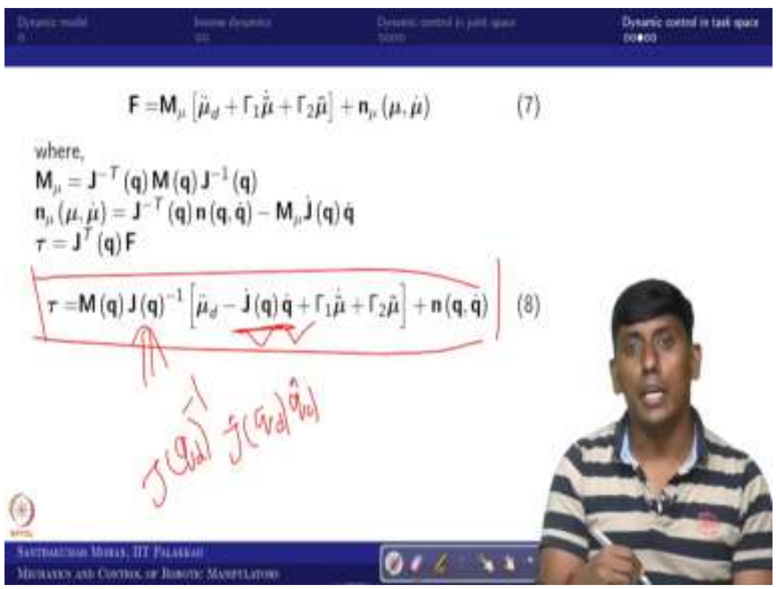


So, what we are trying to see? We are trying to see the motion control in task space. So, for that so what we expected as a given, so the desired end effector position, velocity, acceleration all would be available. Further, so what would be available from the system side? So, actual end effector position and actual joint position; so actual end effector velocity and actual joint velocities would be known explicitly.

Further, we assume that the Jacobian matrix is important, so Jacobian matrix are available to us; and inertia matrix and other effects all available to us. So, then what would be the target to us? We need to find out what would be the end effector forces and movements; so that we can control the system in task space. So, so we expect that t tends to infinity, the mu tilde and mu tilde dot suppose to be 0; and we know what is the syntax, or what is the convention we take mu tilde as mu desired minus mu.

(Refer Slide Time: 12:58)





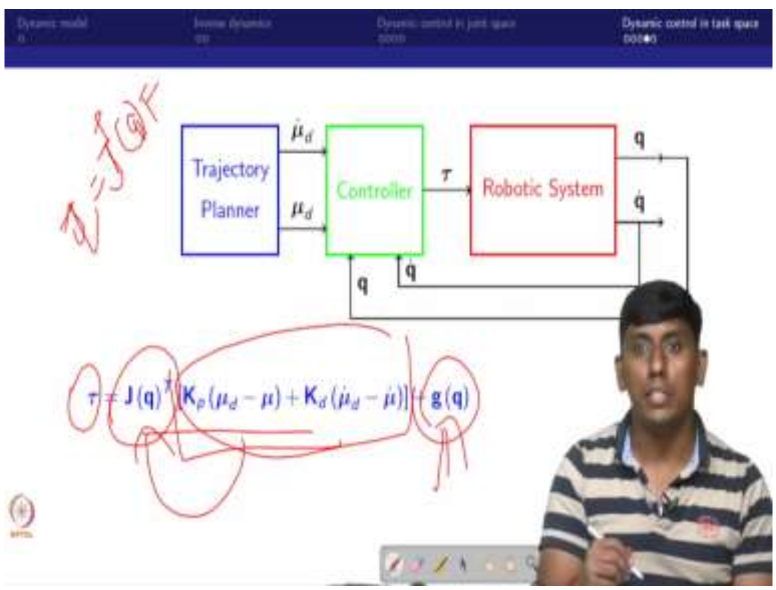
So, we used what we know; so again, we can come back to the second order error dynamics is going to be a stable system; where gamma1 and gamma2 are positive constants. So, only thing, so earlier we have taken q tilde double dot and q tilde dot and q tilde; so here mu tilde double dot. So, we can rewrite this equation, so we know mu double dot; so how we can write?

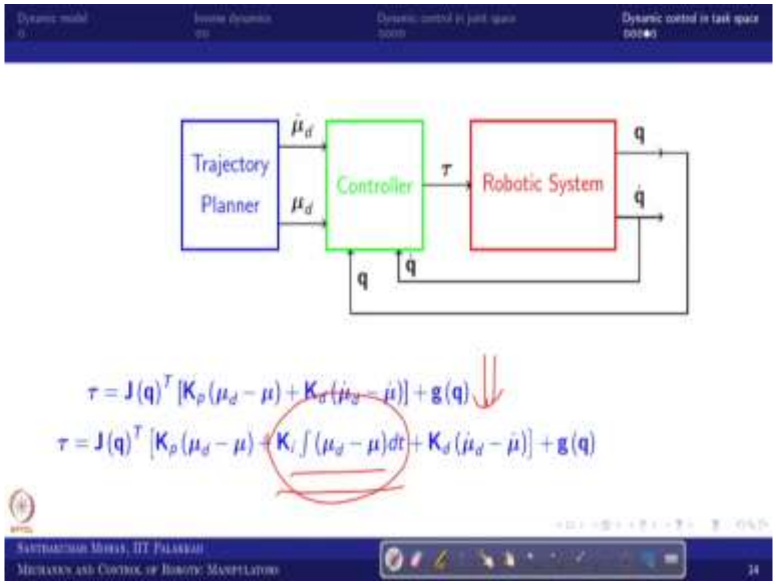
So, we write mu dot as J of q into. So, q dot, so then the mu double dot is J of q into q double dot, plus J dot of q into q dot. So, this can be used, or we can straight away use the direct relation. So, then we can rewrite this here and then mu double dot we can write it from this equation. So, now these two we can equate, then you can see F would come as in this form. But we know certain relations so that is what we are trying to use; so this is what the control input.

But we know already these are the relations, where the M of mu can be written in this form. So, n of mu, mu comma mu dot we can write it in this form; and tau we can write in the form of F. So, now we substitute these all. We try to find out everything in a joint space; because the actuator is connected at joint space, so, we can check it. So, now we substitute, you can see the modified equation comes this way.

So, I already said it need not to be all the time feed forward term and all need to be come here; but we can check it that. So, this have a two options, even you can use J dot of q desired into q desired dot are we can use as a simple feedback. So, either way we can use it, so I am using as like feedback; so that I no need to use it. Similarly, this also some people use, so J of q desired inverse; but we can use actual system itself. So, we will see first so PI and PID like PD and PID.

(Refer Slide Time: 15:11)



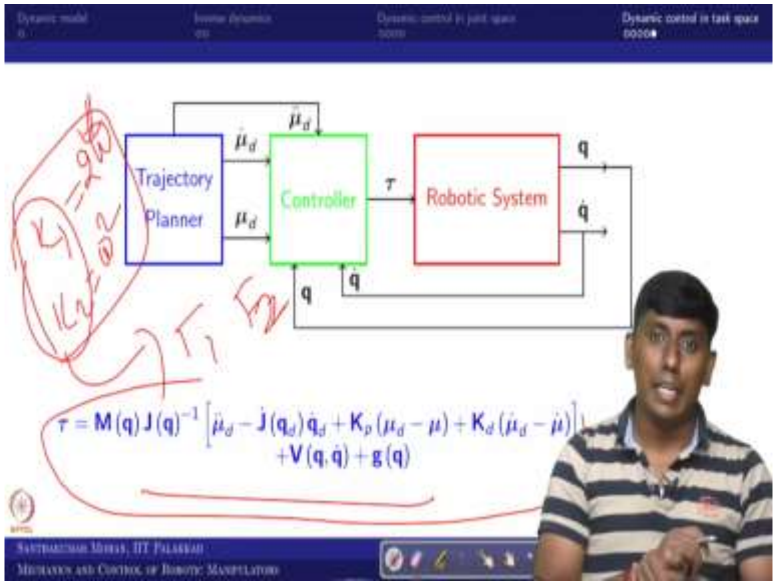


So, then we will go to the computed velocity control or computed torque control in this case. So, we will see this is the controller, the controller required feedback; and required input from the trajectory planner. So, then if it is a PD control, we can use it this way; but you need to give the input in task space. So, we know J is actually like, so the tau is J transpose of, so q into F.

So, we calculate F this way; so, then we added this so that we will come as tau. But we need to always compensate the gravity; but if you do in a real time manipulator, the gravity always compensated. But, when you do a simulation, so then the gravity needs to be compensated. So, that is why we call it is a PD with a gravity compensation control.

Similarly, PID with gravity compensation control can be brought in. So, now this is the I term which is added, so that the steady state error which entered in the PD control would be neglected or converge to 0 by introducing the I term. So, the same way we can extend to the computed torque control.

(Refer Slide Time: 16:31)



So, you can see this is the system and this is the controller; we take feedback, and we give three inputs from the trajectory planner. Both, earlier we give only desired position velocity; now desired acceleration also we added. So, then what happened this is the control law which comes to us; so, this is the computed torque control.

Why this name has come as computed torque control? Because we are computing the torque; so, most of the olden day’s manipulator all are like rotary actuator based. So, that is why the torque need to be calculated; so that too like we use direct drive motors. So, if you compute the torque based on model and the feedback, then we can compensate directly on the actuator level.

So, that is why it is called computed torque control; so, it need not to be called the computed force control; because olden days. So, it was all rotary actuator that is why the computed torque as come as the keyword. So, now the keyword we cannot change; so that is why we are also using computed torque control.

Even you are manipulator have a prismatic joint inside; still we call computed torque control. I hope you are like get some idea about what is dynamic control, and what are the way can be evolve; and how the computed torque control have come. So, what bases it has come, the second ordered error dynamics converge to 0. Now, the choice of gamma1 and gamma2 we take it as per your second order system dynamics.

You want probably critically damped system; then we can take K1 as 2 omega, and K2 as omega square. So, where omega is your output frequency, any how it is like critically damped. So, it won't go any oscillation; but this is the way we can take it. So, now this omega is related with your rise time, so we can make it that way.

So, then the K1 and K2 will come as per that way. So, that the K1, K2 even relate to gamma1 and gamma2 are lambda1 and lambda2. So, now I hope you are getting some kind of idea; so, but if you have a manipulator which is gravity balanced, even simple PD control is quite enough. But you want to go for very accurate precise position following and all, then you have to go for modern control. Even then the model-based control may be having uncertainty I already said; so, then we have to go for robust or adaptive control.

Those things we will see in at the end; right now, we close here with dynamic control. The next lecture we are going to call dual loop or double loop control, where kinematic control and dynamic control we are going to combine. So, most of the control community people call it is back stepping; so that is what we are going to see in the upcoming lecture. Until then see you, bye, take care.