Wheeled Mobile Robots

Prof. Santhakumar Mohan
Department of Mechanical Engineering
Indian Institute of Technology, Palakkad

Lecture - 39
Dynamic Control of Mobile Robots

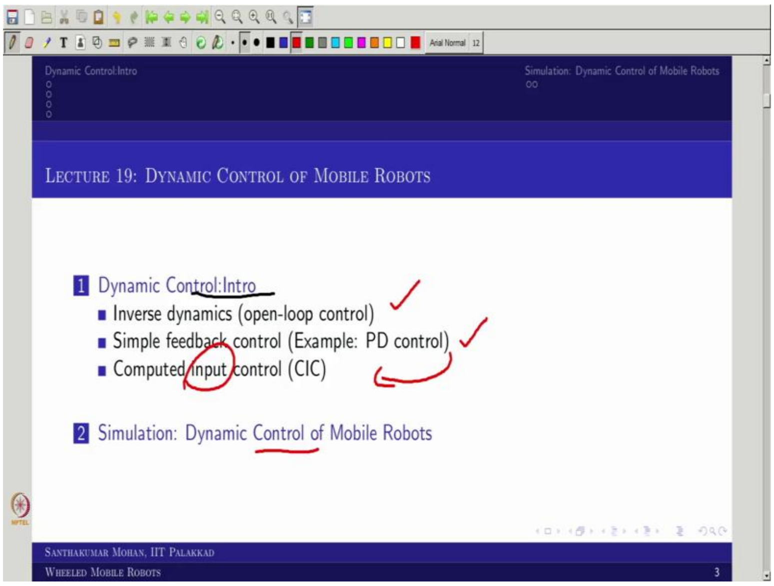
Welcome back to the course on Wheeled Mobile Robot. So, last class what we have seen is actually like kinematic control along with simulation. This particular lecture would be focused on the one another class called Dynamic Control of Mobile Robot. So, here what we are trying to do?

So, we are trying to see how this second order error dynamics can be brought and we will bring one specific you call control scheme called computed input control or you can say computer input which is what you call computed torque or force that we will bring it. So, in the sense, we will try to see how that particular control scheme can be brought and then I will explain how to do simulation by yourself.

So, in this particular, lecture I will not be showing the Matlab simulation, but I will explain how to do by yourself.

(Refer Slide Time: 01:02)

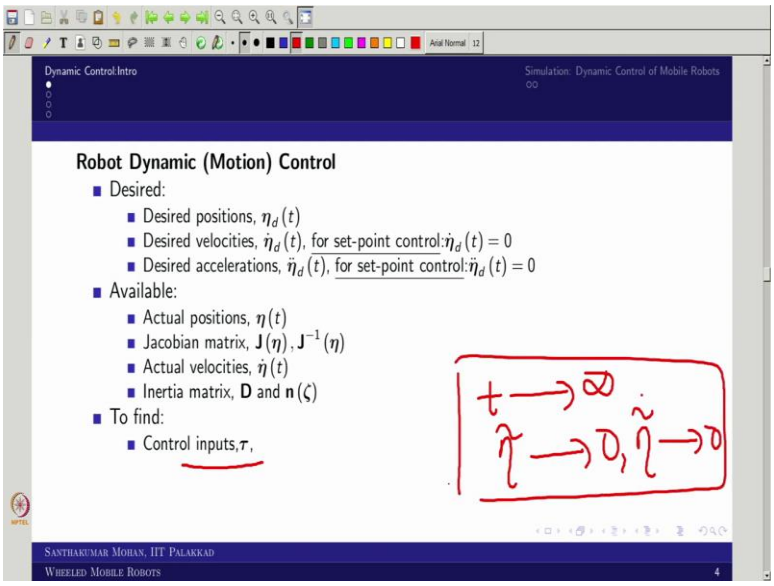
(Refer Slide Time: 01:03)



So, in the sense you will also start learning. So, let us move to the slide, so where we would be talking about so dynamic control; in specific we will see that what is inverse dynamics how that is different from the close loop. Then we will take the simple feedback control based on PD, then finally, we will come back to the what you call computed input control which is generally in robotics what they call computed torque control.

So, the computed torque is the odd word for us because we are not computing all the torque. But if I bring the wheel along with motor then the computed torque would be the proper word, but right now I am generalizing as a computed input control. And the final what I will be showing that how to simulate yourself in you can say Matlab with this particular control scheme.

(Refer Slide Time: 01:49)



So, let us start with the very basic. So, what we did in the earlier case the same way we will see the motion control will start with a given which is what you call desired, available all those things we will see.

So, we will see like what are the desired here. The desired would be here what you call given desired a trajectory which would be consist of η_d, $\dot{\eta}_d$, $\ddot{\eta}_d$. So, these all would be given to us. So, in the sense desired position desired acceleration would be given.

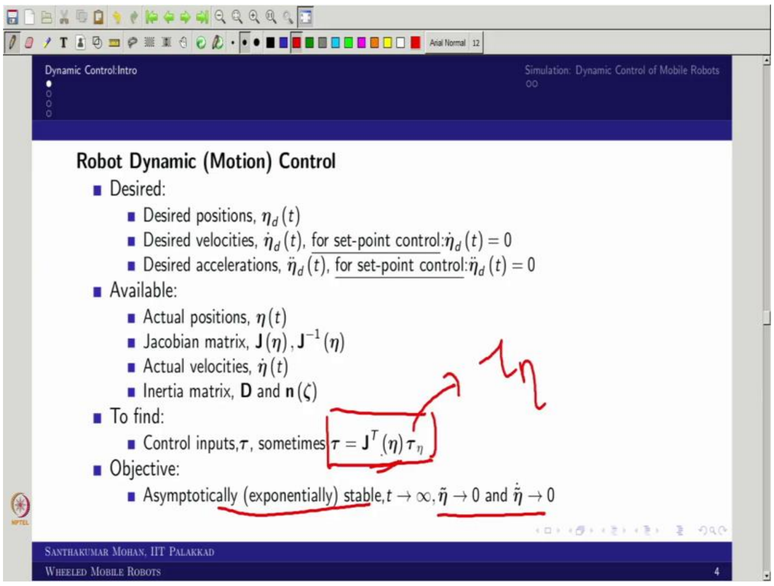
But if it is actually like set point control but you want to do it in dynamic then these two would be 0 that you need to noted down. Then what would be available? In kinematic control, we are assumed that only positions are available. But now it is a second order system, we assume that both position and as well as velocities are available. In the sense actual positions are available and as well as actual velocities are available.

Further we are trying to map between body frame to what you call inertial frame. So, then we assume that the Jacobian matrix is clearly known to us. So, this matrix is available. Further if we are doing in a probably model based control, we already see what is model base and motion base. If it is actually like model base, then the model also available in the sense you would be knowing the inertia matrix accurately and as well as other effects also accurately.

So, these all actually like available. So, then what would be the objective or to find? So, we are trying to find out what would be τ , so that is what you call control input. So, this control input you want to find with the objective of what you call, so $t \rightarrow \propto$, so my error tends to 0, right.

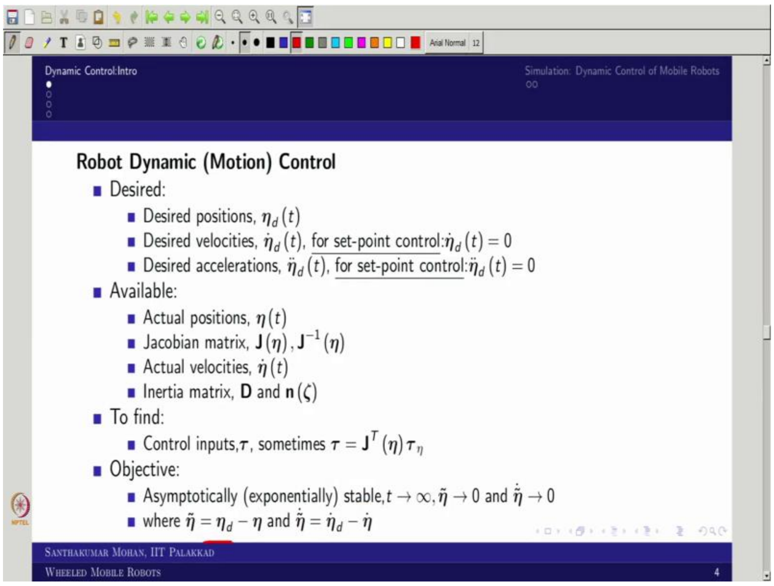
So, this is what we have done for what you call kinematic control but right now it is actually like dynamic control. So, your velocity error also supposed to be 0. So, this is what your objective. In order to fulfill this, we will actually like take it that as subjective ok.

(Refer Slide Time: 03:44)



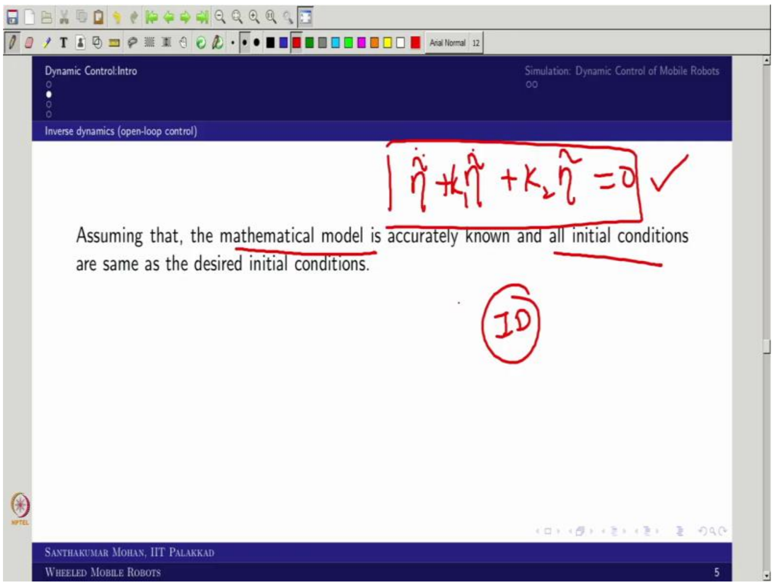
So, now what we can actually like see that the asymptotically stable is what we wanted. So, this is what we are looking at. So, in this case, sometime we may use this. Why? We may actually like try to write this τ_η straight away and then we will apply this $J^T(\tau)$ you can say η as τ . In that sense, we will do everything in one you can say coordinate and then we will bring it to the body as it is. So, this is also like can be done.

(Refer Slide Time: 04:14)



So, in the sense what finally we are trying to see? So, this error supposed to be 0. So, this is what we are interested in dynamic motion control.

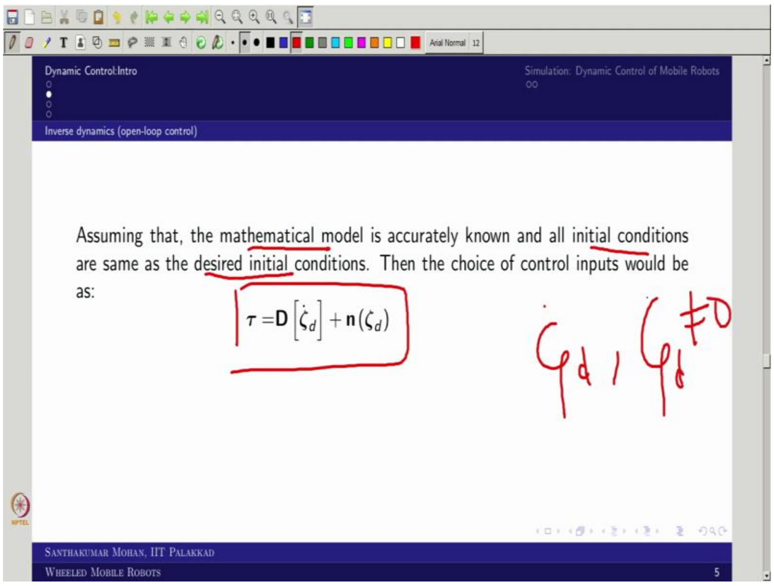
(Refer Slide Time: 04:21)



So, now, if you want to do this what one particular aspect? So, we are looking at this. So, this second order error dynamics what you call, so this I if I actually like make it as 0, so this is the second order error dynamics. If I achieve this, I am already done the required task. So, for that what I required? I required that the mathematic model is accurately known to us, and all the initial conditions are same as you call the desired initial condition. Then what we can do? We can do the inverse dynamics right.

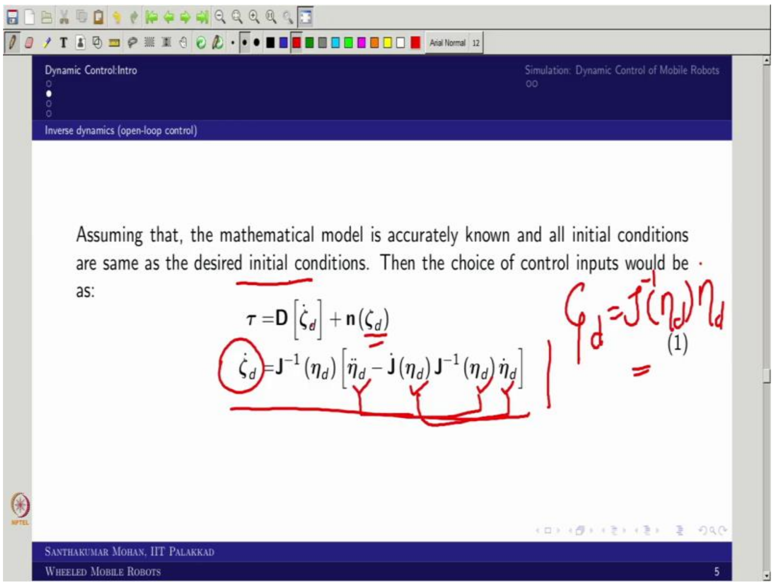
So, that is what initially we start attempt. In fact, in one of the dynamic simulation you can say lecture, I was showing that how this inverse dynamics can be incorporated.

(Refer Slide Time: 05:08)



So, this is what you call inverse dynamics. But the inverse dynamics would be applicable where the mathematical model is accurately known and then the desired and actual initial conditions are same. Further you call this η desired dot and η desired are non-zero then only this would be workable right.

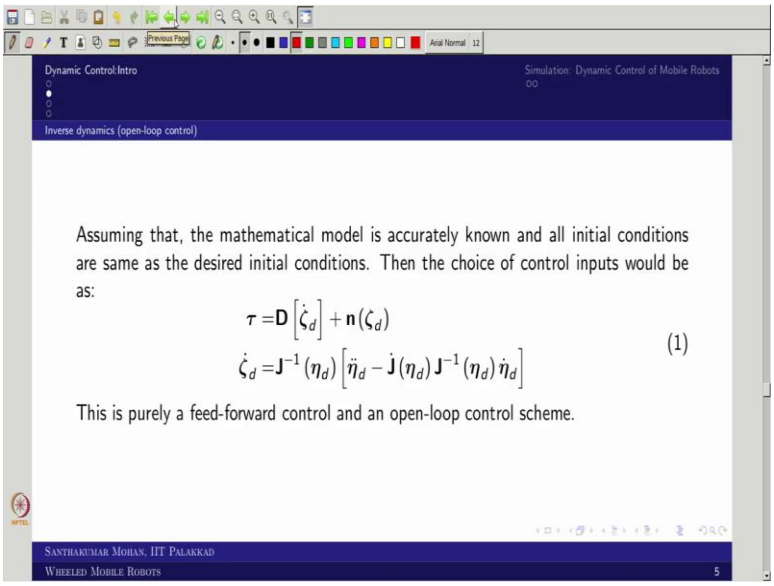
(Refer Slide Time: 05:29)



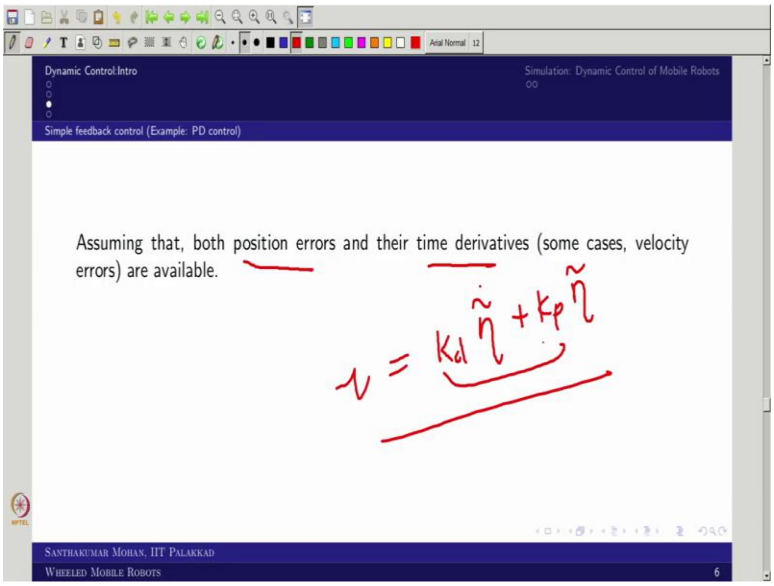
So, now, we will actually like see how that can be incorporated this is what I was saying. So, now, you can actually like bring this η desired dot in terms of what you know ok, so that you can actually bring it, and then you can see. So, these are the what you call desired you call trajectory information right, so that you can incorporate and do it this. Once you know this and then you can do it.

So, in the sense this also you need to write. So, these are actually like available to us and then you are actually like using it. So, these all actually like what you have seen as a inverse dynamics as a open loop which we simply call for feed forward control, but feed forward control will not work for all the cases. So, this condition is not all the time true.

(Refer Slide Time: 06:18)



(Refer Slide Time: 06:23)

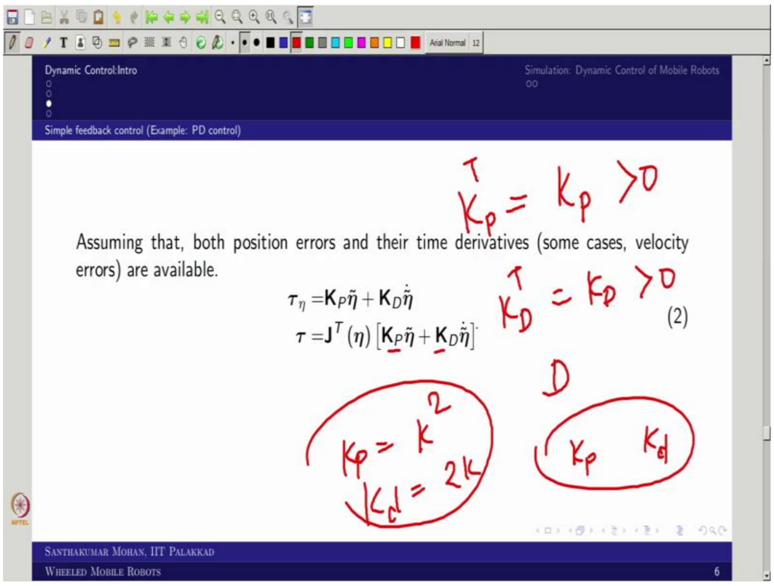


So, then what we are actually like interested? We are actually like interested to see what is the feedback control. So, one of the easiest option, so already in the kinematic or you can say the introduction lecture itself I told, so the first choice of control in robotics is PD control.

Why? The open loop control or open loop system is actually like unstable. But if you incorporate PD control, the close loop control is straightforward as a stable system. So, in the sense the first choice of control is PD, so that is what we are trying to see. In that sense what we are seeing that the position errors and its time derivatives are available to us.

So, if these are available what I can write? So, $k_d \times \dot{\tilde{\eta}} + k_p \times \tilde{\eta}$ d, I can assume as my Ʈ and then I can you can say incorporate. This will work, but only thing this would be actually like based on the choice of k_d and k_p. So, in the sense the k_d, k_p you have to tune. So, most of the cases this tuning is as the bigger issue, so that is why we are moving the other control ok.

(Refer Slide Time: 07:23)



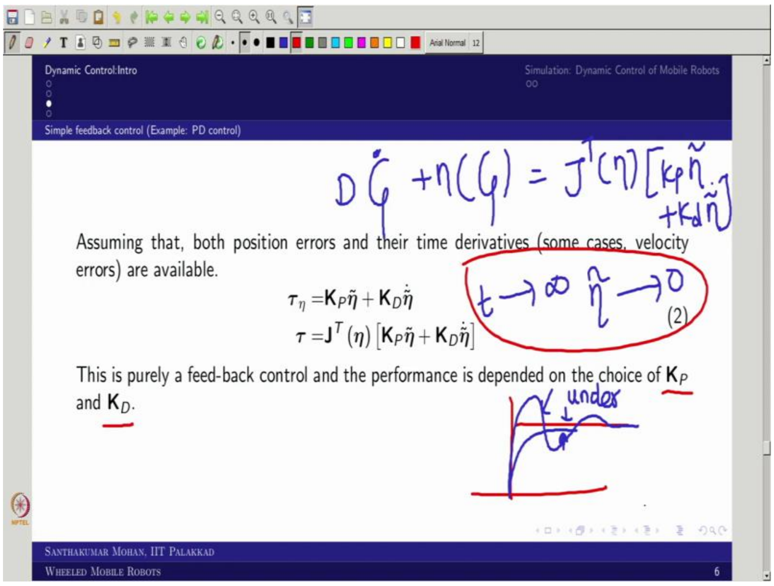
So, how that is actually like coming? This is what the control aspect. So, this is the proportional control and this is the derivative control. And we are actually like assuming that this is not velocity error. We are taking time derivative of position error we have incorporated. So, in that sense what we have to see? So, how this k_d and k_p can be chosen?

So, since the D is actually like one matrix. So, you can actually like take k_d and k_p in such a way that these are actually like somewhat related with one simple parameter called k. This $k_p = k^2$, and $k_d = 2k$. So, then this will give a simplest model what you call critically damped system, but you have to be very clear that these are matrices. So, then what you supposed to know?

So, this k_p matrix supposed to be positive definite and as well as for simplicity we what we wanted actually like symmetric matrix also. In the sense $k_p^T = k_p$, and this is positive. The similar sense you can actually like take it this also ok. The k_d matrix also like supposed to be positive and as well as symmetric matrix.

So, even most of the cases what we usually take a diagonal matrix, which is very simple. So, diagonal matrix with all positive consent that will also work.

(Refer Slide Time: 08:44)

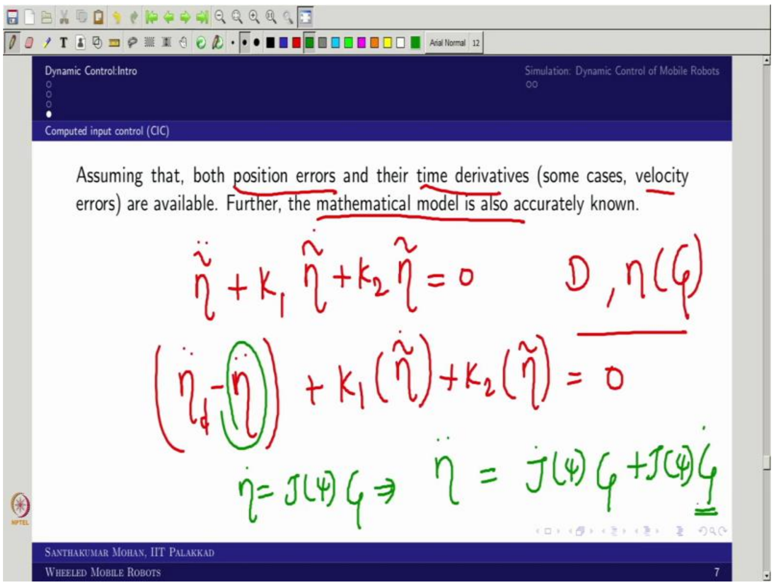


So, now we will actually like move forward. So, what it says? It is purely feedback control, but this performance is purely depend on this choice of k_d and k_p. But this is not the you can say right perspective because if you look at the PD control performance this would be having a you call steady state error. So, this steady state error would be execute, sometime actually like it would be having what you call a overshoot in the sense it would be a underdamped performance.

So, these are actually like suppose to be avoid and as well as we will try to make it the complete error dynamics as a second order error dynamics. In this case, the error dynamics is not that way you see this is what, so you have equation. So, this is what you are actually like substituting as this right.

So, this equation is not giving any you call guarantee that the η tilde tends to 0 when t $\rightarrow \propto$. So, this is not guaranteed right. So, if this is not guaranteed, then what is the point in using the PD control, that is what the whole idea that is why we are bringing the new control called computed torque control in general, here I am putting computed input control.

(Refer Slide Time: 09:53)



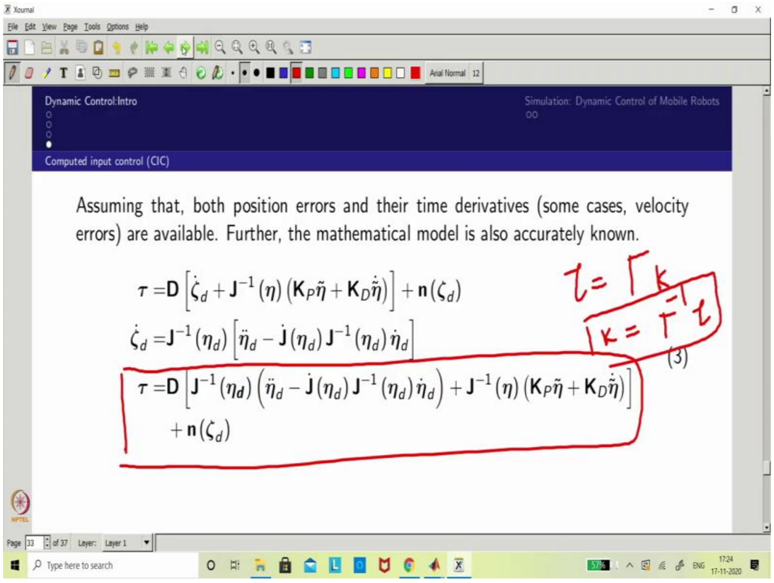
What that means? So, I am assuming that the both position error and its time derivative or sometime here we are going to call velocity error are available. Further, what I am saying that the mathematical model is accurately known. What that mean? The D matrix and n vector is accurately known to us. So, if this is known, so what we are actually like rewriting, this. So, this is what our interest I am assuming that this $k_1\dot{\tilde{\eta}}$ and $k_1\tilde{\eta}$ as this.

So, now, I am rewriting this. So, what this I can write as? So, the error of you can say the position, acceleration, and you call the velocity. So, I can rewrite this. So, I am keeping this as it is. So, I am just putting a bracket just for your understanding. This also like $\dot{\eta}_d - \dot{\eta}$ this is $\dot{\eta}_d - \eta$, but I am not doing it.

Why I am actually like rewriting this way? You can look at this, what is this? This I can rewrite in the other form. So, what is this? This I can write in the form of what you call D matrix. So, for that what we know? The $\dot{\eta}$ I can write as J(Ψ) × η. So, from there what

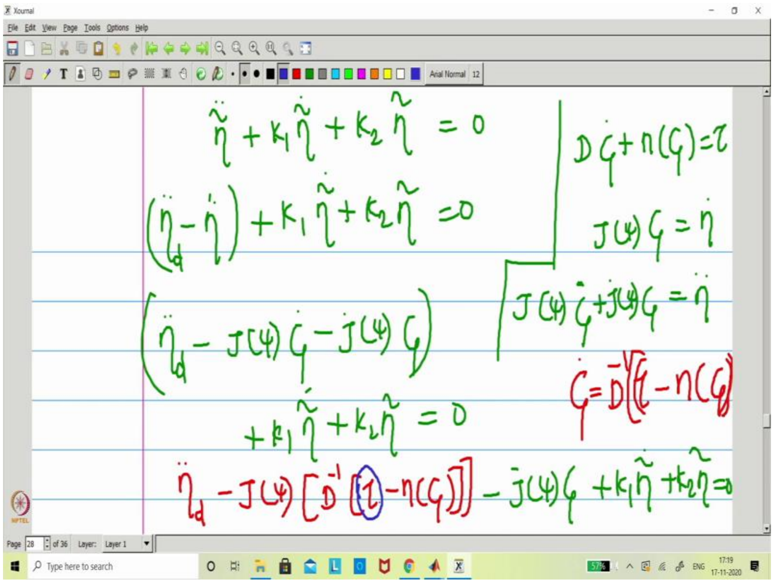
I can do? So, I can do it this right. So, I can do it. So, but what I know? This is actually like having a control input.

(Refer Slide Time: 11:39)



So, that is what I am trying to do. You can see, so this is what I know ok. So, here what I am trying to do? I am trying to do this equation. So, for that, what we are trying to bring? So, probably I will actually like recall that itself. So, this particular slide itself and then I will actually like incorporate one ok.

(Refer Slide Time: 12:03)



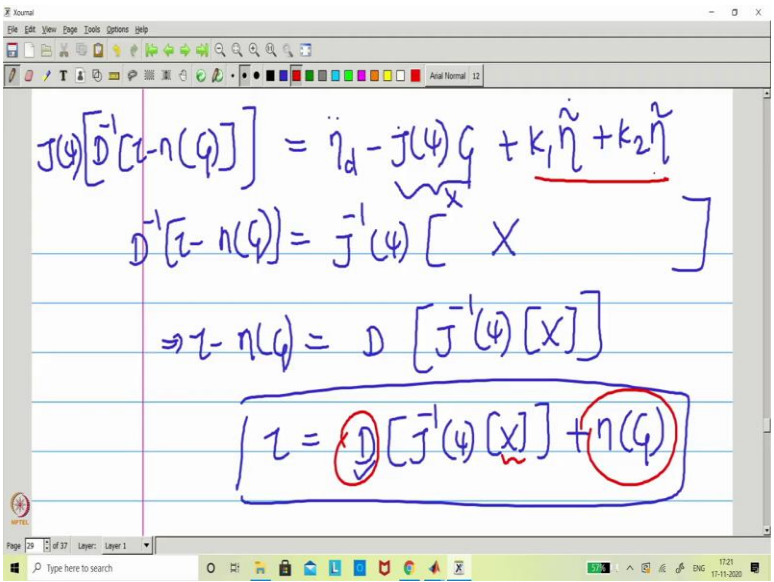
So, what we are actually like interested; interested this right. So, I will rewrite that whatever I have written. So, further what we know? So, this we know. Further we know like $J(\Psi) \times \eta$ is ok. So, then I differentiate this. So, I will get $J(\Psi) \times \dot{\eta} + \dot{J}(\Psi)\,\eta$ of right. So, these are available, I am trying to recall this. So, what I did? So, I did so this. So, I am just rewrite rewriting this as it is. So, then what I can resubstitute this as this form? So, I will rewrite that ok.

So, this is actually like replacing this plus $k_1\dot{\tilde{\eta}} + k_2\dot{\tilde{\eta}} = 0$. So, here what you can naturally again know? From this equation I rewrite that. So, I am writing that. So, this equation I rewrite. The $\dot{\eta}$ sorry $\dot{\xi}$ I can write as $\tau - n(\xi)$ this whole multiply with you can say D^{-1}. So, this equation is available to us. So, I am re substituting that.

So, in the sense, so what that? That is $D^{-1}(\tau - n(\xi))$ right. So, this is the first equation. The remaining I am keeping it as it is. So, what that means? So, I am keeping it as it is. So, I am just putting $\dot{J}(\Psi) \times \xi + k_1$. So, now, what one can actually like noted down?

So, here you can noted down this is what my control input. I can choose this τ based on rewriting this whole equation. So, what I can rewrite this whole equation? I can keep it this.

(Refer Slide Time: 14:37)



So, I can actually like bring that. So, what I can rewrite that equation from this? So, I am actually like rewriting that. So, everything else is the other side. I am keeping it only this term. So, in the sense, what I am doing it? So, the $D^{-1}(\tau - n(\xi))$ I am keeping it. So, the remaining all I am actually like bringing it to other side. So, in the sense what you will get? So, this I am actually like multiplying and putting it.

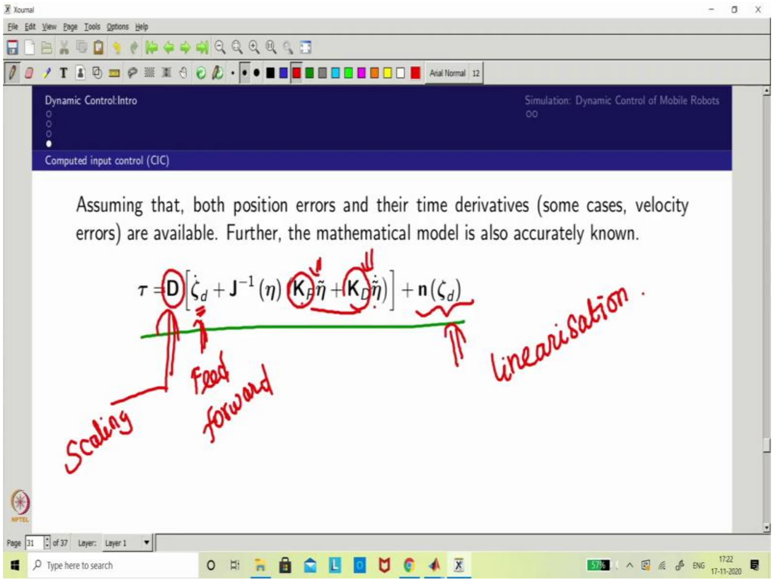
So, even I will actually like put this, what you call j inverse also. So, you can just take it. So, this also like exist. So, now, this is actually like remain as it is, but there is a negative sign. So, I assume that this bring the other side, the remaining all rηin in one another side. So, in the sense $\ddot{\eta}_d - \dot{J}(\Psi)$, then η sorry ξ then what else? So, these all remaining.

So, k_1, so then plus $k_1\dot{\tilde{\eta}} + k_2\dot{\tilde{\eta}}$ 1, I am actually like again rewriting this. So, this whole equation, so, I am actually like putting $J^{-1}(\Psi)$. So, then what remain? So, $D^{-1} \times \tau$ - n(ξ) right. So, I am rewriting, this I call this as capital X. So, this capital X remain.

So, now, what it remains? So, I have take it that. So, D× $[J^{-1}(\Psi)$[X]]. So, then this is actually like τ - n(ξ). So, I am writing τ as. So, D× $[J^{-1}(\Psi)$[X]]. So, then what you have? This; so, now, this is what you call, you have computed the torque based on what you have your model. So, this is actually like your model.

These two are come from model. And this come from your feedback where you can see that the error dynamics becomes second order error dynamics become 0, in the sense both are actually like going to be converged.

(Refer Slide Time: 17:03)

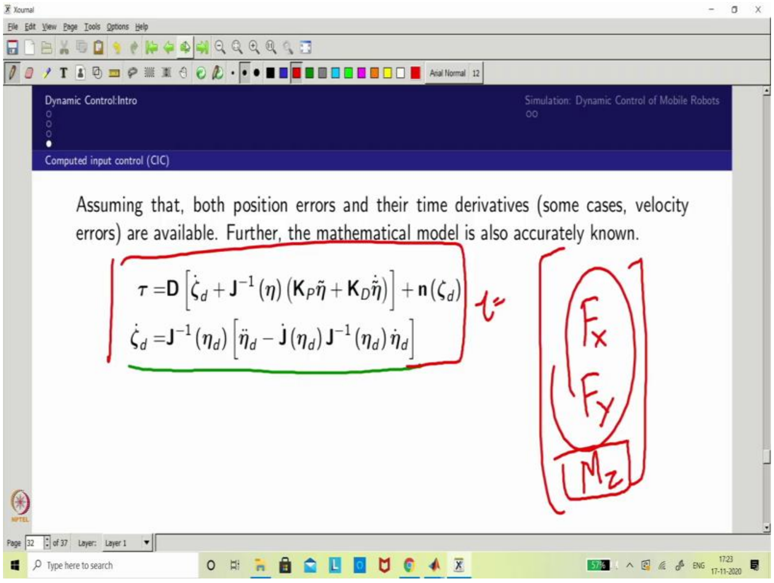


So, that is what we are actually like trying to see. So, that is what we have substituted here. You can see I rewrite that in general form. This is $\dot{\xi}$ and this is actually like your you can say positional information, and velocity information based on error, this is the position error and this is the velocity error and where there would be two gains.

I am assuming that this is a PD control where this is a proportion and derivative control. And here you can see that this is actually like coming from the desired acceleration which is nothing but a feed forward. So, you have actually like got the feed forward term, and this is actually like you are compensating.

So, here actually like you call, so in the sense what you call linearizing your model. So, this is so linearization. So, this is linearization. And this is what you call you can say scaling. So, scaling yourself ok. So, now, what it is giving? So, you are getting the τ directly.

(Refer Slide Time: 18:02)



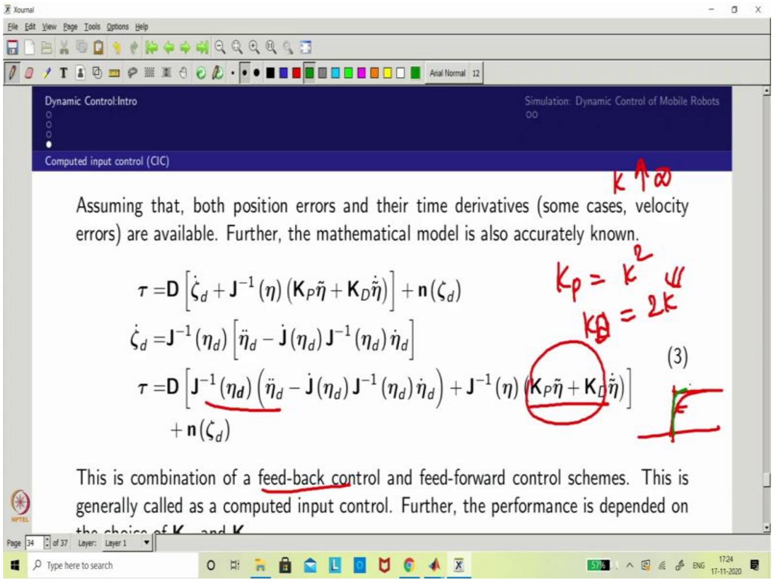
So, this is what we call computed torque control in general. Right now we say that F_x, F_y is not torque right. So, you may feel that the τ vector would be consist of the three quantity as $\begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix}$ where only one is actually like torque, so that is why I modified as computed torque control as computed input control. So, now, you can see that this is the control activity.

So, now, you can actually like take this forward, and you can actually like substitute in the simulation model and then do it. So, now I expand this further. So, what you get? So, you get the overall equation as like this. So, this is what your τ equation. You replace what you have done earlier. So, what we have done earlier. So, $\tau = \gamma \times \kappa$ right.

So, now, you replace that into this and you actually like run the simulation you will get it. So, now, this is one state. The second state once you obtain this τ then you bring it back this $\tau = \gamma \times \kappa$. So, now you wanted to find what is the individual wheel forces. So, you take $\gamma^{-1} \times \tau$.

So, this you can do it. So, this is what you wanted to do it in Matlab simulation. I expect you will do yourself because we have done lots of simulations, and we have actually like modified everything. So, that is what I am expecting you to do it.

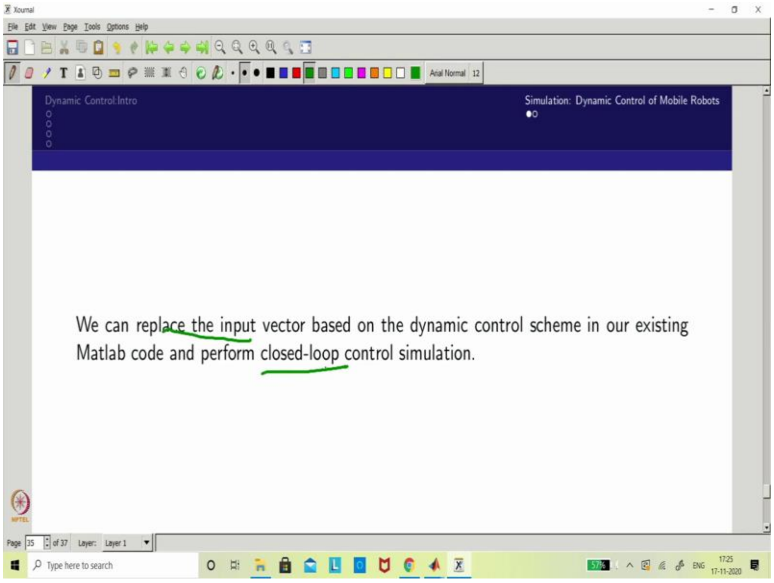
(Refer Slide Time: 19:20)



So, this is a combination of what we are doing. So, you are taking a feedback and as well as feed forward, so that is why it is called feedback and feed forward combined. So, it is actually like computing the input control. Again here the performance is based on the choice of k_d and k_p. Here you can straight away you can take $k_p = k^2$, and k_d as actual like k_d = 2 k, you can straight away take.

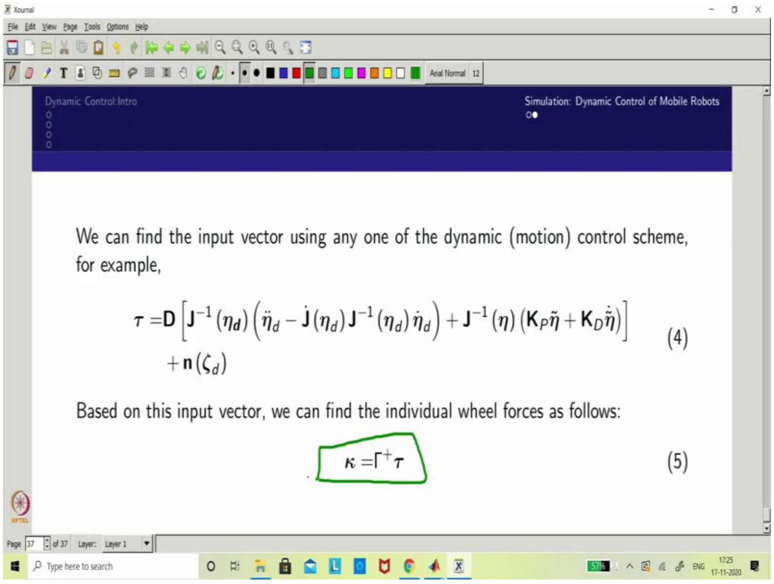
In that sense what it looks like? The performance is actually like critically damped. Based on the k value the raise time would be getting decrease or increase? If you increase the $k \rightarrow \propto$ this right raise time would be shorter in the sense it would become faster. So, it may be faster like this ok. So, now, you can try to do the simulation yourself and get understand.

(Refer Slide Time: 20:09)



So, that is what I am actually like asking you to do. We can replace the input vector based on the dynamic control scheme with the existing Matlab code. I hope you have already the Matlab code and perform the closed loop simulation.

(Refer Slide Time: 20:29)



So, once you do and comfortable in this k_d and k_p choice and you have done without any restriction then you do the further end what that? So, you substitute this and see whether your kappa is actually like how much it is coming. So, based on that, you can actually choose your control input or you can say your wheel motor and all you can choose.

Your wheel motor is not sufficient then you can even try to see what you can actually like make the frictional component. Right now you are taking a plastic wheel, you can take the plastic wheel with one rubber strip.

So, your rubber strip is not sufficient. Then you take it a complete rubber strip as a stick. So, if that is also not done, then you can actually like make a probably a you can say knurling on that in the sense you make a grating. So, then you will get a friction more, so that kind of thing you can do it.

So, that is what I was actually like expecting in this dynamic control. So, with that this dynamic control part is over. So, now, we will move the next lecture with combination of kinematic and dynamic that is what we call cascaded control, some people call it is a back stepping. How it is back stepping? That we will see in the next lecture. Until then see you, bye, take care.