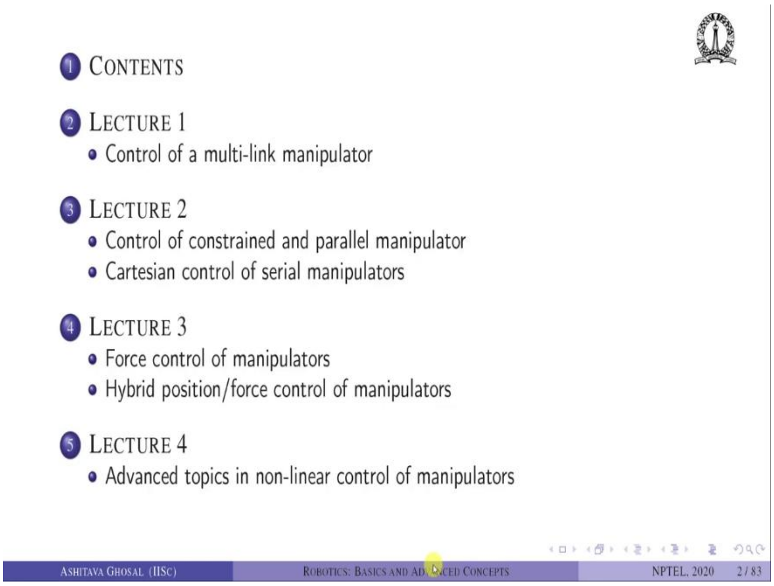
Robotics: Basics and Selected Advanced Concepts
Prof. Ashitava Ghosal
Department of Mechanical Engineering
Indian Institute of Science, Bengaluru

Lecture - 34
Control of a multi-link manipulators

Welcome to this NPTEL lectures on Robotics Basics and Advanced Concepts. In this week we will look at Non-Linear Control of Robots ok.

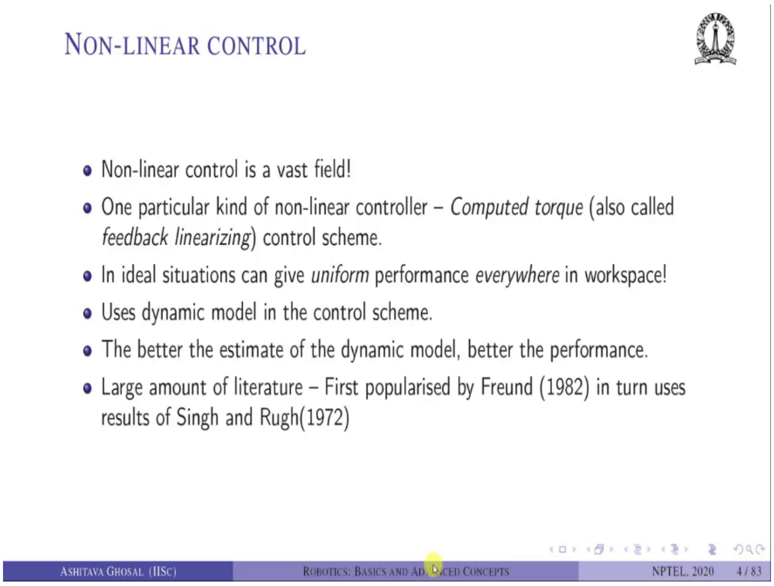
(Refer Slide Time: 00:32)



It will consist of 4 lectures. First, we will look at control of a multi-link manipulator in the second lecture we look at control of constrained and parallel manipulators and also Cartesian control of serial manipulators. In the third lecture we will look at force control of manipulators and also hybrid position and force control of manipulators.

And we will end this week with some advanced topics in non-linear control of manipulators ok.

(Refer Slide Time: 01:08)



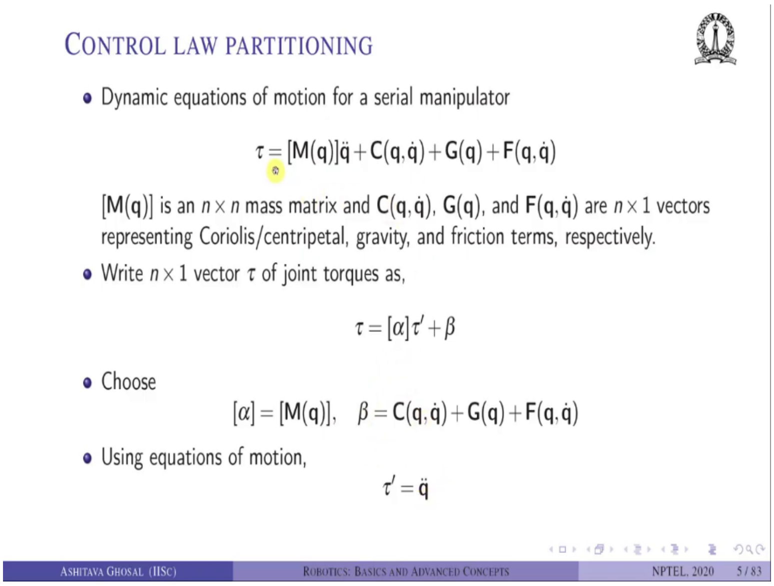
So, let us continue control of a multi-link manipulator. So, the non-linear control is a very vast field, ok we will look at only one particular kind of non-linear controller called the computed torque also called as feedback linearizing control scheme ok.

So, in ideal situations, I will show you that this kind of computed torque or feedback linearizing control scheme can give uniform performance everywhere in the workspace; recall that a non-linear robot, if it is controlled using a linear controller cannot give same performance everywhere in the workspace; however, this computed torque scheme can give uniform performance everywhere in the workspace.

So, this computed torque or feedback linearizing controller uses the dynamic model in the control scheme. So, what do we mean by dynamic model? We need to know the equations of motion of the robots. So, we need to know the terms in the equation of motion, basically mass matrix, Coriolis centripetal term, gravity term and so on.

And I will show you the better the estimate of the dynamic model better will be the performance; there is a large amount of literature on this computed torque control scheme it was first results were in 1972, but it was really popularized by Freund in 1982 ok.

(Refer Slide Time: 02:43)



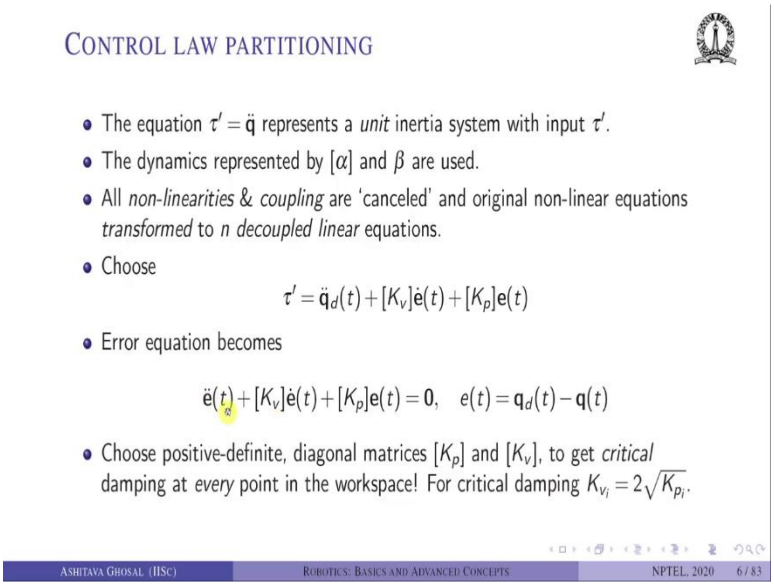
So, let us start so, as I said in the control law using computed torque scheme, it consists of two parts so, that is why it is also sometimes called control law partitioning. And we also need the equations of motion of the serial manipulator. So, the equation of motion of a serial manipulator can be written as, $\tau = [M(q)]\ddot{q} + [C(q,\dot{q})]\dot{q} + G(q) + F(q,\dot{q})$ and if we know some or if we have some idea of the friction, we can add the friction term also ok.

So, the scheme starts as follows. So, we can write this $n \times 1$ vector of joint torques as some $[\alpha]\tau' + \beta$. So, this is like multiplication by some $[\alpha]$, ok plus some bias terms so, scaling and some bias term. If we choose $[\alpha] = [M(q)]$ which is the mass matrix, β as the Coriolis centripetal term plus the gravity term plus the friction term.

Then, if we substitute $[\alpha]$ and β in this equation, we will get $[\alpha]\tau' + \beta = [M(q)]\ddot{q} + [C(q,\dot{q})]\dot{q} + G(q) + F(q,\dot{q})$. So, everything will cancel out, this β, $[C(q,\dot{q})]\dot{q} + G(q) + F(q,\dot{q})$ will cancel out on this side and then, mass matrix is invertible; so, we can remove the mass matrix from both sides and hence we will be left with one equation, which is $\tau' = \ddot{q}$.

So, this is the starting point of this computed torque control scheme or using what is called as control law partitioning ok. And as you can see, what have we done, we have a non-linear system of equations for the dynamics of the robot, we have chosen to implement the torque, which is going into the robot as some $[\alpha]\tau' + \beta$; where $[\alpha]$ is chosen as a mass matrix, β is Coriolis plus gravity plus friction ok. And we are left with $\tau' = \ddot{q}$.

(Refer Slide Time: 05:00)



So, the equation $\tau' = \ddot{q}$ represents a unit inertia system with input τ'. So, see there is no mass matrix Coriolis gravity term. So, it is like as; if $\tau' = I\alpha$, α is the angular acceleration, where I is identity ok. And what have we done? We have used the dynamics represented by $[\alpha]$ and β so, we are chosen $[\alpha]$ as mass matrix, $\beta = [C(q, \dot{q})] + G(q) + F(q, \dot{q})$.

And effectively what we have done is all non-linearities and coupling are cancelled, and the original non-linear equations transformed into n decoupled linear equations. So, we have n decoupled linear equation, $\tau' = \ddot{q}$. Next we can choose $\tau' = \ddot{q}_d(t) + [K_v]\dot{e}(t) + [K_p]e(t)$. So, this is nothing but the PD control; with this acceleration feed forward term added ok.

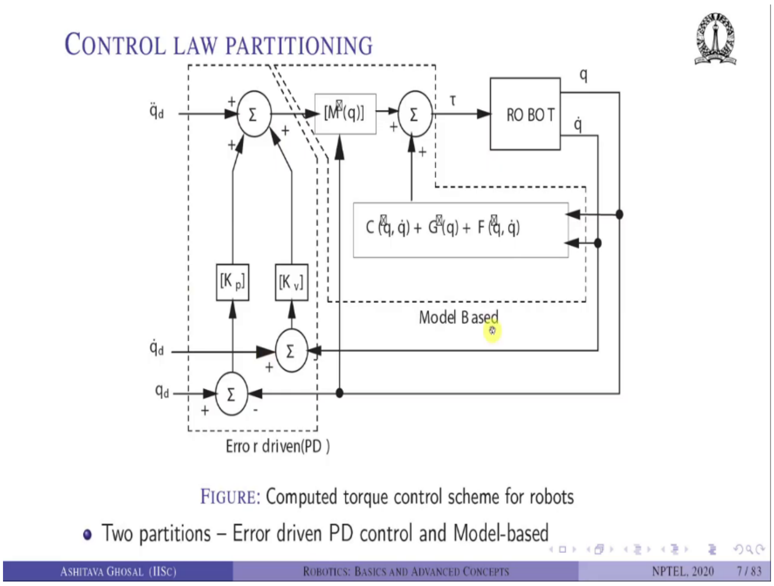
So, if you now, substitute τ' is this, into this $\tau' = \ddot{q}$ and then, take $\ddot{q}$ to the other side, we will be left with the error equation, which is $\ddot{e}(t) + [K_v]\dot{e}(t) + [K_p]e(t) = 0$. So, how many such equations are there? There are n such equations ok and $e(t)$ is nothing but the desired joint, as a function of time $q_d(t)$ minus the measured joint motion as a function of time.

And, if we now choose $[K_p]$ and $[K_v]$, as positive definite diagonal matrices ok. So, then all these e_1, e_2, all the way till e_n they are decoupled ok. And we can choose $[K_p]$ and

$[K_v]$ to get critical damping at every point in the workspace. So, remember, in the case of linear control I had shown you for a single link, that $K_v = 2\sqrt{K_p}$; then, we get critical damping.

So, we can choose critical damping or over damping, whatever we want, to get the desired $e(t)$ as a function of time.

(Refer Slide Time: 07:30)

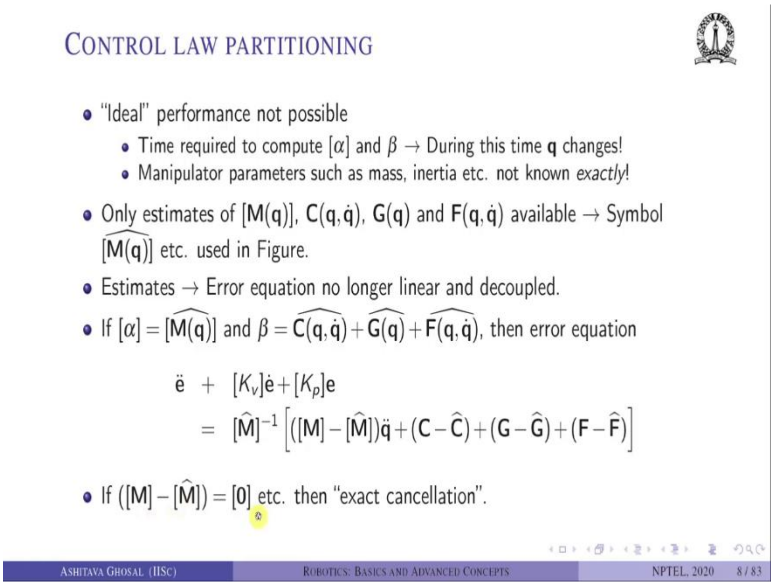


So, in block diagram, what we have is a robot, the input is torque; we measured q and $\dot{q}$ and we take this q and $\dot{q}$ and compute $\widehat{C(q,\dot{q})} + \widehat{G(q)} + \widehat{F(q,\dot{q})}$. So, this $\frown$ here means; that often we do not know exactly, $\widehat{C(q,\dot{q})}$, $\widehat{G(q)}$ and $\widehat{F(q,\dot{q})}$. So, these are estimates of the Coriolis term, gravity term and friction term and the estimates of the mass matrix. We can also take this q and compute $[\widehat{M(q)}]$; mass matrix and q.

So, what you can see is this dotted line here, ok this portion requires the use of the dynamic equations of motion; we need to know what are the model based terms. We can also take this q and $\dot{q}$ so, if you do $q_d - q$ and multiplied by $[K_p]$ ok and then, $\dot{q}_d - \dot{q}$ multiplied by $[K_v]$ and add to $\ddot{q}_d$; at this place we have τ'.

So, basically, what we have is $\tau'[\alpha] + \beta = \tau$. So, the box which is dotted here is error driven, ok it is purely driven by $[K_p]$ and $[K_v]$ so, it is similar to the modified PD control scheme ok. So, there are two partitions; one is which is PD control and one which is model based.

(Refer Slide Time: 09:09)



So, the ideal performance is not possible simply, because we do not ever know the mass matrix, the Coriolis term and the gravity term and friction term; specially, the friction term very well ok. There are also one other property, which is little bit more complicated to understand; we require some time to compute this $[\alpha]$ and β.

So, as you can see this is $[\alpha]$ and this is β so, we need to measure q, $\dot{q}$ and then, use it to compute this mass matrix and $\widehat{C(q,\dot{q})} + \widehat{G(q)} + \widehat{F(q,\dot{q})}$, but it takes some time. So, the torque which you are getting is based on a previous measurement, maybe little bit defined ok. So, hence it will never be exactly cancelling ok.

So, as I said only the estimates of mass matrix, Coriolis centripetal term, gravity and friction term are available; that is why, we have used this $\widehat{\ }$ ok and in the figure the box with the crosses, what is shown as $\widehat{\ }$ ok. So, the estimates, because of the estimates the error equations are no longer linear and decoupled, ok that we can show.

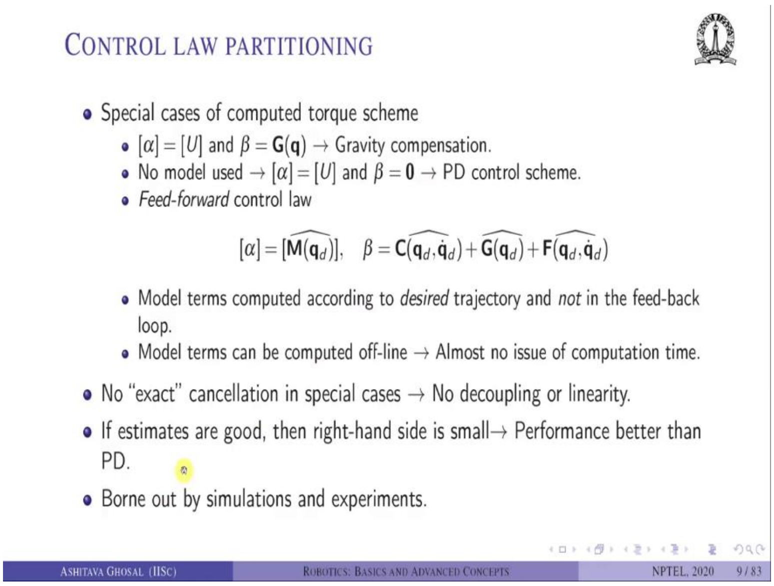
So, for example, if the $[\alpha] = [\widehat{M(q)}]$ and $\beta = \widehat{C(q,\dot{q})} + \widehat{G(q)} + \widehat{F(q,\dot{q})}$ then, you can work out the error equation, which is $\ddot{e} + [K_v]\dot{e} + [K_p]e$, will be now, there will be a right hand side; say it will be $[\widehat{M}]^{-1}\left[\left([M] - [\widehat{M}]\right)\ddot{q} + \left(C - \hat{C}\right) + \left(G - \hat{G}\right) + \left(F - \hat{F}\right)\right]$.

So, $[M]$, C, G and F are the actual dynamics ok whereas, $[\widehat{M}]$, $\hat{C}$, $\hat{G}$ and $\hat{F}$ are the estimated dynamics, that is what you think ok and it is entirely possible; you think the mass is let us say 5 kg, but then somebody has did it a hole, somewhere and the mass has decreased..

So, the estimated mass matrix will be different than the real mass matrix. And as you can see, if $([M] - [\widehat{M}])$, $(C - \hat{C})$, $(G - \hat{G})$, $(F - \hat{F})$ all these things are exact 0 which means what, I know exactly, what is the mass Coriolis gravity and friction terms ok. Then we have again exact cancellation, and we are left with a second order linear ODE in the $e(t)$.

So, what else can we say? If my estimates are good ok; meaning what, this $([M] - [\widehat{M}])$, $(C - \hat{C})$ and so on, are small then what we have is the linear second order ODE subject to a forcing term on the right hand side. So, if the forcing term is small then, the eventual performance of the system; eventual performance of $e(t)$ will be also small ok. So, $e(t)$ will be small, if the estimates are good.

(Refer Slide Time: 12:18)



So, there are some special cases, in control law partitioning; one is if $[\alpha] = [U]$ or $[\alpha]$ is the identity matrix. So, we are not going to assume any mass matrix, we think that it is identity matrix. And $\beta = G(q)$. So, this is called gravity compensation ok. So, if $[\alpha] = [U]$ and $\beta = 0$ then, we can get to PD control scheme. So, $\beta = G(q)$ means; remember the right hand side, there is a $G(q)$ and then, you have $[\alpha]\tau' + \beta$ if $\beta = G(q)$, the gravity terms cancel out from both sides and the system will behave as if there is no gravity.

Likewise, if $[\alpha] = [U]$ and $\beta = 0$ you can show that, you will get back the PD control scheme very straightforward. There is also a very well known control scheme, which is based on the following observation. That we can compute $[\alpha] = [\widehat{M(q_d)}]$, where q_d is the desired trajectory ok. So, $\beta = C(\widehat{q_d, \dot{q}_d}) + \widehat{G(q_d)} + F(\widehat{q_d, \dot{q}_d})$. So, the model terms, $[\alpha]$ and β are computed according to a desired trajectory and they are not in the feedback loop ok.

So, if you know the desired trajectory, we can compute or precompute this mass matrix Coriolis term, gravity term and friction term ok. And hence, there is no issue of computation time ok, it will be just reading from a file ok or reading from a lookup table.

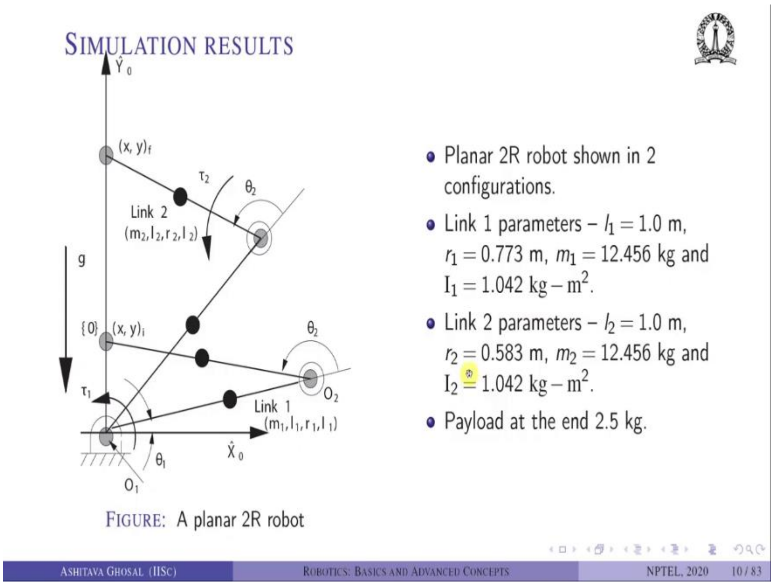
However, there is no exact cancellation in any of these special cases ok, but what is the hope? That q_d and q will be very close to each other; hence the estimates will be very good ok. So, $\left([M] - [\widehat{M}]\right)$ will be very small, $(C - \hat{C})$ will be very small ok.

But, theoretically there is no exact cancellation in any of these special cases, there is no decoupling and there is no linearity ok, but as I said, if the estimates are good then, the right hand side is small and what you can see is; that performance will always be better if you do not use any model.

So, if you do not use any model, which means a PD control scheme then, $[\alpha] = [U]$ and $\beta = 0$. Then, if you use some model; you have $\left([M] - [\widehat{M}]\right)$ so, you have some smaller number ok.

So, performance is always better with some model than with no model, and no model is PD control scheme. So, this can be seen to be borne out by simulation and experiments, which we will show.

(Refer Slide Time: 15:20)

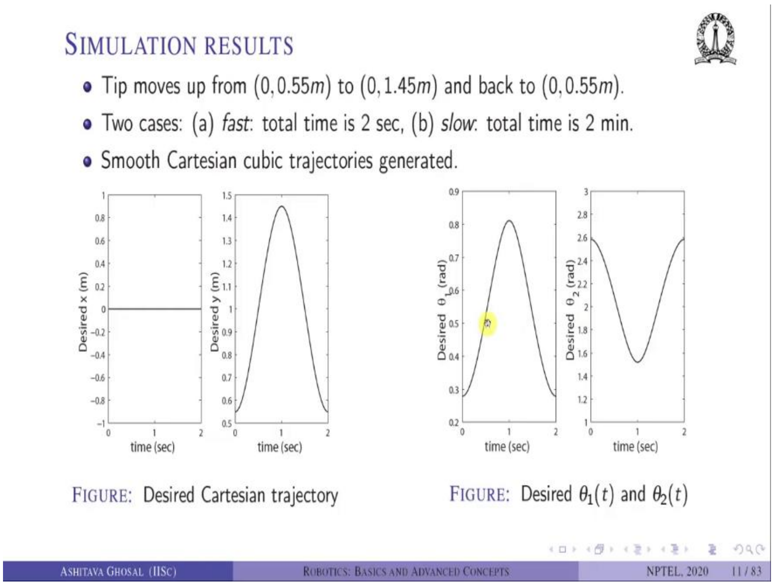


So, let us start with some simulation results. So, we have a planar 2R robot, shown in these two configurations. So, initially it is like this, at some (x_i, y_i) and it goes to (x_f, y_f).

Again just like in the previous week, we have this 2R with link parameters; m_1, l_1, r_1 and I_1 and m_2, l_2, r_2 and I_2 there is the torque acting in first joint, there is a torque acting at the second joint so, we go from (x_i, y_i) to (x_f, y_f) in some time and come back, exactly the same numerical experiments, that we did in the previous week. And we are going to use the same parameter values, for the link 1 parameters and link 2 parameters, and again there is a pay load of 2.5 kg at the end ok.

So, this link length is 1 meter, the C G is located at 0.773, the mass is 12.456 kg and the I_{zz} inertia of this link 1 is 1.042 and so on ok.

(Refer Slide Time: 16:32)

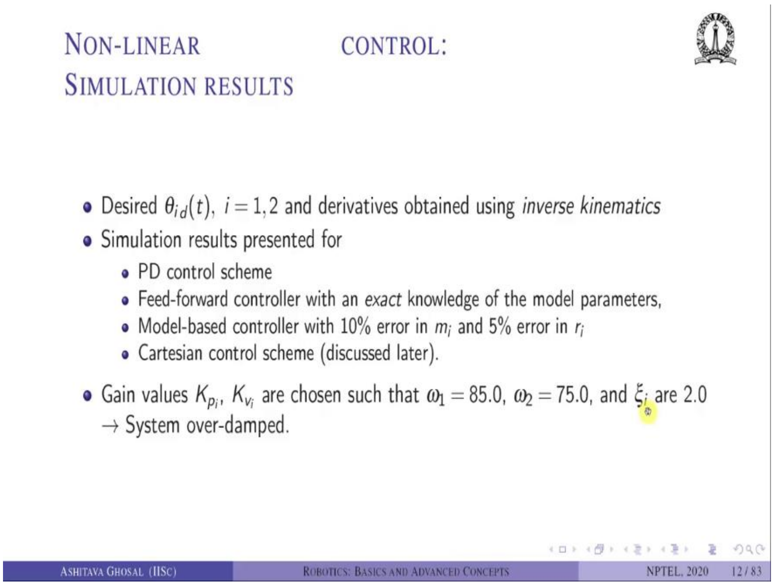


So, as I said, the tip moves from 0.55 meters on the Y axis to 1.45 meters on the Y axis and back to 0.55 meters. So, it is going exactly, against gravity and it is coming down with gravity ok.

So, as we had discussed earlier, there are two cases; one is fast motion, which is total time is 2 seconds, another one is slow, which is total time is 2 minutes. So, the basic idea is, we will use the same results, which we saw for linear control using PD controllers and then we will see what happens, when we use a non-linear control law ok, based on computed torque ok

So, the same set of parameters, same trajectory everything is same. So, we have a smooth Cartesian cubic trajectory so, basically the desired is x should be always 0, y should go from 0.55 to 1.45 and back these are 2 cubic curves, which are patched together; from inverse kinematics, we can find out θ_1 desired and θ_2 desired ok. These also will be smooth curves ok $\mathcal{C}^2$ continuity; $\mathcal{C}^2$ continuous curves.

(Refer Slide Time: 17:55)



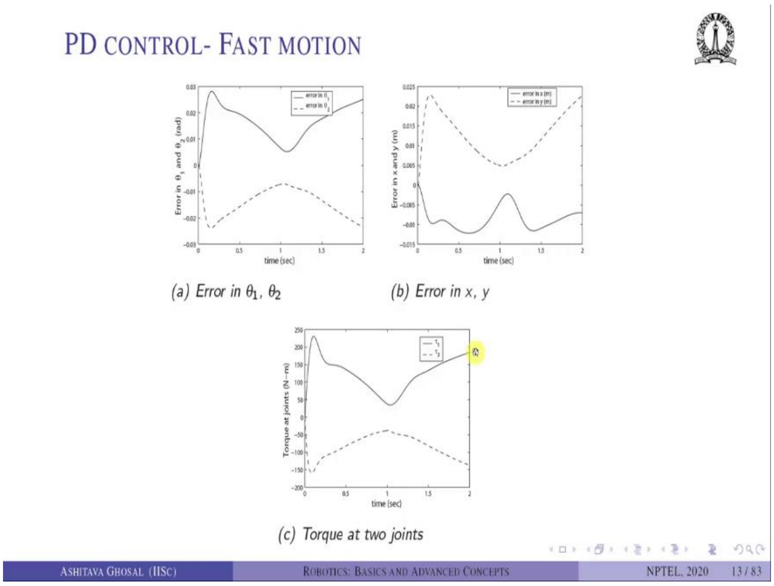
So, as I said, the θ_{1d} and its derivatives are obtained using inverse kinematics and inverse Jacobian. And we will now, present simulation results for a PD control scheme, which is basically a linear control scheme, applied to this robot then, a feed forward controller with exact knowledge of the model parameters. So, although we have $[\widehat{M}]$, $\hat{C}$, $\hat{G}$; the $\widehat{\ }$ is because, we are not using the measured q and $\dot{q}$ and so on, we are using the pre computed desired trajectory.

I will also show you, some results from model based controller with 10 percent error in mass and 5 percent error in location of the CG ok. So, we do not know the masses of the lengths and inertias exactly, let us assume there is a 10 percent error and a 5 percent error in the location of the CG.

Notice that, we are not assuming any error in the link lengths ok. If there is error in the link lengths, then we can never trace the desired trajectory ok. And last, we will look at Cartesian control scheme which is discussed later ok.

We will see what is the equation of motion, when we derived the equations of motion using the Cartesian or end effector coordinates, position and orientation. In all of them, all these above simulation we will choose K_{pi} and K_{vi} as corresponding to $\omega_1 = 85$, $\omega_2 = 75$ and the damping is 2.0. So, basically we have an over damped system, with some chosen natural frequency, we can choose other natural frequencies and damping’s also, but let us see what happens when we choose these values ok.

(Refer Slide Time: 19:51)

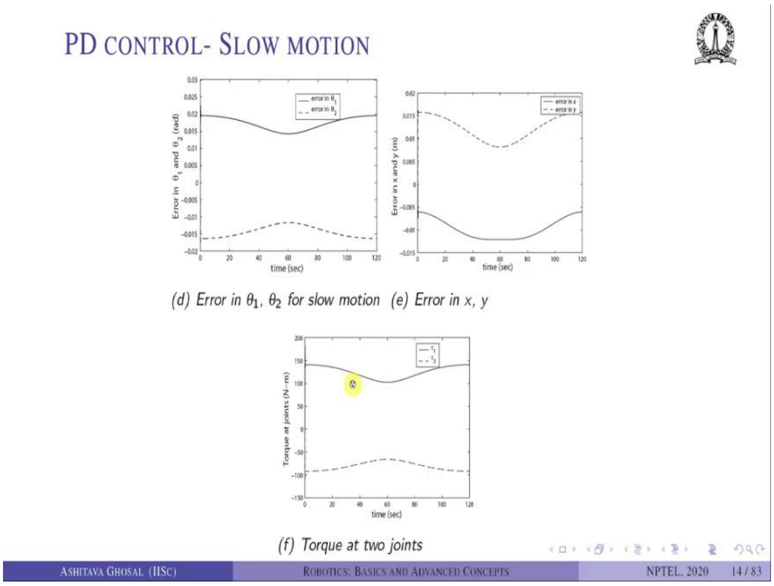


So, as I had shown you, the error for the fast motion is θ_1, θ_2 error. So, it goes like almost 0.03 radians ok. And on the negative side, again 0.025 radians. In x and y also recall that, desired x should be 0; I should just go along the Y axis, the desired trajectory, but we do indeed move along the X axis by some amount and it is of the order of more than 0.02 meters so, like almost 2 centimeters between 2 and 2 and a half centimeters ok.

The y trajectory is also not exactly, following the desired trajectory ok. So, the x is of the order of 0.15 ok so, 1.5 centimeters and in the y direction the maximum error is more; which is of the order of between 2 and 2.5 centimeters. The torque profile looks like this. So, we start from 0, and it goes to some 225 Newton meter for τ_1 and it is lesser for τ_2, it is of the order of little more than minus 150 Newton meters ok.

So, as you can see, important thing to notice here is there is some kind of a peak here and then, it is comes back and then again raises ok.

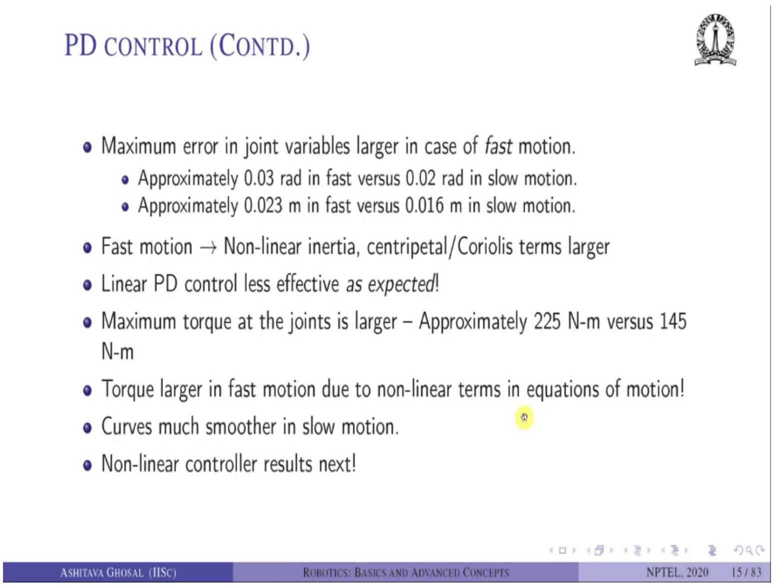
(Refer Slide Time: 21:23)



If you do PD control with slow motion, which was 2 minutes and as I had shown you, the error in θ_1 and θ_2 are much smoother, they are also less ok; the error in x and y are also much smoother and less, and the more importantly the torque at the two joints are smaller ok and they are smoother so, this we had shown earlier.

And the reason was, that if you move slowly, the non-linear terms are smaller; hence their effect is smaller and hence the robot can track better ok, the desired trajectory; the tip of the robot can track the desired trajectory better.

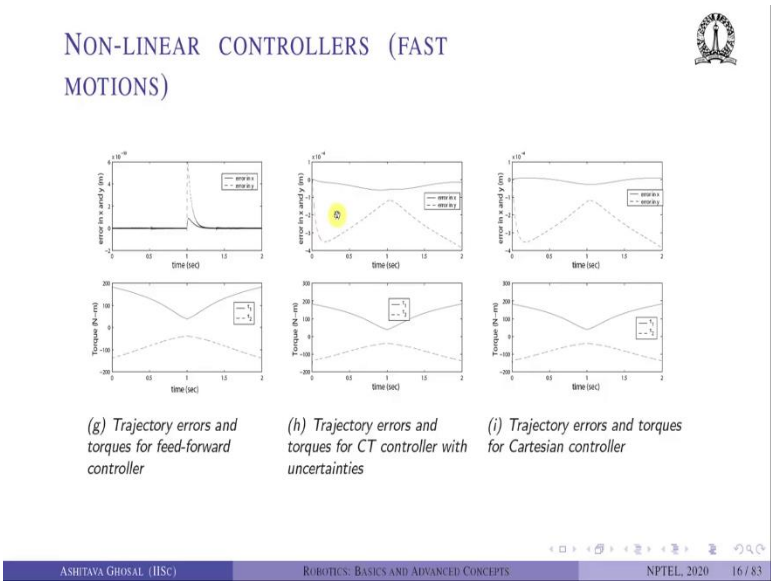
(Refer Slide Time: 22:03)



So, summary the maximum error in the joint variable, in case of fast motion which is what we will look at from now on, is 0.03 radians versus 0.02 in slow motion ok. And 0.023 meters in fast, versus 0.016 meter in slow motion ok. So, as I said, in the fast motion the non-linear inertia, centripetal Coriolis terms are larger ok. Linear PD control is less effective as expected; the maximum torque is also larger, approximately 200 and 25 Newton meter versus 145 Newton meter.

The torque is larger in fast motion again due to the non-linear terms in the equation of motion ok. The curves are much smoother in slow motion and next we will present the non-linear control simulation results. As I said, there are three of them; one is exact cancellation, but feed forward, then there is one with 10 percent error in mass and 5 percent error in location of CG and one is Cartesian control; using Cartesian equations of motion ok.

(Refer Slide Time: 23:27)



So, here are plots. So, let us just go over it. So, and we will look at the fast motion, because in the fast motion for PD control, the errors were larger. So, let us see, what happens when you use non-linear controllers for the fast motion.

So, the error in x and y as you can see is really really small, it is like of the order of 10^{-10} meters. So, in feed forward controller these are the two plots, which show that the error in x and y is very small ok; there is some peak here and it is changing direction, but nevertheless even then it is like 10^{-10} meters. The torque is also much much smoother and very similar to the slow motion torque ok, it is very smooth it comes down and then again increases ok.

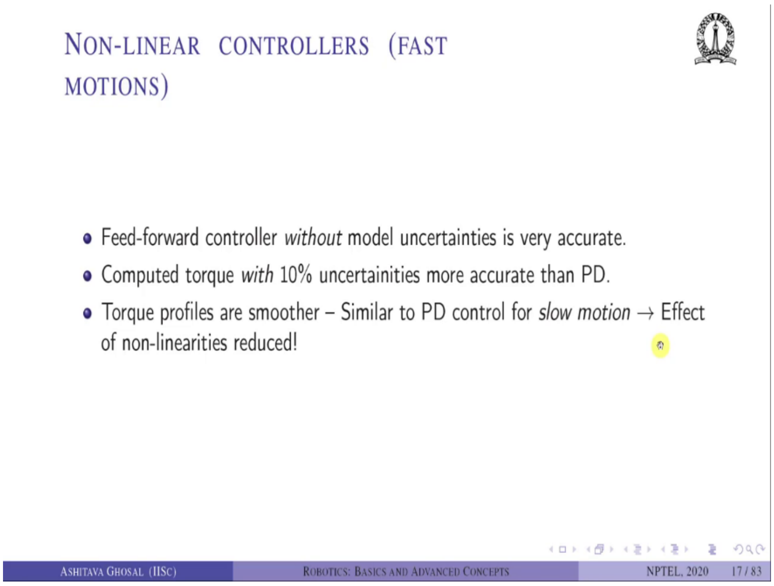
So, as you are going up the torque required is less, because it has acquired speed and then while it is coming down, you have to change the direction of the torque again ok. When you have estimates error of 10 percent and 5 percent, as you can see that the error is much larger than the feed forward with exact cancellation; with exact parameters, nevertheless it is still smaller, which is of the order of 10^{-4} meters in x and y ok.

So, what is 10^{-4} meters? It is like in point something millimeters. The torque is also very very nice and smooth, it is very similar to the torque obtained or the feed forward controller ok and also it is very similar to the slow motion torque, there are no peaks and sharp changes in the torque profile. In the Cartesian controller also it is very small ok, no changes in the torque profile also; and the numbers are very similar for error in x and y.

So, what is the moral of the story? Even with 10 percent error in masses, and 5 percent error in location of CG, the error comes down from something like 2 or 2 and half centimeters to less than a millimeter ok.

And this is shown both in computed torque controller and also in Cartesian controller. And moreover, in computed torque controller with uncertainties. If you know the parameters exactly, then it is much much smaller ok, several orders of magnitude smaller than, when you have an estimates of the mass and location of CG ok.

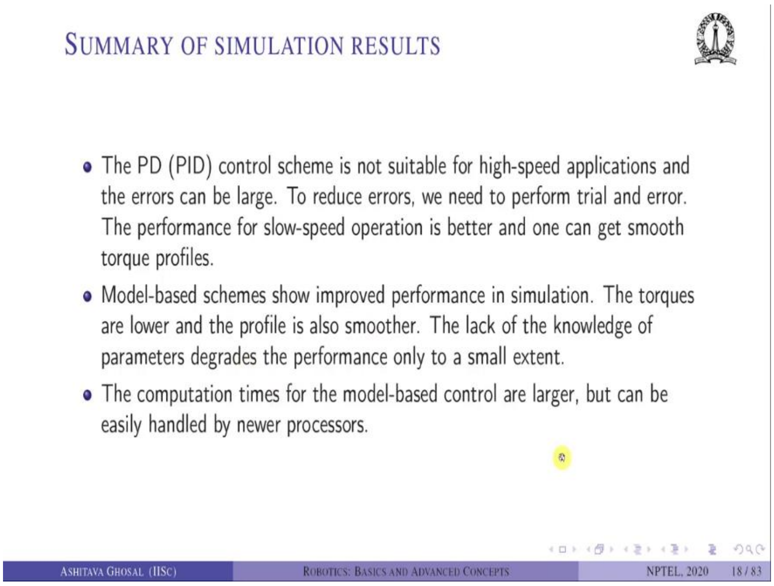
(Refer Slide Time: 26:12)



So, in summary, the feed forward controller without model uncertainty is very accurate. The computed torque with 10 percent uncertainty is not that accurate; however, it is more accurate than PD ok, remember in PD it is of the order of 1 to 2 centimeters. The torque profiles are smoother, it is similar to PD control for slow motion and why? Because the effect of non-linearity is reduced in this computed torque schemes ok. We are somehow cancelling out the non-linear terms in the equations of motion.

So, the 2R robot behaves more like a linear system ok, with computed torque schemes. And it is very similar to the slow motion, because in the PD with slow motion, it was moving slowly the non-linear terms were not so large and hence, the errors were smaller ok. Even one thing to note that, even with PD with slow motion, the errors are larger than the one with computed torque schemes ok.

(Refer Slide Time: 27:29)

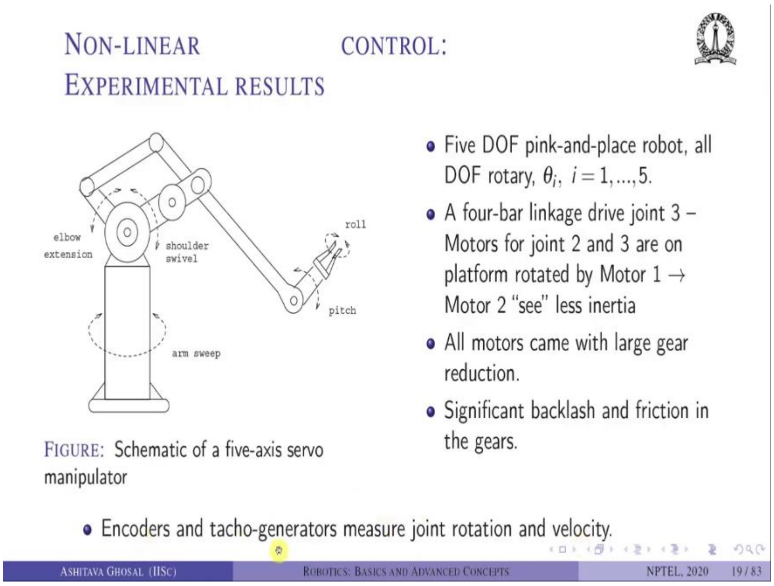


So, summary of simulation results; the PD or PID control scheme is not suitable for high speed applications and the errors can be large ok. To reduce errors, we need to perform trial and error. We can change the controller gains ok. So, I have chosen ω_1 was 85, ω_2 was 75, ξ was 2.0 and then, obtain K_{pi} and K_{vi}; we could play around with that, we can do trial and error to reduce the errors ok.

The performance for slow speed motion is better and one can get is smooth torque profiles. Model based schemes show improved performance in simulation. The torques are lower and the profile of torque is also smoother. The lack of knowledge of parameters degrades the performance to some extent, but not a lot; it is still better than PD without any model.

The computation times for model based controllers are larger, I haven't shown you here, but as you can see we need to compute some 6×6 mass matrix and Coriolis term and centripetal term, but with modern newer processes, which is not very serious issue ok originally, maybe 30 years back, when the processors were not so efficient ok this could have this was an issue.

(Refer Slide Time: 28:55)



Now, let us look at non-linear control for an actual robot. So, I am going to present to you, some experimental results on non-linear control ok. So, this was a robot on which, my two students worked on it on long time back.

So, the robot is a five axis servo manipulator. So, where are the five axis? There is 1 joint, 1 motor which is in the vertical direction then, there are two motors which are located in the same plane here, one is for elbow and one is for shoulder rotation. So, this is called elbow extension and this is shoulder swivel.

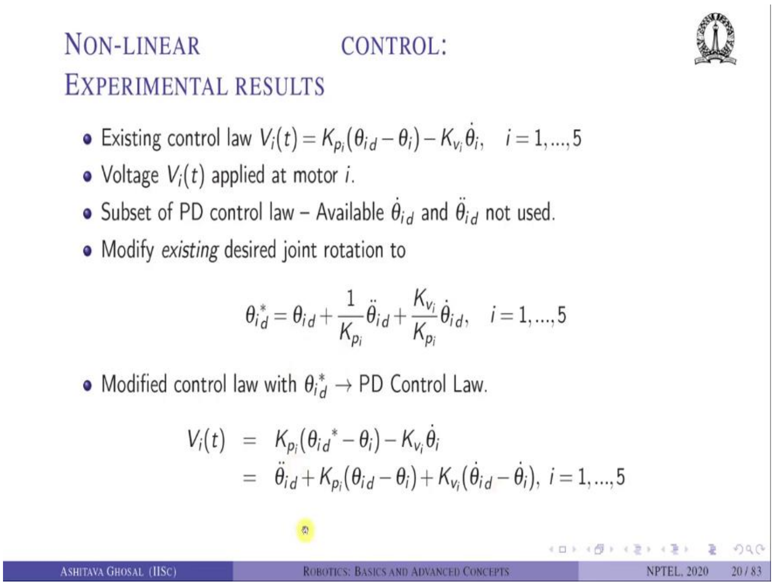
Now, if you have both these motors here, in order to rotate this link we need a mechanism ok. So, basically there is a 4 bar mechanism connecting one of the motors to this linkage. So, one of them rotates this link and one of them rotates the other link and then, this arm will also go up and down ok.

The reason why these two motors are kept on this plane is to minimize the inertia seen by the second motor; first motor sees both these motors inertia, but they are located exactly equal distance from the axis so, that is the nice design.

And then the second motor does not see the third motor inertia. So, in a typical serial robot, the second motor will see the inertia of the third link also in this case not; not so much. But there is a problem or there is a complication, in the sense the drive is not exactly like a serial robots, it is a mixture of serial and parallel robot ok.

The original robot, had motors with large gear reduction, and due to this large gear reduction there were significant back lash and friction in the gears ok. The motors also had encoders and tachometers, tacho generators to measure joint rotation and velocity ok. So, in feedback control we need to measure the θ: q and $\dot{q}$ these were measured using these optical encoders and tacho generators.

(Refer Slide Time: 31:12)



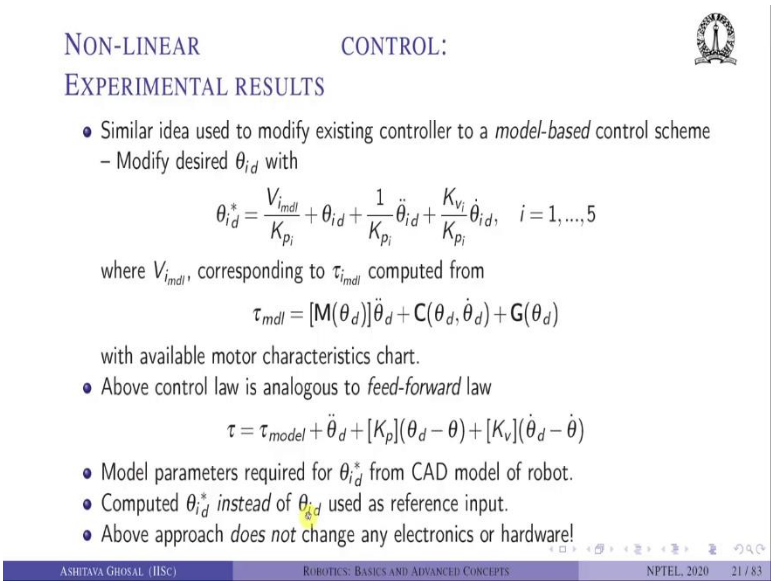
The existing control scheme for this robot, was that the voltage applied at the i^{th} joint is proportional to the error, $\theta_{id} - \theta$ multiplied by a proportional gain minus $K_{vi}\dot{\theta}_i$. So, there is a small difference between what we have seen till now and this, which is that $\dot{\theta}_{id}$ is not being used ok. So, it is $K_{vi}\dot{e}$; $\dot{e} = \dot{\theta}_{id} - \dot{\theta}$ now, $\dot{\theta}_{id}$ is 0. So, hence we have a $-K_{vi}\dot{\theta}_{id}$.

So, this is the voltage, which was applied at motor i ok. The subset of PD control law is being used. So, basically we are not using $\dot{\theta}_{id}$ and $\ddot{\theta}_{id}$. So, first thing that we did was, we modified the existing joint rotation, which is θ_{id} to include the effect of $\ddot{\theta}_{id}$ and $\dot{\theta}_{id}$.

So, it is a clever way of doing it that, instead of θ_{id}; we say, if we supply θ_{id}^* where $\theta_{id}^* = \theta_{id} + \frac{1}{K_{pi}}\ddot{\theta}_{id} + \frac{K_{vi}}{K_{pi}}\dot{\theta}_{id}$.

So, if you substitute back, this θ_{id}^* in this equation instead of θ_i, what you will get is, $\ddot{\theta}_{id} + K_{pi}e + K_{vi}\dot{e}$. So, what have we done? We have modified the desired input, somewhat to obtain the standard PD control of which we are familiar with ok.

(Refer Slide Time: 33:07)



So, we can also modify the existing controller to a model based control scheme, not exactly a computed torque model based control scheme, which cancels the non-linearities, we modify θ_{id} with some voltage due to a model divided by K_{pi} plus the original θ_{id} plus this $\ddot{\theta}_{id}$ divided by K_{pi} and $\frac{K_{vi}}{K_{pi}}\dot{\theta}_{id}$, for each one of the 5 joints ok.

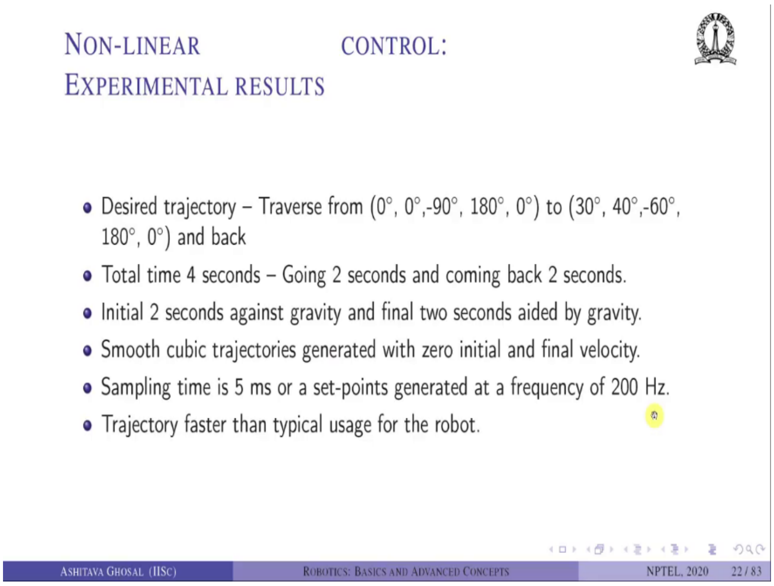
And this voltage due to the model is based on the equations of motion. So, we know what is the desired trajectory, we know $\ddot{\theta}_d$, we know Coriolis term, we know gravity term, we compute the torque which is required to obtain this $\ddot{\theta}_d$ and $\dot{\theta}_d$ and θ_d, that is called τ_{mdl}. And this voltage due to the model is corresponding to this τ_{mdl} ok and this is obtained with available motor characteristic charts.

So, the above control scheme, once you substitute θ_{id}^{*} and τ_{mdl} and $V_{i_{mdl}}$, it will look like $\tau = \tau_{mdl}$, the input to the motor torque is equal to the torque due the dynamics, which is the model part and the PD part ok. So, as you can see it is not $[\alpha]\tau' + \beta$, it is some biased term ok, which is β plus some PD term ok.

So, the model parameters required for θ_{id}^{*} ok like $[M]$, C, link lengths all these things were obtained from the CAD model of the robot. So, this robot was designed in this place. So, which had the CAD model and we could obtain this. So, finally, what happens is, we are going to use θ_{id}^{*} instead of θ_{id}. So, the resistance input is changed ok.

And what is the advantage of doing this? We do not have to touch the electronics of the hardware. All we say is that the desired trajectory is changed and the desired trajectory now, somehow takes into account the $\dot{\theta}_{id}$, $\ddot{\theta}_{id}$ and the model to some extent.

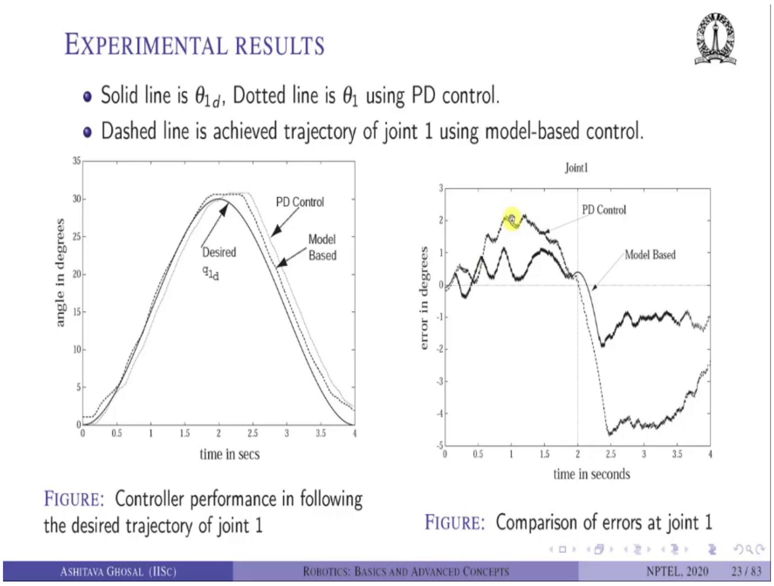
(Refer Slide Time: 35:45)



So, what was the desired trajectory? The traverse from $(0°, 0°, -90°, 180°, 0°)$ to $(30°, 40°, -60°, 180°, 0°)$ and back ok. So, we start from some initial joint angles to some final joint angles and come back. And we assume that, the total time was 2 seconds going from this point to this point and another 2 seconds to come back. So, it turns out that the initial 2 seconds is against gravity and the final 2 seconds is aided by gravity for this robot.

We generated smooth trajectories with zero initial and final velocity, the sampling time is 5 milliseconds or a set point is generated at the frequency of 200 Hertz ok. So, this trajectory is typically faster than the usage of that robot, which was designed for the robot was not designed to show this much distance in 2 seconds.

(Refer Slide Time: 36:46)



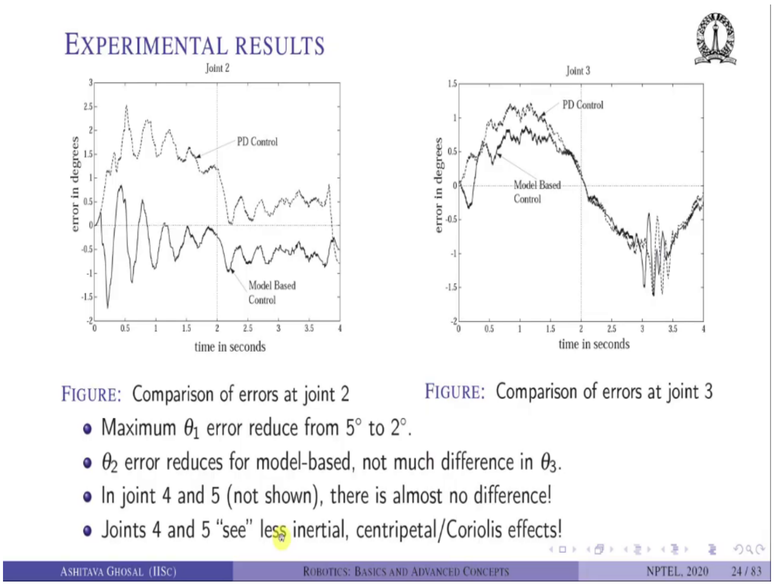
Ok now, let us look at some of the experimental results that we have obtained. So, in this plot here, the solid line is the desired θ_1 ok. So, it is a nice smooth cubic profile up to some place and cubic profile from 2 to 4 seconds. So, it goes against gravity and comes back.

So, the first dashed line ok is the trajectory or θ_1 due to the model, when the model based terms were included. The second small dotted line is what happens, when we do a PD control, basically we modified θ_{id}^* to include the $\dot{\theta}_{id}$ and $\ddot{\theta}_{id}$. And this one is the includes the model. So, what you can see is, the difference between the desired and model based is smaller ok.

The difference between the desired and the PD is larger than the difference between the desired and the model based. And that is specially so, when it is coming down with gravity. The plot on the right hand side shows the difference between q_{1d} and q_1 obtained for model based and q_1 obtained for PD.

So, as you can see, for model based the error is like; maximum error is between 1 and 2 degrees; whereas, for PD it is between on the other side, it is like 4 and half degrees, but while going up it is like 2 degrees ok.

(Refer Slide Time: 38:21)



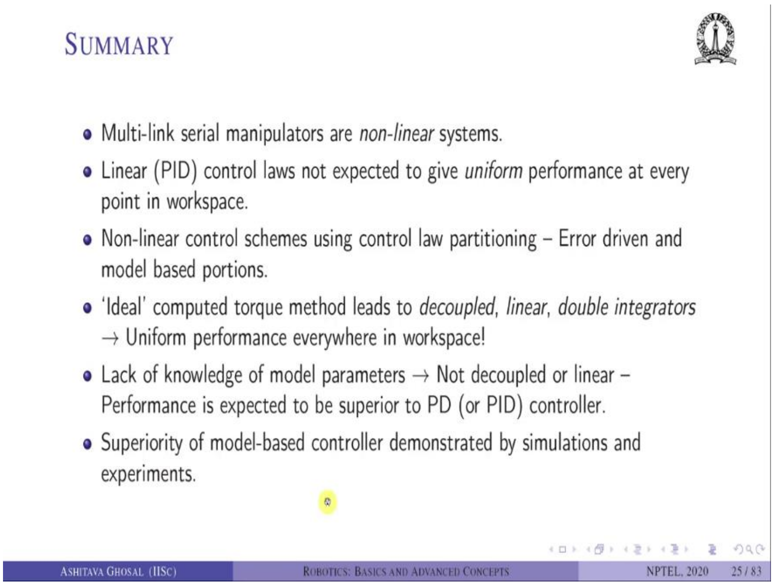
For joint 2 also, we can compare the errors and I am not showing you the trajectory there are also the desired is smooth and record what the joint 2 is doing from the encoder and plot the θ_2 as a function of time, for PD control and θ_2 as a function of time for model based control and then, we subtract θ_2 from the θ_{2d} in both cases.

So, this shows the error between θ_2 in PD control and θ_2 in model based control with respect to the θ_{2d}. And as you can see again, that the model based is smaller ok. So, here it is like 2 and half degrees, this is the maximum is like 1 and a half degrees. Likewise for joint 3, we can plot the error, when we execute PD control and when we execute model based control with respect to the desired trajectory.

So, here you can see that, the difference is not that much ok. So, it is like in one case, it is like 1.25 and this is like 0.75. On the other side it is more or less same. And one of the interesting thing that you can see is at some place there is some resonance ok, there is some oscillations which are happening, although we had chosen the gains such that it is overdamped nevertheless you can see some oscillations ok.

So, the maximum θ_1 error reduces from 5 to 2 degrees, θ_2 error reduces for model based not much difference in θ_3 and in joint 4 and 5, which I am not showing there is almost no difference between PD and model based ok. And there is a reason, the joint 4 and 5 are the last two joints, they do not see too much inertial Coriolis and other effects ok.

(Refer Slide Time: 40:40)



It is moving a very small mass, the last two joints. Whereas the first joint is moving all the links after the first joint so, it sees a lot of inertia, a Coriolis and other effects. So, in summary the multi link serial manipulators are non-linear systems, we know that from the equations of motion. The linear PID control law not expected to give uniform performance at every point in the workspace, as and that you can see.

The non-linear control scheme using control law partitioning ok, which is error driven plus model based portion, performs better. The ideal computed torques method leads to decoupled, linear, double integrators with uniform performance everywhere in the workspace, remember it was $\tau' = \ddot{q}$.

If we do not know the model parameters well; that means, $[M(q)]$ is some $[\widehat{M(q)}]$, $C(q, \dot{q})$ is some $\widehat{C(q, \dot{q})}$ and so on. Then we do not get decoupled or linear set of equations ok, nevertheless the performance is expected to be superior to PD or PID controller as I have shown. And, basically the model based controllers are superior ok and I have shown you, 3 experiments of θ_1, θ_2 and θ_3 experimental data. And similarly, I have shown you for simulation also ok. So, with this we come to an end of this lecture. In the next lecture, we will look at control of constrained and parallel manipulators.