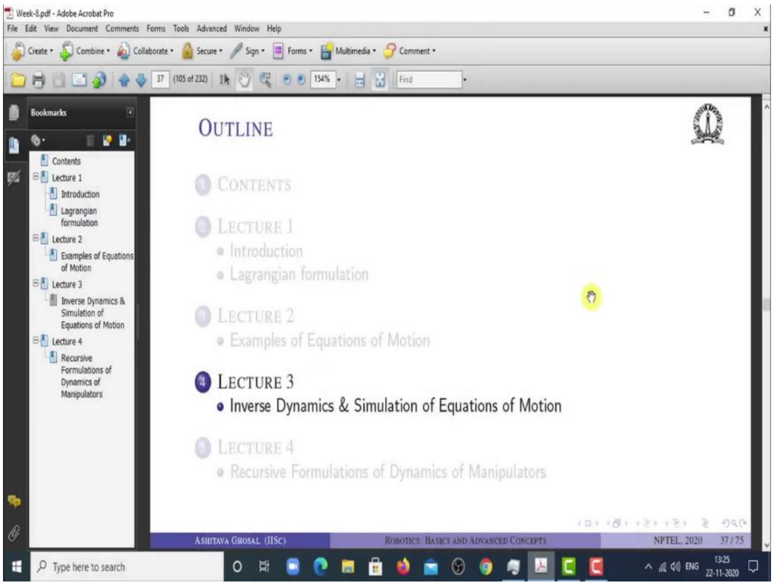
Robotics: Basics and Selected Advanced Concepts
Prof. Ashitava Ghosal
Department of Mechanical Engineering
Indian Institute of Science, Bengaluru

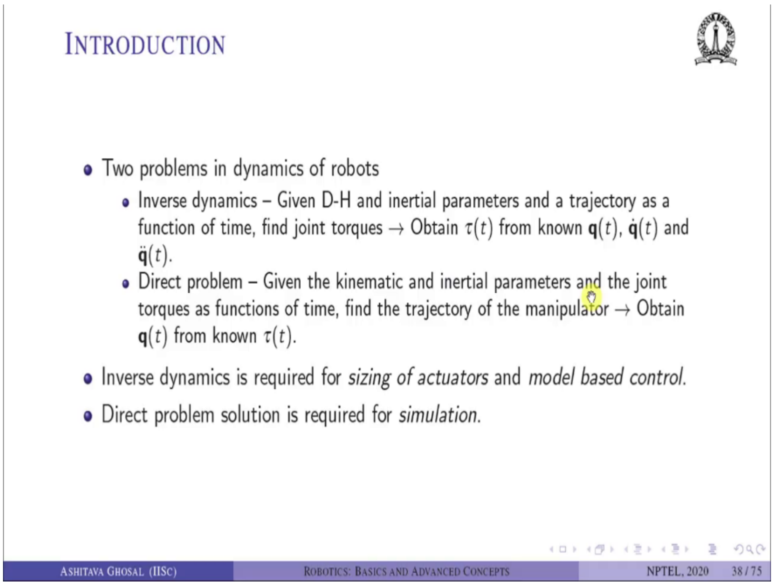
Lecture - 29
Inverse Dynamics and Simulation of Equations of Motion

(Refer Slide Time: 00:19)



Welcome to this NPTEL lectures on Robotics. In the last lecture, I had shown you examples of equations of motion of a planar two-degree of freedom robot and a planar 4-bar mechanism derived using the Lagrangian formulation. In this lecture, we will look at what to do with these equations of motion. Specifically, we will look at two problems in dynamics, one is called inverse dynamics and one is called direct dynamics and the direct dynamics problem involves simulations of equations of motion ok.

(Refer Slide Time: 01:00)



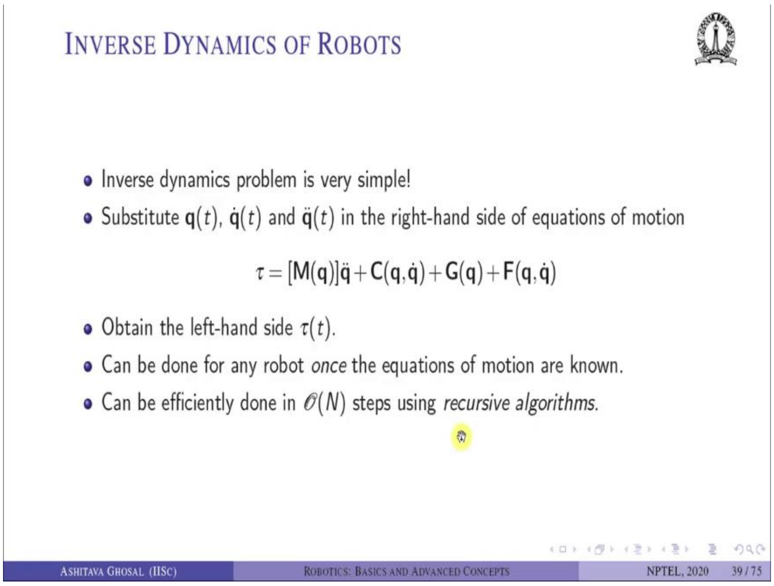
So, let us continue. So, as I said there are two problems in dynamics of robots, one is inverse dynamics which is given D-H and inertial parameters and a trajectory as a function of time, find joint torques ok. So, we need to obtain $\tau(t)$ from known $\boldsymbol{q}(t)$, $\dot{\boldsymbol{q}}(t)$ and $\ddot{\boldsymbol{q}}(t)$. So, $\boldsymbol{q}(t)$ is the trajectory as a function of time and $\dot{\boldsymbol{q}}(t)$ is its first derivative and $\ddot{\boldsymbol{q}}(t)$ is the second derivative.

The direct problem on the other hand is given the kinematic and inertial parameters and the joint torques as a function of time, find the trajectory of the manipulator. So, basically, we need to obtain $\boldsymbol{q}(t)$ from known $\tau(t)$. So, the inverse problem is required for sizing of actuators and model-based control ok. Sizing of actuator means that suppose you want to design a robot and the robot is supposed to perform a set of tasks.

Basically, we have an idea of $\boldsymbol{q}(t)$, $\dot{\boldsymbol{q}}(t)$ and $\ddot{\boldsymbol{q}}(t)$, we can solve the inverse problem, we can substitute $\boldsymbol{q}(t)$, $\dot{\boldsymbol{q}}(t)$ and $\ddot{\boldsymbol{q}}(t)$ into the equations of motion and obtain $\tau(t)$ torques and based on the maximum torques from this computation, we can decide what motors to choose and what should be the torque characteristics of the motor ok.

It is also used in model-based control which you will see later. The direct problem requires simulation meaning what that we are given $\tau(t)$ and we know that the equations of motion are second order ODE's. So, we need to integrate these equations of motion to obtain $\boldsymbol{q}(t)$. So, this is also called simulation.

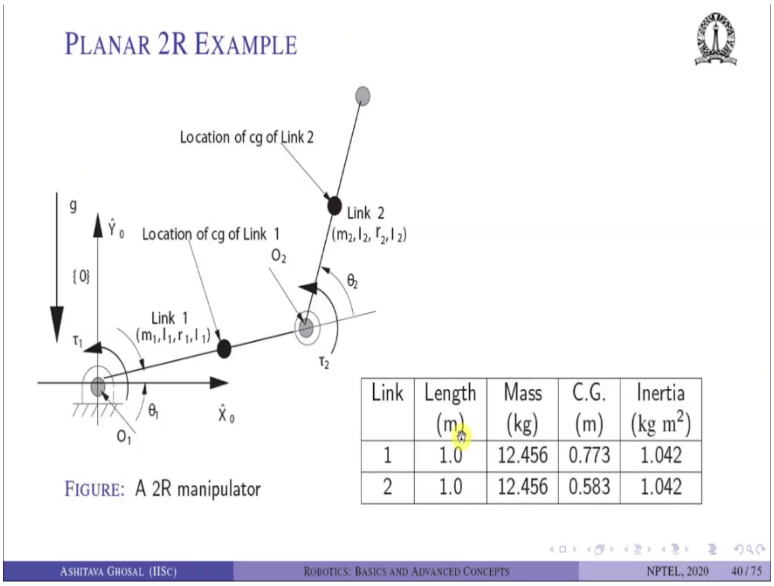
(Refer Slide Time: 02:58)



Inverse problem is very simple why? Because all it requires is that substitute the given $\boldsymbol{q}(t)$, $\dot{\boldsymbol{q}}(t)$ and $\ddot{\boldsymbol{q}}(t)$ in the right-hand side of the equations of motion. The equation of motion is given in general as $\tau = [\boldsymbol{M}(\boldsymbol{q})]\ddot{\boldsymbol{q}} + [\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})]\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{F}(\boldsymbol{q},\dot{\boldsymbol{q}})$.

So, if you know $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$, you can just directly substitute and obtain the left-hand side which is torque as a function of time, ok. So, it can be done for any robot since the equations of motion are known. You can do it very efficiently because we will see later that this can be obtained in $\mathcal{O}(N)$ steps using what are called as recursive algorithms.

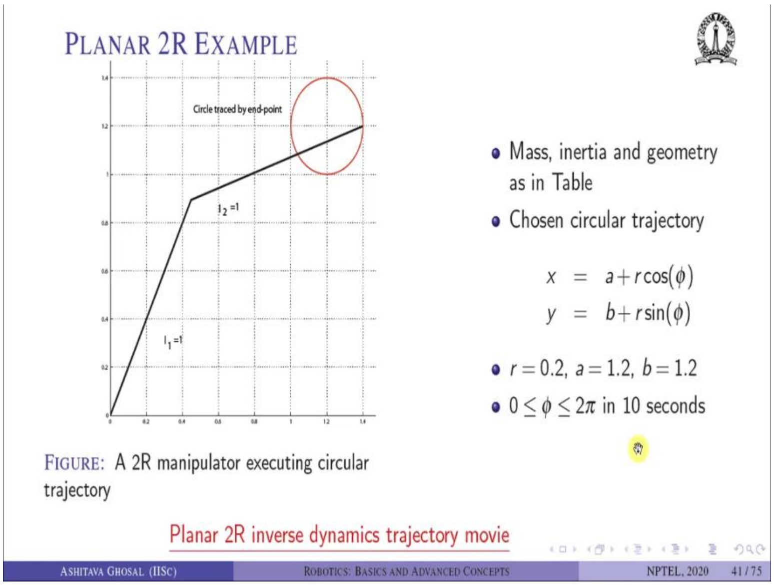
(Refer Slide Time: 03:58)



Planar 2R example: So, again we go back to our very familiar simple example of a two-degree of freedom planar robot with two rotary joints. Again, the mass, length, location of the CG and the inertia are given. The gravity is acting downwards, the location of the CG of the second link is given so, there is a τ_1 which is acting on the first joint and τ_2 which is acting at the second joint. So, in order to compute τ_1 and τ_2, we need to actually first assume some mass, lengths, inertia and location of the CG.

So, in this numerical example, we have assumed that the lengths are 1 meter, the mass is some 12.456 kg, CG is located 0.773 meters away in the first link and 0.583 in the second link and the inertia is 1.042 kg meter square ok. So, these numbers are from data from an experiment ok. So, you might think that is why have we chosen all these funny looking numbers, these are actually from some experiment and literature.

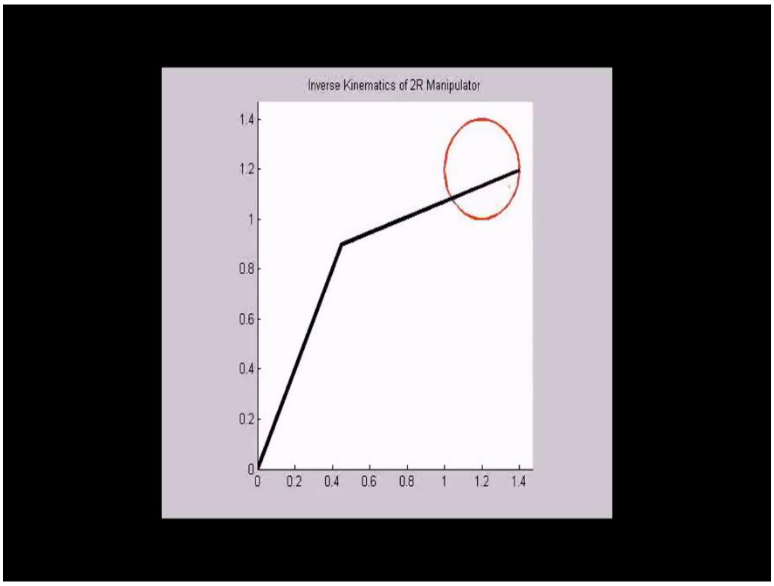
(Refer Slide Time: 05:13)



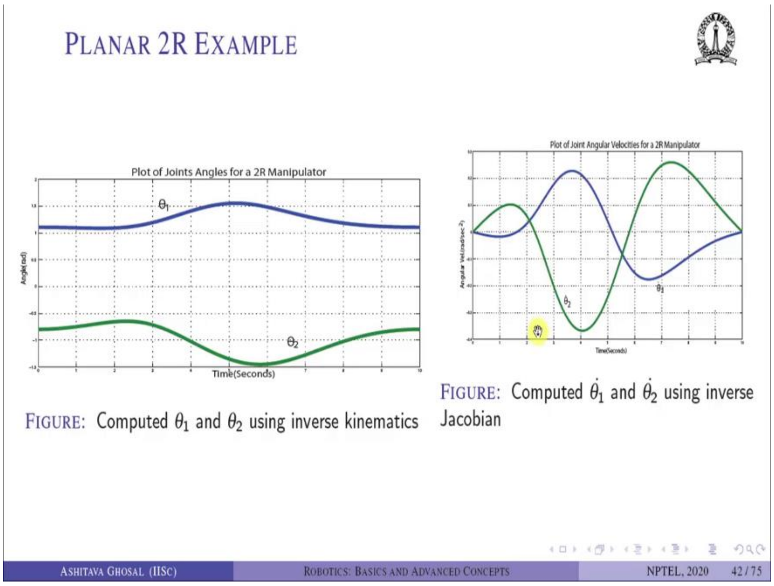
So, the mass, inertia, geometry are as in table ok. We now need to choose a trajectory ok. So, in this numerical example, I have chosen a circular trajectory. So, the tip of these robots to trace a circle which is given by $x = a + r\cos\phi$, $y = b + r\sin\phi$ so, ϕ is the parameter for the circle.

We choose $r = 0.2$, $a = 1.2$, $b = 1.2$, it is located at a distance from the origin and the whole circle lies inside the workspace ok. So, we have this parameter ϕ which varies from 0 to 2π in about 10 second ok. So, later on, I will show you the video of this, but as you can see, this is the example that we are interested in.

(Refer Slide Time: 06:07)



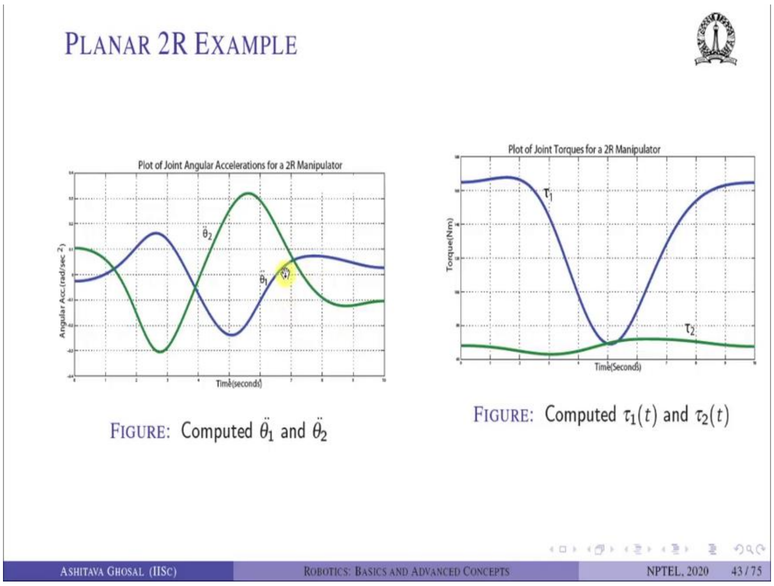
(Refer Slide Time: 06:16)



So, to continue, first we compute θ_1 and θ_2 for that circular trajectory. So, we can see that the angle θ_1 and θ_2 can be obtained using inverse kinematics. So, the y axis is in radians and the x axis is in time so, we go from 0 to 10 seconds, we go 0 to 2π. So, θ_1 plot and θ_2 plot can be obtained.

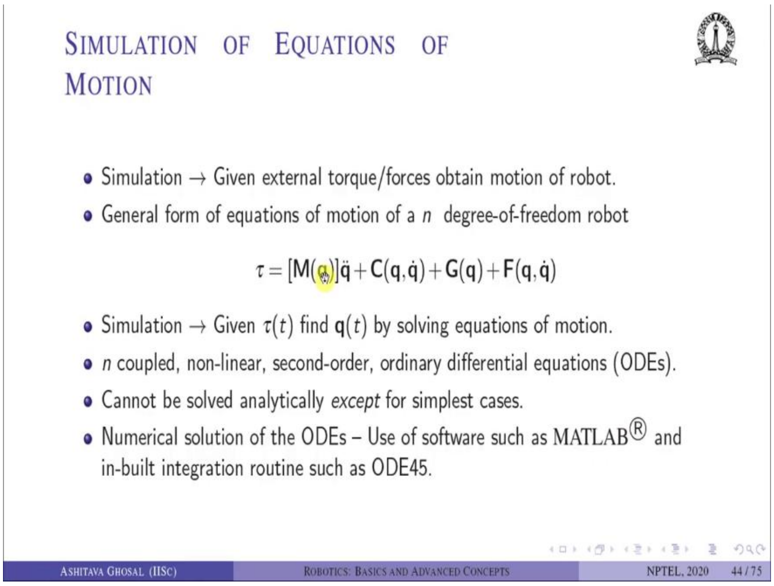
We can also compute $\dot{\theta}_1$ and $\dot{\theta}_2$ using inverse Jacobian. For the 2R manipulator, the Jacobian is simple a 2×2 matrix we can invert the Jacobian and we can find $\dot{\theta}_1$ and $\dot{\theta}_2$. So, the plot of $\dot{\theta}_1$ and $\dot{\theta}_2$ are in blue and green in these two figures.

(Refer Slide Time: 07:07)



We can also find the acceleration because we have an expression which is $\ddot{\theta}$ some $[J]$ inverse something and so on. So, you can find $\ddot{\theta_1}$ and $\ddot{\theta_2}$. Once we have always θ, $\dot{\theta}$, $\ddot{\theta}$, you can just substitute in the equations of motion and plot τ_1 and τ_2. So, as you can see τ_1 looks like this, it is much larger ok. So, it is of the order of 120 or 130 Newton meter, where the τ_2 is much smaller.

(Refer Slide Time: 07:47)



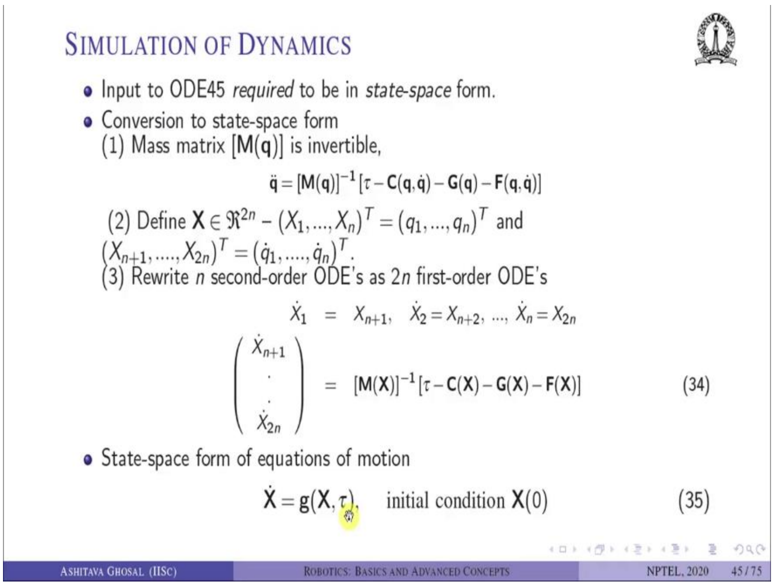
So, this is a simple problem as I showed you as an example, it can be very easily done, there is nothing much to it we just do substitution. The simulation on the other hand of the equations of motion is more interesting and a little bit more challenging. So, what is the problem in simulation? We are given the external torques and forces and you need to obtain the motion of a robot.

So, the general equation of motion of an n degree-of-freedom robot can be given as $\tau = [\boldsymbol{M}(\boldsymbol{q})]\ddot{\boldsymbol{q}} + [\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})]\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{F}(\boldsymbol{q},\dot{\boldsymbol{q}})$. So, in simulation, you have given the left-hand side and we have to solve for $\boldsymbol{q}(t)$ basically, we have to solve this ordinary differential equations. So, these are n coupled, non-linear, ordinary, second-order differential equation, second-order because there is $\ddot{\boldsymbol{q}}$.

We cannot solve these things analytically except probably some very trivial simple case ok. Remember this is non-linear, it has $\sin\theta$, $\cos\theta$, then $\dot{\theta}_1^{\,2}$, $\dot{\theta}_2^{\,2}$ all kinds of non-linearities are there. So, we need to resort to numerical solution ok.

So, we go to MATLAB ok, we use a software tool such as MATLAB which has in-built integration routines ok. So, for example, if you go and look in MATLAB, there is something called ODE45 ok. ODE45 means some 4th order, it is a predictor corrector method, 4th order predict, and 5th order correct Runge Kutta based methods.

(Refer Slide Time: 09:34)



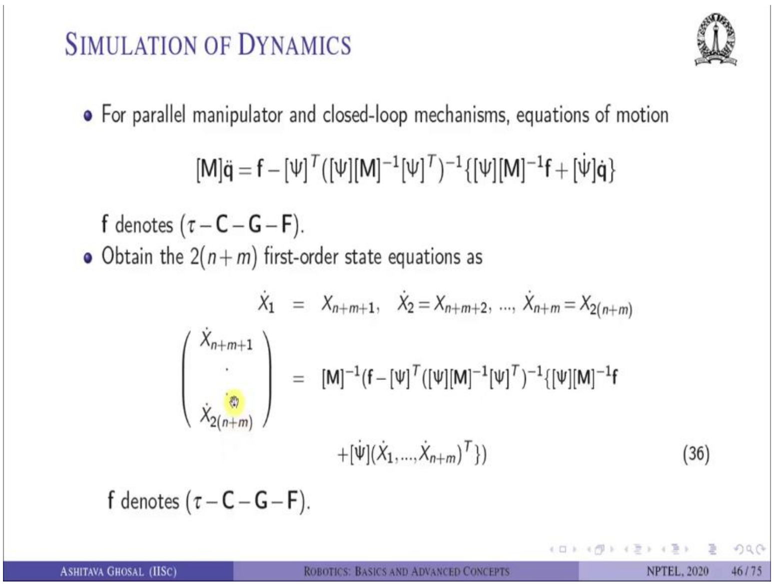
So, the input to this ODE45 in MATLAB requires you to give the equation in state-space form ok. So, what is the state space form? States-space form basically means it is should be of the form $\dot{\boldsymbol{X}} = \boldsymbol{g}(\boldsymbol{X},\tau)$. We need only first order ODE's. So, I need to convert my equations of motion which are second-order ODE's to first order ODE's so, which is possible and it is not very hard.

So, what we 1st do is we rewrite the equation of motion by using the inverse of the mass matrix, the mass matrix is always invertible remember it is positive definite symmetric. So, $\ddot{\boldsymbol{q}} = [\boldsymbol{M}(\boldsymbol{q})]^{-1}(\tau - [\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})]\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{F}(\boldsymbol{q}, \dot{\boldsymbol{q}}))$. Then, we define a state vector $\boldsymbol{X}$ which is $(X_1, \dots, X_n)^T$ which are nothing, but $(q_1, \dots, q_n)^T$ and $(X_{n+1}, \dots, X_{2n})^T$ which are nothing, but $(\dot{q_1}, \dots, \dot{q_n})^T$.

So, basically, we have a n second-order equations, we can rewrite it as $2n$ first-order ODE's ok. So, $\dot{X_1}$ you can see is nothing but X_{n+1}, $\dot{X_1} = \dot{q_1}$ which is X_{n+1} and likewise, $\dot{X_2} = X_{n+2}$ and so on. So, what do we have? We have $(\dot{X}_{n+1}, \dots, \dot{X}_{2n})^T$ which are nothing but $\ddot{\boldsymbol{q}} = [\boldsymbol{M}(\boldsymbol{X})]^{-1}(\tau - \boldsymbol{C}(\boldsymbol{X}) - \boldsymbol{G}(\boldsymbol{X}) - F(\boldsymbol{X}))$. So, we can substitute $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ as $\boldsymbol{X}$ in this and we can get this equation.

So, what are the equations that we have? We have $\dot{X}_1 = X_{n+1}$, $\dot{X}_2 = X_{n+2}$ and so on, and $\dot{X}_{n+1}$ to $\dot{X}_{2n}$ is given by this. So, combining these n equations and another n-equations, we get $2n$ first-order equations which can be written in the form $\dot{\boldsymbol{X}} = \boldsymbol{g}(\boldsymbol{X}, \tau)$. So, remember $\boldsymbol{X}$ contains $\boldsymbol{q}$ and also $\dot{\boldsymbol{q}}$, it is a $2n$ variables ok. So, in order to solve this first order ODE's, we also need initial conditions. So, basically, we must be given what is $\boldsymbol{q}(0)$ and $\dot{\boldsymbol{q}}(0)$ at time $t = 0$ which is $\boldsymbol{X}(t = 0) = \boldsymbol{X}(0)$.

(Refer Slide Time: 12:23)

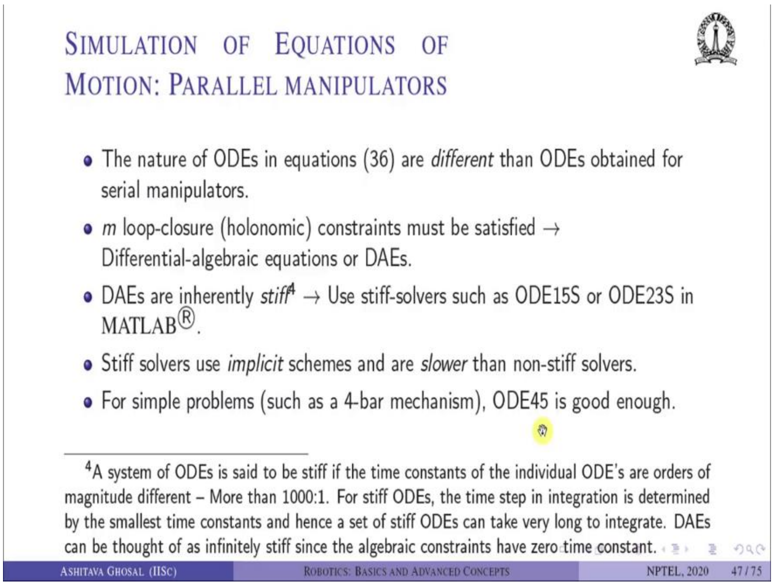


So, for parallel manipulators, we can rewrite the equation of motion again as $[\boldsymbol{M}]\ddot{\boldsymbol{q}} = \boldsymbol{f} - [\Psi]^T([\Psi][\boldsymbol{M}]^{-1}[\Psi]^T)^{-1}\{[\Psi][\boldsymbol{M}]^{-1}\boldsymbol{f} + [\dot{\Psi}]\dot{\boldsymbol{q}}\}$, as $[\Psi]$ is the constraint matrix and $\boldsymbol{f} = \tau - \boldsymbol{C} - \boldsymbol{G} - \boldsymbol{F}$). So, we are dropping this that it is a function of $\boldsymbol{q}$'s or $\boldsymbol{X}$'s.

So, in parallel manipulators, we can get $2(n+m)$ first-order equations. Again, $\dot{X}_1 = X_{n+m+1}$, $\dot{X}_2$ is this, $\dot{X}_{n+m} = X_{2(n+m)}$ and $(\dot{X}_{n+m+1}, \dots, \dot{X}_{2(n+m)})^T = [\boldsymbol{M}]^{-1}\boldsymbol{f} - [\Psi]^T([\Psi][\boldsymbol{M}]^{-1}[\Psi]^T)^{-1}\{[\Psi][\boldsymbol{M}]^{-1}\boldsymbol{f} + [\dot{\Psi}](\dot{X}_1, \dots, \dot{X}_{n+m})^T\}$.

So, again $\boldsymbol{f} = \tau - \boldsymbol{C} - \boldsymbol{G} - \boldsymbol{F}$, this is what $\boldsymbol{f}$ denotes, where τ is given. So, again, we can rewrite this equation as $\dot{\boldsymbol{X}} = \boldsymbol{g}(\boldsymbol{X}, \tau)$ where now the $\boldsymbol{X}$ is $2(n+m)$ variables. So, we have $2(n+m)$ first-order equations for parallel manipulators ok.

(Refer Slide Time: 13:56)



So, the nature of ODE's in a parallel manipulator they look similar, but they are slightly different than the ODE's obtained for the serial manipulator ok. Let us discuss this a little bit. So, the m loop-closure equations must be satisfied that all instance of time ok. So, not only do we have differential equations, but we have m loop-closure equations which are holonomic constraints, they do not include time ok, they do not have derivatives of the variables of the passive joint's variables or the actuated joint variables.

So, this set of some differential equations and some holonomic constraint equations are also called Differential-algebraic equations or DAE's ok. The DAE's are inherently stiff ok. So, what do we mean by stiff? A system of ODE's is said to be stiff if the time constant of the individual ODE's are orders of magnitude different so, more than less than 1000 is to 1 ok. What do we mean by time constant? We have $\dot{\boldsymbol{X}} = \boldsymbol{g}(\boldsymbol{X}, \tau)$.

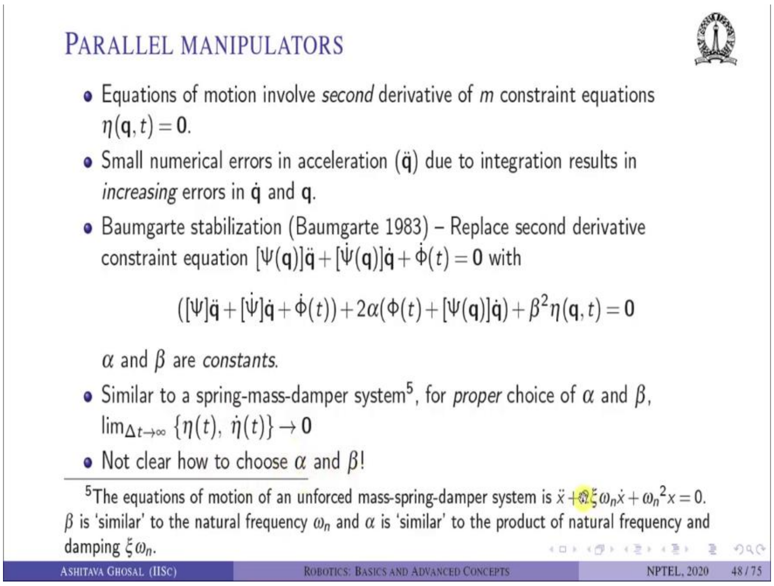
So, this is essentially given an input the $\boldsymbol{X}$ will take some time to reach the state similar to later on we will see what happens in control. So, there is something called time constant for any differential equation. Now, if the time constants for some variables are much much faster or the time constants are much much slower, smaller than the others, then these equations are called stiff.

So, there should be a spread in time constants and if it is more than 1000 : 1, we call them stiff. So, for stiff ODE's, the time step in integration is determined by the smallest time constant and hence a set of stiff ODE's can take very long to integrate. So, we have to take very very small steps if the time constant of some equations are very different than the other.

So, DAE's can be thought of as infinitely stiff since the algebraic constants have zero time constant ok. So, there are these algebraic constraints, holonamic constraints, they have zero-time constant. So, DAE's are inherently stiff, but it is not so serious because we can use stiff solvers such as ODE15S or ODE23S. So, S stands for stiff solvers ok.

So, stiff solvers use what are called as implicit schemes ok, they are not marching forward in time, they solve a set of equation at each time steps and are inherently slower than non-stiff solvers. However, for simple problems such as 4-bar mechanism, ODE45 is good enough. So, we can use ODE45 to solve for the 4-bar parallel mechanism, 4-bar mechanism ok.

(Refer Slide Time: 17:10)



So, parallel manipulator let us continue. The equations of motion involve second derivative of the m constraint equation, this is another aspect. So, if you have small numerical errors in acceleration due to integration, it results in increasing errors in $\dot{\boldsymbol{q}}$ and $\boldsymbol{q}$ because the $\ddot{\boldsymbol{q}}$ is slightly wrong, very very slightly wrong due to the integration.

So, then integral of that which is nothing but summation ok be more, the error will be more, and $\boldsymbol{q}$ will be even worse. So, as a result in parallel manipulator, the $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ will slowly diverge from the real real solution. So, there are various approaches to stabilize this increasing errors ok.

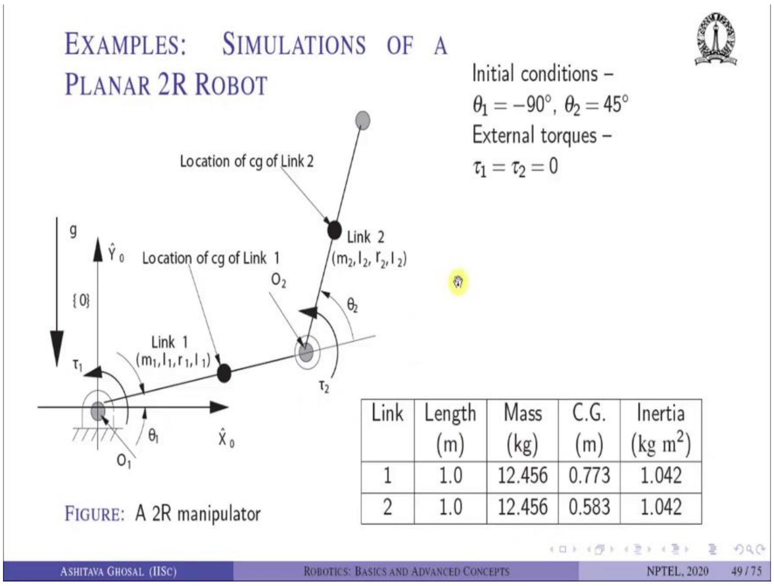
So, one such is called as the Baumgarte stabilization. So, basically, instead of using $[\Psi(\boldsymbol{q})]\ddot{\boldsymbol{q}} + [\dot{\Psi}(\boldsymbol{q})]\dot{\boldsymbol{q}} + \dot{\phi}(t) = 0$, this is my second derivative of the constraint equation, we replace it by $\left([\Psi(\boldsymbol{q})]\ddot{\boldsymbol{q}} + [\dot{\Psi}]\dot{\boldsymbol{q}} + \dot{\phi}(t)\right) + 2\alpha(\phi(t) + [\Psi(\boldsymbol{q})]\dot{\boldsymbol{q}}) + \beta^2\eta(\boldsymbol{q},t) = 0$, where α and β are constants.

So, basically, what it looks like is it is like a spring-mass-damper system. So, this is like $m\ddot{x} + c\dot{x} + kx = 0$. So, if you choose properly α and β, both $\eta(t)$ and $\dot{\eta}(t)$, its first derivative both will go to 0 just like a spring-mass-damper, if you choose the damping and the spring properly, the oscillations will die down, ok.

So, hence, if you use this Baumgarte stabilization, where we have chosen somehow α and β, then η will be satisfied, $\dot{\eta}$ will also be satisfied and $\ddot{\eta}$ is of course, satisfied because if these two are satisfied ok. So, similar to a spring mass damper system exactly. But there is a problem, we do not know how to choose α and β ok. In a spring mass damper system just to go back, we have $\ddot{x} + 2\xi\omega_n\dot{x} + \omega_n^2 x = 0$.

So, β is similar to the natural frequency ω_n and α is similar to the product of natural frequency and damping ok, but this is not exactly same because it is very complicated so, we do not know how to choose α and β, but nevertheless Baumgarte showed that this kind of stabilization can work, and we can make sure that the constraint equations are satisfied at each instant of time.

(Refer Slide Time: 20:26)



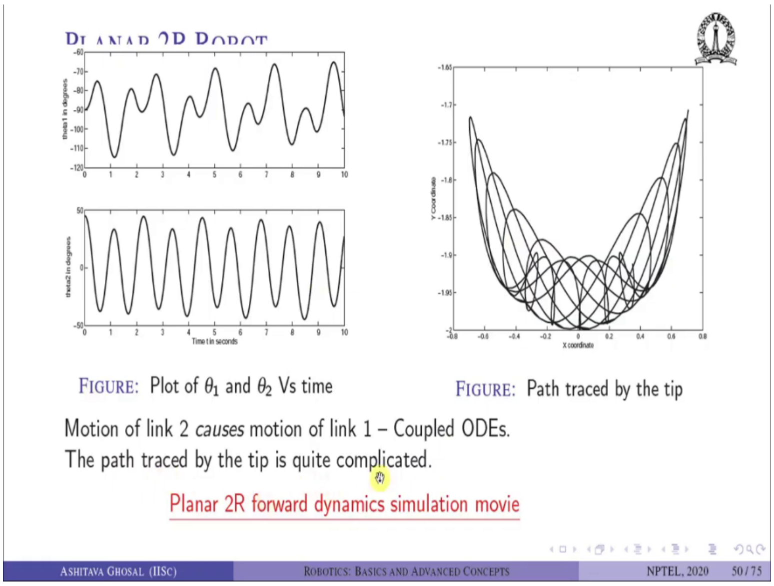
Link	Length (m)	Mass (kg)	C.G. (m)	Inertia (kg m^2)
1	1.0	12.456	0.773	1.042
2	1.0	12.456	0.583	1.042

So, let us take some examples of simulation. So, we have a planar two-degree of freedom robot, again we have this two links, there is a τ_1 and τ_2, there is a gravity vector which is acting down along the negative Y axis and we have m_1, l_1, r_1 and I_1. For link 2, you have m_2, l_2, r_2 and I_2. So, we take the same lengths and inertia which was done for the inverse dynamics problem.

So, lengths is 1 meter, mass is 12.456 kg and so on ok. So, the initial conditions are θ_1 is -90 degrees, θ_2 is 45 degrees, $\dot{\theta}_1$ is 0, $\dot{\theta}_2$ is 0 ok. So, what does it look like? It is like a two-degree of freedom robot which is hanging down below so, it is hanging down below with minus 90, but the 2nd link is displaced by 45 degrees in the vertical direction plus 45 degrees, ok.

And then, let us assume that the τ_1 and τ_2 are 0. So, what is it? It is like a double pendulum. So, those are few who have done any course in dynamics, you know what is a single pendulum which is one link with a bob at the end, but you can also have two links connected at the joint and this is called as a double pendulum. So, this planar 2R with τ_1 and τ_2 equal to 0 is nothing but a double pendulum.

(Refer Slide Time: 22:05)

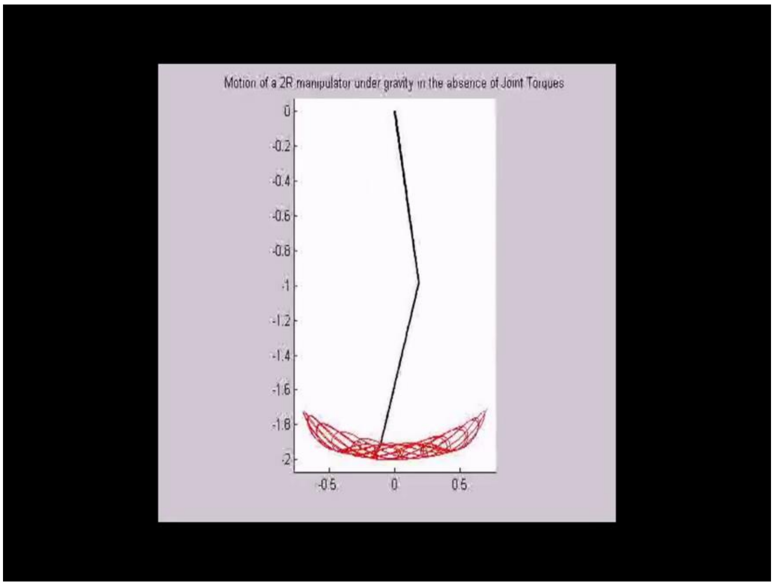


So, we can solve the equations of motion for this planar 2R robot and we can plot θ_1 and θ_2 as a function of time as I said, we do it for 10 second. So, you can see the plot looks quite interesting even the θ_2 is interesting, it is not exactly at the same frequency ok.

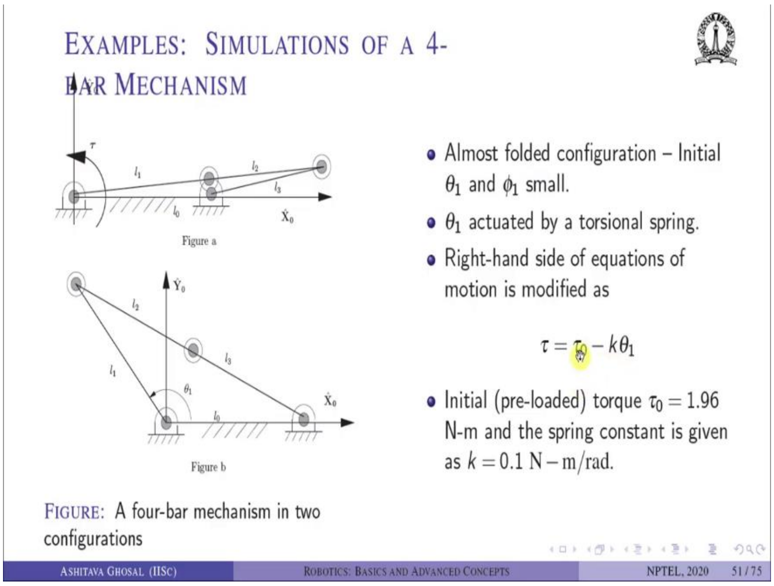
We can also plot x and y as a function of time, and we can see that the tip of this robot which is starting from here, it will go down, it will come up and it does all kinds of strange motions ok. So, what is the first important thing that the motion of link 2 causes a motion of link 1 ok.

So, remember the 1st link is hanging vertically down and the 2nd link is displaced by 45 degrees and there are no torques which are acting on this link. So, intuitively, one might think that the only the 2nd link will oscillate, but that is not true the motion of the 2nd link causes the motion of the 1st link and the path traced by the tip is quite complicated.

(Refer Slide Time: 23:20)



(Refer Slide Time: 23:47)



Let us look at at the example of the 4-bar mechanism. So, we will start with the 4-bar mechanism which is almost completely folded ok. So, this comes from one of these problems, which we had done earlier where a mechanism was initially folded and then, you actuate one of the joints and then, you see how long it takes to unfold or to deploy.

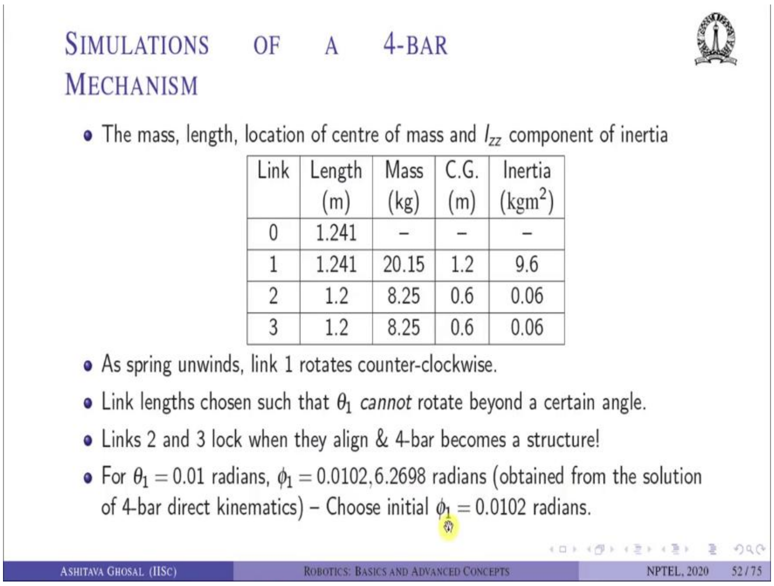
So, this is l_1, l_2, l_0 and l_3, this is the 4-bar mechanism, there is a torque which is acting and since it is almost folded configuration, this θ_1 and ϕ_1 are very small ok.

This θ_1 is actuated by a spring, there is actually no motor, but there is a spring and the right-hand side of the equation which is the torque, τ_1 corresponding to θ_1 is given by some $\tau_0 - k\theta_1$.

So, basically what is happening is there is a free torque or an initial torque which is τ_0 which is 1.96 Newton meter, k was chosen as 0.1 Newton meter per radian and as θ_1 increases, the torque which the spring applies is reducing ok. So, I want to see what happens when you have this 4-bar mechanism, and you apply this torques on the spring.

So, what you can see is in the way this links are drawn, after a while the second and the third link will become straight ok, they will become a line and then, this θ_1 cannot increase any more. So, this is called as the deployed configuration ok, the link lengths are chosen; such that it will not rotate fully, θ_1 will not rotate fully.

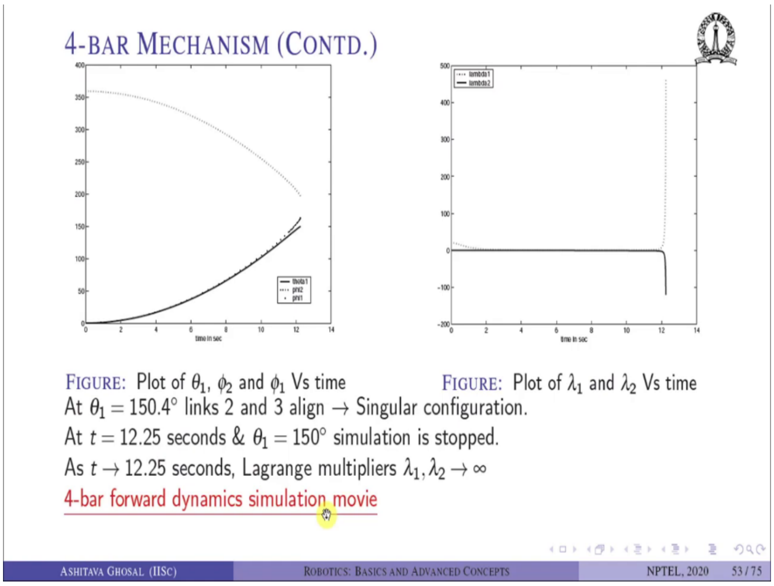
(Refer Slide Time: 25:42)



So, what are the some of the questions that we can ask ok? First let us look at what are some of the dimensions and mass and inertia properties of the link. So, this is chosen in this form 1.241, CG is at 1.2 and so on, inertia is 1st link is very large, the others are very small. So, as a spring unwinds, link 1 rotates counter-clockwise as I showed you. Link lengths are chosen such that θ_1 cannot rotate beyond the certain angle ok.

So, link 2 and 3 locks when they align, and 4-bar becomes a structure. So, after some angle, link 2 and 3 are parallel and there is a mechanism which will lock that joint. So, we initially start with some very small angle $\theta_1 = 0.01$ radians ok, ϕ_1 is this obtained from the solution of the 4-bar kinematics and we choose some initial value of ϕ_1 so, we cannot start with 0 and 0 because that is a singular configuration, and the integration will not start ok.

(Refer Slide Time: 26:47)



So, what are some of the questions that we can ask? I want to know at what angle it locks number 1 and more importantly how much time it takes to lock for given mass, length, geometry, spring constant and everything. So, we can simulate this 4-bar mechanism, we can plot θ_1, ϕ_1 and ϕ_2. So, θ_1 is this dark line, ϕ_2 is this other line and this is ϕ_1 ok.

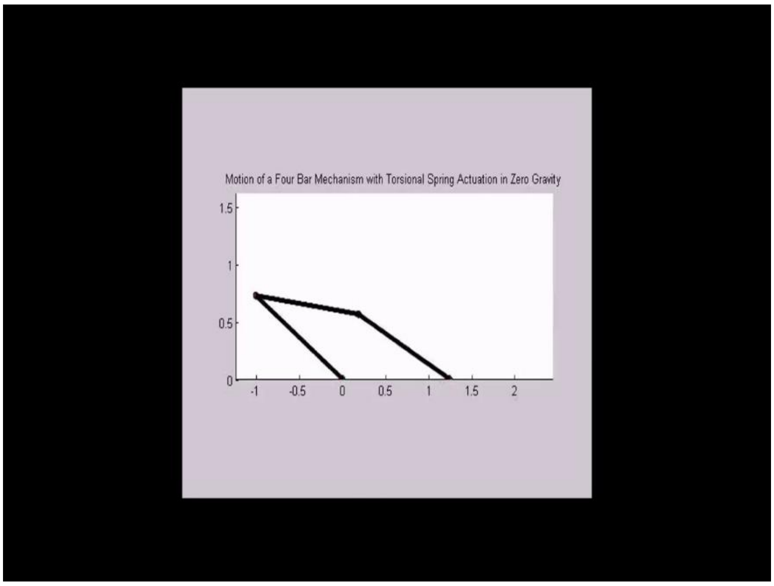
So, what you can see is around 150.4 degrees, links 2 and 3 align. So, this is a singular configuration, for this we do not need to solve the equations of motion, we can just do geometry, you can draw the triangle and measure the angle. So, at $t = 12.25$ seconds, θ_1 is 150 degrees and the simulation is stopped why?

At $t = 12.25$, you can see that the Lagrange multipliers are going off to infinity ok. So, remember in a 4-bar mechanism, we introduce this Lagrange multipliers for the loop closure constraints and then, we solve for the Lagrange multipliers and then, we substitute back in the equations of motion and then, solve for the equations of motion numerically.

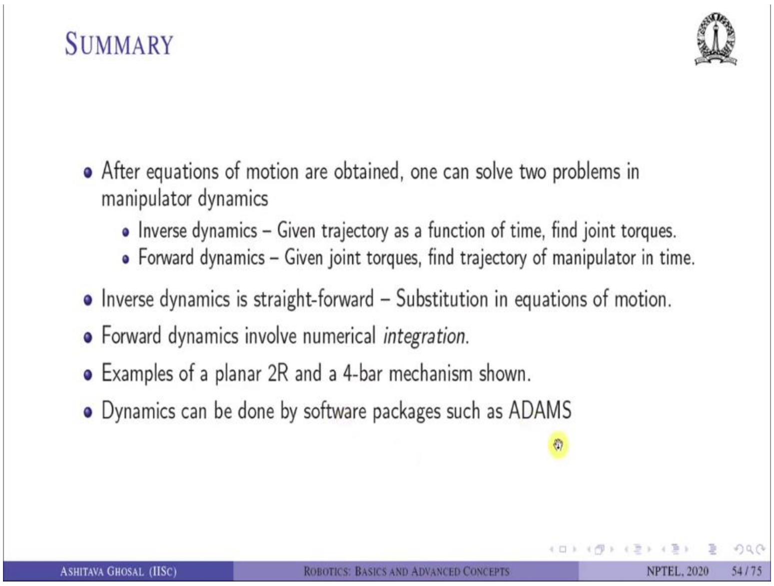
So, if the Lagrange multipliers are going off to infinity, there is nothing much we can do. So, and it turns out that this is around 12.25 seconds is when the two links, link 2 and link

3 are becoming aligned and it locks ok. So, later on I will show you this 4-bar forward dynamic simulation movie.

(Refer Slide Time: 28:37)



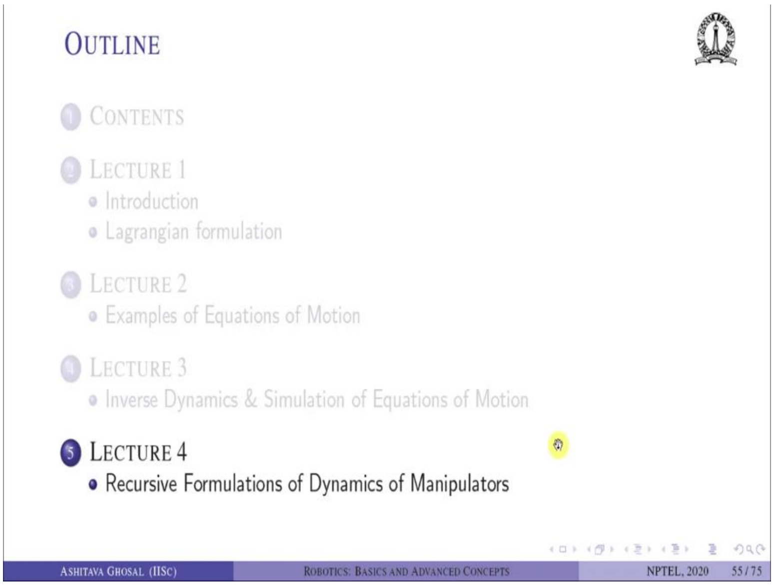
(Refer Slide Time: 28:46)



In summary, after the equations of motions are obtained, one can solve two problems in manipulated dynamics, first is inverse dynamics which is given the trajectory as a function of time, find joint torques. Then, we have forward dynamics or direct dynamics given the joint torques find the trajectory of the manipulator in time. Inverse dynamics is straight-forward, it just requires substitution of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ in the equations of motion.

Forward dynamics involves numerical integration and I have shown you examples of a 2R planar manipulator and a 4-bar mechanism and I showed you how we can get the forward dynamics, how we can simulate the equations of motion of these two examples. The dynamics can also be done by software packages such as ADAMS ok. So, in ADAMS, we can also simulate the mechanism of the manipulator, which is input to ADAMS ok.

(Refer Slide Time: 29:55)



So, with this, we will stop this lecture which delt with inverse dynamics and simulation of equations of motion. In the next lecture, we will look at the more abstract way of finding the equations of motion efficiently ok, this is called as a Recursive Formulation of Dynamics of Manipulators.