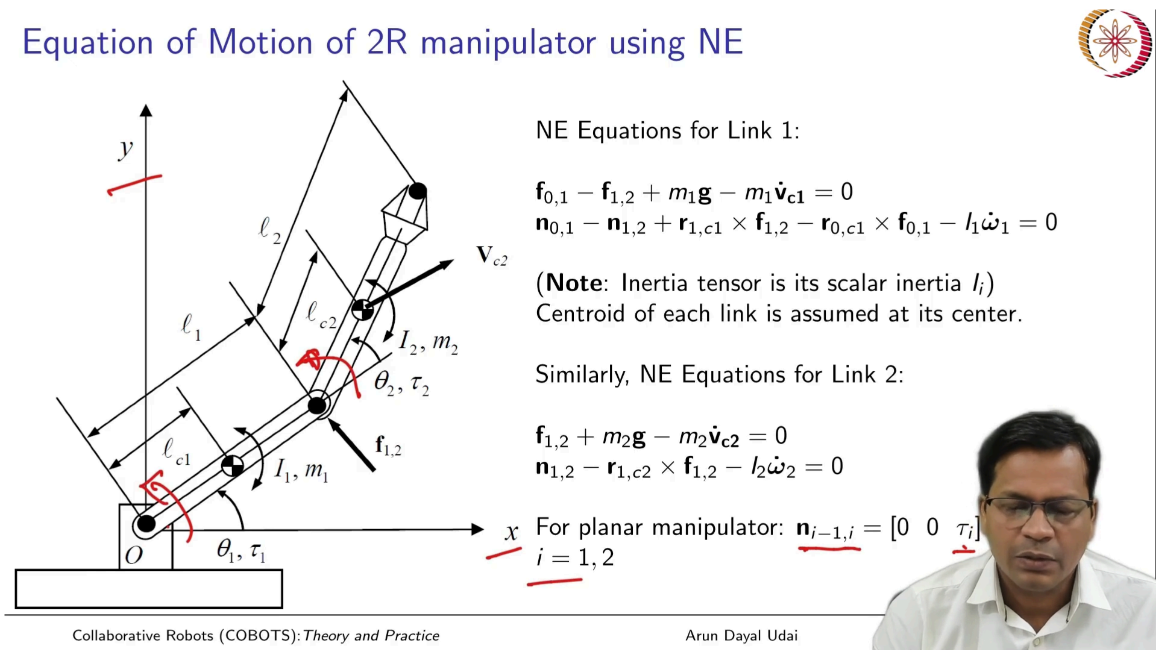
NPTEL Online Certification Courses
COLLABORATIVE ROBOTS (COBOTS): THEORY AND PRACTICE
Dr Arun Dayal Udai
Department of Mechanical Engineering
Indian Institute of Technology (ISM) Dhanbad
Week: 06
Lecture: 27

Dynamic Equation of Motion of a Two-Link Manipulator using Newton-Euler (NE) Approach

Welcome to the fourth lecture of the course, which is on the Dynamic Equation of Motion of a Two-Link Manipulator using the Newton-Euler Approach. So, let us begin.



This is our two-link planar arm that is shown here with the x-axis and y-axis shown here. The first link is attached here. The second link is attached here at the tip of the first link, and this is your end effector. f1,2 is the interaction force between the links, link 1 and link 2. Similarly, there would be f0,1, which is between the ground and the ground-fixed frame and link 1, and this is your f from the external force that is not considered here, which is considered as 0. Vc2 is the centre of mass velocity. So, these are the centre of mass locations. So, if I assume the links to be homogeneous, then it resides exactly at the centre. So, l_{c1} and l_{c2} are the distances of the centre of mass from one end of it. l1 and l2 are the link lengths. Theta 1 and theta 2 are the joint angles, and the corresponding joint torques are tau 1 and tau 2. m1 and m2 are their masses, and I1 and I2 are the moments of inertia about the centre of mass of those links.

So, forming the Newton-Euler equation here for the first link first. So, the sum of all the external forces on the link should equate to the mass into the acceleration part of it. So, that is what it is very trivial and simple.

$$f_{0,1} - f_{1,2} + m_1 g - m_1 \dot{v}_{c1} = 0$$

f0,1 is the force from this direction on this link. f1,2 is the interaction force between the links, link 1 and link 2, and m_1g is the gravitational force on this link. So, the sum of all the forces, if it is unbalanced, creates this mass into acceleration, that is vc1 dot, that is the acceleration. So, when we bring it to all one side, it should be equate to 0.

$$\mathbf{n}_{0,1} - \mathbf{n}_{1,2} + \mathbf{r}_{1,c1} \times \mathbf{f}_{1,2} - \mathbf{r}_{0,c1} \times \mathbf{f}_{0,1} - I_1 \dot{\boldsymbol{\omega}}_1 = 0$$

Similarly, the moment balance equation can be n01, which comes from outside, n12 that, from between the links and the forces which were there that, will create a moment at the centre of mass locations. So, f1,2 into r1, c1, so that is this. Similarly, f0,1, which is here from outside that is from the ground fixed frame to the link 1 f0, 1 cross r0, c1. So, that is this distance here that is the vector here, and all these should now equate to the moment of inertia into angular extraction, and when it is brought to the same side, it is equating to 0. So, this is Newton Euler equation for the link 1.

So, note here inertia tensor is a scalar quantity I_i, I1 and I2. The Centroid of each link is assumed at its center.

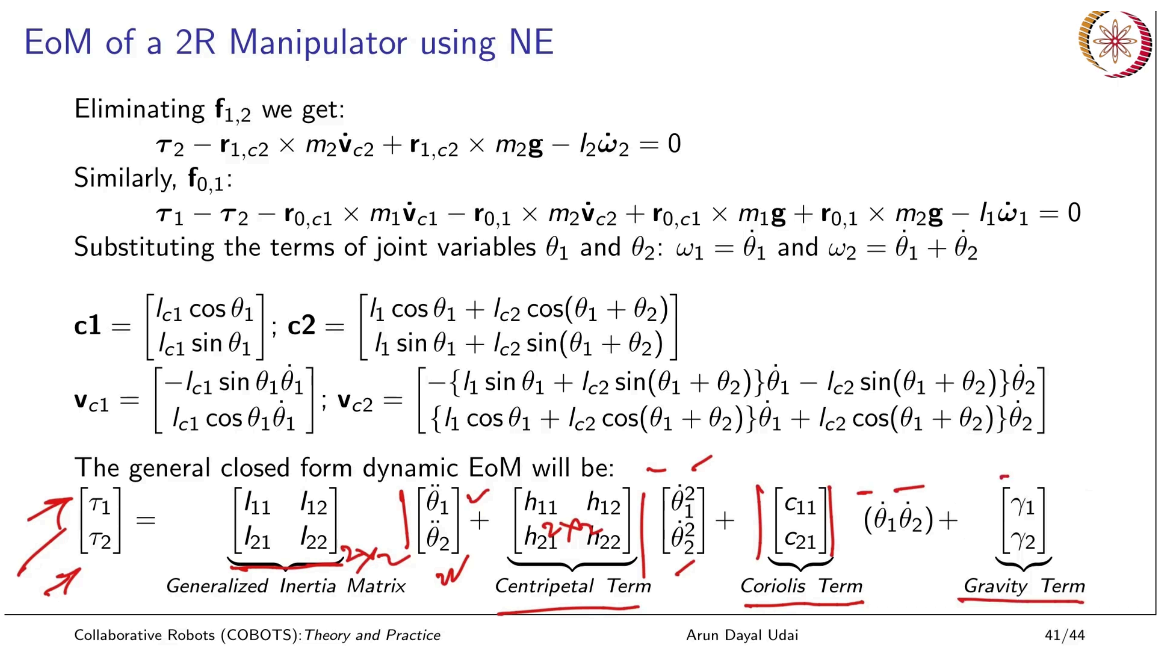
Similarly, Newton Euler equation for link 2 can be written as this.

$$\mathbf{f}_{1,2} + m_2 \mathbf{g} - m_2 \dot{\mathbf{v}}_{\mathbf{c2}} = 0$$
$$\mathbf{n}_{1,2} - \mathbf{r}_{1,c2} \times \mathbf{f}_{1,2} - I_2 \dot{\boldsymbol{\omega}}_2 = 0$$

So, it is f1,2 that is here. There is no external force that acts on it, and m2g is the force due to gravity acting at the centre of mass location, so m2g. So, the sum of all three forces, if it is in motion, should equate to m2vc2 dot, that is, mass into acceleration. So, this forms your first equation.

The second equation is the moment balance equation. Here, you have a moment coming from here, a moment coming from here. This is 0, and this has some interaction moment, which is there, okay? That is n12, which is visible here. The forces will create some moment as well, so that can be like this: f12 into r1, c2, so that is the component here, and that's all. So, there is no other moment that is there. That should create a moment of inertia into the omega 2 dot, that is angular acceleration. So, that should now equate to 0. So, this is the pair of equations which is for link 2.

So, using these two, we will try to move ahead and extract the unknowns that are there. So, for a planar system, n i minus 1 i may be written as 0, 0, tau1 transpose. As you know, it is along the z-direction, which is perpendicular to this plane. x, y, and z is perpendicular to this plane. So, that is here. So, 0, 0, tau1 transpose. So, tau1 and tau2 both are here. So, one of them is here, and the other one is here for both the links. Now, let us use these two pairs.



So, now eliminating f1, 2, we get this, that is from the equation pair. For this one, I have extracted f12 and substituted it here; or otherwise, both are the same, and I got to tau2.

$$\tau_2 - \mathbf{r}_{1,c2} \times m_2\dot{\mathbf{v}}_{c2} + \mathbf{r}_{1,c2} \times m_2\mathbf{g} - I_2\dot{\boldsymbol{\omega}}_2 = 0$$

I could directly obtain tau2 and, similarly, f01 from the first pair of equations, that is, this one. f01 and f12 come from here, that goes here and f01, which is here. So, all those can be extracted now, and you get to this.

$$\tau_1 - \tau_2 - \mathbf{r}_{0,c1} \times m_1\dot{\mathbf{v}}_{c1} - \mathbf{r}_{0,1} \times m_2\dot{\mathbf{v}}_{c2} + \mathbf{r}_{0,c1} \times m_1\mathbf{g} + \mathbf{r}_{0,1} \times m_2\mathbf{g} - I_1\dot{\boldsymbol{\omega}}_1 = 0$$

That is, tau1 minus tau2 is equal to the whole of this. So, both tau1 and tau2 can now be solved using these two equations. And now, I can also obtain C1. This is the mass centre location of the first link. C2 is the mass centre location for the second link. So, this is quite trivial. That is in frame 1. These are in frame 1. So, that is why it is so long. So, C1 and C2, I have used this also when I substituted in the root equations, and similarly, Vc1 and Vc2 are just derivatives of this. The C1 dot is Vc1, C2 dot is Vc2, which can be quickly derived like this.

$$\mathbf{v}_{c1} = \begin{bmatrix} -l_{c1}\sin\theta_1\dot{\theta}_1 \\ l_{c1}\cos\theta_1\dot{\theta}_1 \end{bmatrix}; \ \mathbf{v}_{c2} = \begin{bmatrix} -\{l_1\sin\theta_1 + l_{c2}\sin(\theta_1+\theta_2)\}\dot{\theta}_1 - l_{c2}\sin(\theta_1+\theta_2)\}\dot{\theta}_2 \\ \{l_1\cos\theta_1 + l_{c2}\cos(\theta_1+\theta_2)\}\dot{\theta}_1 + l_{c2}\cos(\theta_1+\theta_2)\}\dot{\theta}_2 \end{bmatrix}$$

So, substituting all these into this equation and solving for tau 1 and tau 2, we can write the general closed-form dynamic equation of motion like this.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \underbrace{\begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix}}_{Generalized\ Inertia\ Matrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}}_{Centripetal\ Term} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{bmatrix} + \underbrace{\begin{bmatrix} c_{11} \\ c_{21} \end{bmatrix}}_{Coriolis\ Term} (\dot{\theta}_1\dot{\theta}_2) + \underbrace{\begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}}_{Gravity\ Term}$$

So, where this is a vector-matrix form, it is packed in a column format. Tau 1 and Tau 2 will come like this. The generalised inertia matrix is a 2x2 matrix formed out of the terms containing the angular acceleration, that is, theta 1 double dot and theta 2 double dot. Both of them come as a column here. Similarly, the centripetal term is again a 2x2 matrix, and it contains all the angular velocity squared terms, that is, theta 1 dot squared and theta 2 dot squared. So, that is arranged here. This, again, is 2x2, and the Coriolis term, that is, the product of angular velocity terms, is arranged like this. The last term is the gravitational term. That is the gravitational torque, the torque due to gravity. Those terms are arranged here. So, you can express tau 1 and tau 2, which are extracted out of these two equations, and by substituting all these, you can write that in this form.

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \underbrace{\begin{bmatrix} I_{11} & I_{12}\\ I_{21} & I_{22}\end{bmatrix}}_{Generalized\ Inertia\ Matrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} + \underbrace{\begin{bmatrix}h_{11} & h_{12}\\ h_{21} & h_{22}\end{bmatrix}}_{Centripetal\ Term}\begin{bmatrix}\dot{\theta}_1^2\\ \dot{\theta}_2^2\end{bmatrix} + \underbrace{\begin{bmatrix}c_{11}\\ c_{21}\end{bmatrix}}_{Coriolis\ Term}(\dot{\theta}_1\dot{\theta}_2) + \underbrace{\begin{bmatrix}\gamma_1\\ \gamma_2\end{bmatrix}}_{Gravity\ Term}$$

EoM of a 2R Manipulator using NE

Terms of Generalized Inertia Matrix (GIM) are:

$I_{11} = m_1 l_{c1}^2 + I_1 + m_2(l_1^2 + l_2^2 + 2l_1 l_2 \cos\theta_2) + I_2$ ← Combined MI with joint 2 is immobilized

$I_{12} = I_{21} = m_2(l_{c2}^2 + l_2 l_{c2}\cos\theta_2) + I_2 I_{22} = m_2 l_{c2}^2 + I_2$ ← Interaction between two joints

$I_{22} = m_2 l_{c2}^2 + I_2$ ← MI with link 2 about joint 2

Centripetal Terms are:

$h_{11} = 0,\ h_{12} = -m_2 l_1 l_{c2}\sin\theta_2$

$h_{21} = m_2 l_1 l_{c2}\sin\theta_2,\ h_{22} = 0$

Coriolis Terms are:

$c_{11} = -2m_2 l_1 l_{c2}\sin\theta_2$ and $c_{21} = 0$

Gravity Terms are:

$\gamma_1 = m_1 g l_{c1}\cos\theta_1 + m_2 g\{l_1\cos\theta_1 + l_{c2}\cos(\theta_1+\theta_2)\}$

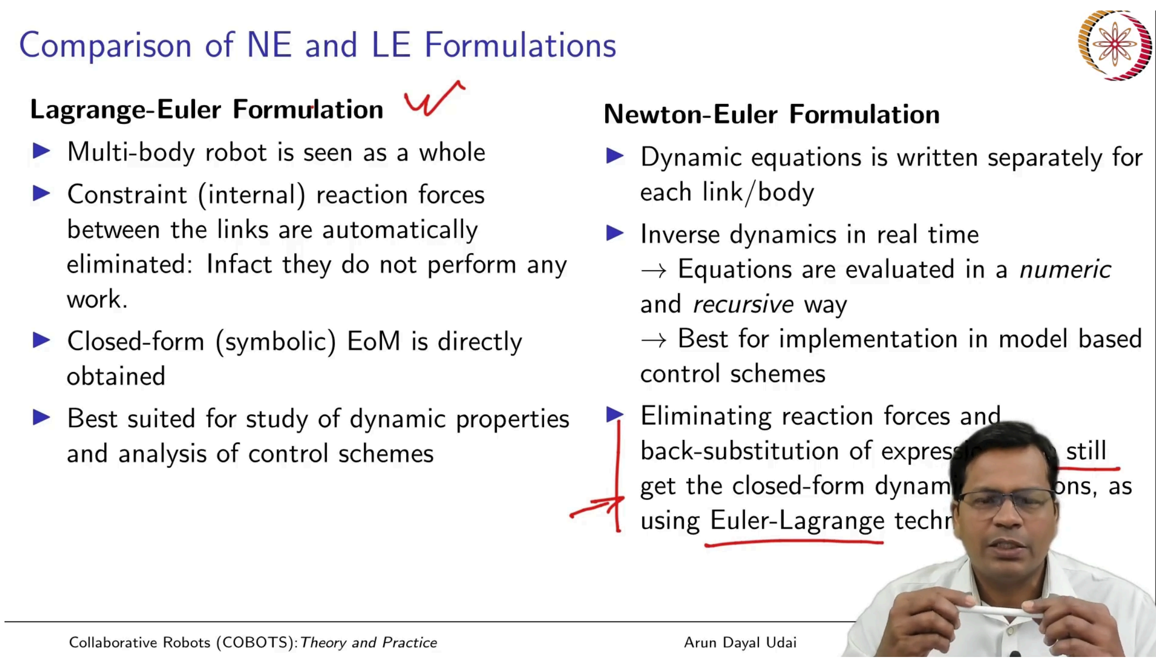
$\gamma_2 = m_2 g l_{c2}\cos(\theta_1+\theta_2)$

Collaborative Robots (COBOTS): *Theory and Practice* Arun Dayal Udai

So, by doing so, you directly get the terms of the generalised inertia matrix like this: I11, I21, and I12 (both are the same) and I22. So, what is the significance of this? Let us say you have your link, which is like this—the first link and the second link. If I freeze this second link over here at an angle theta 2, the torque that comes here is because of the combined motion of links 1 and 2 together in this configuration. So, that is the first term of your matrix here in the generalised inertia matrix. This is the combined moment of inertia of joint 2, which is immobilised. It is not moving with respect to link one. That is why this is such a big term. Again, the term here, I22 is very simple, okay? You just have the moment of inertia, which was given about the mass centre location, that is, I2. We just have transferred it from here to the axis of rotation, which is over here, and these two are interactions between the joints. They are the product moment of inertia, and that is given by this. So, this is the physical significance of having these terms in place. So, the diagonal matrices are the principal moment of inertia, and these are the product moment of inertia.

So, likewise, you can continue if there is a third link. So, this will be link 3 alone. This will be due to links 2 and 3, and this will be due to links 1, 2, and 3, and the remaining will be the product moment of inertia, the remaining part.

So now, the centripetal terms are here: h11, h21, h22, h12. All four are different. They are here. Coriolis terms are here: c11 and c21, that is here, and gravity terms are directly here. So, what does this represent? Basically, this is directly the gravitational torque, irrespective of whether the robot is in motion or not. This term is always there. Even if it is not moving, this is the amount of torque that must be supplied continuously at the joints. So, this is what we obtained while we were doing gravity compensation in robot statics. So, this is exactly the same term. You just set the angular acceleration and velocity term velocity term equal to 0 in the equation of motion, and you get these terms. So, you are left with gamma 1 and gamma 2. So, this is the gravity term.

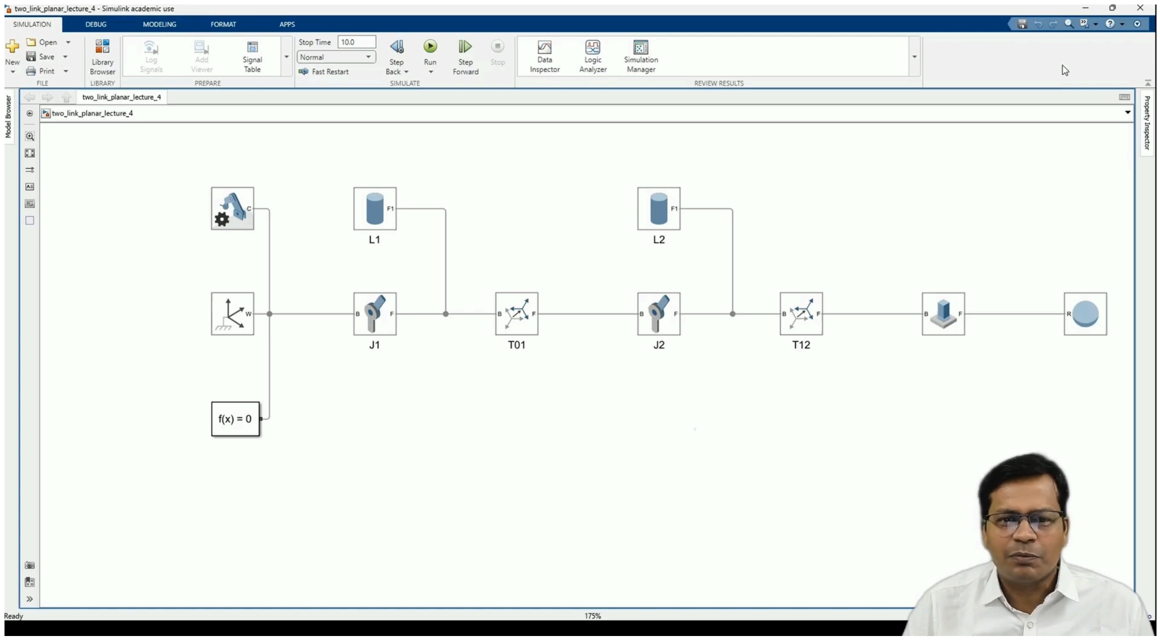


So now, let us compare the Newton-Euler and Lagrange-Euler formulations. So, what does the Lagrange-Euler formulation do? It is a multi-body robot as a whole. What did you do? You basically calculated the Lagrangian for the entire system together, then took partial derivatives, applied the LE formulation, and derived the torque. So, the constraints that are there, that is, the internal reaction forces f01 and f12, which are the reaction forces between the links, are automatically eliminated. In fact, they don't do any valuable work. So, that is the reason it gets automatically eliminated.

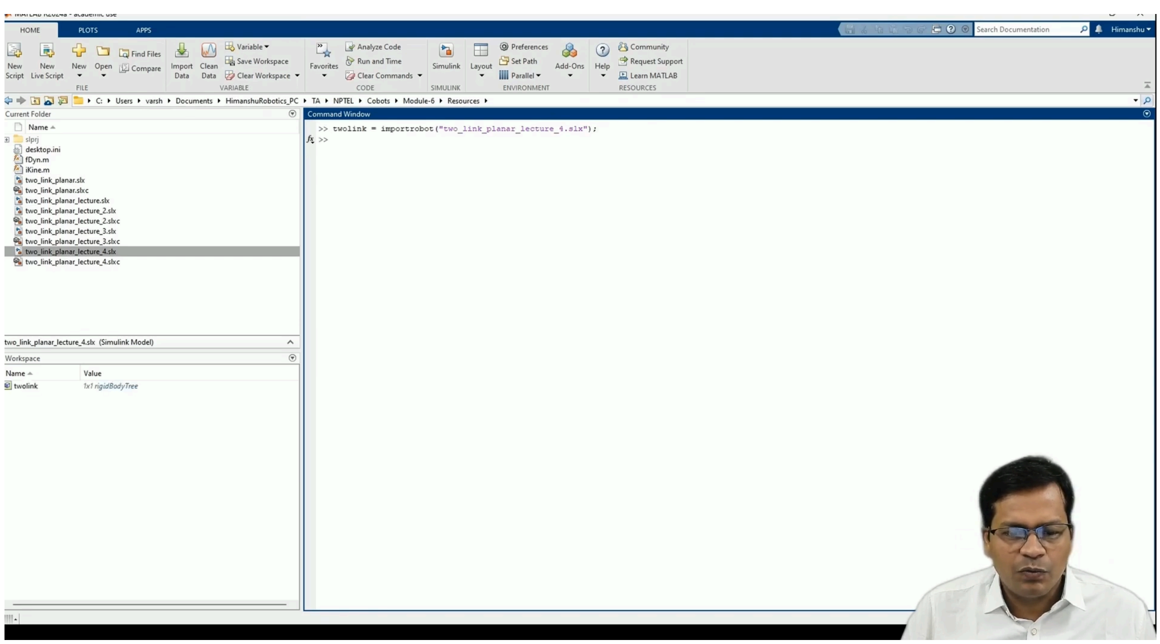
The closed-form symbolic equation of motion is directly obtained without solving them. Without solving them, you don't have pairs of any equation for each of the links. Solving them together gives you something, but in this case, you can directly obtain that.

It is best suited for the study of dynamic properties and analysis of control schemes. Whereas in the case of Newton-Euler formulation, the dynamic equation is written separately for each link or body, you saw that just now.

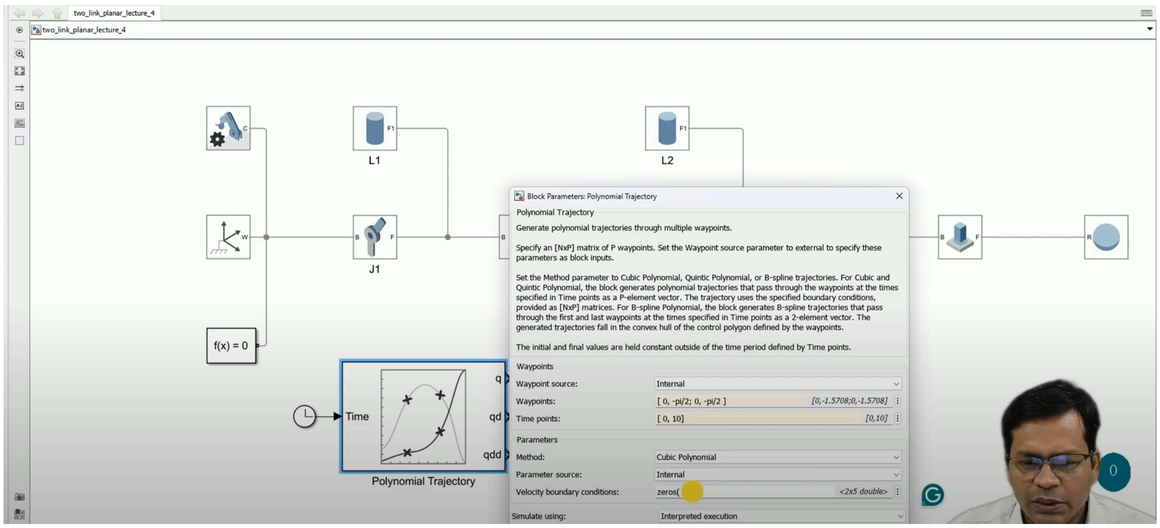
Inverse dynamics in real-time: equations are evaluated in a numeric and recursive way. Any approach allows doing recursive formulation also, so that is the reason they can be done in no time. It can be done in a recursive way and very fast. Best for implementation in model-based control schemes. So, in model-based control schemes, you tend to calculate the torques during the runtime of the robot. For a given trajectory, you keep calculating the torque and feed it to the robot. So, you need to do dynamic solutions online. So, you need to be very, very fast. That is the reason Newton-Euler supports that very easily. So, it is best implemented in model-based control schemes and eliminates reaction forces and back-substitution of expression, and we still get the closed-form dynamic equation of motion as in the case of the Euler-Lagrange solution. So, we get the same result here, but additionally, we also get the interaction forces. These interaction forces are very, very helpful when we try to do any FEM analysis while designing any link, let us say. So, you know the forces and moments from both sides; you know the gravitational forces. So, all the link forces are known, internal forces as well as external forces of the link. So, that is the reason while designing the link, it becomes very, very useful.



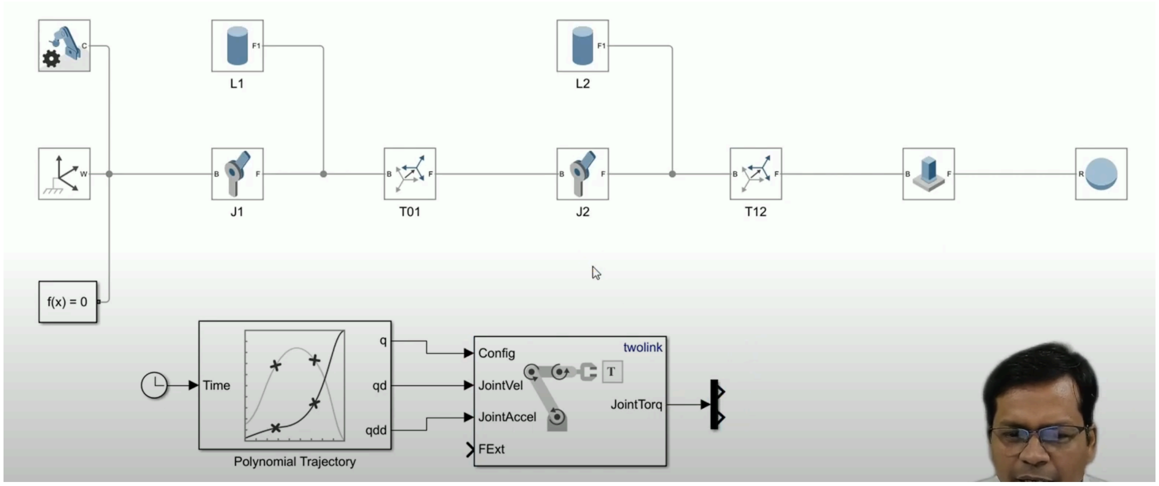
Now, let us see if we can extend the simulation blocks that we have prepared earlier if it can do the inverse dynamics simulation also. Inverse dynamics basically here I would mean you have a trajectory in hand, and corresponding to that, you calculate the joint torques using the equation of motion and feed it to the robot to see if it follows the planned trajectory.



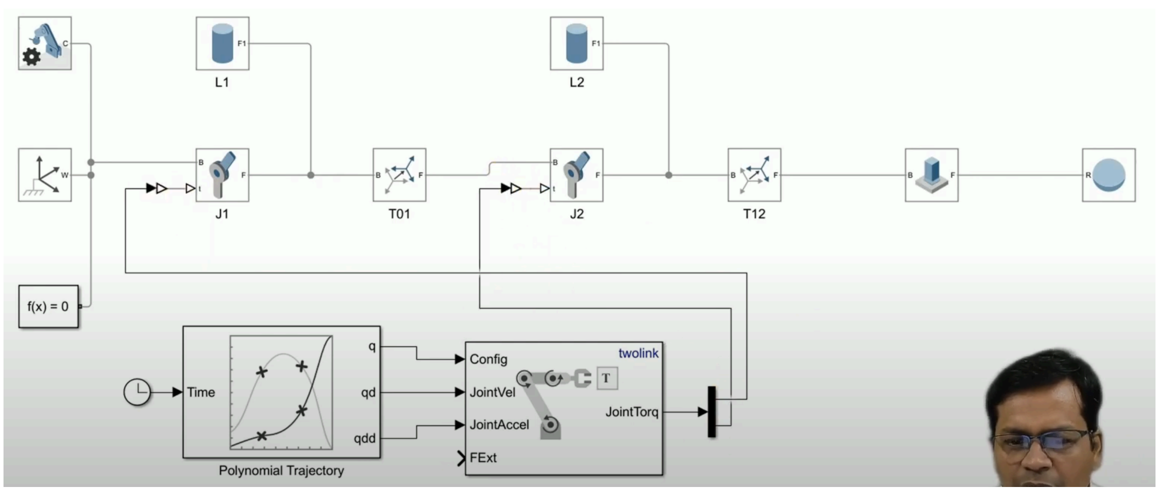
So, yes, let us continue. So, what am I doing? I am using the same simulation model. This time, again, I will just create the object by importing the simulation model of the two-link manipulator. What have we done earlier? So, it is the name of the simulation file, which is 'two-link planar lecture 4.slx.' Import that. It creates the object in the workspace. This is a rigid body tree, basically.



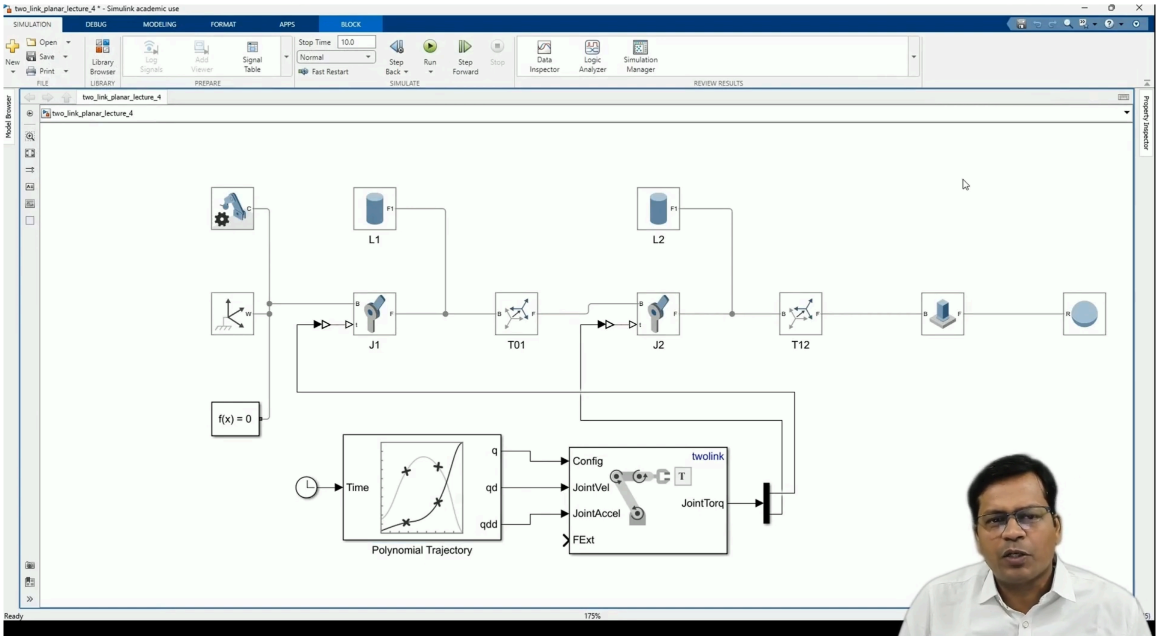
So, now I will insert the polynomial trajectory block. This takes input as time simulation time and calculates the trajectory. So, these are the waypoints for joint 1. It is 0 to pi by 2. Using cubic polynomial trajectory, you can see the parameters are: method is a cubic polynomial. For the second joint, it is 0 to π/2 once again. This is the way for the time points 0 to 10. So, from 0 to 10 seconds, which is the total simulation time, it will go from 0 to π/2, the first joint, as well as the second joint go from 0 to π/2. The velocity boundary condition it starts with zero velocity. So, you see, it gives me joint rates as well as joint angles. So, it is calculating polynomial trajectory as joint angle, joint rate, and joint acceleration.



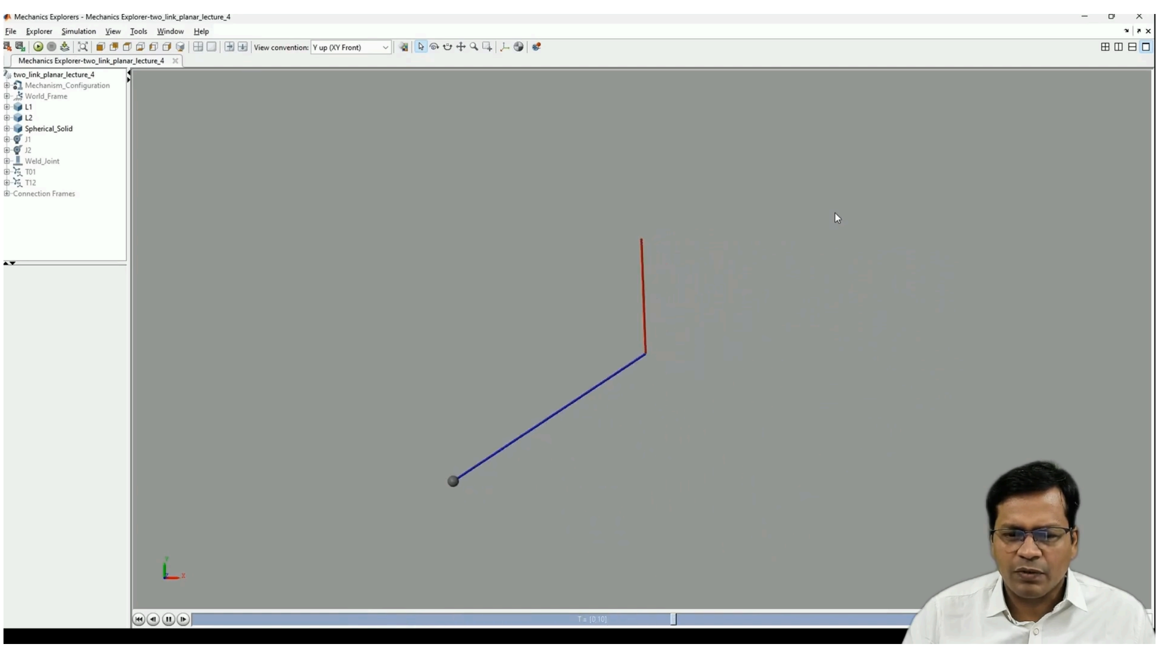
Now, I will perform robot inverse dynamics. It takes input for the configuration, which is the joint configuration, that is, the joint angle, joint velocity, and joint acceleration. It will use the equation of motion of this robot. So, I will make it take the data of this robot from the tool link tree, which we have created in the workspace. So, joint torque is calculated here, and again, I will transfer these joint torques to the joints. So, this time, the joint torque is provided by the input, and the motion is automatically computed.



Again, simulating the physical system data converter. These are physical values that will go to the Now, I will run this simulation, and you will see 0 to π/2 for both the joints I have fed, and it is moving in a cubic polynomial trajectory for both joints. So, it goes like that.



These angles can be edited; you can try changing that, also. 0 to 5π/3 for the first one and 0 to 5π/6 for the second one. Run it once again; it creates a torque profile corresponding to the joint angle profile, and we are feeding it to the robot joints and observing the motion. So, exactly, it is following that.



You can keep on changing for multiple values. 3π/4 and π/2. Run it again. 3π/4, and the second one is π/2. The starting and ending time for both of them is 10 seconds. So, you see, we have done dynamic simulation as well using the two-link manipulator system. So, we will complete the dynamics here. We have seen all the algorithms how to compute the dynamic equation of motion and we have seen how to simulate them.

In the next module, we'll start using them also for our force control algorithms. That's all. Thanks a lot.