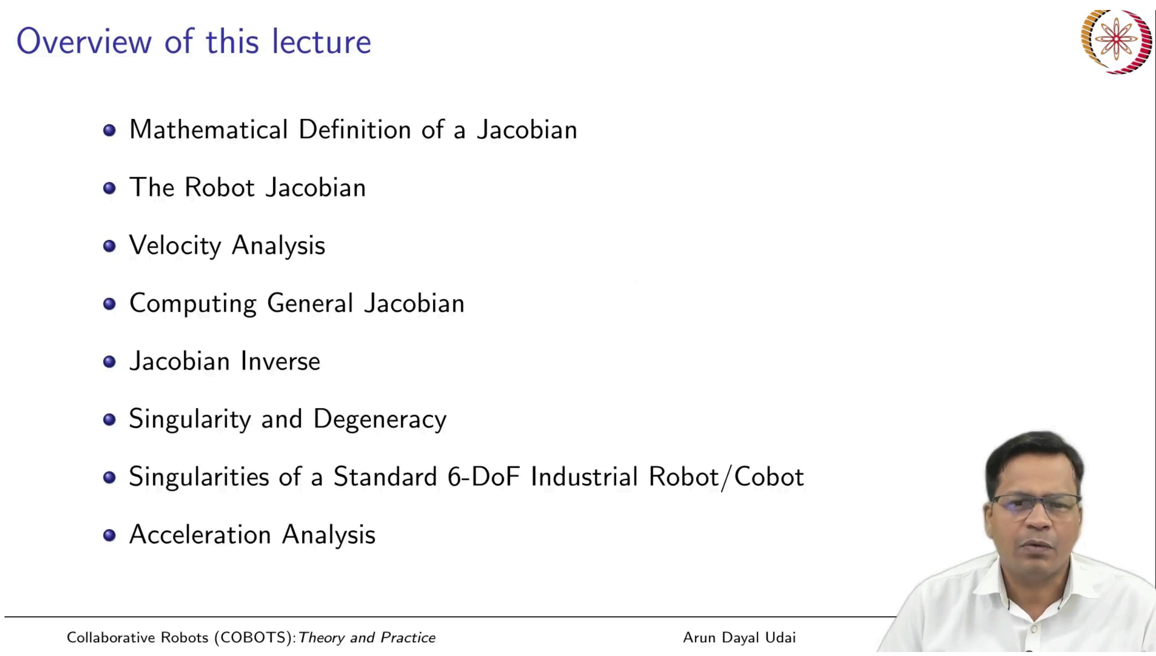
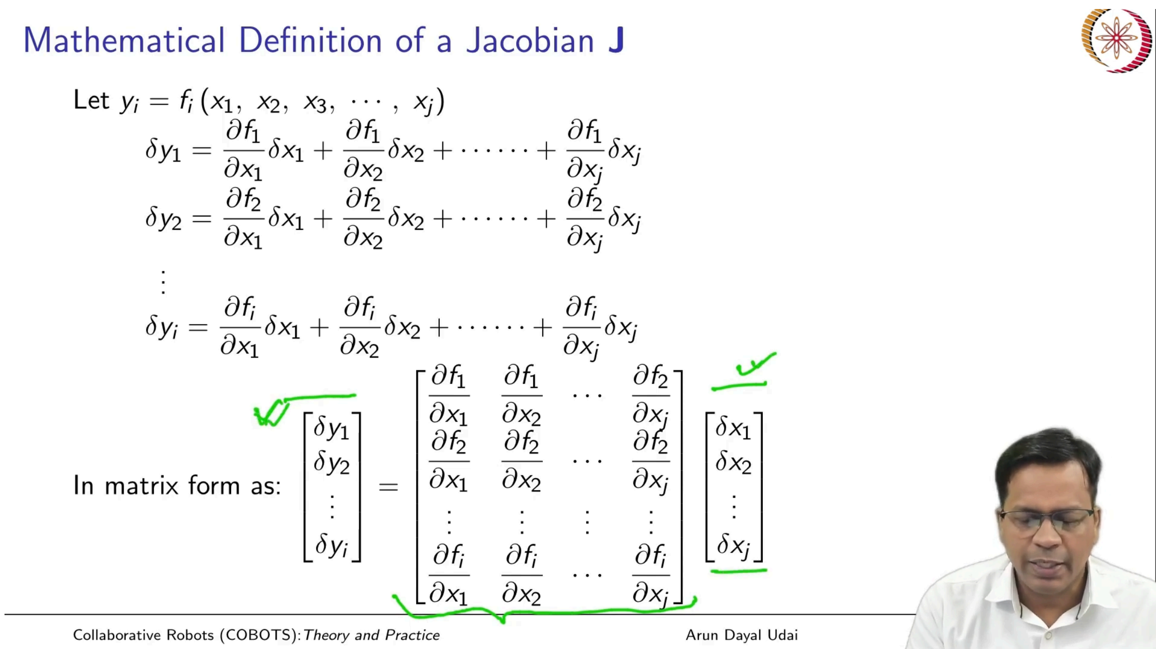
NPTEL Online Certification Courses
COLLABORATIVE ROBOTS (COBOTS): THEORY AND PRACTICE
Dr Arun Dayal Udai
Department of Mechanical Engineering
Indian Institute of Technology (ISM) Dhanbad
Week: 04
Lecture: 18

Differential Motion Analysis: Robot Jacobian, Velocity and Acceleration Analysis



Welcome back to the fourth lecture of the course, Collaborative Robots: Theory and Practice. So, in this lecture, I will discuss the mathematical Jacobian. What is the definition of a Jacobian? The robot Jacobian, velocity analysis, how to compute a general Jacobian, Jacobian inverse, how to do that. Singularity and degeneracy, what it is. Singularities of a standard serial arm, which is a six degrees of freedom, industrial cobot or a robot, will do acceleration analysis as well.

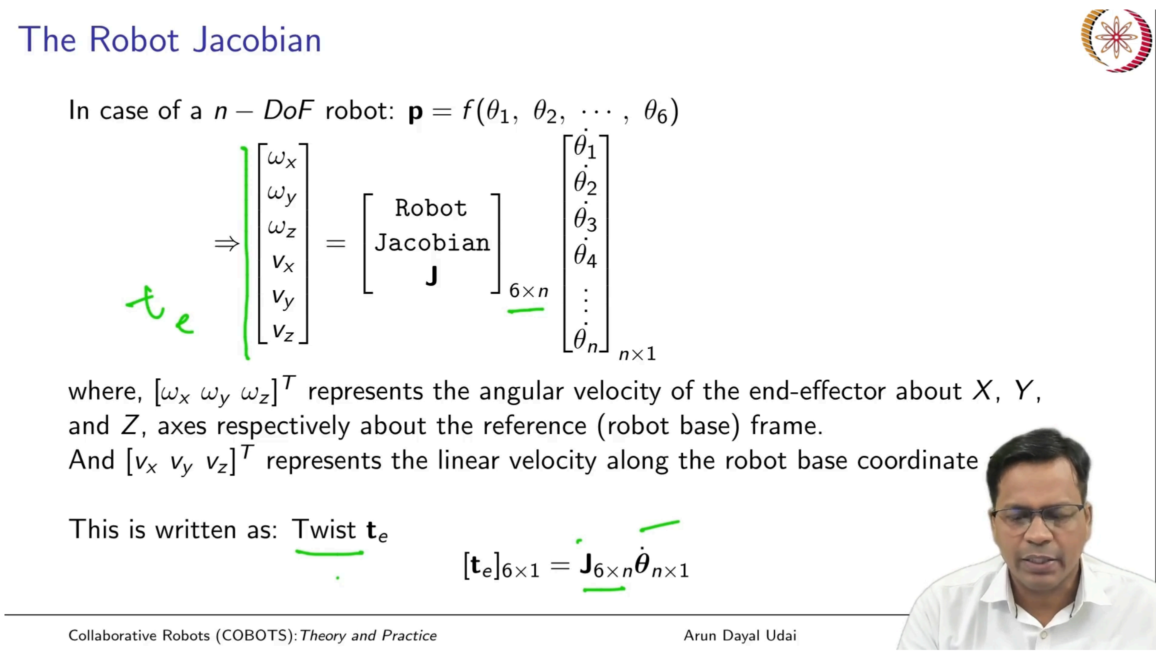


So, let us begin. So, what exactly is a Jacobian (J)? Why are we talking about that here? Because the Jacobian relates two different rates. Let us say you have y, which is a function of x1, x2, x3, x4, and xj. So, these are all the input variables on which the output variable, that is, y, depends. So, this may be expressed like this.

$$\delta y_1 = \frac{\partial f_1}{\partial x_1}\delta x_1 + \frac{\partial f_1}{\partial x_2}\delta x_2 + \cdots\cdots + \frac{\partial f_1}{\partial x_j}\delta x_j$$
$$\delta y_2 = \frac{\partial f_2}{\partial x_1}\delta x_1 + \frac{\partial f_2}{\partial x_2}\delta x_2 + \cdots\cdots + \frac{\partial f_2}{\partial x_j}\delta x_j$$
$$\vdots$$
$$\delta y_i = \frac{\partial f_i}{\partial x_1}\delta x_1 + \frac{\partial f_i}{\partial x_2}\delta x_2 + \cdots\cdots + \frac{\partial f_i}{\partial x_j}\delta x_j$$

Delta y is equal to del f by del x1 delta x1, del f by del x2, delta x2, plus so on till this. Similarly, if there are multiple outputs that are related to the same inputs, that is, x1, x2, x3, x4, and so on. So, they all can be written like this.

If I pack them together in the matrix form, all the output variables are here. All the input variables are here. This is the matrix of partial differentials of f, which relates the input to the output. Basically, input rates to the output rates. So, it relates to two different rates.

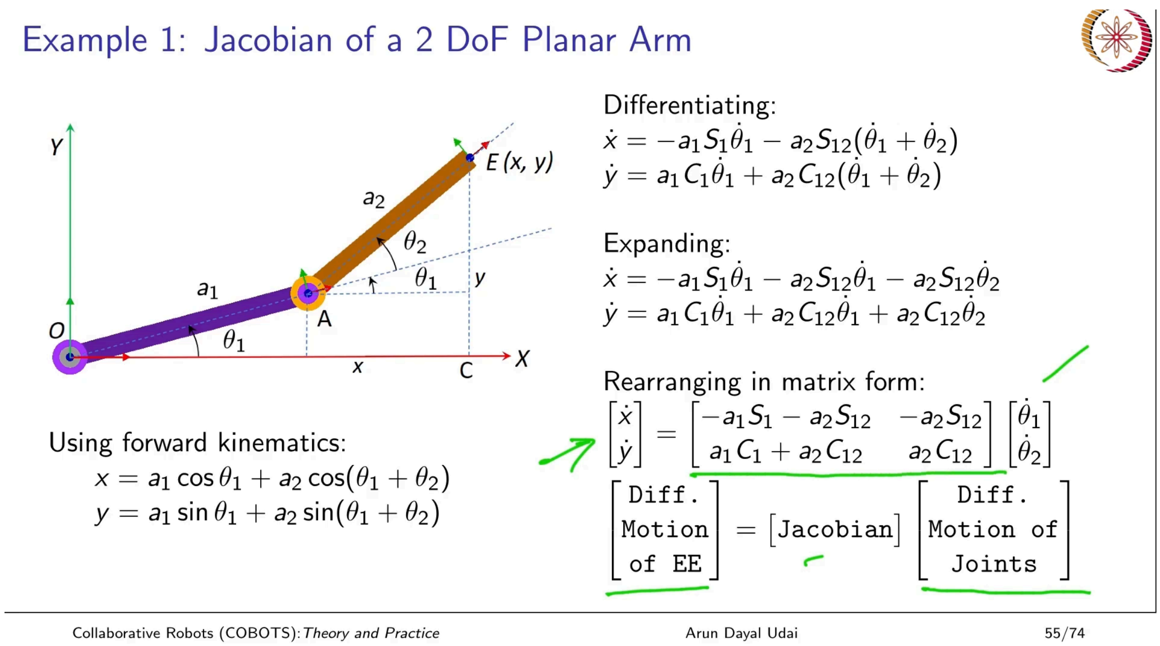


So, in the case of robots, when you have an end effector position, which is a function of the input joint angle. So, the end effector pose of the robot depends on theta 1, theta 2, theta 3, and theta 6 in the case of 6 degrees of freedom robot, or it can be 7 degrees of freedom robot, in which case it will be up to theta 7. So, in that case, the end effector rate means the angular velocities of the end effector and the linear velocities of the end factor that is related using a 6 cross n Jacobian, where n is the degrees of freedom of the robot, and theta 1, theta 2, theta 3, theta 4, and up to theta n are the input joint variables. So, joint variables can be in terms of rotary joint variables or prismatic joint variables. So, where omega x, omega y, omega z represent the angular velocity of the end effector about the x, y, and z axes of the reference frame of the robot, and vx, vy, vz are the linear velocities along the robot base coordinate axes.

So, this is written as a twist. A twist is a vector which is a column vector of this.

$$[\mathbf{t}_e]_{6\times 1} = \mathbf{J}_{6\times n}\dot{\boldsymbol{\theta}}_{n\times 1}$$

So, this is a twist vector of 6 cross 1 dimension, and the Jacobian here would be of 6 cross n dimension, which is the Jacobian matrix that relates the output rates to the input rates, and this is the joint variable. So, that is theta n cross 1. So, this is how the Jacobian is defined for a general robot of n degrees of freedom.



So, now let us calculate the Jacobian for a two-degrees-of-freedom Planar Arm. This is one of the very simplest arms which can be derived by hand calculation also. So, we'll see that now. So, a1 and a2 are the link lengths. Theta 1 and theta 2 are the joint angles. x and y are the end effector coordinates. So, now x and y can be written using forward kinematics, which can be evaluated here also by taking simple projections. So, this is a1 c1, a2 c1 2, a1 s1, a2 s1 2. So, the sum of these two will be x and y. So, this is forward kinematics.

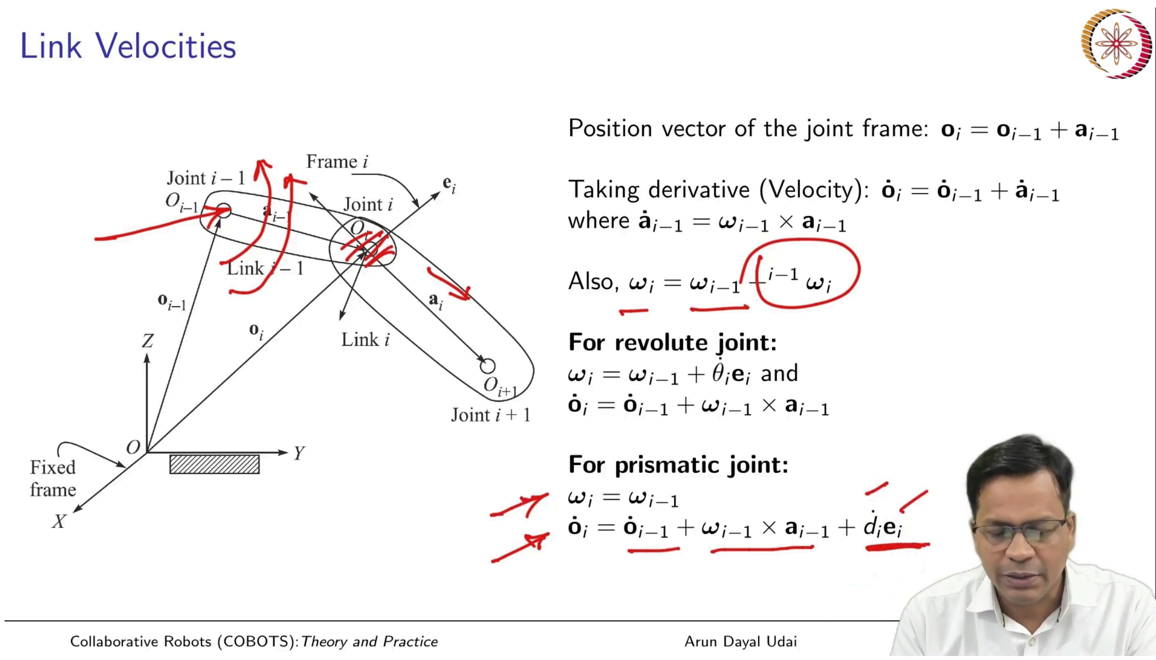
So, if I take the differential of this, taking the derivative on both sides will give me this.

$$\dot{x} = -a_1 S_1 \dot{\theta}_1 - a_2 S_{12}(\dot{\theta}_1 + \dot{\theta}_2)$$
$$\dot{y} = a_1 C_1 \dot{\theta}_1 + a_2 C_{12}(\dot{\theta}_1 + \dot{\theta}_2)$$

If I expand that, I will get this.

$$\dot{x} = -a_1 S_1 \dot{\theta}_1 - a_2 S_{12}\dot{\theta}_1 - a_2 S_{12}\dot{\theta}_2$$
$$\dot{y} = a_1 C_1 \dot{\theta}_1 + a_2 C_{12}\dot{\theta}_1 + a_2 C_{12}\dot{\theta}_2$$

Psi S12 here I would mean sine theta 1 plus theta 2, similarly for cos 12 and S1 would mean sine of theta 1. So, if I rearrange them in the matrix form, it comes to x dot y dot is equal to a 2 cross 2 matrix into theta 1 dot theta 2 dot. Theta 1 dot and theta 2 dot are the angular velocities of the joint, joint rates, and x dot y dot is basically the end effector linear velocities. In the case of a 2R manipulator, it can only move in two linear directions. So, this is your x-dot direction, and this is your y-dot direction. This cannot rotate because you do not have that degree of freedom in this type of robot. So, what is this basically? So, this is the Jacobian. So, you see now it is relating the differential motion of the joints, that is this, to the differential motion of the end effector. And the term which is here is actually the 2x2 matrix, which is nothing but a Jacobian, which relates the joint rates to the end effector rates.



Let us do a velocity analysis now. So, link velocity is first. So, the position vector of the joint frame. So, let us say this is a Link i of link length a_i that terminates with joint i plus 1 and begins with o_i frame, and before that, you have a link i minus 1 that begins with o_i minus 1 frame, and the link length vector here is a_i minus 1. So, this is the configuration of two consecutive links which are attached and floating in space as it is shown. o_i vector is the position vector of the o_i frame, and o_i minus 1 vector is the position vector of the i minus 1th frame, with respect to the base frame, which is O, which is shown here, that is, the fixed frame. So, the position vector of the joint frame is given as o_i, which will be equal to o_i minus 1 plus the link length vector.

$$o_i = o_{i-1} + a_{i-1}$$

So, if I can draw it here, so o_i is equal to o_i minus 1 plus this vector. That is, using the vector triangle, you can quickly obtain it; you can write it like that. So, taking the derivative of this, what do I get? o_i dot is equal to o_i minus 1 dot plus a_i minus 1 dot.

$$\dot{\mathbf{o}}_i = \dot{\mathbf{o}}_{i-1} + \dot{\mathbf{a}}_{i-1}$$

So, what are these terms if you can see them closely? So, how much is a_i minus 1 dot? Let us isolate this link here. So, if this is a_i minus 1, it is rotating about this joint by an angle or velocity omega i minus 1. So, what is the velocity that is associated with it? It is the velocity along this direction. That is because of the joint velocity of this frame oi minus 1 at this frame. So, it rotates like this and creates a velocity like this. So, that is in the tangential direction. So, this is a_i minus 1 dot. Okay. So, because this is not a prismatic joint, it will not go along this direction. So, that is the velocity associated with this, and the angular velocity omega i will be omega i, that is, the angular velocity over here, will be equal to the angular velocity up till here plus the angular velocity of this joint. So, this is the joint angular velocity, that is, the angular velocity of i with reference to i minus 1, and omega i is the angular velocity of frame o_i minus 1 till here and this. So, this can be written as this.

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + {}^{i-1}\boldsymbol{\omega}_i$$

So, now, if I put them in the revolute joint. So, it comes out to be this.

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \dot{\theta}_i \mathbf{e}_i$$

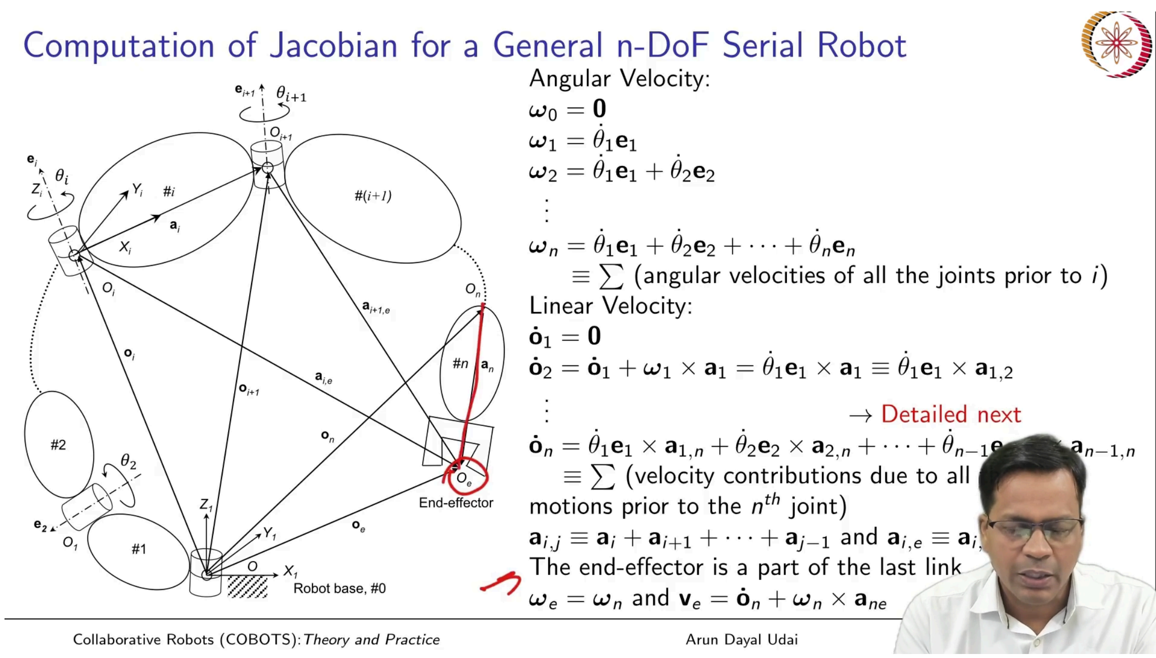
So, omega i is equal to it is exactly this equation, omega i minus 1. So, this is the joint rate, this is the magnitude of the joint rate, and this is the direction which is along the axis of rotation. So, this is equal to this, and now this equation is written here as just I have substituted this to this, and it comes out to be this.

$$\dot{\mathbf{o}}_i = \dot{\mathbf{o}}_{i-1} + \boldsymbol{\omega}_{i-1} \times \mathbf{a}_{i-1}$$

This is for the revolute joint. In the case of the prismatic joint, because the joint itself does not have any angular velocity. So, the next link velocity is exactly equal to the previous link velocity angular velocity, and now you have linear velocity because of three different components.

$$\dot{\mathbf{o}}_i = \dot{\mathbf{o}}_{i-1} + \boldsymbol{\omega}_{i-1} \times \mathbf{a}_{i-1} + \dot{d}_i \mathbf{e}_i$$

The first one is linear velocity over here and linear velocity due to the rotation of this link and linear velocity because of the expansion of the joint. So that is due to the expansion of di, so that is always there in this case. You see, because there is no, this is frozen. So, whatever is the angular velocity of this becomes the angular velocity of this link as well. But this is expanding also. So, if this is expanding, you get what now? You have di dot, which is the rate of change of displacement along the direction of motion. The direction of motion is given by e_i. This is in the case of a prismatic joint. So, these are the link velocities.



Let us see if we can calculate the Jacobian using the vector-matrix approach for a general n degrees of freedom serial robot. Moving on with a similar analogy, the way we did it here, omega 0, angular velocity. So, this is the structure of this robot, a general robot. This is link 1 attached to frame 1, and it is attached to the ground. This is the next link.

This is the next velocity, angular velocity at the joint. Next joint, next link, next joint, next link, so on and so forth, till it terminates at the end effector, which is here. So omega 0, because it is attached to the ground, it is equal to 0. How much is omega 1?

$$\omega_1 = \dot{\Theta}_1 \widehat{e}_i$$

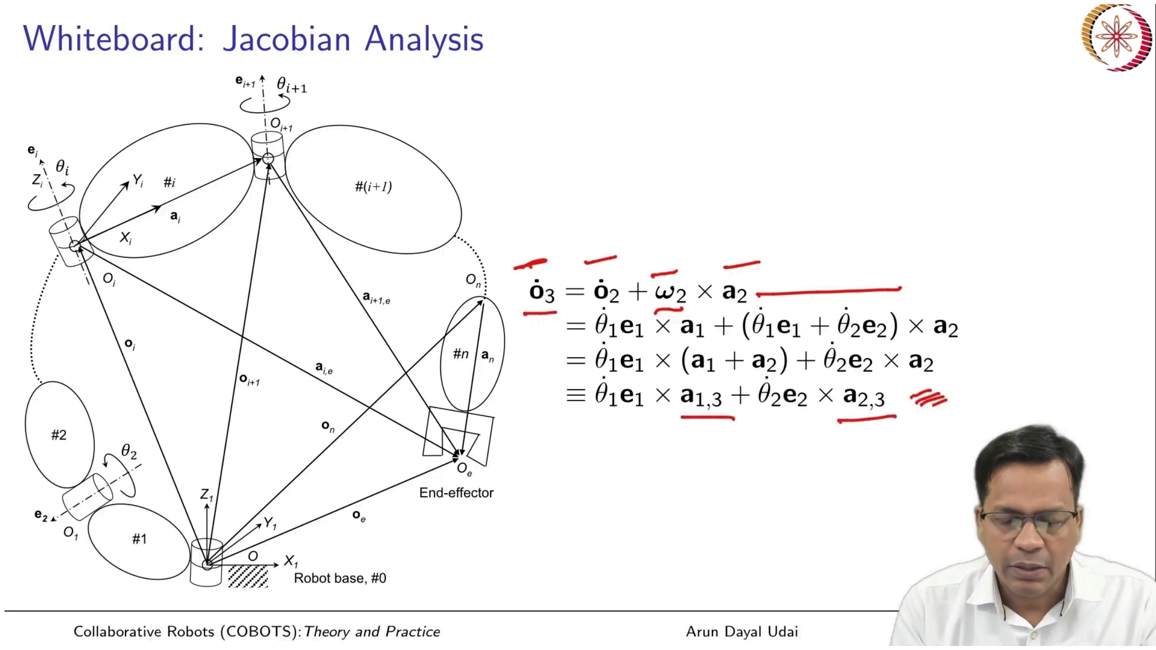
Omega 1 is equal to theta 1 dot, that is, the magnitude, and along the axis vector. So, this is your omega 1.

$$\boldsymbol{\omega}_2 = \dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2$$

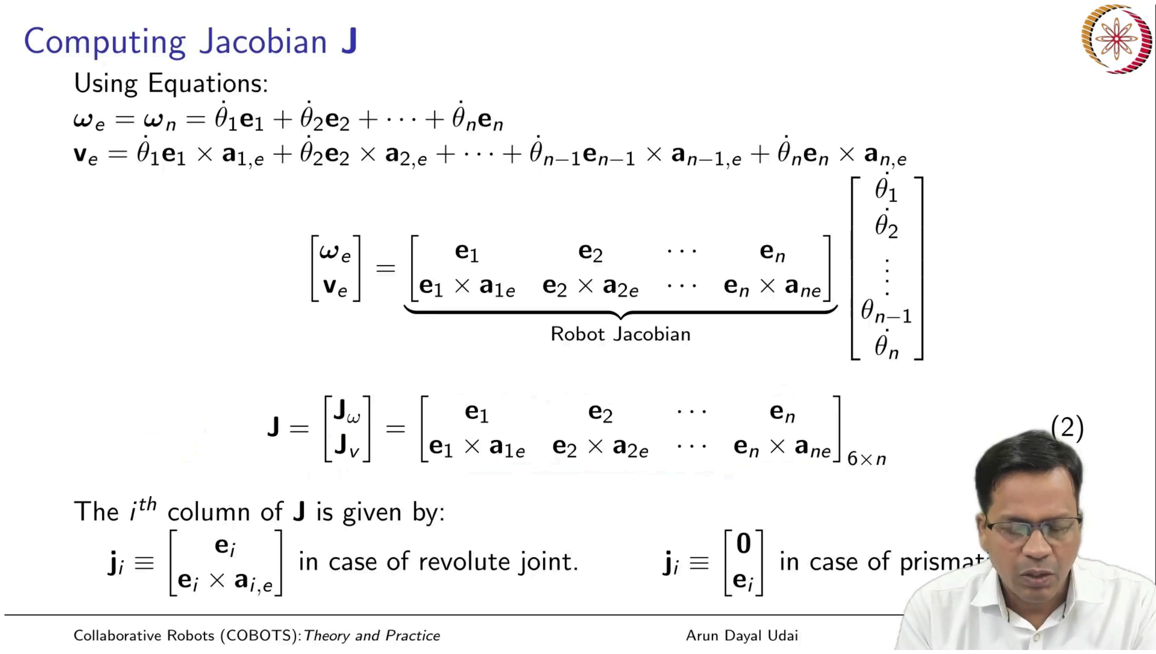
Similarly, omega 2 will be equal to omega 1 plus the angular velocity of joint 2; this is the magnitude and axis vector 2. So, that is how it will propagate, and finally, omega n will be equal to this. So, this is quite trivial for the angular velocity case. For linear velocity, what you see here is the angular velocity at the nth link will be equal to the angular velocities of all the joints prior to i. If it is for i, the angular velocity is due to the angular velocity of all the links that come prior to this.

Now, the linear velocity in this case, again, you see the ground because it is attached to the ground. The first linear velocity would become equal to zero. The next linear velocity will be equal to the first linear velocity and the linear velocity due to the angular velocity of the first link, okay? So, that creates a tangential linear velocity that is here, and if I can club it together, it is omega cross r. So, this is omega, and this is the radius. So, this is this distance, omega cross r. And in case this continues further, okay, so it goes like this. So, the final omega n would be equal to this. And what is this actually? It is the velocity contributions due to all the joint motions prior to the nth joint. Let us say you are at the nth joint, that is over here. So, whatever is the linear velocity over here, it is because of you see, the rotation of all the links prior to this. So, because of this angular velocity over here, because of this angular velocity over here, okay, and because of all the links which are here. So, the contribution of all the angular velocities and the distances. So, the angular velocity here and the distance from 2 to n. In this case, it is 2. So, it is 2 to n. Omega 2, 2 to n. So, so on and so forth. So, all of them it will be added together, and it will have multiple linear velocities, and all will be added to get the final linear velocity.

So, a_{ij} is basically this, i to j is this, and these are the notations that I have used here. Similarly, the end effector is a part of the last link. So, this is the part of the last link, basically. Let us just evaluate the term which comes here because this was quite easy, this is quite trivial, now this one.



So, the O3 dot will be equal to O2 plus omega 2 a2. So, how much is omega 2? Omega 2 can be substituted directly here. If I put that here, that is from Here I have copied it. So, this omega 2 comes here, and you expand it, and you can derive it till here. So, a2,3, that is from frame 2 to frame 3. So, this is how it is: if it is 3, it is 1 to 3 and 2 to 3. And these are the theta 1 dot and theta 2 dot. So, this is all you keep doing, and finally, for the n-link, it is like this.



So, now. These are the two equations that we have obtained.

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_n = \dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2 + \cdots + \dot{\theta}_n \mathbf{e}_n$$

This one is for angular velocity,

$$\mathbf{v}_e = \dot{\theta}_1 \mathbf{e}_1 \times \mathbf{a}_{1,e} + \dot{\theta}_2 \mathbf{e}_2 \times \mathbf{a}_{2,e} + \cdots + \dot{\theta}_{n-1} \mathbf{e}_{n-1} \times \mathbf{a}_{n-1,e} + \dot{\theta}_n \mathbf{e}_n \times \mathbf{a}_{n,e}$$

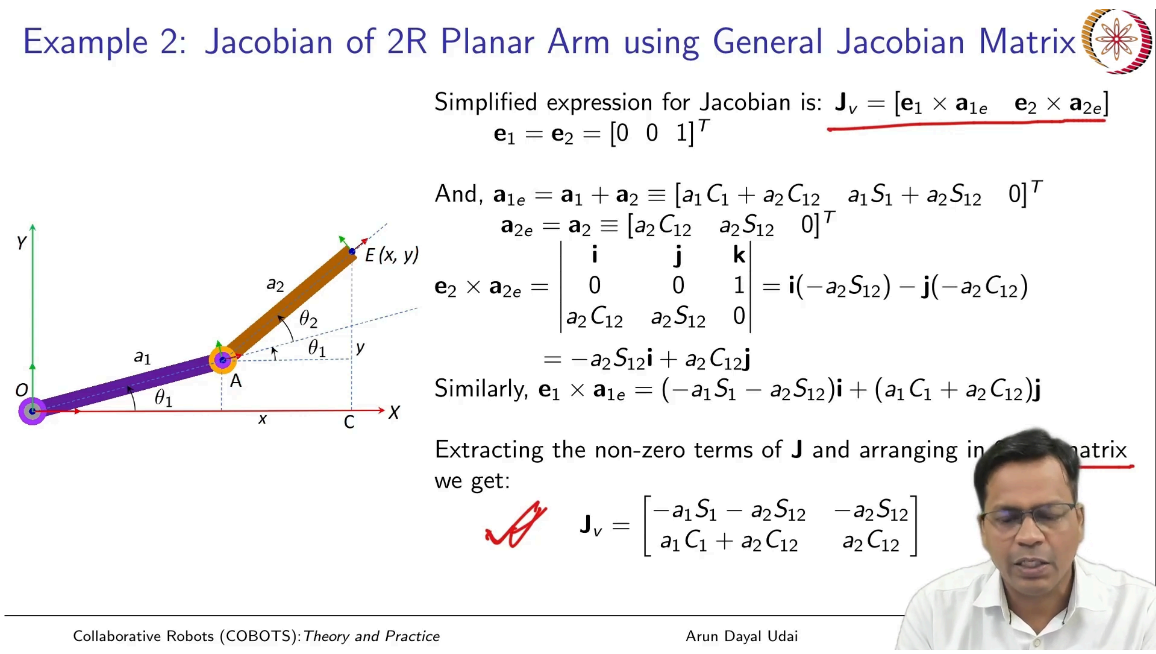
And this is for linear velocity. If I pack them together again, so these two come here, and you have theta 1 dot, theta 2 dot, theta n dot. So, all are arranged here, extracted from all the terms,

$$\begin{bmatrix} \boldsymbol{\omega}_e \\ \mathbf{v}_e \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_n \\ \mathbf{e}_1 \times \mathbf{a}_{1e} & \mathbf{e}_2 \times \mathbf{a}_{2e} & \cdots & \mathbf{e}_n \times \mathbf{a}_{ne} \end{bmatrix}}_{\text{Robot Jacobian}} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_{n-1} \\ \dot{\theta}_n \end{bmatrix}$$

And this becomes your Robot Jacobian. The dimension of this would be e here is 1, 2, 3, and again, this is a vector 2, 3. So, a total 6 cross n, this is n cross 1, this is 6 cross 1. So, this is all; these are the dimensions, and here is your robot Jacobian. It comprises All axis vectors and the distance vectors from the link i to the end effector.

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_\omega \\ \mathbf{J}_v \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_n \\ \mathbf{e}_1 \times \mathbf{a}_{1e} & \mathbf{e}_2 \times \mathbf{a}_{2e} & \cdots & \mathbf{e}_n \times \mathbf{a}_{ne} \end{bmatrix}_{6\times n}$$

So, this is your Jacobian, and it has basically two components: j omega and j v. This is responsible for linear velocity. The upper row is responsible for angular velocity. Got it. So, the column of J in the case of a revolute joint is like this. In the case of a prismatic joint, the velocity vector would be because there is no angular velocity; this goes to 0, and you have linear velocity the same as that of the previous link. So, that makes this as e_i.



So now let us use this and see if we can do it for a 2R planar arm using the general Jacobian matrix, not taking the differential. So, the simplified expression could be because it can only have linear velocities; it has no control over angular velocities. So, we will consider just the J_v component of the Jacobian. So, that is e_i, a_{ie} type.

So, e1 and e2, that is, the axis vector, will be along the z-axis only because, you know, all the joint axes are along the z-axis, so that is over here; it is perpendicular to the plane of this paper, so z is out of the plane of this. So, you have z1, you have z2, both are here, so that is here.

$$\mathbf{e}_1 = \mathbf{e}_2 = [0 \ \ 0 \ \ 1]^T$$

So, they can be written as e1 is equal to e2 is equal to 0, 0, 1. They are unit vectors, so it is 1. So, I have transposed it to make it compact, but actually, it is like this. So, these are the z-axis, and then a1e, that is, frame 1 to end effector, from 1 to end effector. So that is equal to a1 vector plus a2 vector, and that can be quickly written as this.

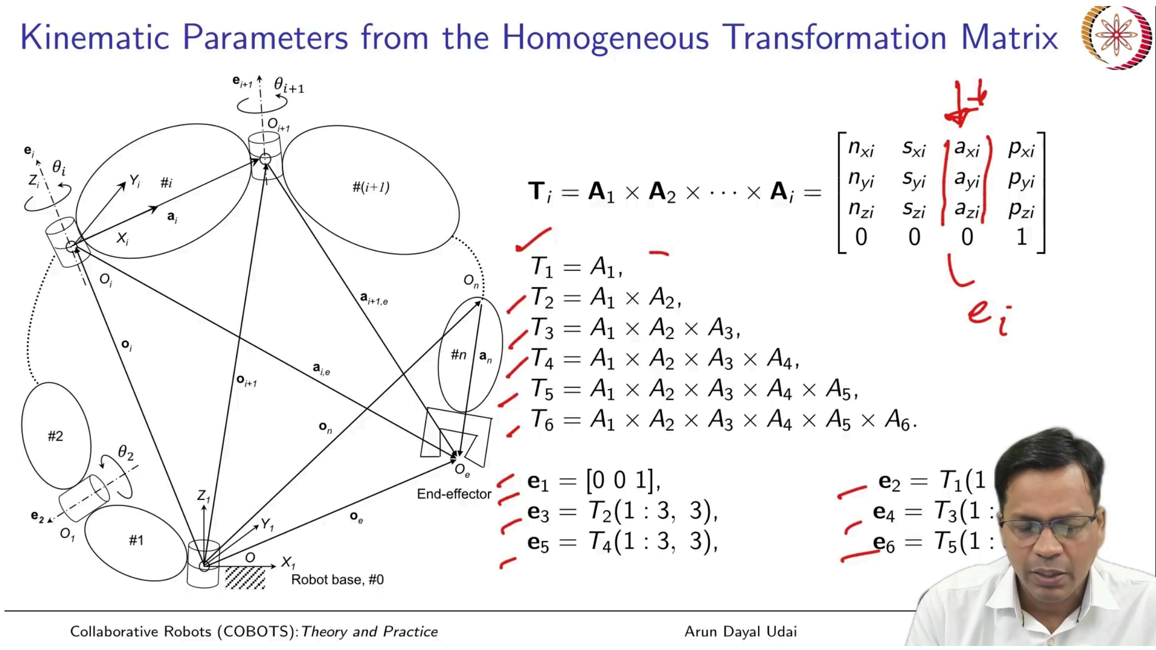
$$\mathbf{a}_{1e} = \mathbf{a}_1 + \mathbf{a}_2 \equiv [a_1 C_1 + a_2 C_{12} \quad a_1 S_1 + a_2 S_{12} \quad 0]^T$$

First component   Second component

This is using trigonometric projection; you can write it. So, it is basically this and this; this is the first component, this is the second component, and along z, it is equal to 0. So, this is a1e; similarly, a2e from 2 to the end effector is the a2 vector, and that can quickly be written as this and this. So, this over here, this over here, z becomes equal to 0.

Now, taking the product, so e2, a2e will be equal to i, j, k; you know how to take the cross product of this. So, the component will go like this. So, e comes from here, and from here, you take 2e; it comes out to be this, and similarly, ei x a1e cross-product would take you to this. Again, you take the first value from here and a1e from here, and you take the cross product; you get to this.

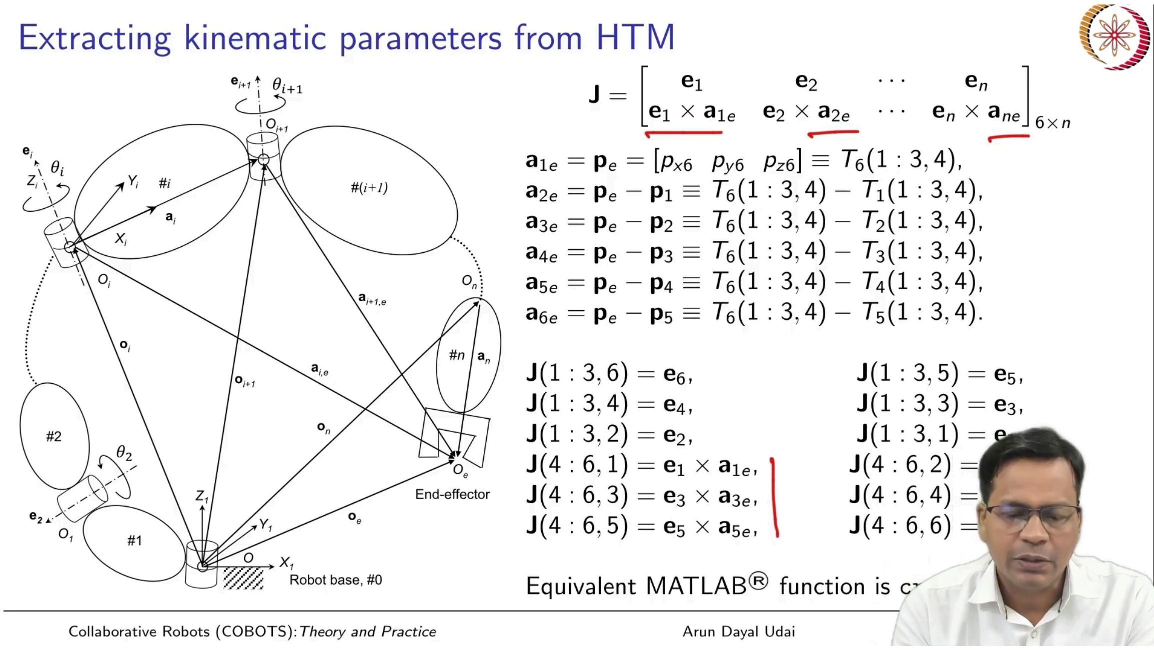
Now, if I arrange it, extracting the non-zero terms of j and arranging them in a 2x2 matrix, I get exactly the same Jacobian the way we derived it earlier in a 2x2 matrix, which we derived by simply taking derivatives of the positions. So, it comes out to be the same. So, remember this because it is a planar system, and it does not allow any rotations; we have taken it like this. Otherwise, we would have also taken the ei component, e1, and e2 component.



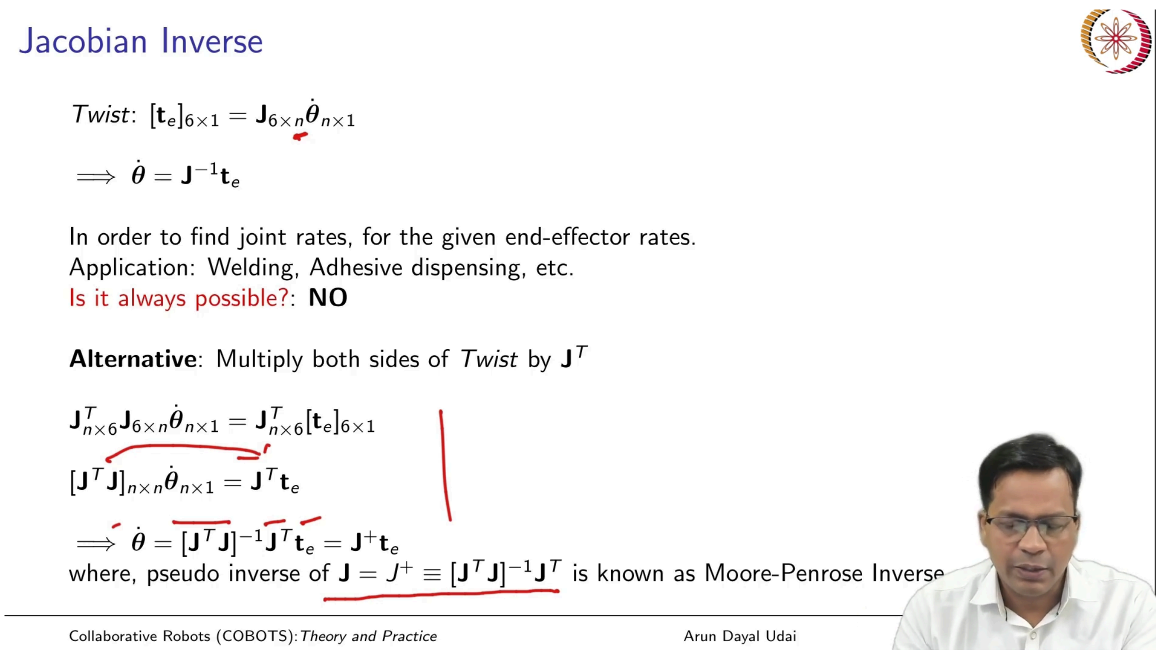
So now, can we extract all these parameters that I have used, e1, a1e, a2e, using general homogeneous transformation matrices? That is what we will try out here. So, extract the Kinematic Parameters from the Homogeneous Transformation Matrix. So, Ti is equal to, you know, this A1. Whenever I write 1, it is with respect to 0; 2 is with respect to 1. So, if it is with respect to i, it is with respect to i minus 1. Got it? Okay. So, this is what is here. It is, you know, all the products will finally give you the final homogeneous transformation matrix. Got it?

So, now T1 is equal to this. T2 will be equal to a1 and a2, T3, and so on and so forth. You do this. So, this is the overall forward kinematic transformation. These will take you to the frame, which is just one prior to the sixth one, that is, for the fifth frame. This is for the fourth frame. This is for the third frame, likewise.

So now the last column, e1 is the column which is here. It is the z-axis vector. You know the component of z along x, along y, along z lies here. So, this essentially is nothing but the axis vector, okay, the axis vector. So, that can be extracted from all of this. So, if it is for the first axis, you have to use this. For the second axis vector, third axis vector, fourth, fifth, and sixth axis vector, for all the consecutive t i is this column, this third column, the first three rows will give you the axis vectors.



Now this is the final term, you know, you have to find out aie is also. So, that can be obtained from the position vector, so that also is available here, you see. This is the position vector of any ith frame with respect to the O. Okay. That is the fixed frame. Okay. So, I will use that here. So, a1e, a2e will basically be. So, let us say I want to find out aie. This vector. So, how will I do it? I will simply use this vector and this vector. This is nothing but the end effector vector. This is obtained from all the aIs that are here from here to here. If I take the product of this, whatever I get, the homogeneous transformation matrix for this joint, I, from there I, can extract this one. So, using this vector triangle, I get this. For all the links, I can do it in a similar way. So, all the a_{1e}s can be obtained now. This we have already seen. Okay, now we'll put them in place in this matrix. Look, whatever we have extracted, ei is an aie, and similarly, the cross products are put over here. Okay, this is the way you can extract all the homogeneous transformation matrix elements and use them to calculate the Jacobian. Got it?



So, finding out the inverse is again very trivial. You already know the twist is given as the Jacobian into theta dot.

$$\textit{Twist}: [\mathbf{t}_e]_{6\times1} = \mathbf{J}_{6\times n}\dot{\boldsymbol{\theta}}_{n\times1}$$

So, if I take the inverse, it should give me this.

$$\dot{\theta} = \mathbf{J}^{-1}\mathbf{t}_e$$

This is very trivial, provided this is an invertible matrix, that is, a square matrix only. So, in order to find the joint rates, basically, what we do in quite a lot of applications like welding, adhesive dispensing, profile following, maybe okay, or if it is painting, you have to, you already have an input of end effector velocity, and you have to find the joint velocities, corresponding joint velocities, for the given end effector velocity. So, how will you do it? All these applications will demand this particular equation. So, the Jacobian inverse will be required.

Is it always possible? You know, it is not possible because it may or may not be a square matrix. So, an alternative approach would be, if this is the equation, you know, multiply both sides of it by the Jacobian transpose, so you get to this, okay, this over here, and this side is here, okay.

$$\mathbf{J}^T_{n\times 6}\mathbf{J}_{6\times n}\dot{\boldsymbol{\theta}}_{n\times 1} = \mathbf{J}^T_{n\times 6}[\mathbf{t}_e]_{6\times 1}$$

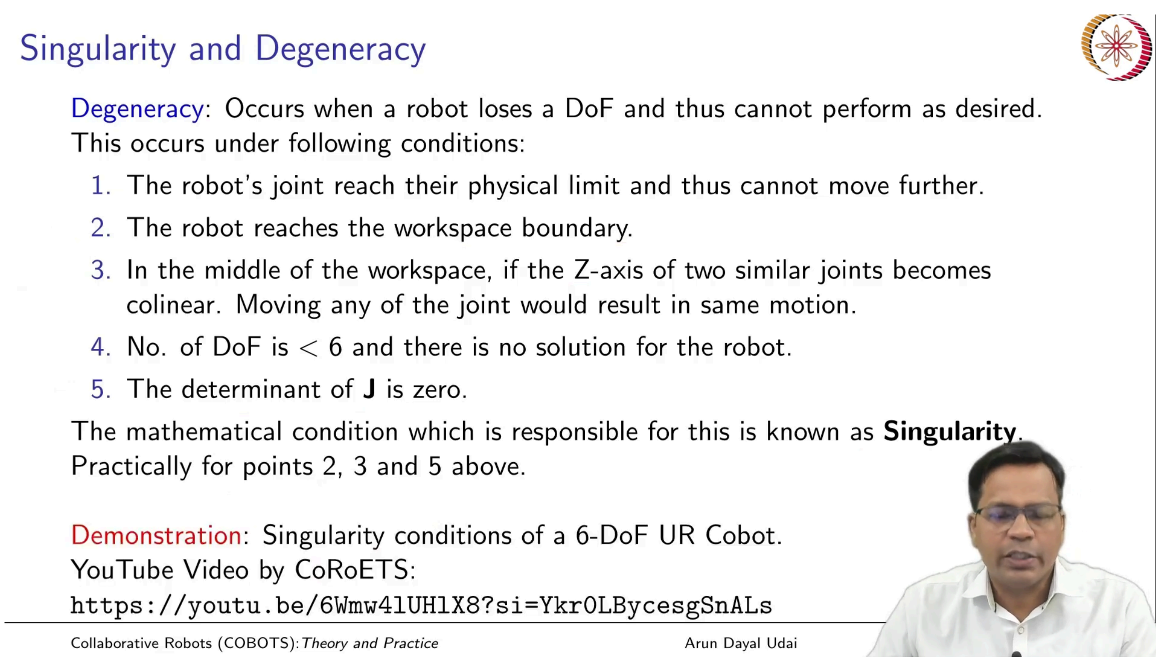
Now, J transpose J is an n cross n matrix. So, if you take the product, it becomes a square matrix.

$$[\mathbf{J}^T\mathbf{J}]_{n\times n}\dot{\boldsymbol{\theta}}_{n\times 1} = \mathbf{J}^T\mathbf{t}_e$$

Now, this is invertible. So, theta dot now can be by taking this as the inverse to the other side.

$$\dot{\boldsymbol{\theta}} = [\mathbf{J}^T\mathbf{J}]^{-1}\mathbf{J}^T\mathbf{t}_e = \mathbf{J}^+\mathbf{t}_e$$

Now, this is here. Jacobian transpose TE. And this term is known as the pseudo-inverse. It is also known as the Moore-Penrose inverse. This is quite easily possible in the case of a 7-DoF robot. You can use this approach where you have n not equal to 6. In that case, you can use this to take the inverse. There are limitations to this also. We will talk about this later. This is how you find out the Jacobian inverse.



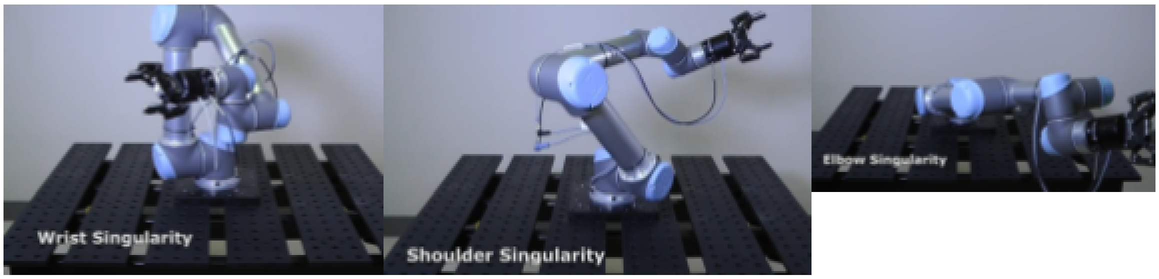
Now, there are a few terms that are very important here. One of them is Degeneracy, which occurs when a robot loses a degree of freedom and thus cannot perform as desired. So, what are those conditions?

The first one is the robot loses joint reach to its physical limit and thus cannot move further. There may be some joint limitations because of the motor fixture, the architecture of the system, or because of the joint limits or motor limits. So, it cannot go further, and it cannot. If it is expected to go into a fully stressed condition, but because of the architecture, let us say it cannot move. So, that is a physical limit. So, it cannot move further, and that is leading to this condition again.

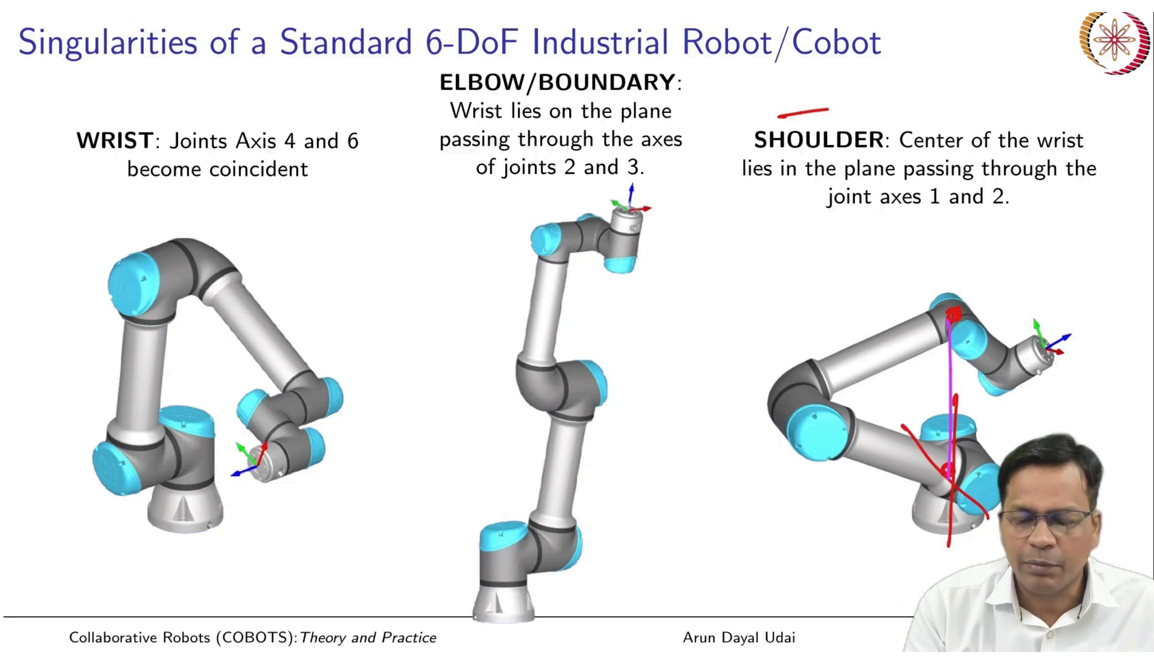
The second one is when the robot reaches the workspace boundary beyond which it cannot go. So, those are the conditions where your robot can go degenerate. The next one is in the middle of the workspace if the z-axis of two similar joints becomes collinear, okay? Moving any of the joints would result in the same motion. So, that is the case when it is again degenerate. So, if the degree of freedom is less than 6, as in the case of a SCARA robot, it cannot roll, and it cannot pitch. So, there is no solution for the robot along those directions. So, that is also a degenerate system, and when the determinant of the Jacobian becomes equal to 0, so as you have expected, the twist is equal to j theta dot.

If I have to find out theta dot, what did we do? Jacobian inverse of the twist of the nth factor. That is what we know. So, the Jacobian inverse is basically the adjoint of J divided by the determinant of J. So, if the determinant of J becomes equal to 0, in that case, what happens? This is not possible, so you cannot extract theta dot for the given affected trajectory. So, in that case, the determinant becomes equal to 0, and you cannot calculate the joint rates. So that is one of the important cases because this is widely used in robot control also. So, it is not always the inverse kinematics approach where we find out the joint angles. We also sometimes use it to find out the joint rates using this approach.

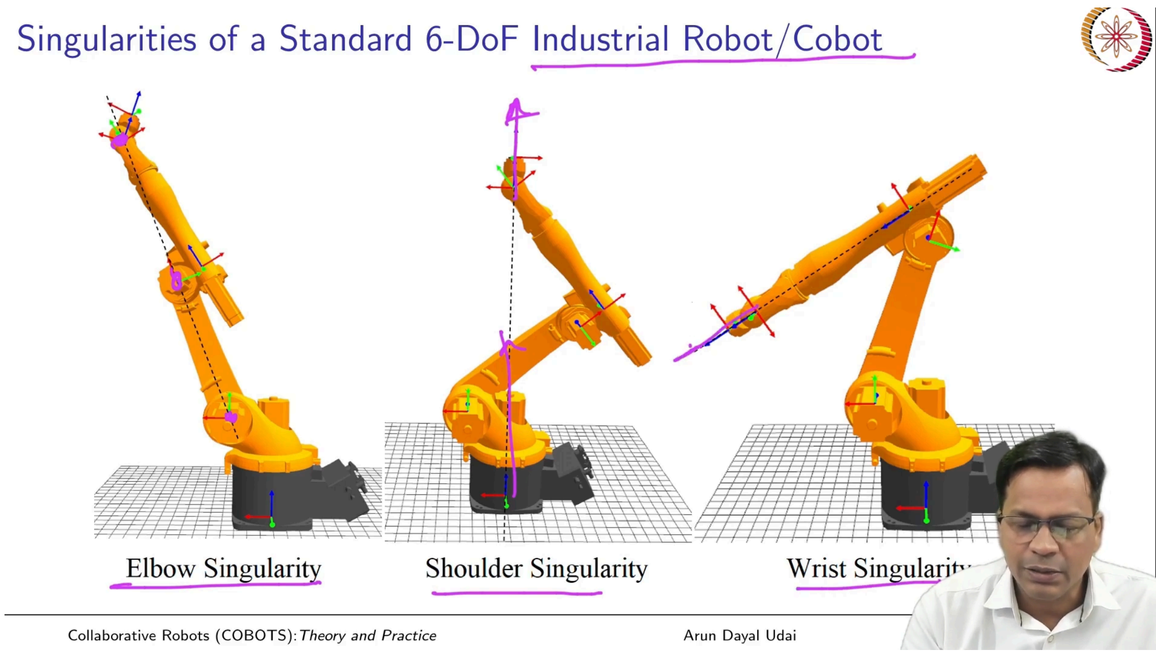
The mathematical condition which is responsible for this is known as Singularity. So, basically, the determinant becoming equal to 0 is the singular case, mathematical singularity, and that is what is the case for point number 2, point number 3, and point number 5. So, the determinant becomes equal to 0 or in the middle of the workspace, if the z-axis of two similar joints becomes collinear mathematically, even J becomes equal to 0. The determinant of J also becomes equal to 0. The robot reaches its boundary in quite a lot of cases. In those cases also, the determinant becomes equal to 0.



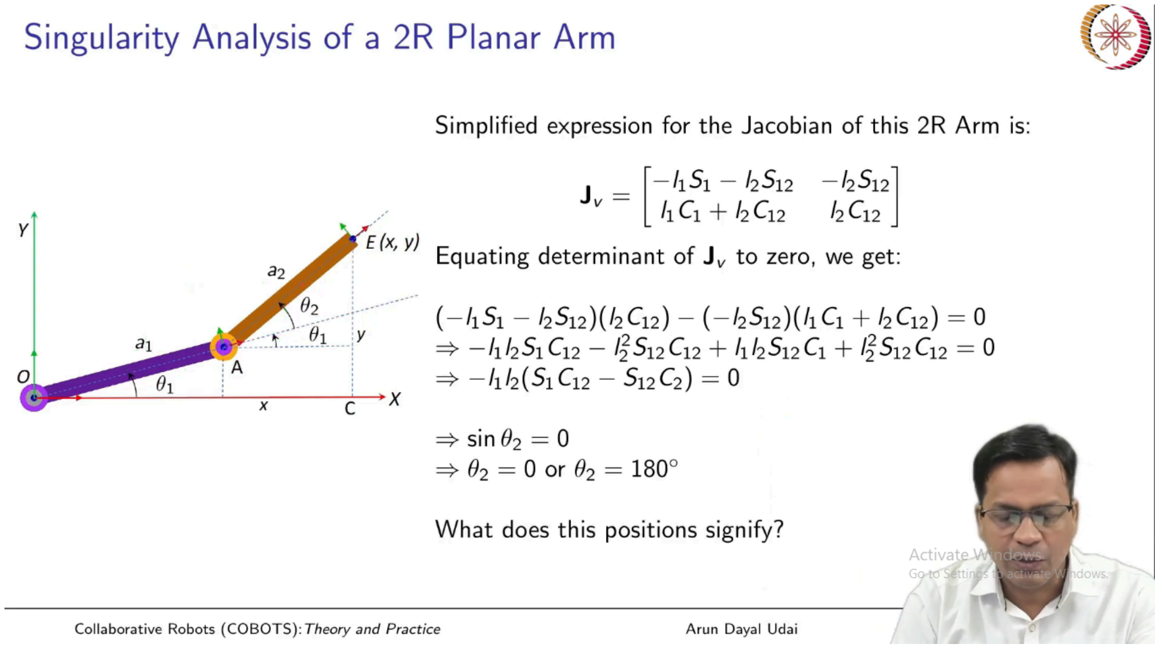
I will just show you a small video of the UR robot, which does have a singularity. The link to this video I have mentioned here. This is wrist singularity. This is wrist singularity. You see, in between, it became singular, and the robot halted for a moment, then it moved forward. There is an infinite solution for that particular force. This is wrist singularity again. This is shoulder singularity. You saw, in between, it had an infinite solution again. Got it? This is elbow singularity. It is in the case of a fully extended condition that is, the robot reached its boundary. Got it? So, these were the cases.



The Singularity of a standard 6 DOF industrial robot or a cobot. So, in the case of UR, you saw these were a few. Wrist singularity, axis 4 and axis 6. So, where are they? Axis 1, axis 2, 3, axis 4 and 6. Both are coincidental. They are parallel to each other, okay? In this case, elbow or boundary singularity, you see this, this, and this came in a straight line. So, the wrist lies on the plane passing through the axis of joint 2 and joint 3. This is joint 2, joint 3. The wrist lies on the same plane, okay? So, that is a boundary singularity. It is also known as elbow singularity, and the final one is the shoulder singularity; the centre of the wrist lies in a plane passing through the joint axis 1 and 2. So, this is joint axis 1 and 2. So, it lies exactly above this. So, this is again a place where this whole robot can rotate about this axis, and that is a shoulder singularity.



That is there in the case of a standard industrial robot or a cobot also. This is exactly having the same architecture as that of FANUC 35iA that I have shown earlier. So, this is elbow singularity. You see this, this, and this comes in a straight line now. Shoulder singularity: you have this axis getting aligned with the last axis, two axes, and this singularity here is when this axis comes in line with the fourth one. The fourth and sixth get in line. So, these are a few singularities that are there.



So, now singularity analysis of a 2R Planar Arm. Now that we already have this matrix in hand, I will just check where it goes Singular. So, the simplified expression for the Jacobian of a 2R arm is this.

$$\mathbf{J}_v = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \end{bmatrix}$$

So, the determinant of this and equating it to 0 gives me this equation.

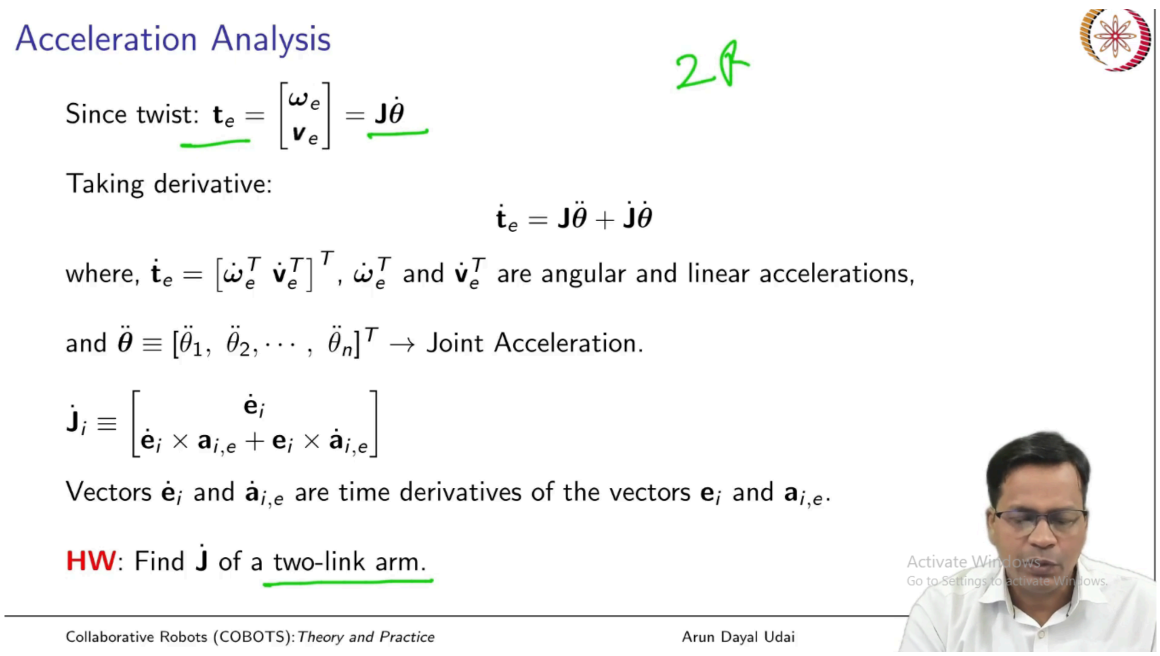
$$(-l_1 S_1 - l_2 S_{12})(l_2 C_{12}) - (-l_2 S_{12})(l_1 C_1 + l_2 C_{12}) = 0$$

If I solve it, what I get is this, and finally, this leads to sin theta 2 equal to 0 or theta 2 equal to 180 degrees.

$$\Rightarrow sin\Theta_2 = 0$$

$$\Rightarrow \theta_2 = 0 \text{ or } \theta_2 = 180°$$

So, these are the two places where it is singular. So, where are they? Theta 2 becomes equal to 0, which means this and the angle becomes equal to, so this is here, so the whole robot goes to a straight line, okay? Straight linear, okay? In that case, it is a boundary singularity; that is what is happening, and when it is folded, theta 2 is equal to 180 degrees. It is fully enfolded. The second link comes here. Again, it is singular. So, these are the two physical conditions in which it is having singularity. It does not depend on theta 1, you see. So, anywhere where it is finding the boundary, everywhere it is going singular. So, these are the significance of these two solutions.



Let us do an Acceleration Analysis now. We have checked upon the velocity a lot, okay? So, how to find out the accelerations here? So, this is what you already have in hand; that is, the twist is equal to the Jacobian theta dot.

$$\mathbf{t}_e = \begin{bmatrix} \boldsymbol{\omega}_e \\ \mathbf{v}_e \end{bmatrix} = \mathbf{J}\dot{\boldsymbol{\theta}}$$

$$\dot{\mathbf{t}}_e = \mathbf{J}\ddot{\boldsymbol{\theta}} + \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}$$

Taking the derivative of this would give this: the acceleration twist dot is equal to J theta double dot plus J dot theta dot, okay? So, what is a twist dot? It is basically angular acceleration and linear accelerations of the end effector. Theta double dot is the joint acceleration that is here. Jacobian is Jacobian; you already know it, okay? Jacobian dot can be evaluated using you already know Jacobian is equal to ei ei cross aie. So, this is what the various columns of Jacobian are, okay? So, if you take the derivative, the ith column will be equal to this. So, first into the derivative of the second plus the second into the derivative of the first. So, that is the way you know derivatives are taken.

So, it is just written like this. So, this becomes the J_i dot that is substituted here. Theta dot is the angular velocity of the joint. Theta double dot is the angular acceleration of the joint. J is Jacobian. J dot is found out like this. That is how you find out the end effector accelerations, angular and linear.

Can you try finding out J dot for a simple two-link arm? Now, I want you all to do this exercise as homework. So, just twist Jacobian times theta dot. Do it for a 2R planar manipulator. We already have the Jacobian in hand for that. I would like you to do it.

That is all for this lecture. So, in the next lecture, we will do inverse kinematics of a 7-degree-of-freedom KUKA LBR iiwa robot. We will also discuss what Null Space is. That is all. Thanks a lot.