NPTEL Online Certification Courses

COLLABORATIVE ROBOTS (COBOTS): THEORY AND PRACTICE

Dr Arun Dayal Udai

Department of Mechanical Engineering

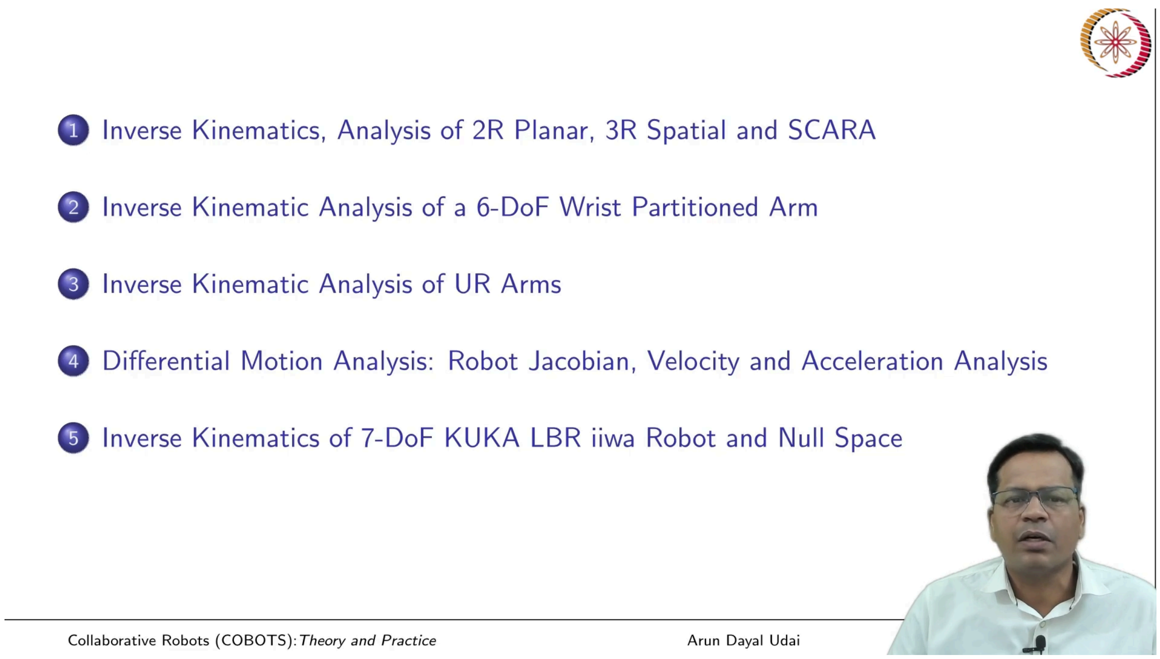
Indian Institute of Technology (ISM) Dhanbad

Week: 04

Lecture: 15

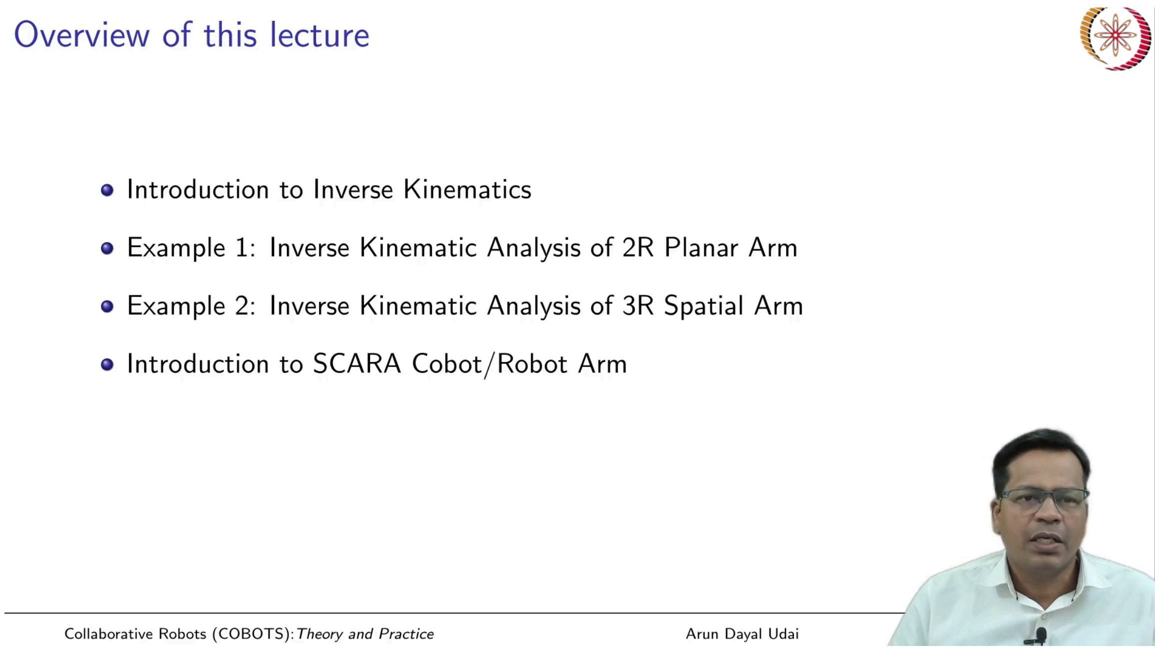
Inverse Kinematics, Analysis of 2R Planar, 3R Spatial and SCARA

Welcome to the fourth week of the course- Collaborative Robots: Theory and Practice.

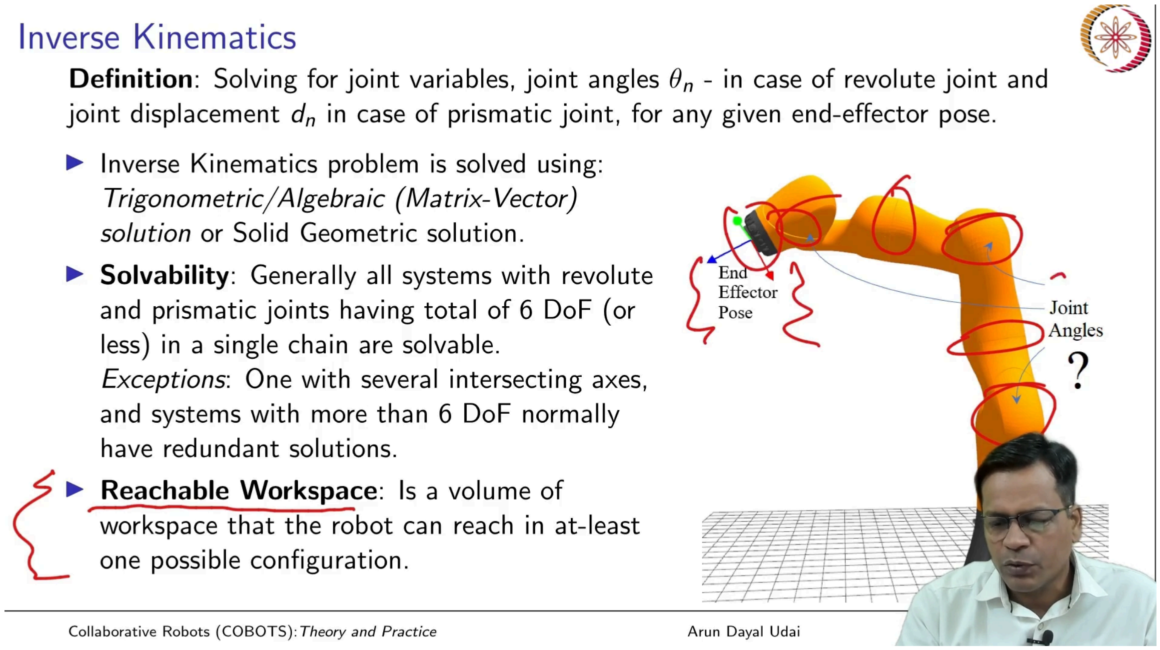


This week will focus on Inverse Kinematics, Velocity, and Acceleration Analysis. We will have five lectures in this module. We will start with an Introduction to Inverse Kinematics. We will continue with the Analysis of a 2R Planar Arm, a 3R Spatial Arm, and a SCARA Robot. In the second lecture, I will discuss a 6-degree-of-freedom Wrist Partition Arm. We will perform inverse kinematics of that. We will discuss the Inverse Kinematics Analysis of a UR Arm, which is a universal robot arm with 6 degrees of freedom, in lecture 3. Before we move on to the inverse kinematics of 7-degree-of-freedom robots like the KUKA iiwa robot and discuss null space, I will cover differential motion analysis. In the fourth lecture, we will discuss Robot Jacobian, Velocity, and Acceleration Analysis. In the fifth and final lecture, we will perform inverse kinematics of a 7-degree-of-freedom KUKA iiwa robot and finally discuss Null Spaces.

So, let us continue with the first lecture of this module, which is on Inverse Kinematics, Velocity, and Acceleration Analysis of the course Collaborative Robots: Theory and Practice.



In this lecture, I will begin by introducing you all to inverse kinematics. I will continue with examples. The first example will be on a 2R Planar Arm. I will do an Inverse Kinematic Analysis of that. In the second example, I will do an Inverse Kinematic Analysis of a 3R Spatial Arm, and I will continue further to 4 degrees of freedom, SCARA Cobot or a Robot Arm, and I will do inverse kinematics of that as well.



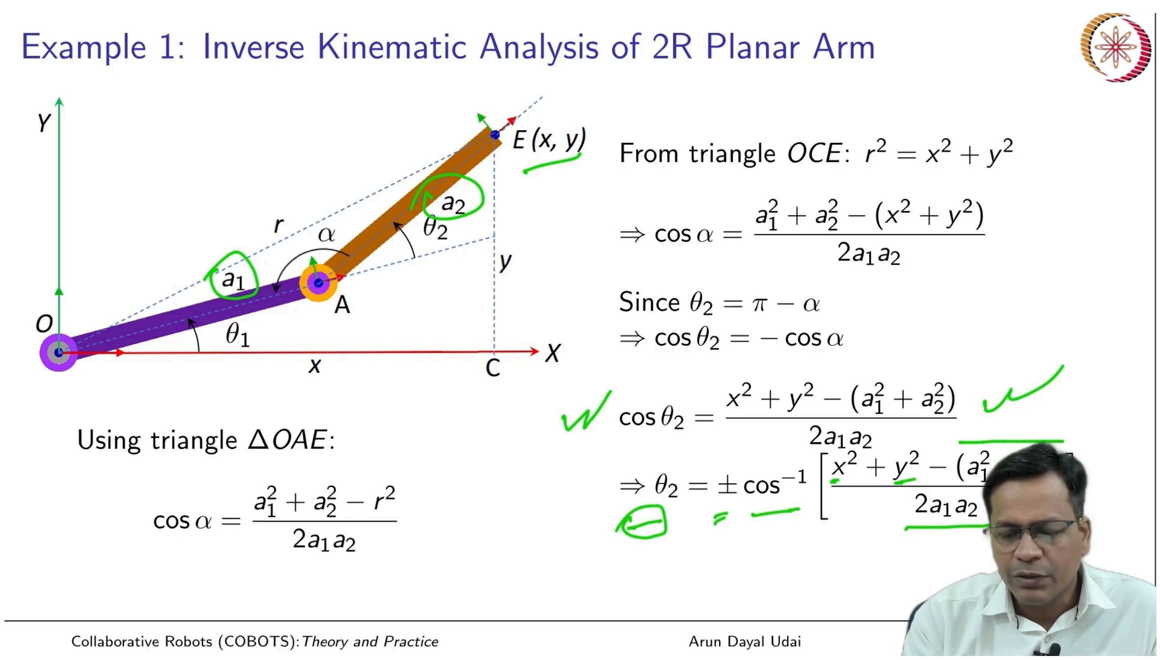
So, let us continue with the introduction first. So, what is inverse kinematics? So, solving for joint variables, that is the joint angle (Θ_n) in the case of a revolute joint robot and joint displacement (d_n) in the case of prismatic joints of a robot for any given end effector pose. Let us say you have a robot like this, okay? You have the first link that comes here, and you have the second link that goes like this. Let us assume it is a planar arm. So, if you know the x and y, that is the end effector position of this robot, so this is the x-axis, and this is the y-axis. If you know the position of this, what is my joint angle? So, the first joint angle that comes here, let us say it is theta 1, and the second joint angle is theta 2. The link lengths are given as l1 and l2. So, given the end effector position, finding out the joint angles that will actually take it to that particular position. So, solving these problems is known as inverse kinematics analysis.

The inverse kinematics problem is broadly solved using two major ways: the Trigonometric and Algebraic approach, that is, using matrix and vector approaches, and the second one is the Solid Geometric Approach. We will not be doing this. We will focus on the trigonometric and algebraic solutions for all the problems that I will be discussing.

Solvability: Generally, all the systems or all the robots with revolute or prismatic joints having a total of 6 degrees of freedom or less in a single serial chain are solvable. That may not be true for quite a lot of robots that we will see later. So, exceptions are one with several intersecting axes, and systems with more than 6 degrees of freedom normally have redundant solutions or may even have infinite solutions. Those are not solvable.

So, what is a Reachable Workspace? So, in order to get to a solution first for inverse kinematics, the robot should lie in a reachable workspace. So, it is a volume of workspace that the robot can reach in at least one possible configuration. It may not be possible that you go there and orient it in all the possible ways, so at least one possible configuration it should go there. If the robot is in this reachable workspace, definitely We should try for getting an inverse kinematics solution.

So, this is a robot. It is the KUKA iiwa which is shown here. So, these are the joint angles that it has, okay? Various joint angles are there. So, the end effector pose will be given, and I am required to solve for the joint angles. So, in the case of cobots, they mostly have revolute joints and will solve for joint angles.



So, let us begin with a traditional problem, that is, inverse kinematics analysis of a 2R Planar Arm. This is one of the simplest arms that is there in quite a lot of robots as a subset, even if they have multiple links. So, any portion of that may have a 2-link arm which is like this. What is it? It is a planar manipulator, just like your engineering drafter, as you have seen. So, this is your link one; the first link goes here, and you have a second link that goes like this. So, you have an end effector that lies here. The end effector has coordinates x and y. So, this is your x-axis, and this is your y-axis. This is joint angle theta 1 and joint angle theta 2. Theta 2 is a relative angle. The first joint makes an angle with the horizontal axis; that is, with respect to this fixed frame, it makes theta 1. The second joint makes an angle with respect to the extension of link one. Link two makes an angle of theta two. Link lengths are a1 and a2, which are given. So, the entire geometry of this system is known. Okay, what additionally is known? Where does it lie? It lies at E(x, y). So now, I have to get the output of this inverse kinematics algorithm. So, what is it? It is basically the solution of this robot for the given position, that is, x, y. I need to get the angles, that is, theta 1 and theta 2. So, this is how inverse kinematics is defined. So, we will do that now. So, using triangle OAE, where are they? It is O, this is A, this is E. So, this is the triangle which comprises of link length A1, link length A2, and you have this is R, this is O, this is A, this is E. If this is the triangle, using the cosine rule, I can get the cosine of alpha. Alpha is here. If you can see it here, this is your alpha. The cosine of alpha is here. So, a1 squared plus a2 squared minus r squared by 2 a1 a2. This is your length angle. This is the angle alpha.

$$\cos\alpha = \frac{a_1^2 + a_2^2 - r^2}{2a_1a_2}$$

So, moving forward from triangle OCE, where are they? O, C, and E, that is the end effector position. You know, using the Pythagoras theorem, you can get r square is equal to x square plus y square.

$$r^2 = x^2 + y^2$$

So, this is your x and y. So, if I substitute this r square here, what do I get? The cosine of alpha is equal to a1 square plus a2 square minus x square plus y square by 2 a1 a2.

$$\Rightarrow \cos\alpha = \frac{a_1^2 + a_2^2 - (x^2 + y^2)}{2a_1a_2}$$

So, now, since theta 2 is equal to pi minus alpha, you can get it. So, this is theta 2 is equal to pi minus this alpha angle. So, this is the angle, so pi minus alpha.

$$\text{Since } \theta_2 = \pi - \alpha$$
$$\Rightarrow \cos\theta_2 = -\cos\alpha$$

So, the cosine of theta 2 is equal to the minus of the cosine of alpha. So, I can just put cosine alpha there. So, what do I get? The cosine of theta 2 is equal to this,

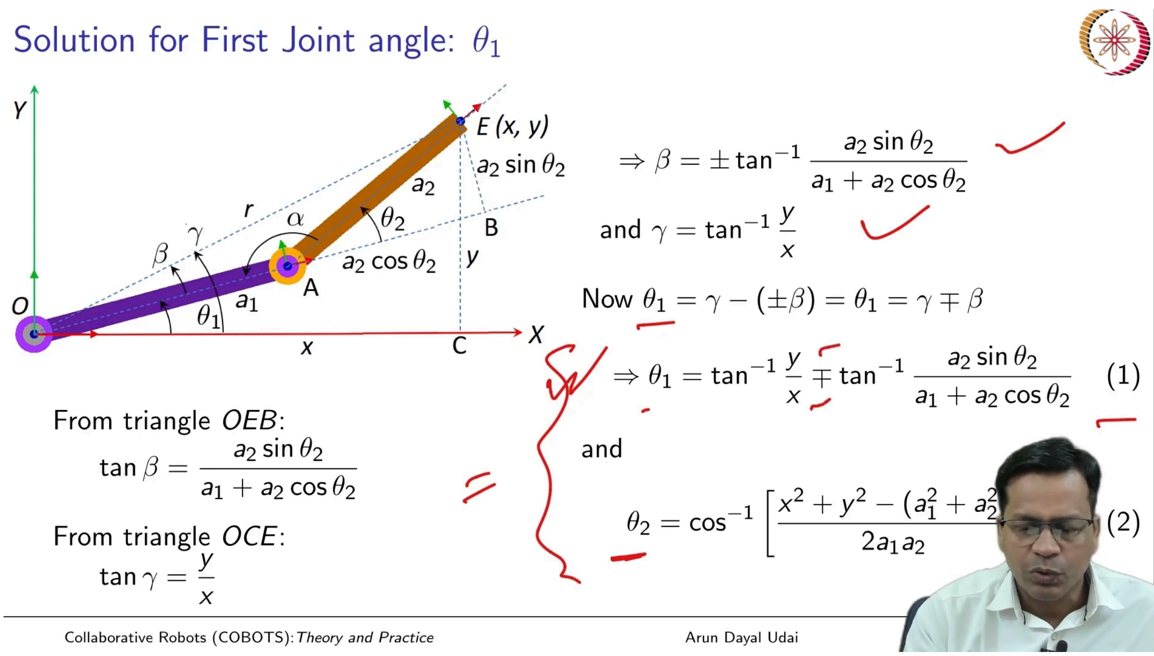
$$\cos\theta_2 = \frac{x^2 + y^2 - (a_1^2 + a_2^2)}{2a_1 a_2}$$

And theta 2 is equal to plus or minus cosine inverse of x square plus y square minus a1 square plus a2 square by 2 a1 a2.

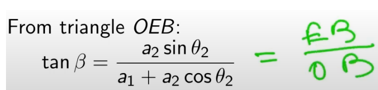
$$\Theta_2 = \pm \cos^{-1}\left[\frac{x^2+y^2-(a_1^2+a_2^2)}{2a_1 a2}\right]$$

So, now I could get theta 2 for the given position x-y and the link lengths a1 and a2. So, these are the known parameters. Theta 2 was unknown, and we could get it.

Please note, so cosine theta 2 is this. So, theta 2 will be plus or minus of cosine inverse. Why? Because the cosine of minus theta is also equal to cosine theta. So, that is an inverse. So, that is what I should be getting. So, this is how I get theta 2.



Now, how to obtain theta 1 (Θ_1), that is the joint angle 1. Now, from triangles OEB, that is the right-angle triangle you see. You could clearly note here, yes, if this is a2 so that this side will be, this perpendicular will be a2 sine of theta 2, this angle is theta 2. Similarly, AB will be AB; that is, this one is a2 cosine of theta 2, ok. So, now, the edge this is actually equal to EB by OB. How much is OB, basically? OB is the a1 plus a2 cosine of theta 2 that goes to the denominator, and the perpendicular is the a2 sine of theta 2. So, how much is that? That is equal to the tan of beta. So, the beta is here; the beta angle is here. So, the tan of beta is equal to EB by OB.



Similarly, You can quickly get, using triangles OCE again, that it is equal to tan of gamma.

$$\tan\gamma = \frac{y}{x}$$

Gamma is here; that is the full-angle gamma. Tan of gamma is equal to y by x.

So, now using these two, I can quickly write beta is equal to plus or minus. Again, it will be plus or minus because if it is minus theta 2, you saw in the earlier case it was plus or minus. So, if minus theta 2, it will be minus; for plus, it is plus. So, beta is equal to plus or minus tan inverse of this.

$$\Rightarrow \beta = \pm \tan^{-1} \frac{a_2 \sin\theta_2}{a_1 + a_2 \cos\theta_2}$$

So, using this, I obtained this again from the second one: tan gamma y by x. So, I could get gamma is equal to tan inverse of y by x.

$$\gamma = tan^{-1} \frac{y}{x}$$

So, now, using these two, I can get theta 1. So, theta 1 is here, which is related to gamma and beta as gamma minus beta is equal to theta 1. So, gamma minus beta; beta can be plus or minus. So, I can get theta 1 is equal to gamma minus and plus beta.

$$\theta_1 = \gamma - (\pm\beta) = \theta_1 = \gamma \mp \beta$$

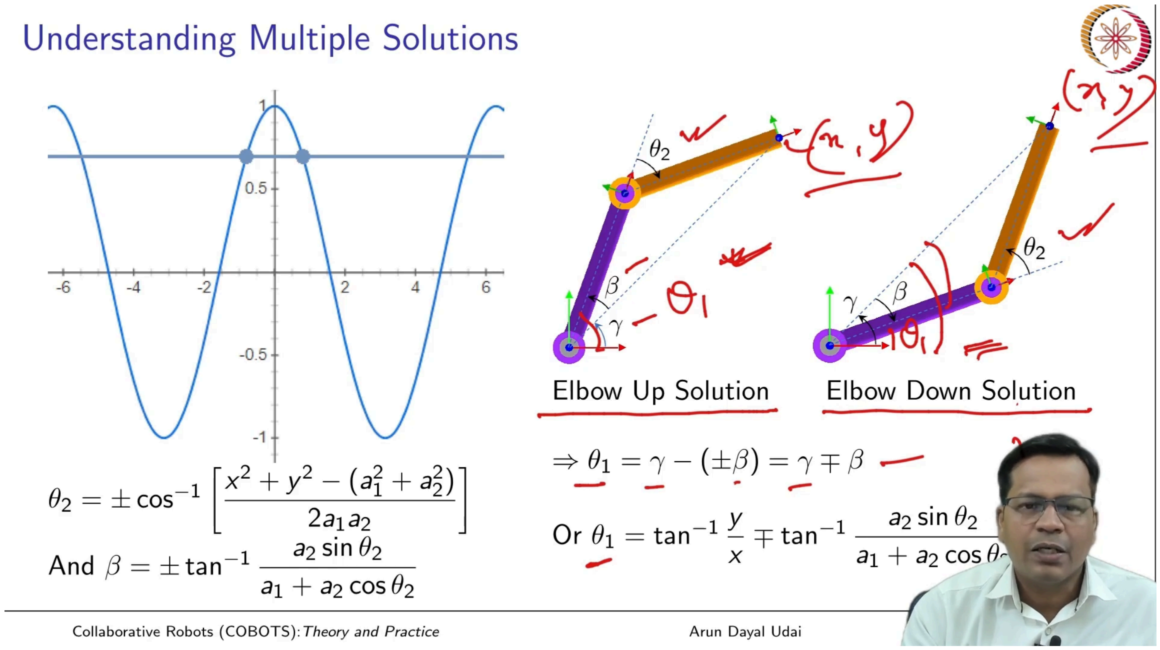
So, theta 1 is now equal to if I put the values directly from here; I could get theta 1 is equal to this.

$$\Rightarrow \theta_1 = \tan^{-1} \frac{y}{x} \mp \tan^{-1} \frac{a_2 \sin\theta_2}{a_1 + a_2 \cos\theta_2}$$

And from the previous result, we already got theta 2, which comes like this.

$$\Theta_2 = cos^{-1}[\frac{x^2+y^2-(a_1^2+a_2^2)}{2a_1a2}]$$

So, you see this points out that these two actually point out to a multiple solution for plus and minus. You have at least two solutions, which are clearly visible here. Can we reach there?



Now, let us understand the multiple solutions. So, you got what? You got theta 2 is equal to plus or minus cos inverse of this-

$$\Theta_2 = \pm\, cos^{-1}\left[\frac{x^2+y^2-(a_1^2+a_2^2)}{2a_1a2}\right]$$

And beta is equal to this.

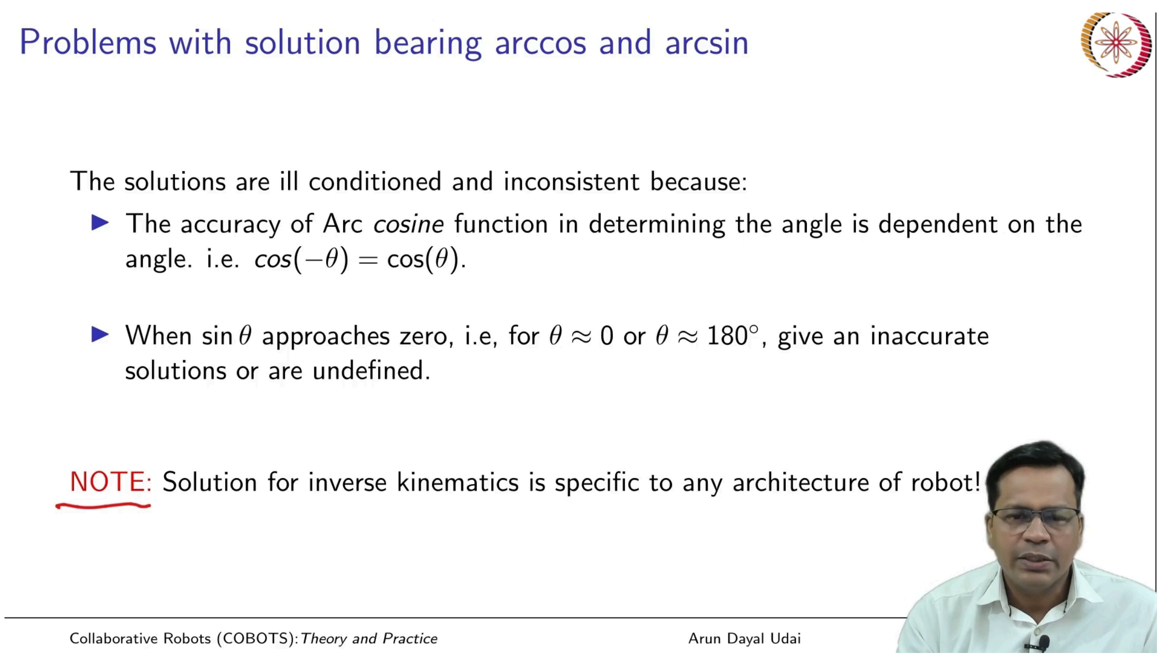
$$\beta = \pm \tan^{-1}\frac{a_2 \sin\theta_2}{a_1 + a_2\cos\theta_2}$$

So, you can now clearly see here this is the plot of cosine. When you see here, you see you have cosine having two different values for the value like cosine of plus or minus of theta leads you to the same value, that is the same value. These are leading to the same positions.

So, now, what do these two solutions mean? You see, first of all, theta 1 is equal to gamma minus plus and minus beta, so gamma plus or minus beta becomes gamma minus plus beta, and theta 1 is equal to this-

$$\theta_1 = \gamma - (\pm\beta) = \gamma \mp \beta$$

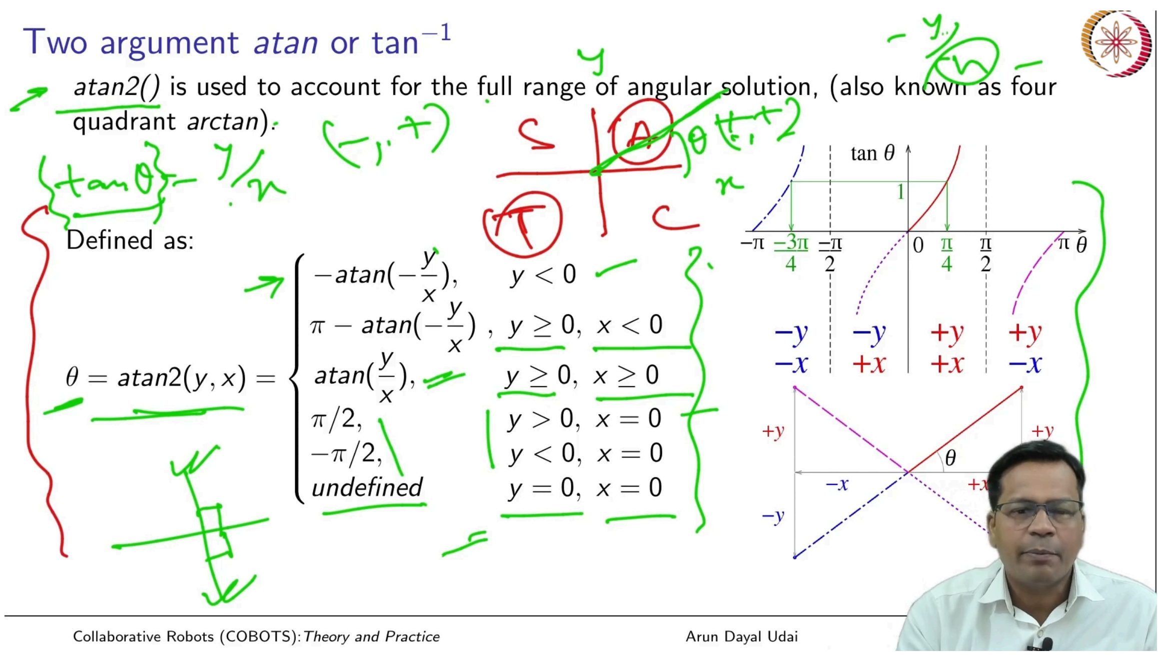
So, you see, one of them is actually gamma plus beta, okay? So, that is your theta 1. In the second case, it is gamma minus beta. So, that becomes equal to theta 1. So, these are the two cases, okay? So, when theta 1 is like this, the configuration of the robot is like this. So, in that case, theta 2 is negative, okay? So, when it is in this position So, theta 2 is positive. So, one of them is positive; the other one is automatically negative. So, in this case, it is known as the Elbow-Up Solution, and this one is the Elbow-Down Solution. Both of them are located at the same end of the vector position, that is, x, y. Got it? So, for the same x, y position, it could be either the elbow-down or elbow-up case. Depending on the solution. So, you see, right at the first example, you could notice that even a two-link system can have multiple solutions.



Now, let us see some of the problems that are arising because of arc cosine and arc sine solutions. So, you saw here this one was a tan solution, but theta 2 has a cosine solution. So, what is the problem with this? So, the accuracy of the arc cosine function in determining the angle is dependent on the angle itself. So, as you see, cos minus theta is equal to cos theta.

So, the solutions are ill-conditioned and inconsistent because of this. In programming, when you start programming for inverse kinematics, you should be very particular about the solution, and the system should be well programmable. Can we put any condition so as to get to the accurate solution, the exact solution? So, that is the reason it should be well-defined.

So, in the case of sine, you see sine approaches 0 for theta nearly equal to 0 or nearly equal to 180 degrees. So, this gave an inaccurate solution, and again, they are undefined at times. Okay? So, you see, again, using sine would also be a problem. So, what is the solution for this, that we will see. So, please note again the solution for inverse kinematics is specific to any particular architecture of the robot. If a robot is an inline robot like KUKA iiwa or it is a robot like a UR arm or like in a standard inertial arm. So, each and every robot has its own solution. Unless they are kinematically similar. Only the dimensions are changing, then you can use the same general solution. But, there is no general solution in the case of inverse kinematics of a serial robot. Each robot may be different, and you need to solve them independently.

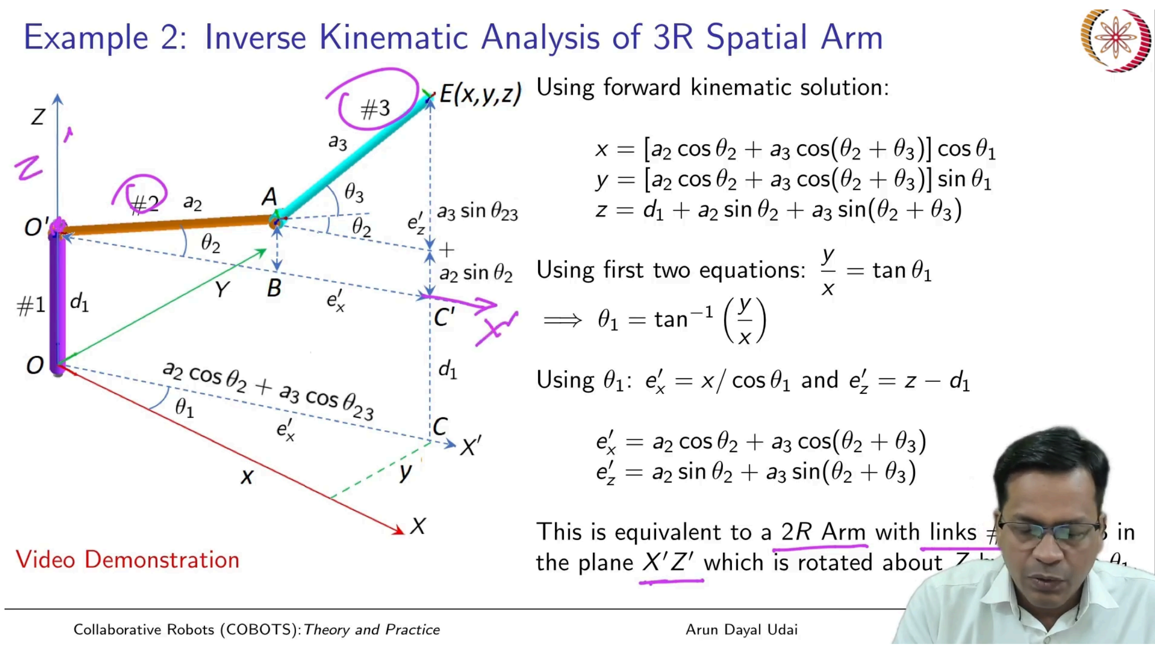


So now, let us move and see how those sine and cosine issues can be solved. So now, You saw I attempted to get the result in the form of tangents. So, you see, in the case of tangent, you can put conditions. Let us say you have all sine, tan, and cos. So, tan is positive here; tan is positive here; otherwise, it is negative. So, based on the location of your arm, If your arm is located here, so it is plus, plus for x and y. If it is here, it is having a different scenario. So, it is negative and positive for the x and y coordinates. So, depending on the different quadrant of your arm location, you have a different result for the tangent, and this can be easily set in the case of a tangent because tangents are directly governed by x and y coordinates, unlike sine, where it is y by the hypotenuse. So, hypotenuse does not change its sign; only y will change its sign. So, you cannot put the condition to it. But in this case, you can clearly understand if the robot arm is in the positive quadrant or the second quadrant, third quadrant, or fourth quadrant. So, depending on them, you can clearly define your theta using programming also.

So, these are the conditions that you can put depending on positive positive of x theta lies here. So, tan because you see tan theta tan theta is equal to y by x. Tan theta will have the same value for plus plus of these two plus x and plus y minus x and minus y. It will have the same value. So, if the tan value is the same but the input values for x and y are in different quadrants, you should get different values for theta. So that is quite easily possible using these types of conditioning. So, in this case, it is known as the eight and two solution and it is a ready-made function available in quite a lot of programming languages like MATLAB or Python. Libraries are an inbuilt function, so this accounts for the full range of angular solutions, also known as the four-quadrant R tangent solution.

So, these are the conditions you can go through, and for each condition for y less than 0, you have to consider this value of theta. For y greater than and x less than this, both are greater than or equal to 0; you can take directly. Again, there are some explicit conditions in which one of them becomes equal to 0, especially the denominator. So, when the denominator is equal to zero, you can have a solution like it is either over here or here. So, in that case, 90 degrees or plus 90 degrees, you can directly check the condition without evaluating the tangent, and you can write it as plus pi by 2 or minus pi by 2.

In case both of them are 0, you need not evaluate; you directly say it is undefined. These are the plots for the tangent that I have shown and the quadrants also. So, you tend to try to get the solution in a tangent way so that it can be programmatically made unambiguous.



Now, let us move ahead to my second example, which is the Inverse Kinematic Analysis of the 3R Spatial Arm. So, we have already done the forward kinematics of that. So, you can already directly get these equations from there. So, what is this directly? So, you can directly even get through trigonometric projections. So, this is an arm that looks like this. I'll just show you once here. So, this is the first joint that is moving. Okay, it is coming back. This is the second joint. This is the axis of motion. Parallel to the ground plane, and the final one is, again, an axis that is parallel to the ground plane. That is for the joint angle 3, the third joint. So, yes, this is x, this is y, this is z. I can obtain them directly using trigonometric projections also, but I have just copied it from the forward kinematic solution. So, you can check it.

So, this is what x. x is here. How much is that? So, you see, this is the two-arm planar arm, which is link 2 and link 3. That lies always in a plane, and the plane rotates about axis 1 by an angle theta 1. So, this lies in a plane. So, this is your plane. So, this arm will always lie in a plane. So, this is a planar 2R arm with link lengths a2 and a3. Theta 2 and theta 3 are the joint angles.

So, ex dash, ex dash is here. So, ex dash is equal to a2 cosine theta 2 plus a3 cosine theta 2 plus theta 3, and that is ex dash.

$$e'_x = a_2 cos\Theta_2 + a_3 cos(\Theta_2 + \Theta_3)$$

Similarly, ez dash over here; instead of y along the vertical axis, you have z. So, ez dash is equal to a2 sine theta 2 plus a2 sine theta 2 plus theta 3. So, you got x dash and ez dash.

$$e'_z = a_2 sin\Theta_2 + a_3 sin(\Theta_2 + \Theta_3)$$

So, those two are the coordinates for these 2R planar arms in their own base frame, that is, O dash. Now, using Ex dash, that is this one. So, if I project it along the x-axis, I get this cosine theta 1, that is x. First equation. Similarly, sine theta 1 will be The second one, that is y, and z will be the sum of this vertical distance plus from here to here. So, it went a little clumsy. So, this is the clear one for the same. So, this is how you can directly obtain through trigonometric projection, or you can directly do it using forward kinematics. So, this is how the link lengths, joint angles, and the end effector positions are related. Now, what is given is E(x, y, z), which is the location of the end effector position. I have to find out the joint angles of theta 1, theta 2, and theta 3.

The first two equations are equation one, equation two, and equation three.

$$\begin{aligned} x &= [a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3)] \cos\theta_1 \\ y &= [a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3)] \sin\theta_1 \\ z &= d_1 + a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3) \end{aligned}$$

From the first two, I can quickly write theta one is equal to the tan inverse of y by x. If I divide the second equation by the first equation, okay, so y by x gives me a tan of theta 1. Again, theta 1 would be equal to atan2 y comma x.

$$\Theta_1 = tan^{-1}(\frac{y}{x}) = atan2(y, x)$$

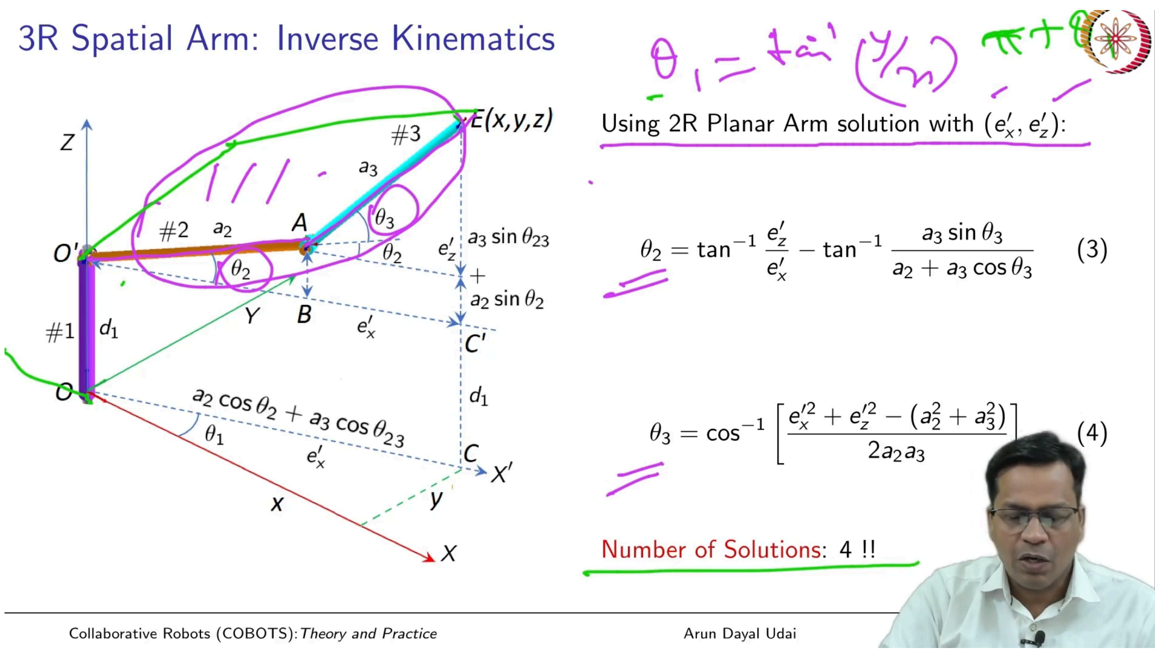
So, you have to use our tangent here to get the angle theta 1. Okay, so now, using theta 1, I can write, I can substitute theta 1 here and here, and I can write ex dash is equal to x by cosine theta 1. Similarly, from this equation, I can write. Z minus d1 is equal to ez dash.

$$e'_x = x/\cos\theta_1 \text{ and } e'_z = z - d_1$$

So, that is the distance between the ez dash and the ex dash. What I am trying to get is basically the local coordinates, that is, ex dash and ez dash, that is, the end effector x dash and y dash in this plane, this plane so, that I can consider this subset as a 2R planar arm, and I can solve for joint angles theta 2 and theta 3. So, in order to solve this, this becomes a local 2R manipulator, and I will solve for it. So, these are the coordinates for that for this 2R manipulator.

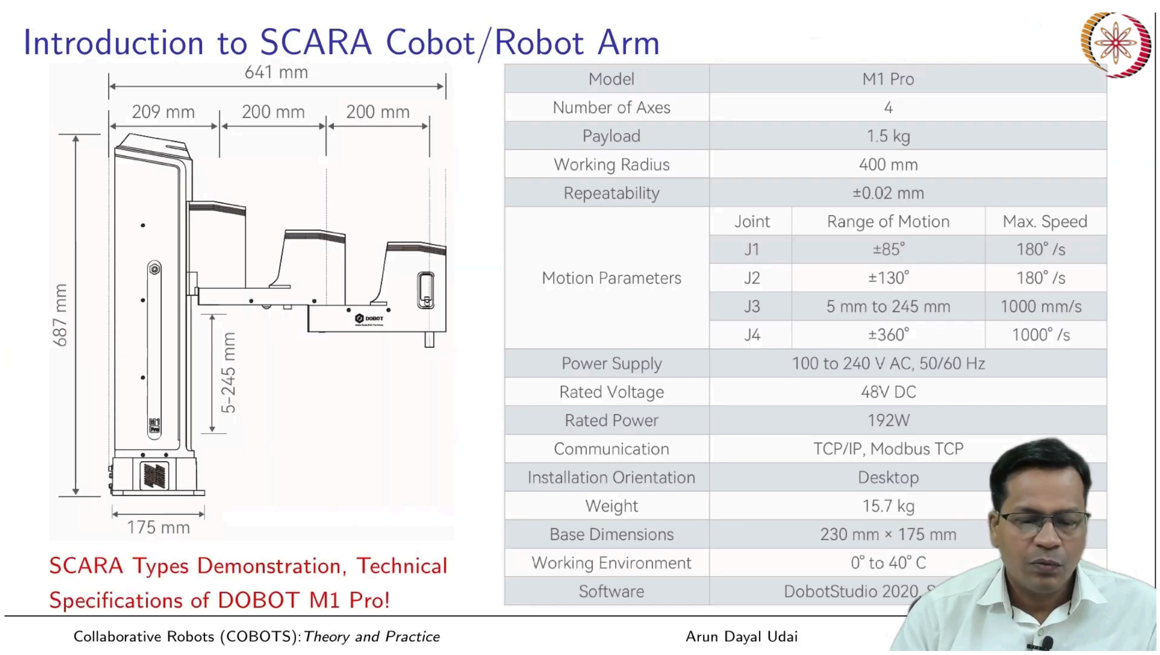
$$e'_x = a_2\cos\theta_2 + a_3\cos(\theta_2+\theta_3)$$
$$e'_z = a_2\sin\theta_2 + a_3\sin(\theta_2+\theta_3)$$

So, this is equivalent to now a 2R arm with links 2 and 3 in the plane x dash z dash. So, where are they? This is your x-dash, and this is your now z-dash from here. So, now I will use this and solve for this.

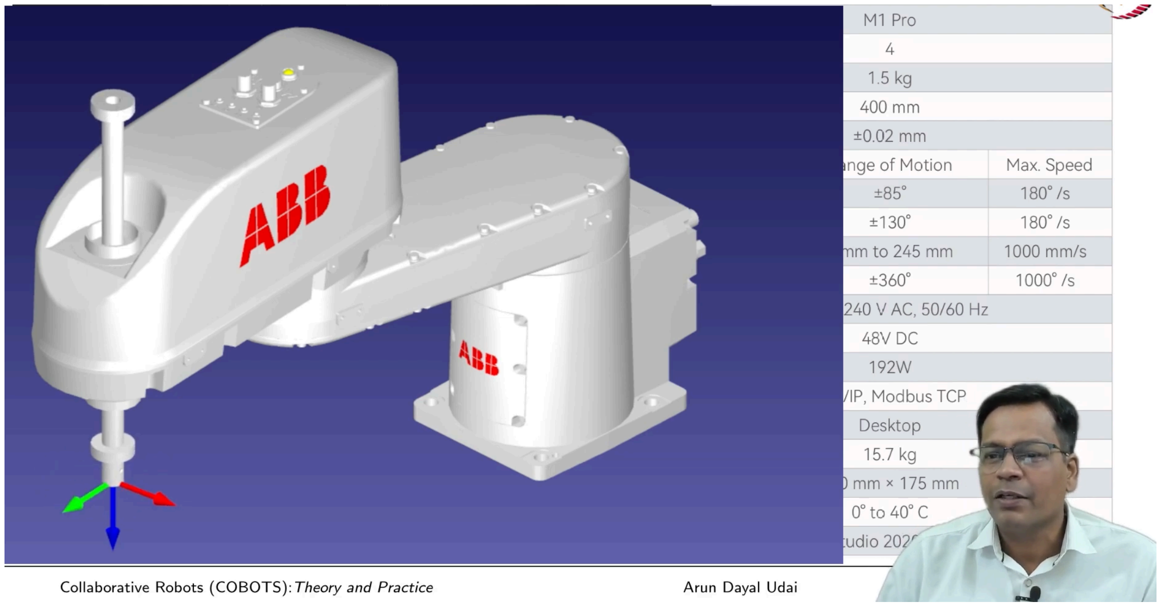


So, using this 2R planar arm solution that we have done earlier. I will simply put ex-dash and ez-dash in place of x and y, and I can get theta 2 and theta 3. That is this angle and this angle, and I could get all the angles. So, theta 1 was equal to the tan inverse of y by x, theta 2, and this is theta 3. So, all the joint angles are solved. So, the second subset is very, very helpful here; we have already solved this one actually, the 2R planar arm. So, I have used it directly here, and the same we will be doing for many other arms also. So, the 2R manipulator is very, very important, and that helps us to quickly solve many other robots as well as the way we have done it here.

So, now how many solutions does it have? You have already seen the second link, and the third link can have two positions. This is the elbow down, which is drawn here. This could be the elbow-up position. Again, theta 1 could be theta 1 or pi plus this theta 1. Is it not? So, this can be right at the front, or you can go 180 degrees behind. Theta 1 goes 180 degrees behind from the existing position, and from behind, you can do elbow up and elbow down. So, in total, there are four solutions. So, this is how you need to evaluate any given solution for multiple solutions also.



Now, let us see what is a SCARA robot. This is one of the most popular kinds of robots in the industry. I will just quickly switch to that window and try to show you here.



So, this is a standard SCARA robot. You see, it has the first motion. It can do like this. That is the revolute axis that is perpendicular to the ground. The second one is something like this. Again, it is perpendicular to the ground where it is attached. The first and the second axes are parallel to each other, and the third one is also perpendicular to the ground, and it is prismatic this time. It goes up and down in a linear way. So, it is linear displacement. The fourth one is a revolute axis again. So, you have at least three revolute axes here, and all the axes are parallel to each other and perpendicular to the ground.

This is a standard robot. So, in this case, What you can see here is why this robot is very popular because it can do all the motors for the first and second axis lie on the ground, and the second joint is connected mostly using a belt. So, you have a low mass that is moving. Both the motors are situated at the fixed base, okay? Over here, in the fixed base, that makes it very light. So, they have less moment of inertia so that it can have very high speed, okay? They have very high speed and are not influenced by gravity. So, you can put it anywhere, and gravity does not affect any of the joint torques.

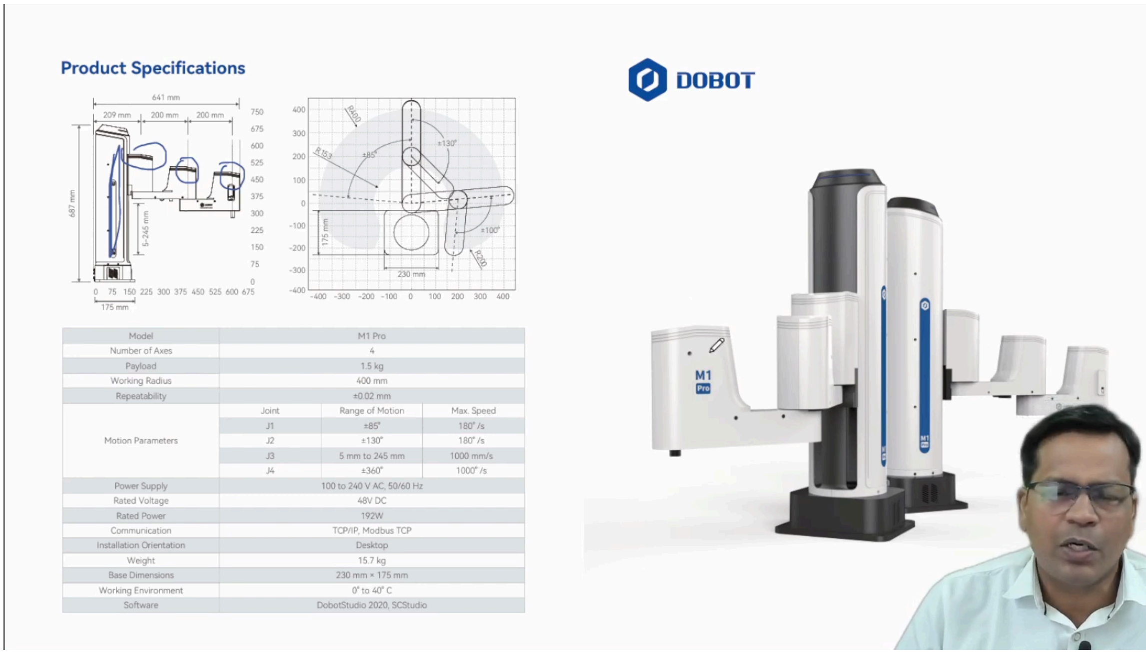
So, you do not have to move against gravity anywhere except the prismatic link, which is only affected by gravity. All three rotary axes are free from gravity. So, that is the reason they are very fast, they are very quick and they have a low moment of inertia. So, that is the reason it is very popular in the industry for pick and place, vertical pick and place operations in the industry, like PCB assembly. You can pick up a chip and place it on the board, and you just need some motion, like yaw motion. You do not need many other motions. So, this is the yaw motion to orient your chip. It can be picked from a place and put it anywhere. So you pick it, you place it, pick it, place it. So that way it is very, very fast.

So, the robot that I am going to show here now. It is a different kind of SCARA in which the prismatic axis is at the first joint. So, I will just show a small, quick video.

In this case, you see the first joint is a prismatic axis. The second, third, and fourth ones are revolute axes. Both are equally good. So, this is the first and second joint shown in motion. So, this is the second and third axis, which is shown in motion.

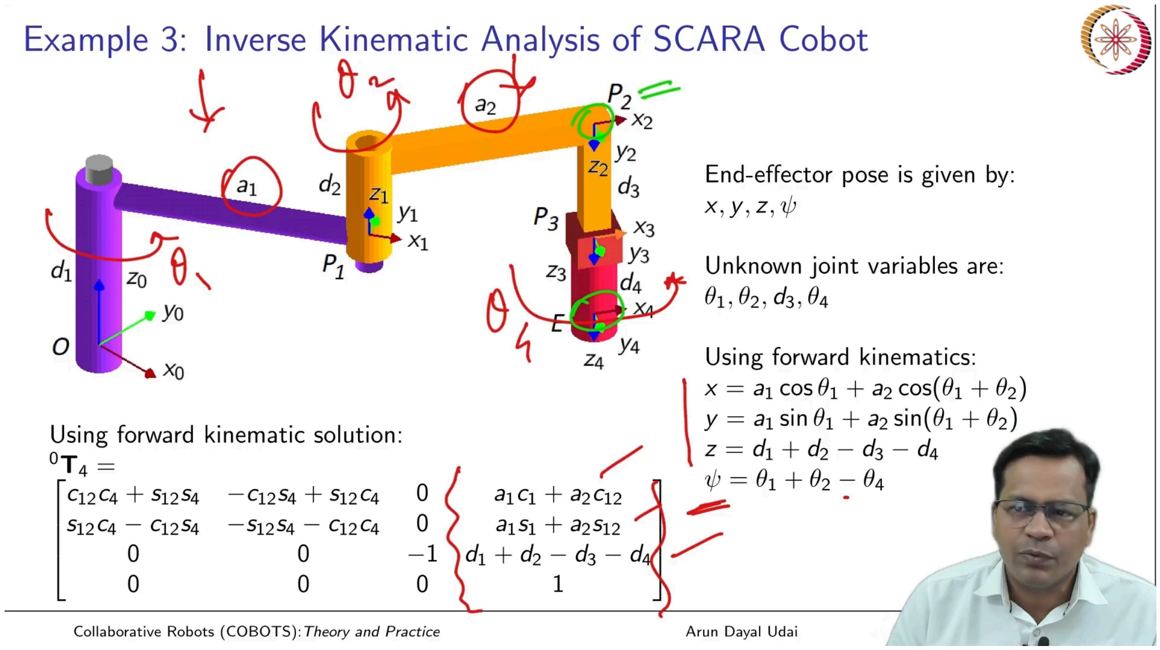
So, you see, it shows all the axes here. This is the second joint moving, and the first joint is also moving. So, this is how it can operate. So, the only difference from the standard ABB SCARA robot is it has the first axis which is prismatic. Both are equally good; we will see using the kinematic equation also.

So, let us come here. So, this is what your SCARA is again. Let me just show you the technical specifications of this.



You can see it. So, yes, you see the first axis, this one from here to here. So, this is your linear axis. This goes up and down. So, you have another one, which is the prismatic axis and a revolute axis, which is here. You have a revolute axis here, and you have a revolute axis here. So, three are revolute axes. The first one is a prismatic joint.

This is how it looks like. So, these are the specifications. Nowhere is it written which standard it follows. I do not know if it is purely a cobot because a cobot has to follow ISO 15066. So, that standard is not mentioned here. So, any robot that allows you to move by hand is not a cobot. So, mind it. So, this may not be a cobot because it does not mention the standards that it follows. It may not be safe for humans all the time. So, yes, but because it is a low-powered robot and a desktop robot, it may be safer.

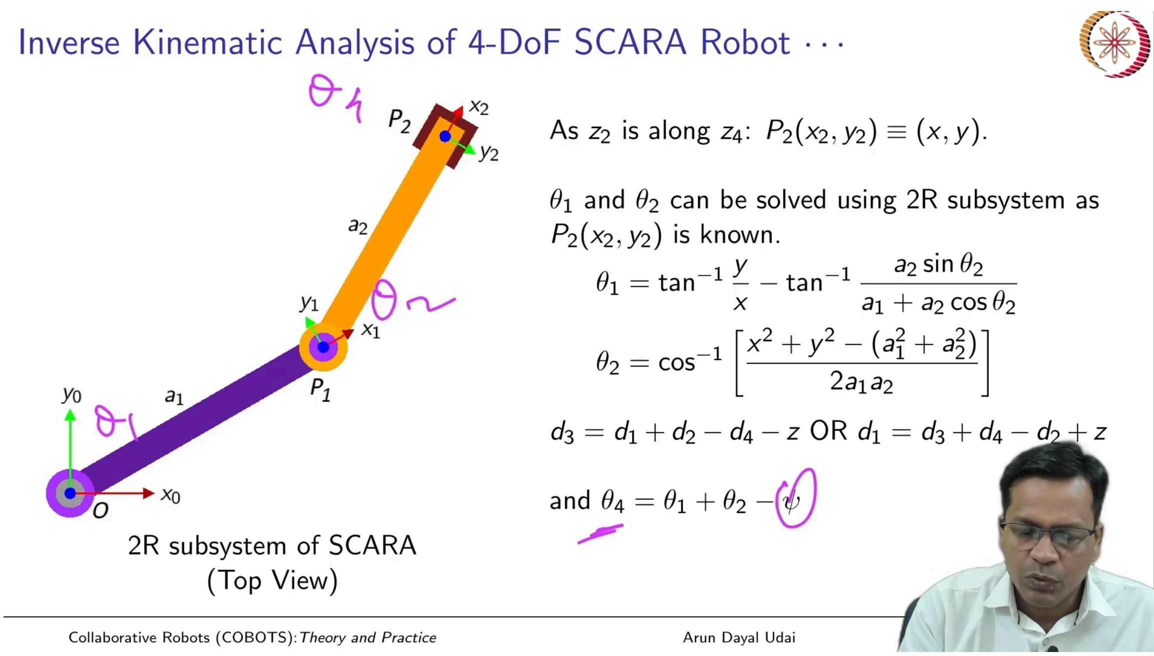


So, now let us go and do inverse kinematics for this. So, inverse kinematics of SCARA again, I will start with the forward kinematic equation that we have obtained earlier. So, this is your forward kinematic equation for the end effector the homogeneous transformation matrix is written here, which gives you the orientation and position of your end effector frame with respect to the base.

So, what is known is the end effector pose, that is, the position XYZ of this end effector and orientation. This robot cannot only go to any XYZ because it is a 4-degree-of-freedom robot, and you have a rotation. So, this last frame can also have a rotation axis. Okay, so that is the angle psi, so that is written here. So, you have to give the complete configuration of this robot by mentioning the end effector position and the angle psi. Okay, what is to be obtained is joint variables, that is, theta 1, theta 2, d3, and theta 4. In the case of this robot, the one which is shown here, d3 is over here. This is the prismatic joint, but in the case of the other SCARA, that is, the Dobot, you have d1, which is variable. Equally good, both are vertical axes, so you have only one vertical displacement. So, it doesn't matter if it is at the beginning of the robot or at the end of the serial chain, at least in this case. That is not always true, but in this case, you can have either one of them as a prismatic joint, and it is equally good.

So, now, using forward kinematics, I can quickly write this over here. So, you have x, you have y, you have z, which is x, y, z, and you have psi, which is nothing but the sum of all the angles. So, that is theta 1, theta 2 and theta 3. The sum of all the angles will actually make it to psi. But in the case of theta 4, it is actually counterclockwise, which is positive because it is pointed downwards. So, if you look from the top, it is like this. So, that makes it negative here.

So, now, I will use this to solve the inverse kinematics. So, if I look at it from the top, what do you see? What do you see here? So, it looks like a 2R subsystem. If I look at it from the top, if I go back, what do you see? From the top, you see a1 and a2 are the link lengths for which theta 1 and theta 2 are the joint angles. So, whatever is the position of my end effector, the same is the position of the P2 point, which is vertically above the end effector. Is it not? If you look at it from the top, it becomes like this. So, in this case, it is exactly equal to the same coordinate, which is there for the end effector. So, if it is x, y here, then this also will have the same coordinate x, y over here.



So, this is again a 2R subset as Z2 is along Z4. So, Z4 and Z2 are in the same line. So, that makes it like that. So, this is it. So, now you can quickly obtain. So, you have already solved for it. a1 and a2 are the link lines. You have theta 1, and Theta 2 are the joint angles. They can quickly be obtained using the 2R subset again, and you got this-

$$\theta_1 = \tan^{-1}\frac{y}{x} - \tan^{-1}\frac{a_2 \sin\theta_2}{a_1 + a_2\cos\theta_2}$$
$$\theta_2 = \cos^{-1}\left[\frac{x^2 + y^2 - (a_1^2 + a_2^2)}{2a_1a_2}\right]$$

How much is d3? So, you see, if you just go back and see your slide. So, only d3 is variable. d3 is variable; d1 is constant. d2 is also constant, which is not varying. d4 is also constant, which is here. d1, d2, and d4 are constants; those are not varying. So, what is varying is only d3, and that is making the change in Z.

If I write it here directly, I can write d3 as d3 is equal to d1 plus d2 minus d4 minus z,

$$d_3 = d_1 + d_2 - d_4 - z$$

Or, in the case of Dobot, I can write it like this-

$$d_1 = d_3 + d_4 - d_2 + z$$

So, in that case, you have d1, which is a variable, and you can write it like this. So, this is in the case of Dobot, and this is in the case of the ABB robot that I have shown. Many other industrial robots are of this kind only.

Now, if you have to get theta 4, you can quickly write it as equal to theta 1, theta 2, and the final angle that you also had. That is the variable theta 4;

$$\Theta_4 = \Theta_1 + \Theta_2 - \psi$$

Psi is already given, which is the end effector orientation. The yaw angle. So, you just use it, and you can get theta 4. So, you have solved for theta 1, theta 2, d3, and theta 4. So, this is how a SCARA robot is solved.

So, that is all for this lecture. In the next lecture, I will discuss the Inverse Kinematic Analysis of a 6-degree-of-freedom Robot, which is a Wrist-Partitioned Arm.

Thanks a lot.