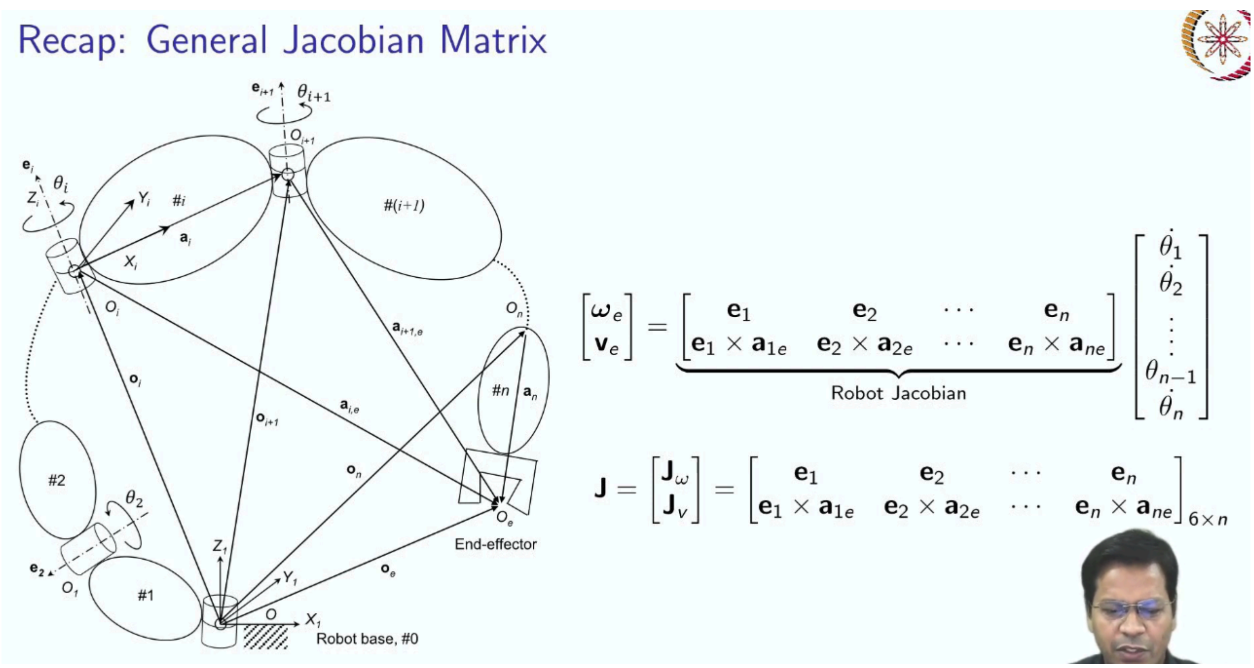
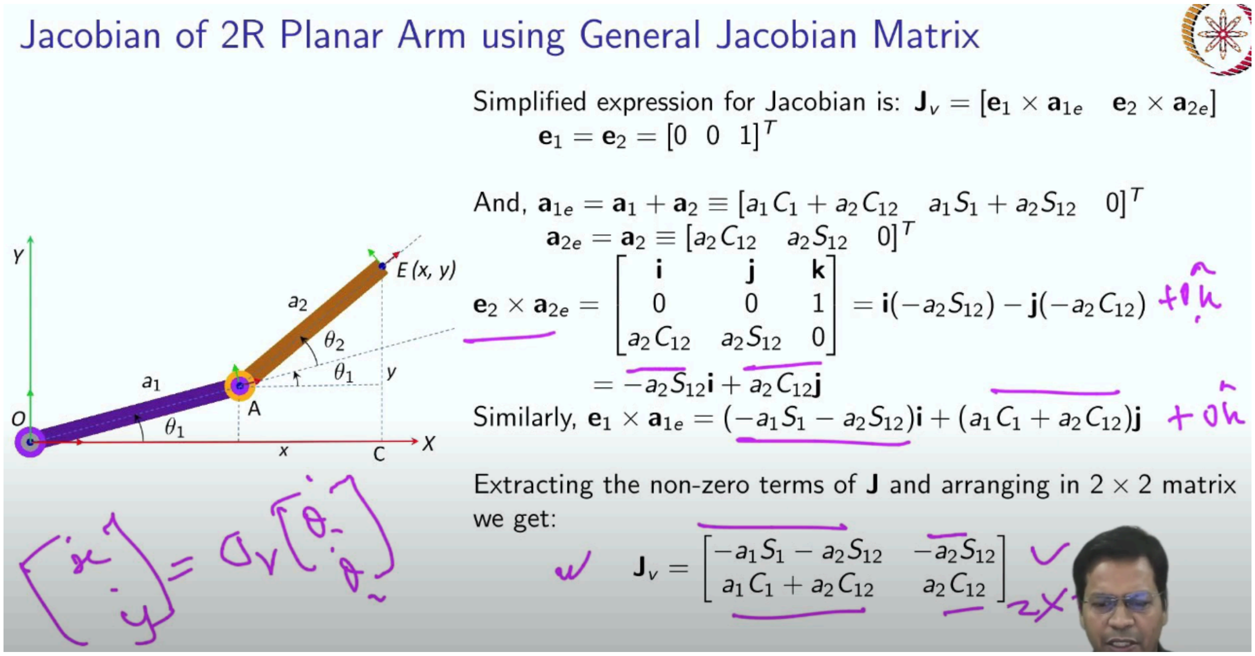
NPTEL Online Certification Courses
Industrial Robotics: Theories for Implementation
Dr Arun Dayal Udai
Department of Mechanical Engineering
Indian Institute of Technology (ISM) Dhanbad
Week: 05
Lecture: 25

Jacobian (2R). Jacobian: Inverse, Singularity and Acceleration Analysis

Hello everyone. So, in the last class, we discussed Jacobian. That relates the joint rates to the end effector rates. We saw how a Jacobian can be calculated for a 2R planar arm. We also saw a vector matrix approach for calculating the Jacobian of generals and degrees of freedom of the robotic arm. Moving further today, we will be discussing general Jacobian formulation further, and we will do Jacobian for a 2R manipulator using a vector matrix approach. We will understand the singularity of a robot, and we will also do an acceleration analysis. So, let us move further.

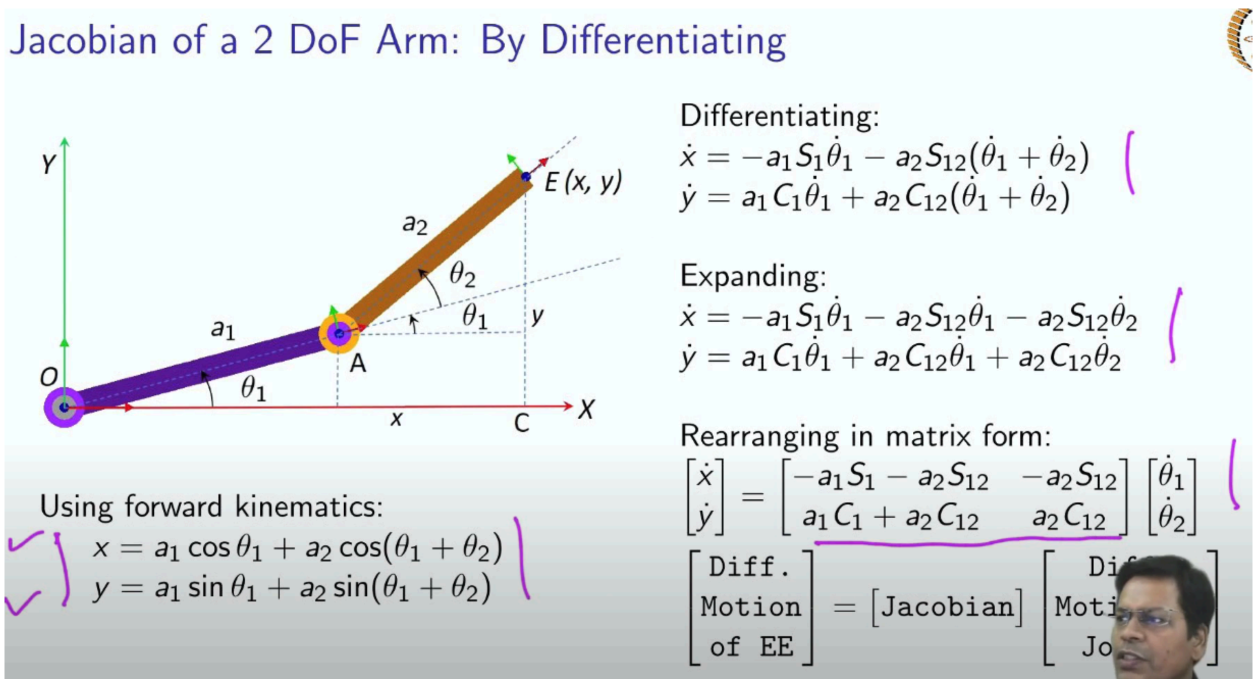


So, this was our general Jacobian matrix. So, let us just quickly recapitulate what was there. So, a general Jacobian matrix can be written like this: This consists of an end effector twist that is here, which is nothing but a column matrix of 6x1. So, this is the Jacobian matrix, and these are the joint rates. So, in general, it can be written as j omega Jr, which is nothing but a matrix, which can be written like this: So this becomes your j omega, and this becomes your Jv. So, it is ei on the top three rows, and ei cross a1, ea2, e_{an}. Likewise, it has in this row. So, what were all the variables? ei was the joint axis vector, so this was e2, e1, e3, right. So, it was like that. For ith link, it is ei, which was oriented like this. So, these were all unit vectors, that is, along the axis of that link. And then you have ei a1e cross product. So, what is that? 1e is nothing but the position vector that connects like this. So, this is 1 to e, Then you have 2e, Then you have ai to e. Likewise, it will continue. So, you understood what the general Jacobian matrix is.



So, let us just use this and see if we can calculate the 2R planar manipulator Jacobian using the general Jacobian matrix. So, our 2 degrees of freedom planar manipulator looks like this. Let us just talk about the variables. What are they? So this is a1 which is the link length, and a2 is the second link length. Theta1 is the joint variable, that is, the joint angle angular displacement. That is measured like this. Then you have relative angle theta2. That is again measured by the extension of the first link. So, this is like this. This is all planar. So, simplify the expression for Jacobian because you know it is just a positioning robot. There won't be any angular velocity. Part of the Jacobian j omega will be absent, So let us not calculate it also. So, it becomes a simplified Jacobian. So, it has just e1, a1, e2, a2e. Those elements are to be found out for this particular robot. So, what are e1 and e2? You know e1 and e2 are directed perpendicular to this plane. It is perpendicular to the x-y plane, So it is. It may be given as 0, 0 and 1. So, it is directed like this. It is perpendicular to the plane of this. So, both become the same. One is placed here, and the other one is placed here. So, that is e1 and e2. So, now, what is a1e? a1 to the end effector. Where is your end-effector? It is here, So let us just connect it. a1e. So, this is the vector. This is nothing but a1 vector plus a2 vector.

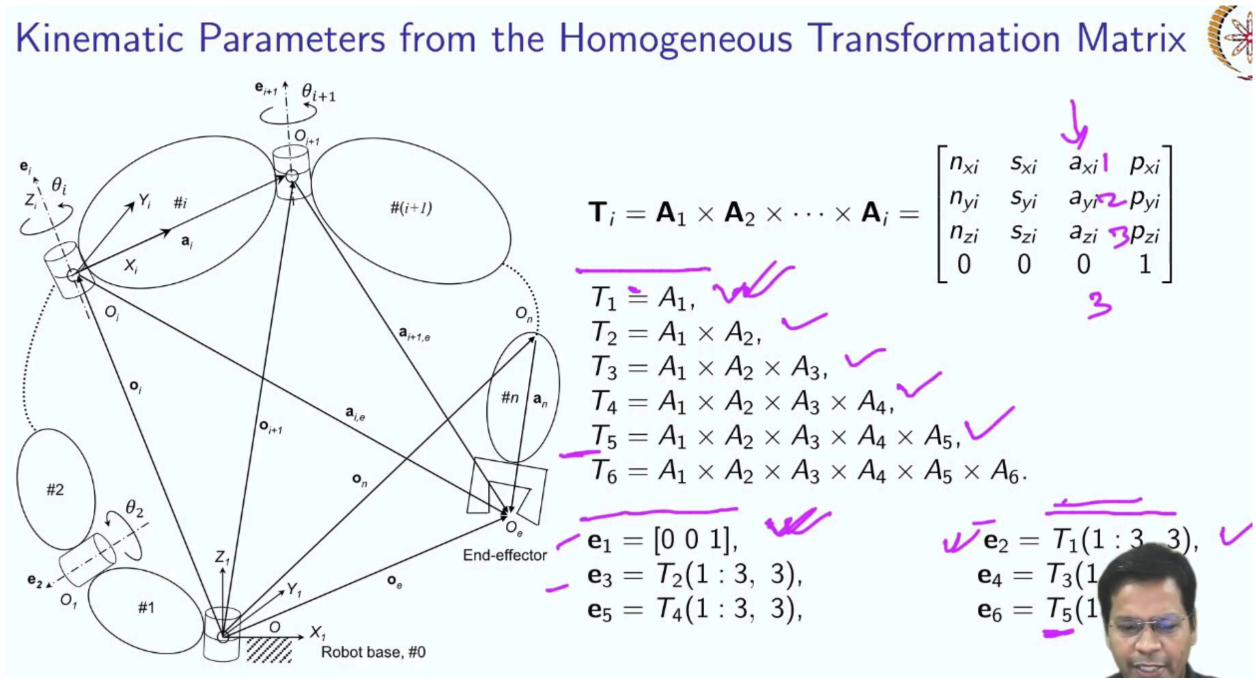
So, you already know what they are. So, how much is this? This is a1c1. This is a1s1. Similarly, what is this? It is a2c12, and this one is a2s12. Got it? So, if you calculate the x coordinate of the end effector, that is a1c1 plus a2c12. So, that is your x coordinate of the end effector. So, that is. I am talking about this point which is here. Similarly, this point is a1s1 plus a2s12 and because it is a planar manipulator, so if you have a point over here. So, with a coordinate set as zero. So, this becomes your vector: a1e, a1 to e. Similarly, a2 is nothing but this distance, It is. This vector goes from here to here, and that is very simple. It is simply a2cos12. That is this, and a2s12, that is this, and you have zero here. So, these are the two variables which are now known. So, we now know this. We also know this for both links. So, I can quickly take the cross product e2 x a2e. I have started from this end. So, ink, you can put it like this: Standard method of taking cross products. So, it becomes 001. So, this is your e2, it is 001 and a2e that goes here. So, taking the cross product, you can quickly obtain this. Similarly, you can take the cross-product of e1 and a1e. So, what is that? If you do it similarly, using e1, that is this and a1e, that is this, and you can quickly obtain this. Got it? So, now I can put them together in the form of a matrix. I have just extracted out the zero term, so that was the k vector, which is absent here. Similarly, k, which is absent here, is 0k, so I have simply eliminated it, and I have written Jv as this. So, this is your Jacobian. So, if you write them in a column manner, so it is minus a2s12, it is here, okay, and a2c12, it is here. This comes here, and this one will go here. So, this is your Jacobian. This relates the joint rates theta1 dot and theta2 dot to the end effector rates, that is, x dot and y dot. Got it? So, this is 2 cross 2 matrices. Very simple.



So, now let us just compare what we have obtained earlier using a simple differentiation way. So, we did it simply by differentiating the forward kinematics equation; that is, x is equal to this, and y is equal to this. Taking derivative, expanding and rearranging in the matrix form. It gave me this.

$$\begin{bmatrix} -a_1 S_1 - a_2 S_{12} & -a_2 S_{12} \\ a_1 C_1 + a_2 C_{12} & a_2 C_{12} \end{bmatrix}$$

You see, it is the same. It is the same. So, this is how you can use a general Jacobian matrix for n degree of freedom serial chain system to calculate the Jacobian for your robotic arm. So, if you have all the information, you can quickly do it.

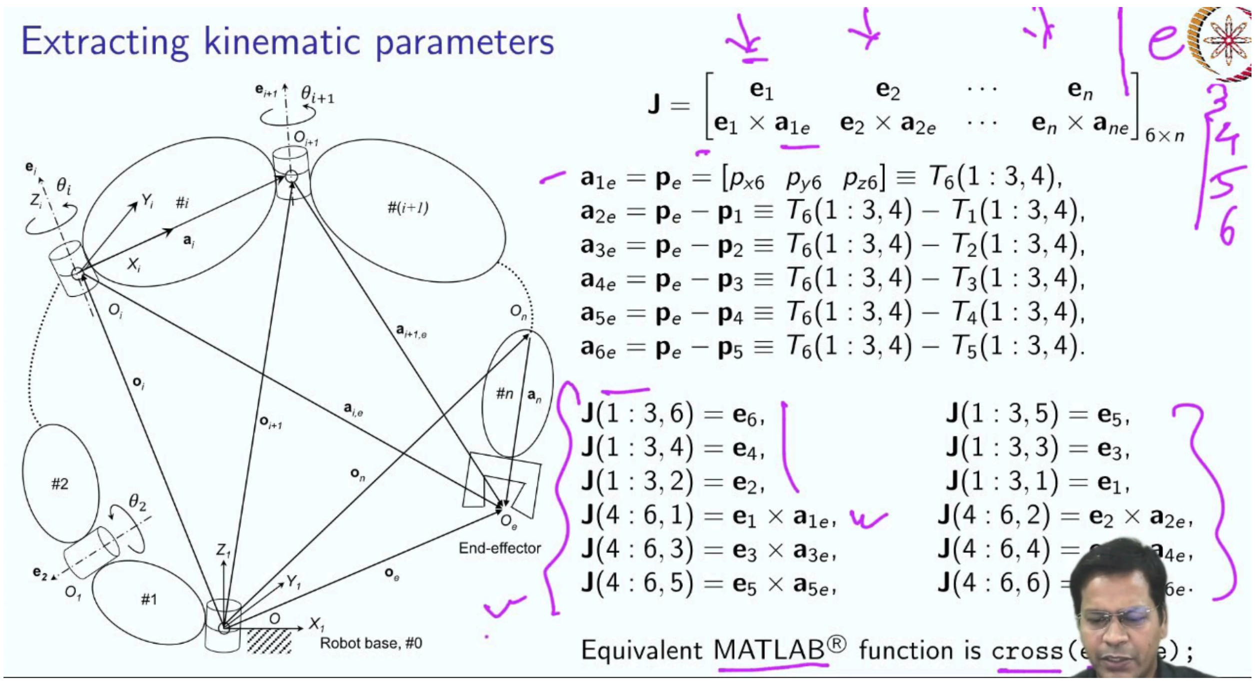


So, now let us work out where to extract all those parameters. Basically, ei and aie, were the parameters which were required to do Jacobian. So, let us just find out where they are. So, you know, these are all z-axes about which your link is rotating. So, if we talk about theta2, it is about this which was measured. So, if we just simply can look at the homogeneous transformation matrix, what are the information which are there? So what is this? This is nothing but the frame, which is attached to the ith link at the end of the ith link.

We talk about Ti. So, that basically includes all the transformation matrix: A1, A2, and so on, and so forth, till Ai. What is this effect? Actually, it is i minus 1 i. So, what was this? This was giving me the position of this frame (o_{i-1}). If you take it till i, it should give you this. So, what is that? That is nothing but the position vector of this. That is attached at the end of the link. So, this is attached at the end of the link. So, this is this one. So, it quickly gives me i plus 1.

Similarly, you can do it for all. So, if it is T1, it gives you the location of this. If it is T2, this one will give you the location of this. If it is for the sixth link. let us say this is the six degrees of freedom robot. So, it will give you the position of I assume this also includes the tool which is there all the end affected dimension, which is there because it is permanently mounted without any join, at the end of the last link. So, if that is included, this is your position that will be given by T6. So, one, two, three, four, five, six. At the end of the sixth link, you get this position. This, you know, is always equal to zero. It is attached to the base frame of the robot. So, you start from zero, go till n for n degrees of freedom robot. So, you will quickly get the vectors which are directly connected from here to this, here to this, like that. So, you will obtain them all. So, this is just giving you this information.

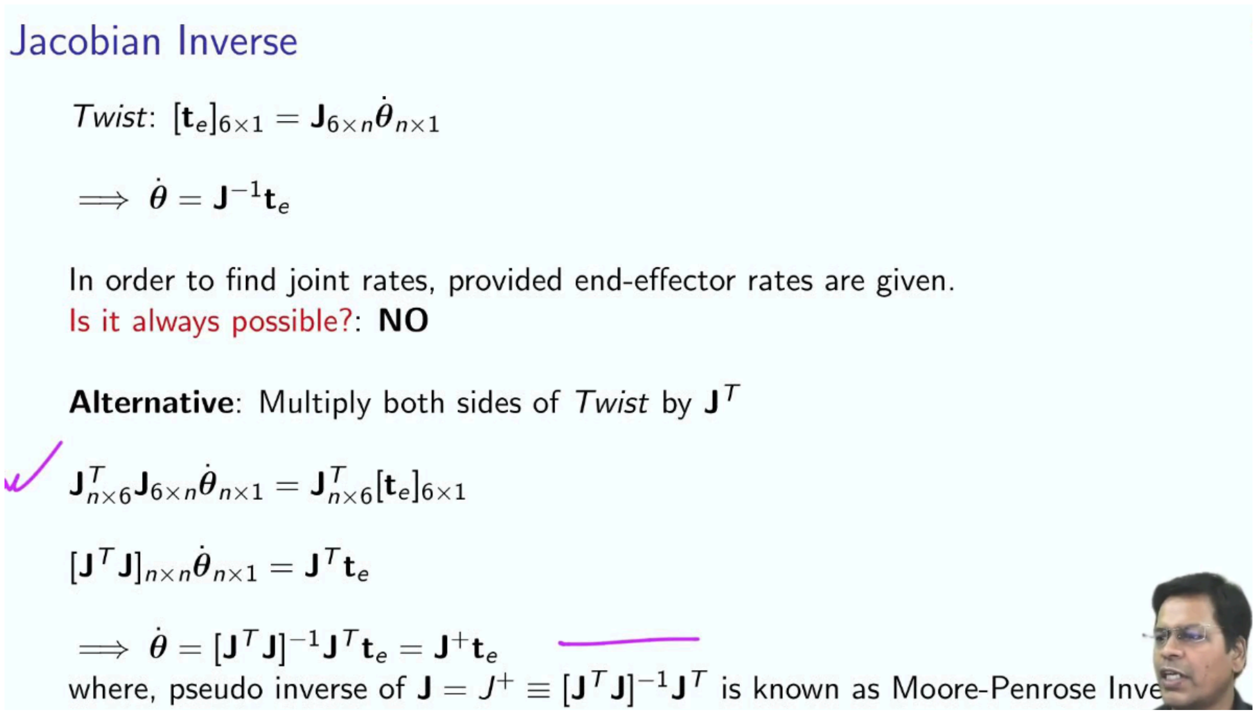
Now, let us move ahead. So, what is this? This is e1. e1 can be straight away written as zero, zero one. Why? Because that is the first axis which is placed at the base of the robot. So, you know, the displacement axis is along the z-axis. So, in the case of row three, it is the same. In the case of prismatic, it is also along the z-axis. So, E1 becomes zero, zero one. What is e2? e2 is nothing, but that is placed at the end of the first link, so you can quickly extract it from here at the end of the first transformation matrix. So, that is also equivalent to A1. So, it is this row of this. So, it is one, two, or three elements of the third column. What if? So that is here. And similarly e3, e4, e5 and e6. e6- you have ended till here because you know this is the one, the second last one. So, that is exactly giving you the location, which is here. So, that is your e6 axis. Six vectors are placed at the end of link five, so that is where you will obtain it.



Now, you just see if you know all these terms. So, this is what is required. So, e1 is known. Now you have to calculate a_{1e}, a2e, a1e. So, where can you obtain that? You know this can be obtained directly by T6, which is a homogeneous transformation matrix. That comes then there a1, a2, ai, ai plus 1 and finally an. So, until you take the product of all of them, you will reach this point. So, if it is six degrees of freedom, robot, you have to take product till six, and at the end of six, you have this frame. So, that will give you the tip position. So, that is T6. So, this is your a1e. How you will calculate is one to e. So, that is nothing but a frame one on how to calculate a2 to E. Let us see if this is your two to e. How will you obtain it? So you have to take the difference. So, you already know this. This end effector location is already known from the T6. This column is basically giving you one to e so that you already know. So, this you have already obtained.

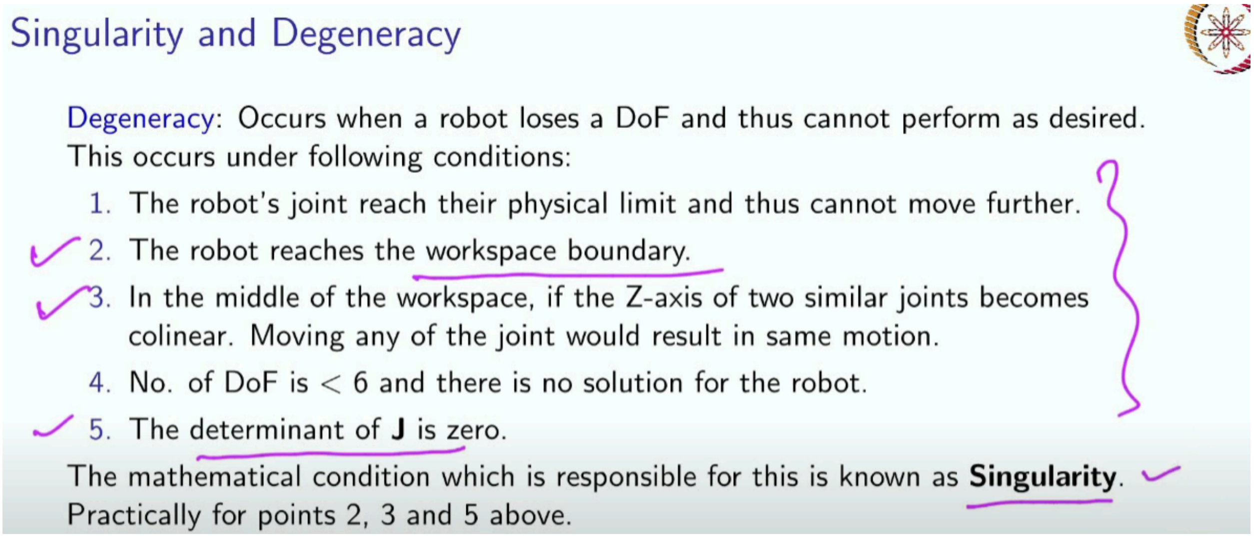
That is pe. that is the end effector location. Now you already know this also. Now, that is basically the T1. So, if it is T1, you have reached here, T1. This is so it is the fourth column, the first three-row elements. So, that is your vector, which is this, got it? So, now, using this vector triangle, you know this, you know this, you can quickly calculate that is a2e so that that is pe minus p1. So, this is your p1, got it? P's are all-if it is homogeneous- transformation matrix. This is Px, Py and Pz one. So, this is the column that I am talking about. This will give you any transformation matrix. So, this will give you the position vector of the corresponding frames, which are placed at the end of the length. So, this is it. This is P1.

Now, let us move ahead and see how you can calculate for, let's say, if it is a3e. That is this from here. So, how will you calculate it? So you have to take the product of a1- a2. That means you have reached this point. So, this vector is now known. You already know this vector, so using a vector triangle, you can calculate this very easily. So, that uses T2, that is, a1 into a2. So, that is how you can obtain any of them you can obtain. So, this is how you will obtain all the vectors. And now you already know a1e, a2e. These vectors are known. This was known previously. So, now you have to take cross vectors. So, what are these?
Jacobian, if you remember this, the first three one and the next three one. The first three were e, and the next three were e cross, a1e, a2e, a_{ie}, like that. So, you have simply put those values into this matrix. These are just element-to-element transfers using the cross-product. So, cross-product results will go to elements in rows four, five, and six, in rows one, two, and three. This will go, got it. And for all the degrees of freedom will come like this. So, this is how it is done and the equivalent function in MATLAB. It is cross cross-product of e6 a6c. It is just a method to take the cross-product. If you know these two vector, you can quickly do that. So, this is how it is extracted out of your standard kinematic transformation matrices, and you can do Jacobian, so this is how Jacobian can be obtained using any programming technique.



Now, Jacobian inverse- why is that required? You already have this in hand. Twist vector-that is nothing but a vector containing angular velocity and linear velocity. Over here, which is known as the twist, is equal to Jacobian times of joint rates. If it is n degree of freedom, Jacobian is 6 cross n, and this is n cross 1. And degrees of freedom will have n joint variables, and this is always 6 cross 1 because you know, in the end, you have six variables, only three for position and three for rotation. So, that is there always. So, if you have to take the Jacobian inverse, that is, to obtain joint rate.s. This is already given. So, if you know the end effector rates, can you calculate the joint rates? So, this is the way you can do it. So, this is quite okay, as long as it is six degrees of freedom row dot. J is a square matrix: six cross six, so you can quickly take the inverse. So, this is at least true for this. But is it always possible? As you have seen, no, why, if it is more than six degrees of freedom row dot, it is not that trivial. So, what you have to do, you have to find out the Jacobian inverse because it is not a square matrix. You cannot do it. So, just start with this equation: multiply both sides with Jacobian transpose on both sides of this equation. So, what I have obtained is this here, and this is additionally here. So, now, what is this? This is j transpose j, not. This is a square matrix. It is n cross 6 and 6 cross n. j transpose j will give you n cross n matrix. So, this is now invertible. This is as it is. Jacobian transpose is here and the twist vector: okay. So, that is here.

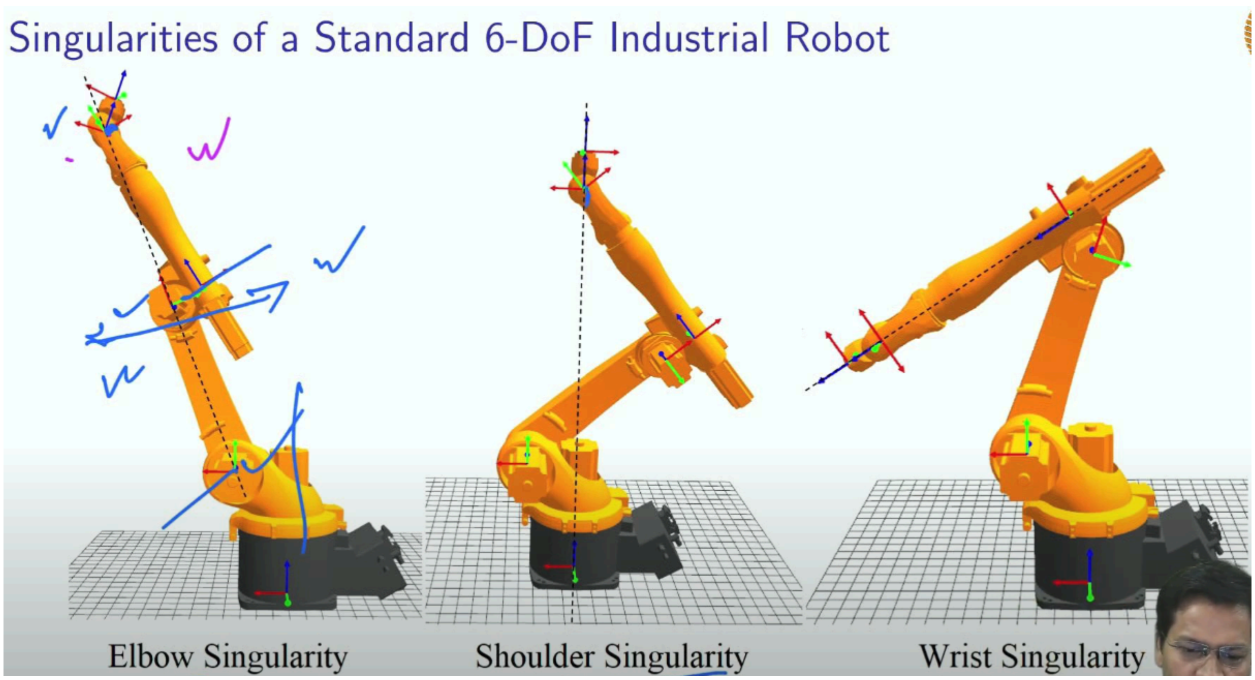
Now, I can take the inverse of this. So, now theta dot is j transpose j inverse j transpose twist. So, this is known as j plus, commonly known as a single inverse of j. It is also known as a Moore-Penrose Inverse. That is my standard method of finding out the inverse for a non-square matrix. So, this is what can be used for a robot for which degrees of freedom are more than six. If it is less than six, maybe five or four, a similar technique is there that can be adapted. So, this is the way. Even with this, it is not always so invertible if the matrix can be singular. In that case, this also is not doable, so what is that? So let us discuss that.



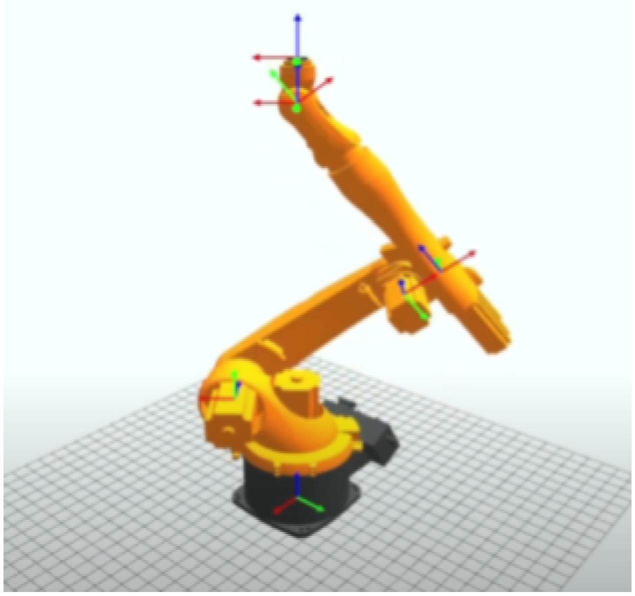
So, that is known as a Singularity. There is another term which is closely related to similarity. That is known as Degeneracy. It occurs when a robot loses a degree of freedom and thus cannot perform as desired. So, what is that? What are the conditions when this can happen? So the robot joint reaches its physical limits and thus cannot move further. You have fully extended your robot arm. And now you cannot do certain functions. It is not always true that you see a similar matrix in a few of these conditions. So, that is there. So, it is the robot that reaches its workspace boundary. This also is a case, the case which I have just discussed. Physical limits are the point limit. Your joint cannot go. Let us say this joint cannot straighten up more than this, so this becomes its joint limit, and thus it cannot do further action, it cannot move further. So, that was your first point. But reaching the boundary space is the extreme joint angle you have already obtained to go to a certain point in the workspace boundary. So, that is the second point. This is where have maximum reach of the robot is obtained. This is not the case where, kinematically, it may be possible, but your joint is not allowing you to go further. So, this is the case one.

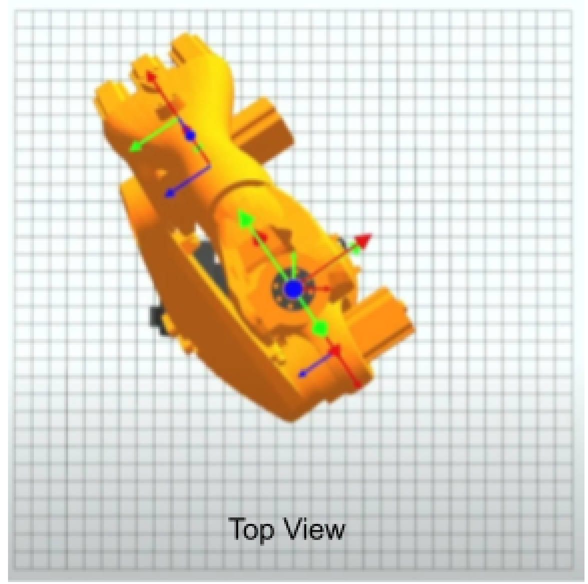
So, the third one is in the middle of the workspace. Also, if the z-axis of two similar joints becomes polynomial, I'll show you by example if the z-axis of two similar joints becomes polynomial. So, whatever you rotate, you see the same action. Let us say, if this link goes vertically over, vertically over this, okay, if it is like this, so what has happened? But if so, if you rotate this, or you rotate this, you are going to see the same type of motion. Well, because you have this axis and this axis becomes polynomial. So, this is also a case when it has become a degenerate robot. The number of degrees of freedom of the robot is less than six. That is maybe a case with discard. When it is a four-degree-of-freedom robot, you have seen that it cannot roll, it cannot pitch, but it can do XYZ translation, and it can rotate about the vertical axis. So, it can do four motions. Two of them were absent, that is, the roll and pitch motion of the end effector. So, it doesn't have any solution over there. So, that is also causing some sort of degeneracy. So, we'll talk about this. So, the determinant J is zero. This is exactly the case of singularity.

The determinant of J is zero. That means it is non-invertible. That is, you cannot obtain a joint rate for a given end effector rate. That is the end effector rate, Jacobian inverse, theta dot you wanted to obtain. Because this is not invertible, you cannot obtain the Jacobian inverse. So, me mathematical condition, which is responsible for this, is known as the singularity, the condition that you saw, which does not allow you to take the Jacobian inverse. That is what is similarity. So, you see, practically for point number two, three and five, two, three and five, you see they are a singularity, whereas degeneracy is. All these conditions lead to a degenerate rule. So, effectively, only these are when the determinant of J is zero in the middle of the workspace. Also, if both the axes become polynomial, they become singular, and mostly, in this case, when it reaches the extreme of the workspace boundary, your joints may become in a straight line. So, I'll show you by example also.



Now, standard singularity: six degrees of freedom, the industry of robots can have. So, you see what is this? Your wrist has come to its extreme position. It cannot go further. So, if it is, if it was an L-shaped link like this. So, this becomes your diagonal and that diagonal is now coming in line with this. So, all the frames are now in a straight line. So, you see, there is the axis in the dotted line I have shown. You see this, this and this tinfoil wrist-all the three are in a straight line. So, this is known as elbow singularity. So, this is your base joint, this is your shoulder joint, this is your elbow joint. So, it is the elbow that can oscillate. That can oscillate. Still, it can. This will be almost like no motion at all, so that is a very dangerous situation which can happen. So, this is your elbow singularity. What is shoulder singularity? It is exactly the position that I showed you when your last link axis. This axis is in line with this axis.

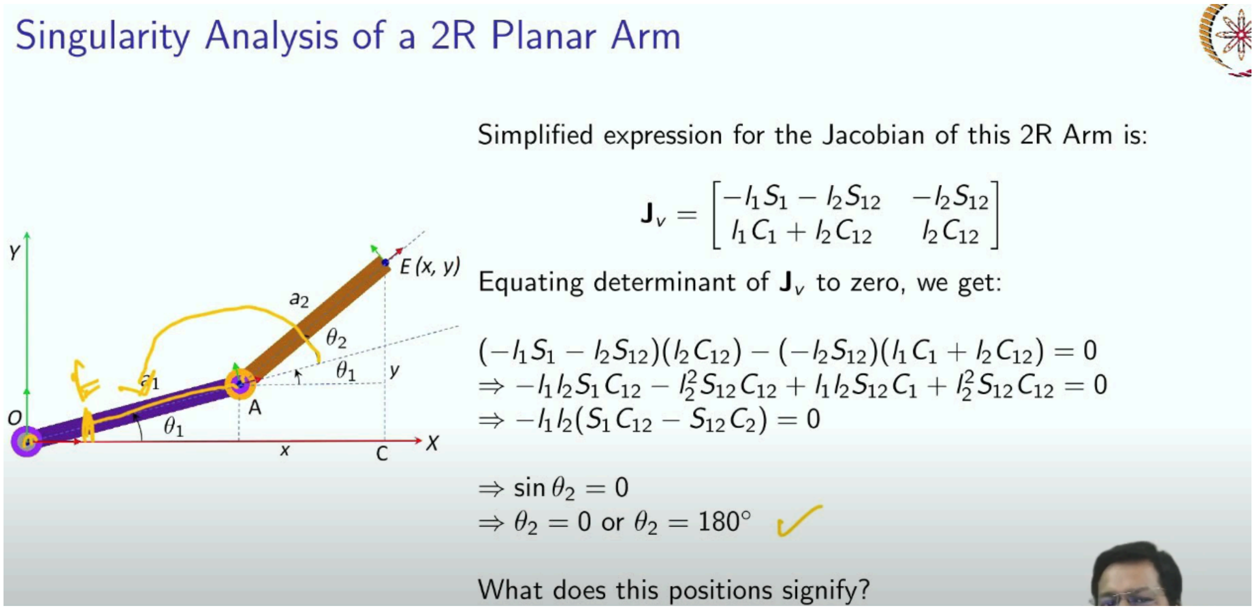




Let me demonstrate at least one of them. This one: you see what is this? You see, you can move the robot like this, you see, and the effector is statically at. The same position. The end effector is lying in the same position, and the position is not changing. And the same will be the case if you rotate the last axis. The last axis and the base axis are both in a single line. So, if you look from the top, you see it is not causing any displacement. This is the last axis motion, and the same will be the case when it rotates from the base. So, this is a typical case of shoulder singularity.

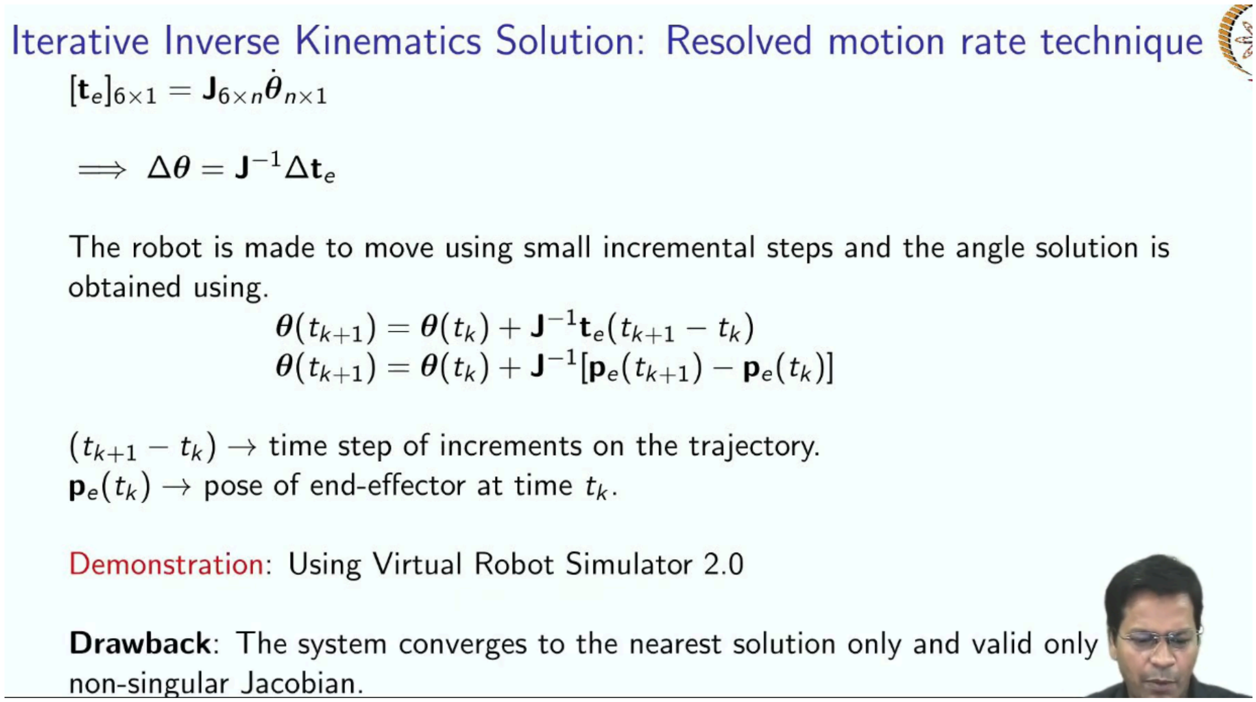
Wrist singularity: what is this? In this case, you have axis four in line with the last axis, but if two of them are in a single link, what is this? It is the frame which is here, that is like this, which is able to rotate it like this whole of the wrist. Now, you have the last axis, can also make the same movement, and both are in a single line. So, this is known as a wrist singularity. I can show you, even with this robot. You see this axis, the axis over here and the axis over here.

Both are in line. So, you can move like this: okay, that is there, and you can move like this. Both are in a single line. Okay. So, moving any of these will give you the same sort of motion. So, this also is a singularity, which is known as a wrist singularity. So, these are some of the similarities that industrial robot normally sees. Other than these, you already know if it reaches its boundary, like this, something like this. So, you can see something like this. So, it can be quite probable that it is also singular.



So, now let us analyse our planar arm, once again, for singularity. So, you know, this is your Jacobian, this is your Jacobian. You have just now derived for this. So, in order to make it singular, the determinant of this should become equal to zero. So, equating the Jv determinant to zero, I get this. You simply take the determinant of this, okay, this into this, this into this, taking subtraction in between, so that can be equated to zero. Solving it further, I got l here. This is trigonometrical equations. You can simply reduce it further, and you can obtain sine theta 2 is equal to 0.

You remember s1 is sine theta 1, c1 is cosine, theta 1, s12 is sine, theta 1 plus theta 2 and likewise. So, you see, sine theta 2 is equal to 0. So, this gives you a solution like this: theta 2 is equal to 0 or theta 2 is equal to 180 degrees. In both these positions, this angle becomes equal to 0. That means it is in a straight line it has reached here. So, it is a type of boundary singularity it has obtained, or you may call it the elbow is in a straight line to the base and the end. So, it is elbow singularity. Also, when it cannot go further, so this is our position. When it is a singularity in the case of theta 2 becoming 180 degrees, this is totally inward folded. So, this has come down till here and your link has come here. There, your end effector comes here, okay, so in this case, also, you have the end effector, elbow and the base joint. All three are in a straight line. And then again, you lose your degrees of freedom, and you become singular. So, this is also the case. So, this is the physical significance of that. This is how you can analyse for similarity of any given robot like this.

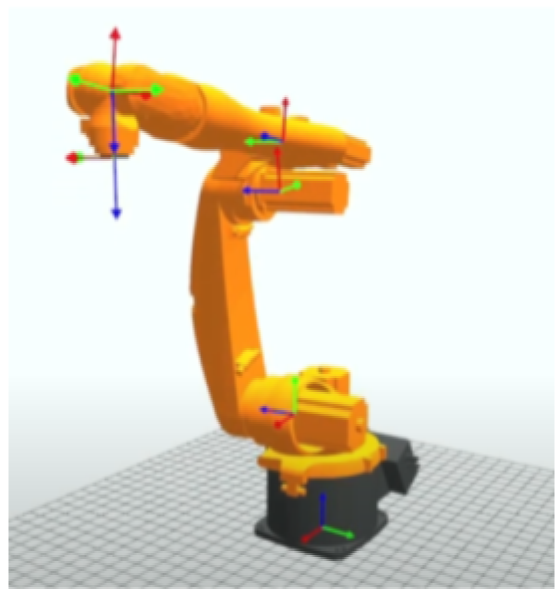


So, Iterative Inverse Kinematic Solution. That is also known as a resolved notion rate technique. Why is that used? You already know this expression. That is first equal to Jacobian into theta dot. This is a very well-known equation now for you because it relates the joint rates to the end effector rates. So, provided you are given end effector rates, can you know that? So you know you can take the inverse. So, that can be written as delta theta is equal to Jacobian inverse delta of twist. To understand it better, let us just say it has some delta x, delta y and delta z; maybe these other things become equal to zero. So, I am not talking about that for simplicity. So, this is your end effector velocity. So, that comes here. Velocity, because it is just a rate and with the same time variable, both are taking differentiated, so it is the relation holds true still. So, it is this: okay, this can be. The robot can be made to move using this, using small increments. So, this equation, you know, is true for very, very infinitesimally small steps. If you can move, you can do delta theta. Let us say your robot is already in a certain position. So, this is the place you can calculate your theta one, theta two and theta three. You can calculate your Jacobian. You can take the Jacobian inverse also for this. So, if at all you want to have this as a variable. This is not a constant. It changes at one, changes for each and every location of the robot, and every pose, it changes. So, this is a variable with theta angle somewhere in between. It's a non-linear relationship. So, that is there. So, this is your end effector, so this is your end effector rate. So, if you want to make it move with small incremental displacements like this and every small amount of time, so, you can calculate what could be the small amount of increment in theta one, delta theta two and delta theta three, so you can go to that new position, that is, if it is attached new position. So, you know now if you change by delta thetas in all the jumps you can take. It to a new position. So, this is what we want. So, without actually doing inverse kinematics and getting to the new joint angles, okay, you are able to calculate your new joint angles using this. So, you are able to displace your robot by a certain distance, a small distance, and you can make it move by incrementally changing your joint angle. Mind it, this is true only for very small displacements. So, this is your new equation. So, that is theta time of k plus one; that is, the next point is the first point plus j, the inverse of this.

$$\boldsymbol{\theta}(t_{k+1}) = \boldsymbol{\theta}(t_k) + \mathbf{J}^{-1}\mathbf{t}_e(t_{k+1} - t_k)$$

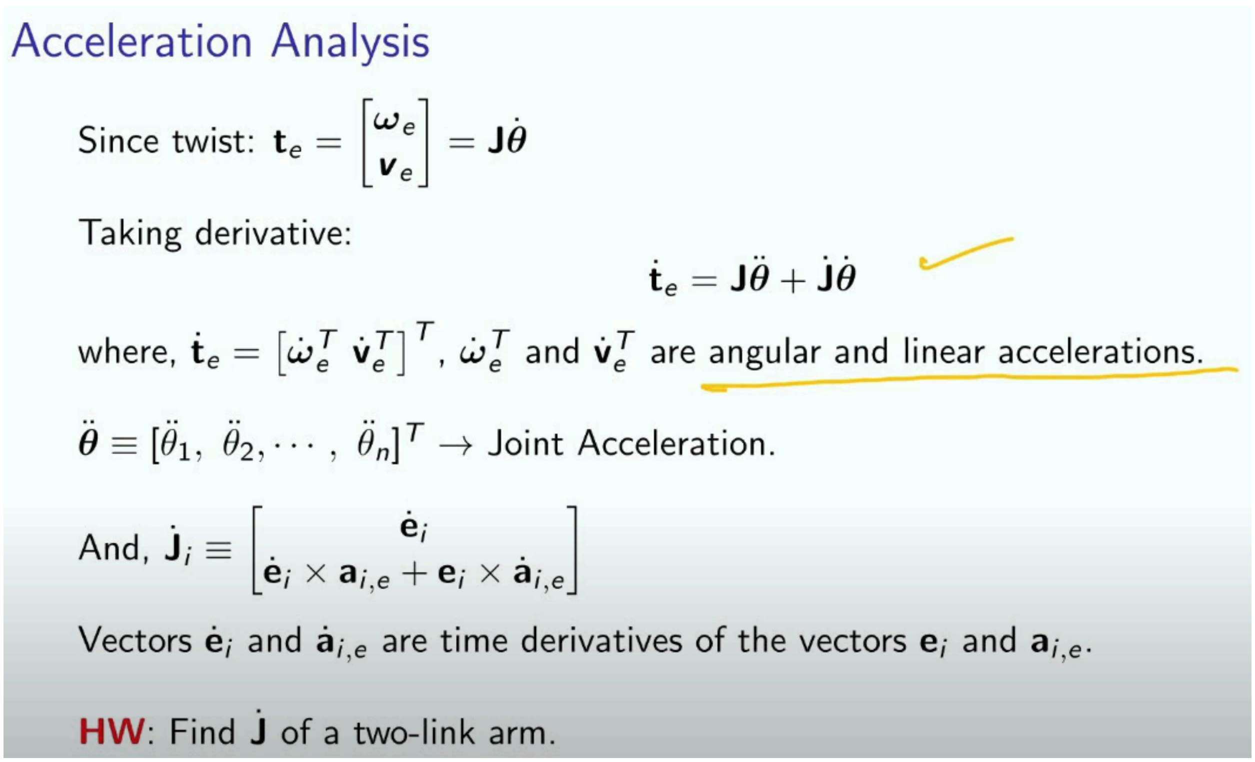
This is the delta theta. So, this is the delta theta that you are trying to add to the previous joint angle. Getting to the new joint angle for change in position.

So, it can be further written as. So, this is your change in the position of the end effector. This is where you want to go, and this is where you were. So, taking the difference and multiplying it with the Jacobian inverse gives you a delta theta vector, and that is added to this initial joint angle. Get to the new joint. So, this is all so that you can move this way also, so this is your time step. This is your pose of the end effector mind. It is force, and it is not just a change in position. It is a change in orientation as well. So, that is with time. So, I can show you this using the Virtual Robot Simulator that I was just using. I can make it like this.



Let us first try moving in Cartesian with this. Only let us see if it moves. So, if I want to move it along x by a certain distance of 20, I have fed here. So, if I start it, you see, it has just gone wrong. It says some error has popped up, and finally, it is unable to move. So, it cannot move from a singular position. So, that is, the limit is always there because this type of technique uses the Jacobian inverse, and this is unable to move. Let me just displace it by a certain angle, so I will not make it singular now. Let me take it to certain other, some other position. So, can I move it? Now, you see, it is not a singular one. Now, I will try to move by the same distance and see if it is able to move. Let me give a little more displacement. So, now you see, this is moving. You see, so I have given along y I have moved by 50. Let me move also along, z guys. Let's say 70. So, this is using the number of steps. I have taken 100. So, this is able to move. You see, it is moving along x, along y, along z, so it is able to perform. So, this sort of motion can be achieved using a small incremental step. For such a small distance like 70 mm, I have divided by something around 100 steps. So, it is a very, very small displacement it is doing, and that is why it is able to do it. It can reach where I have commanded it to go. So, this type of method works, at least for a small incremental distance and slope velocities you can very well attain this. So, this was coming back to my normal slide.

So, this is it. But the drawback you already know is that the system converges only to the nervous solution. Let us say you wanted to go to an inverse kinematic solution out of the eight different solutions. It won't take you there, and it will take you to the nearest one. So, that is sometimes beneficial also, and it is only valid for non-singular Jacobians. You cannot start. You cannot pass through a place which is singular.



So, let us do acceleration analysis also. So, we already know a twist is the Jacobian type of theta dot. So, this is a common equation that we have started quite a number of times today. So, this is it. So, if you want to obtain acceleration, we'll be using this. So, this is your velocity matrix. Derivative dot, you can write. So, it is j theta dot. So, it becomes j theta double dot plus j dot Theta dot.

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

It is just differentiation, nothing else. So, the same way, the way you differentiate two variables, the same way you can differentiate even vectors or even matrices. So, j is a matrix, and theta is a column. It is called a vector here, angular and linear accelerations. This is what you wanted to obtain, and whether things you needed here is j dot. So, what is theta double dot? It is nothing but joint acceleration. So, in order to obtain end effector acceleration, you need to know the joint acceleration also.

Let us say you are holding something at your end effector, and, if at all, it is moving with an acceleration. I want to be very sure that, with that acceleration, whatever the pseudo forces are generated due to inertia okay, it should not come out of the gripper. So, that is why sometimes acceleration analysis is also very, very important. So, this is it. And then, the j dot can be calculated like this: taking the derivative of ei cross aie.

$$e_i \times a_{ie}$$

That is what you did, so this was your general any with terms of your j matrix. So, you just take the derivative of this as well as this. So, taking a derivative of this can be expanded like this: okay, so now you know j dot. Theta, the double dot, is already known. So, these terms are known. So, now you can obtain the end effector angular and linear accelerations. Now, you can use this expression again on two degrees of freedom arm, and you can calculate twist rates, that is, linear and angular solutions.

That's all, thanks a lot. So, with this we'll end today and this module also. And the next module is very, very new for you all. So, that is industrial robot installation and commissioning, that is when you buy a new robot. What are the things that you do? That's all, thanks a lot.