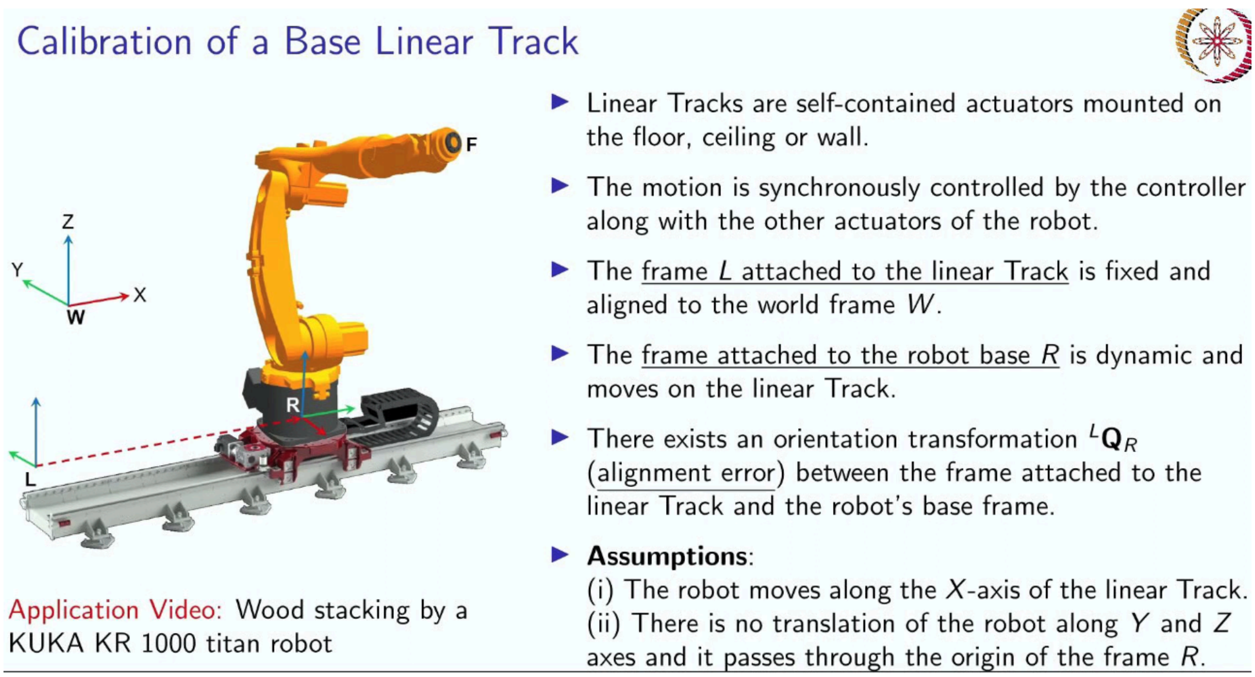
NPTEL Online Certification Courses
Industrial Robotics: Theories for Implementation
Dr Arun Dayal Udai
Department of Mechanical Engineering
Indian Institute of Technology (ISM) Dhanbad
Week: 07
Lecture 32

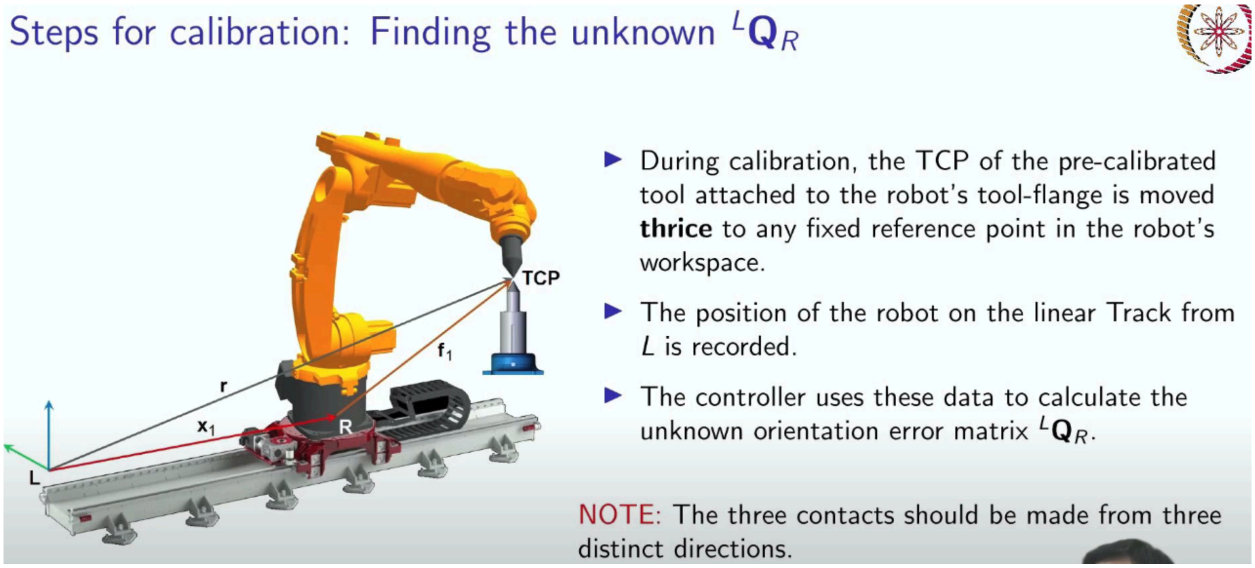
Base Linear Track and External Turn-Table Calibration

Hello, and welcome back to the module Calibration of Industrial Robot Systems. In the last class, we studied fixed tool calibration, in which we studied external TCP and workpiece calibration. Those were mostly static systems, and they were not controlled by the robot controller. In today's class, we will further extend the calibration of industrial robot systems, in which we will be doing base linear track and external turntable, which are also known as rotary axes. These are dynamic systems that extend the degrees of freedom of the robot and are controlled by the robot controller itself.

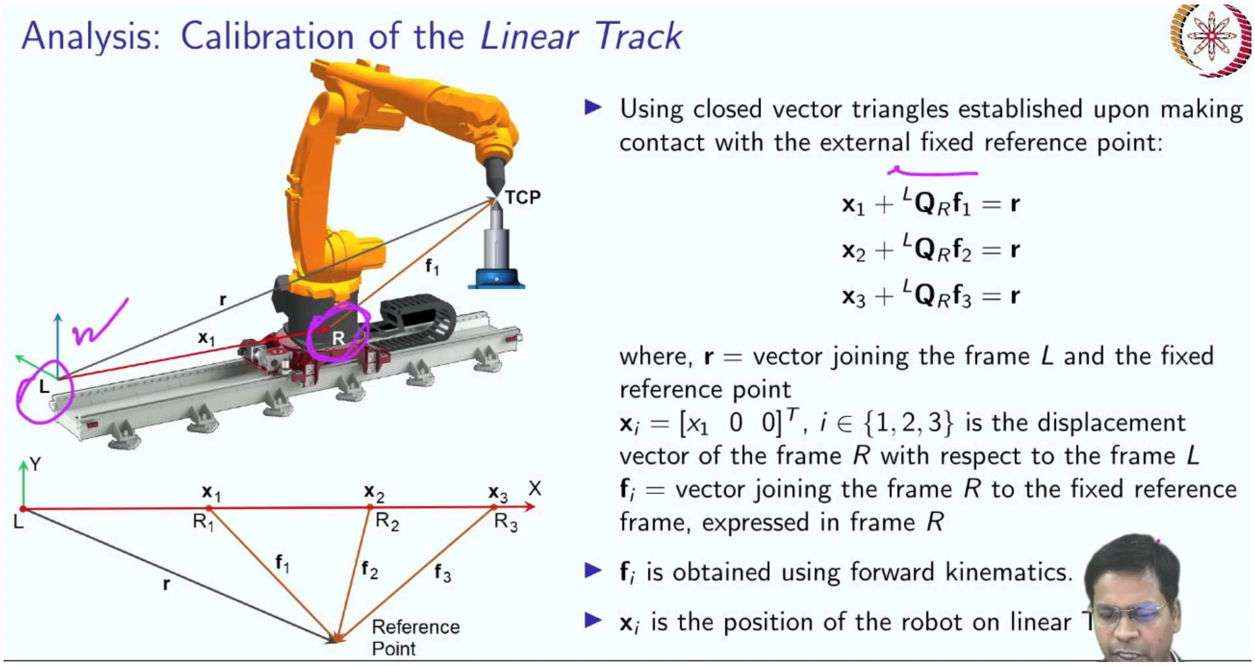


So, let us start. So, the calibration of the base linear track. What does it look like? You just see a robot is no longer grouted permanently to the platform on the ground. Rather, these are mounted on a linear track, just like a railway track That also has provision to cover all the cables and transport all the cables along with the motion that it can do. These are not static systems; rather, they are. These are like, just very much like a prismatic joint which can move the base of the robot, on which the whole of the robot can also move. So, let us just see a quick video that will demonstrate it. Well, this is a robot. You see, it is mounted linearly. It is moving, the base, the whole of the base is moving, and on top of that, the whole of the robot is also moving. These are basically meant to extend the workspace of the robot. A robot, let us say, can certainly go certain within its reach of the arm. If the base itself moves, it further extends that reach. This is a huge robot. The KUKA KR 1000 that is a Titan robot with an almost 1-ton payload capacity. Doing more stacking. Hope you got it. So, let us continue further. So, yes, the linear tracks are self-contained actuators. They are themselves actuators. They are mounted on the floor or the ceiling, or they can be on the wall, also.

So, that has been provisioned, just like a flag on which your robot can be mounted, So the motion is synchronously controlled along with the other axis of the robot. This controller also has provisioned to attach an external axis, So this becomes one of the external axes also. So, the motion is synchronously controlled by the controller along with the other actuators of the robot. The frame L is attached. So, let us just start putting the nomenclature here. So, I am calling the linear rail a frame L, which is attached to the linear track, which is fixed and aligned to the world frame. The frame attached to the robot base will call it R- is a dynamic system and moves on a linear track. So, this base frame of the robot, that is, R, will move on top of the linear rail. So, there exists an orientation transformation that we will call LQR. Q is the error matrix, that is, the orientation error matrix between L and R. Normally, we try to put the robot frame aligned to the linear rail frame, but it is not always possible due to some manufacturing defects or mounting defects or some other errors between the linear rail frame and the robot base frame. So, that is what LQR is, we will assume. So, what are the assumptions that we will put here? Now, The robot moves along the X-axis of the linear track. So, in every displacement that we can make to this linear rail, the robot will be displaced by the same displacement, and I am assuming it is moving along the X-axis of the linear rail. My second assumption is there is no translation of the robot along the Y and Z axis In reference to the linear rail. There is no offset along the Y and Z axes and X, and if it is there, it is only the active displacement, which is actuated by the linear rail joint. It is very much like a prismatic joint.



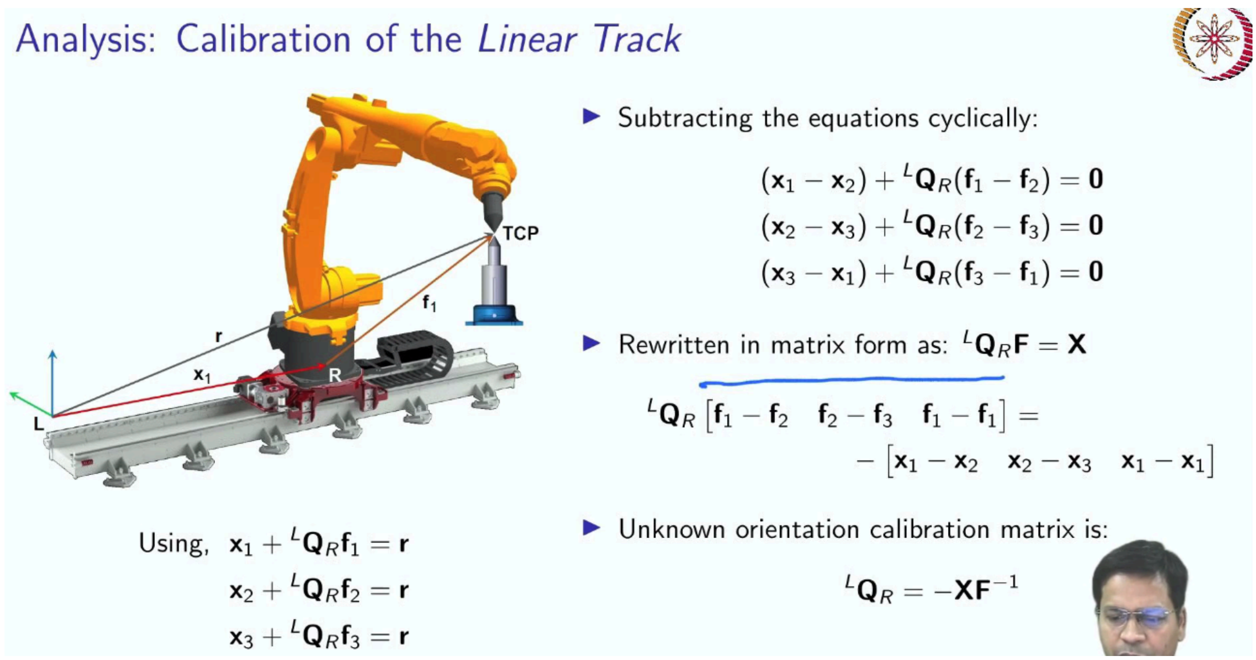
So, the steps for calibration are finding the unknown. That is $^{L}Q_R$, that is, the orientation error between the L and R frame. So, that is what is our job now, that we will do in this calibration. So, during calibration, the TCP of a pre-calibrated tool. So, there is a tool which isAttached to the robot's tool flange, and it is already calibrated. So, wherever the tip of this TCP will go, the tool centre point will go. We will quickly know the location of that TCP by the robot's forward kinematics. So, this tool flange-calibrated tool will now be moved thrice to any fixed reference point in the robot workspace. So, there is a fixed reference point in the robot workspace. So, this robot is now made to touch from three distinct directions to the same location. The position of the robot on the linear track from L is recorded, that is X. That is the displacement which is here. So, it can be X1, X2, X3, like that. So, that is recorded. The controller uses these data to calculate the unknown orientation, that is, the error matrix, LQR. This is what this calibration is all about. So, the three contacts should be made from three distinct directions. They should not be very close to each other. That will cause. Later on, when you see the calibration approach for this, you will find that it is going to create some sort of singularity. So, you won't be able to get to the solution. So, that is why this is one at the very beginning of the calibration step.



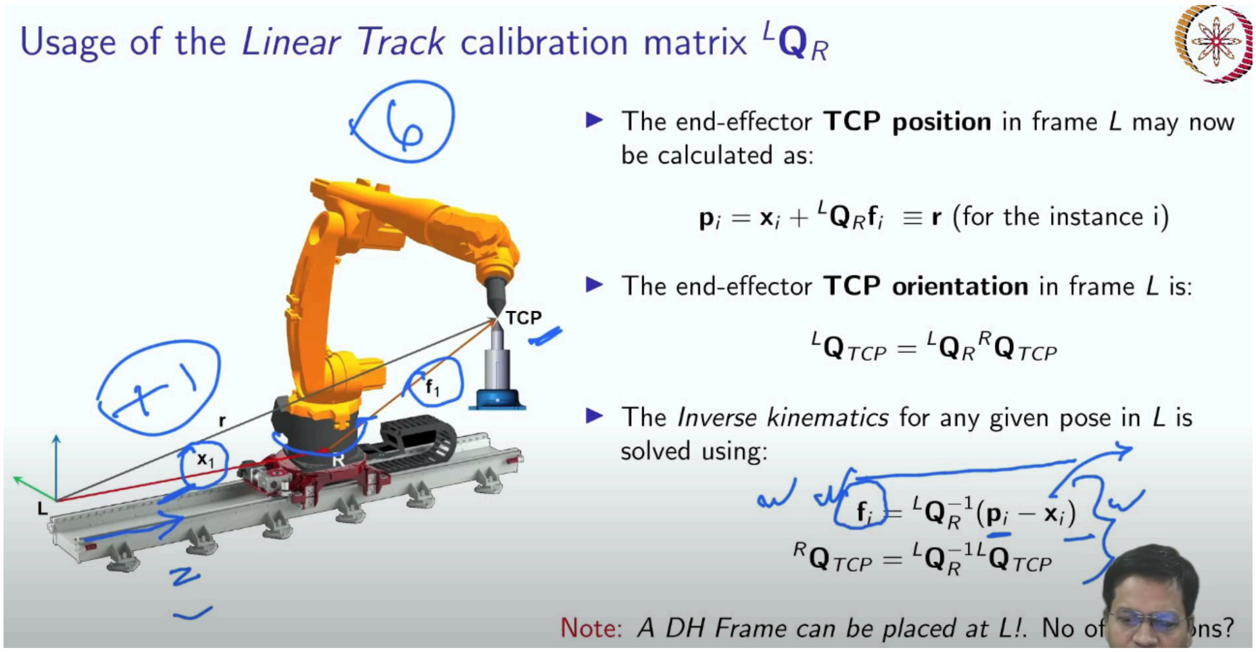
So, let us now analyse the calibration of the linear track. So, how does it work? So, you see what you did now. You simply approached it from one of the locations. So, what exactly did you do? You moved it along linear displacement along the linear rail. The base has moved by a distance of x1 from L. So, from L to R, linear displacement along this X axis of the linear rail is X1 from R to TCP which is the distance which is quickly obtained by the robot's forward kinematics. That means the TCP position with respect to the robot base, as well as its orientation of the TCP, can be quickly obtained using robot forward kinematics. So, because this is already a calibrated robot, you can quickly obtain the TCP location; that is, the f1 vector can easily be obtained. But in the robot's base frame. Now, this is your r. What is r? r is unknown because, you know, this fixed tool, which is there as a reference, which is kept outside the robot in the robot's work space, is free to move. You can search, and you can find any good place to touch your TCP from different locations, three different locations. So, this is an arbitrary reference point.

So, during the closed vector triangles that were established upon making contact with the external reference point, so you see, the robot was here x1 and using its forward kinematics f1, it touched upon this point. The robot is now here. Forward kinematics reached here. The robot was again here x2, and forward kinematics f2 reached here. So, every time you can obtain f1, f2 and f3 in the robot's base frame, R1, R2, and R3 locations. r is an unknown. X1, X2, and X3 these 3 are distances from L. So that is in frame L. If you call it a vector, if this is a vector which is represented in L frame, so it is [X1 0 0], [X2 0 0], [X3 0 0], it can be expressed like that. So, you can quickly write it like this: three different equations. Let me explain just one of them. You see what is happening here. So, this is. You just pick up any one vector triangle. Let me wipe it off a bit. Let me just take one of these triangles. So, see what is happening. So, R is the sum of the X1 vector and F1 vector. But you see, F1 is in this frame, whereas X1 and R are in this frame. So, from here to here, you know there is an error. There exists an orientation error. So, let us suppose you have a transformation matrix that gives you that orientation transformation given by LQR.

This is an orientation transformation from L to R. So, multiplied with LQR. Now, f1 will be in the L frame, that is, the linear rail frame. So, that is what I am doing. So, this is in a linear rail frame. This also is in frame L. This also is in frame L. So, all together, this is your closed vector triangle equation and the same for all three instances that you did. That is from here x1, x2, x3. Got it? So, r is nothing, but it is the vector joining the frame L and the fixed reference point. This is the tip of this reference tool. f1 is the vector joining the frame R to the fixed reference frame expressed in frame R. So, f1 is in frame R, Got it? So these are the two important elements which are here. This is what we have to find out. f1 is obtained using forward kinematics. The magnitude of f1. The direction of f1 is in the R frame. If it is obtained from forward kinematics, you get it in the R frame. But by multiplying with the transformation matrix from L to R, you can get it in the L frame. That is what we did. So, X1 is the position of the robot on the linear track, the position on the linear track. So, it is simply [X 0 0]. So, it can be written like this.



So, copying back these three equations that we have obtained. Let us move ahead, Subtracting cyclically, subtracting these. So, this minus this will give you the first one, and then this minus this will give you this one and then you have this minus this will give you this one. Got it? So that is how it is obtained. So, you got three equations, and they are independent of R. So, you see, this R has gone. That is unknown, So it has been eliminated here. Now you are left with this that is known through forward kinematics. This is known because you are displacing the robot on the L frame, which is a linear rail frame. So, you know this already. This is something which is unknown and that is there in all the three equations, so let us compactly put that like this: That is in matrix form, So I can quickly write it as taking this common: So this is here, this is here, this is here. That is one full matrix and similarly, if you take it on the other side, You are remaining with a negative sign. Here and again, those are put like this: So X, any vector X, is X00, and scalar values are there. Similarly, forward kinematics will have FX, FY and FM. So, that is a 3x1 vector. So, all aligned in this way, this is a 3x3 vector. This is also 3x3. Orientation error is also a 3x3 vector. So, it is like this. So, you see, now you can write it as you just take the inverse of this. Multiply to both sides of this equation, and you will get LQR as this. So, you have obtained the orientation calibration matrix. That is nothing but an orientation error that is there between L and R frames. So, that was our objective, and we have obtained that. Got it? So, it is quite trivial. Now, you are capable enough to handle any external TCP point.

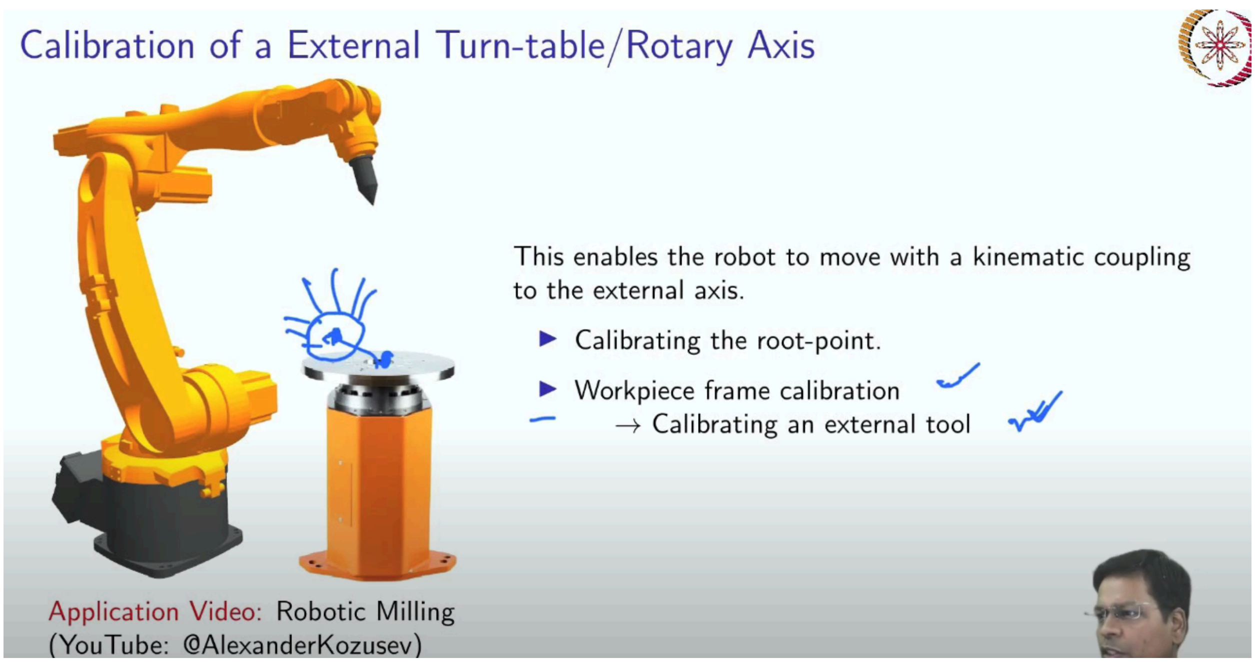


Now, the end effector TCP in frame L may be calculated as this: So you can simply calculate any point f which is in this frame. You can now get that point in this frame. So, if you reach TCP using your robot, you have this. This was something that we have identified just now. This is the linear displacement that is there. So, now the same point is this one: Got it? So this is equivalent to R, as we have just seen. So, for any instance I can quickly find out the location of this tip of your robot in L frame. Got it? So the end effector TCP orientation in frame L is L to R, R to TCP. So, effectively, what you will get is L to TCP, that is, the orientation of the TCP in frame L. So, this is your orientation, this is your position, Got it?

$$^{L}Q_{TCP} = {}^{L}Q_{R}\ {}^{R}Q_{TCP}$$

The inverse kinematics. Let us say I want to go to some position given by pi. So, how do you get there? You simply know using the same equation. I have obtained this. So, this is what is to be done using inverse kinematics. So, you have obtained fi. This vector is now extracted out and using this, if you already know how much is the orientation that you want, L to TCP, So this was your calibration matrix. So, you got what R to TCP. So TCP orientation with respect to R. So, that becomes the input for the inverse kinematics algorithm. So, you know the orientation and position in frame R. So, that is what goes to the inverse kinematics algorithm, and you can do inverse kinematics. You can calculate the joint angles to reach there.

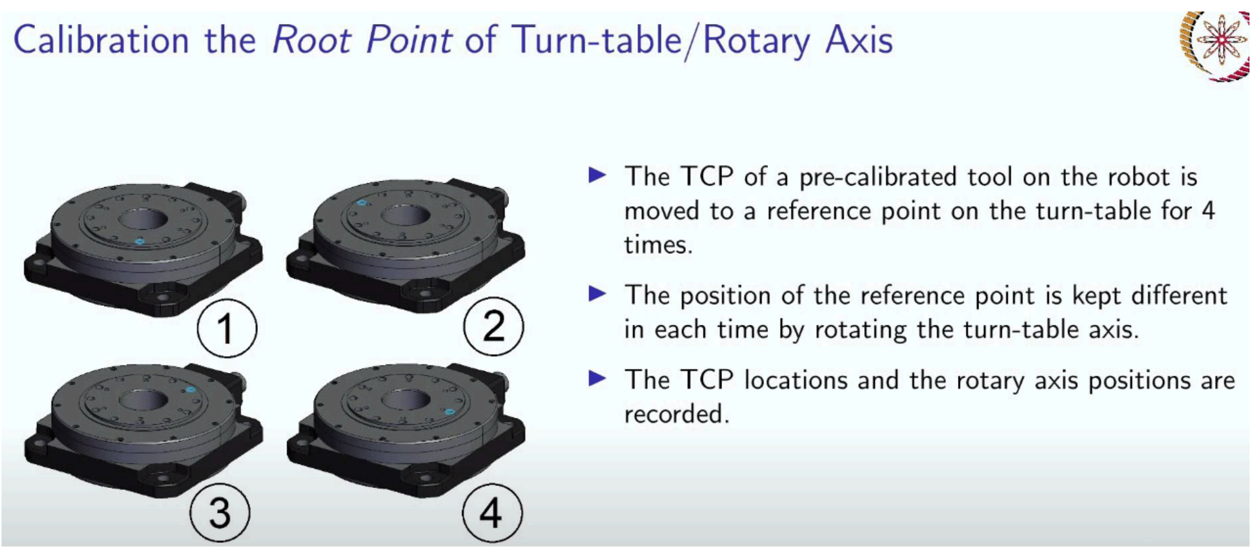
Now that we have assumed X over here, the whole of the system can be assumed, like 7 7 degrees of freedom robot, and you can express them in DH parameters, and you can do forward kinematics as well. That is quite easily possible. But normally, in industry, it is trended to use this as X. That is the reason I have taken it as X. Otherwise, you are free to choose it as X, Y or Z, whatever is okay. Normally, in the case of DH, you should take this as the Z axis. You know that that is a prismatic joint. So this can be your prismatic joint. The first revalue joint will be moving on top of this, So that is quite easily possible. So, how many number of solutions does it have? This inverse kinematics now has 7 degrees of freedom system. Six within this robot and plus one from this linear range. It is a seven-DoF system. What you can attain is a six DoF, that is, 3 for translation and 3 for orientation. So, it should have redundant solutions. So, how many solutions does it have? Just look at this carefully. You see, in this equation is possible to have a solution for every value of X. So, in order to go to a particular position for any value of X if it is within the reach of the robot. So, whatever the value of X, you should be getting f, is the TCP position with respect to the robot base, and it is solvable. It has eight different solutions for each f1. So, that is quite easily possible and this orientation also. So, your orientation changes with respect to the robot, but your orientation may be stable with respect to the linear frame, L frame. So, your robot will move, but your L frame will. Your TCP orientation will not change With respect to the R position. It changes, but it doesn't change with respect to this frame. So, hope you got the insight of this.



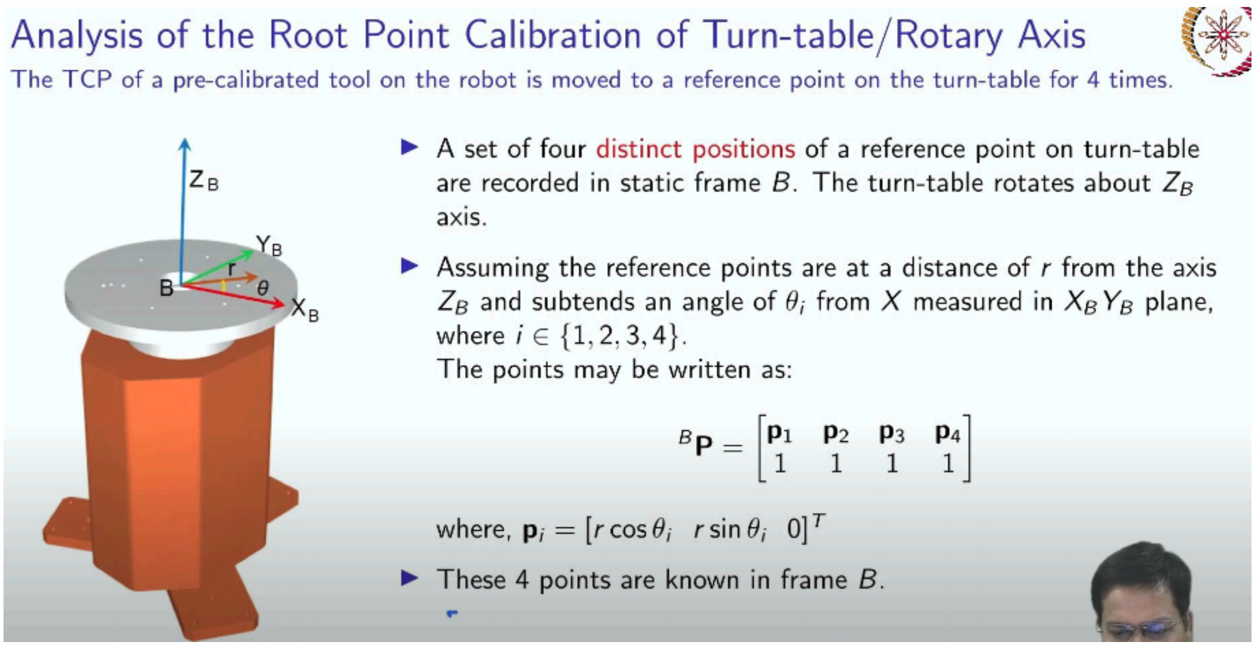
Now, let us move ahead. So, now we will be doing calibration of the external kinematic system. To begin with, we will do a Turn-table. That is also known as the rotary axis. So, what does it look like? It looks very much like this. You have an axis which is placed outside of the robot. It is scouted to the ground. It is fixed. This tabletop can rotate. So, it has an axis about which it can rotate, and it has a location also through which it passes. If at all, it is a circular disc, so that should be the centre of the circle, and there is an axis about which you can measure the angle of this rotary disc.

So, let us quickly see the video again. So, you see, this is one of the applications which is of CNC machining which the robot is doing? It is a KUKAN robot again. It is doing its job nicely, you see. So, the whole of the system needs to be calibrated. You should know where the centre of this tabletop is and also the location of this table with respect to the robot in order to express them in the robot's frame. So, that is what our job is. So, if this is your R frame, you should know where the centre of this table-top turntable is located.

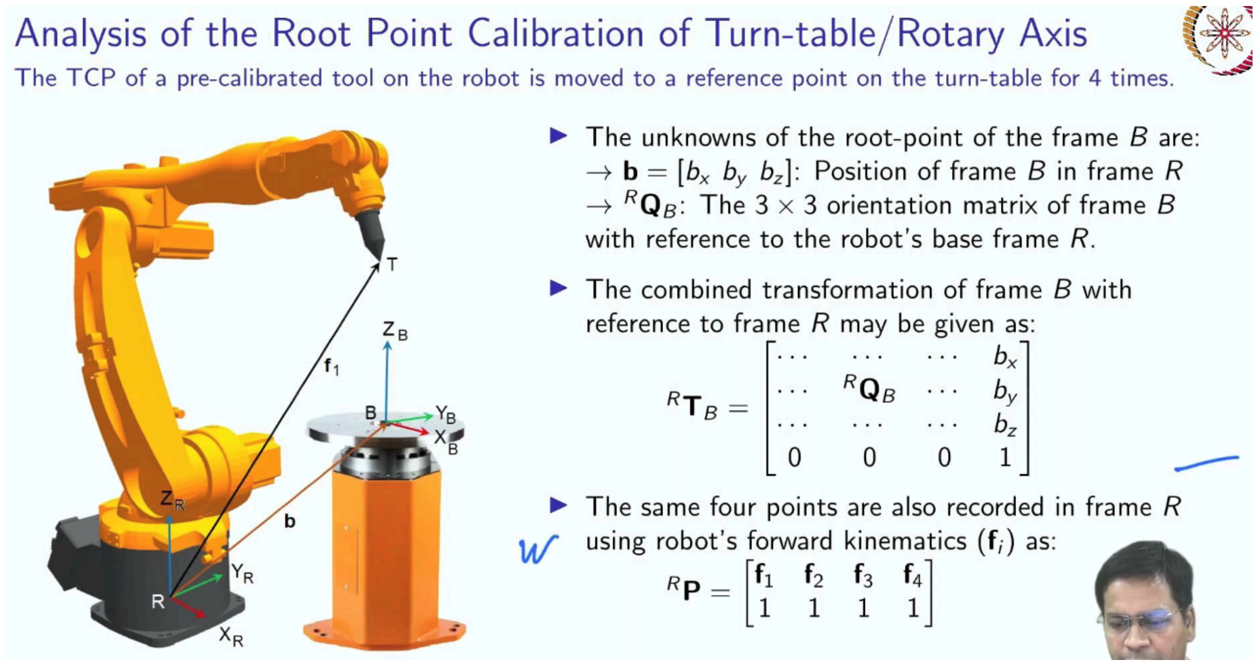
So, let us begin. So, this enables the robot to move with a kinematic coupling to the external axis, as I told you. This becomes the 7th axis and that goes to the controller also. So, the controller can now include this information also to its kinematics, and it can control all of them synchronously. So, what are the things that go on here in calibration for such an arrangement? It is calibrating the root point. The root point is the point which is located at the centre of this turntable. So, what does it include? It basically includes the vector that connects this. You should know this vector, where it is located, the position vector of the centre, as well as the orientation of this table-top. If this is your orientation, you should know how this axis is aligned. So, this job is basically done by root point calibration, and the second one is workpiece frame calibration. If at all you have an object which is kept on top of this, the centre of that with respect to this should also be known because you have to machine on this. So, that is on this frame. It is not at the frame which is located here. So, that is the second job that we will be doing. If at all, there is a tool which is there on the rotating external turntable So, that also can be taken care of. That is very, very rare, but that also is one of the jobs, if at all. You have a tool in the workspace which is mounted on top of this rotary axis. Anyway, let us move ahead quickly.



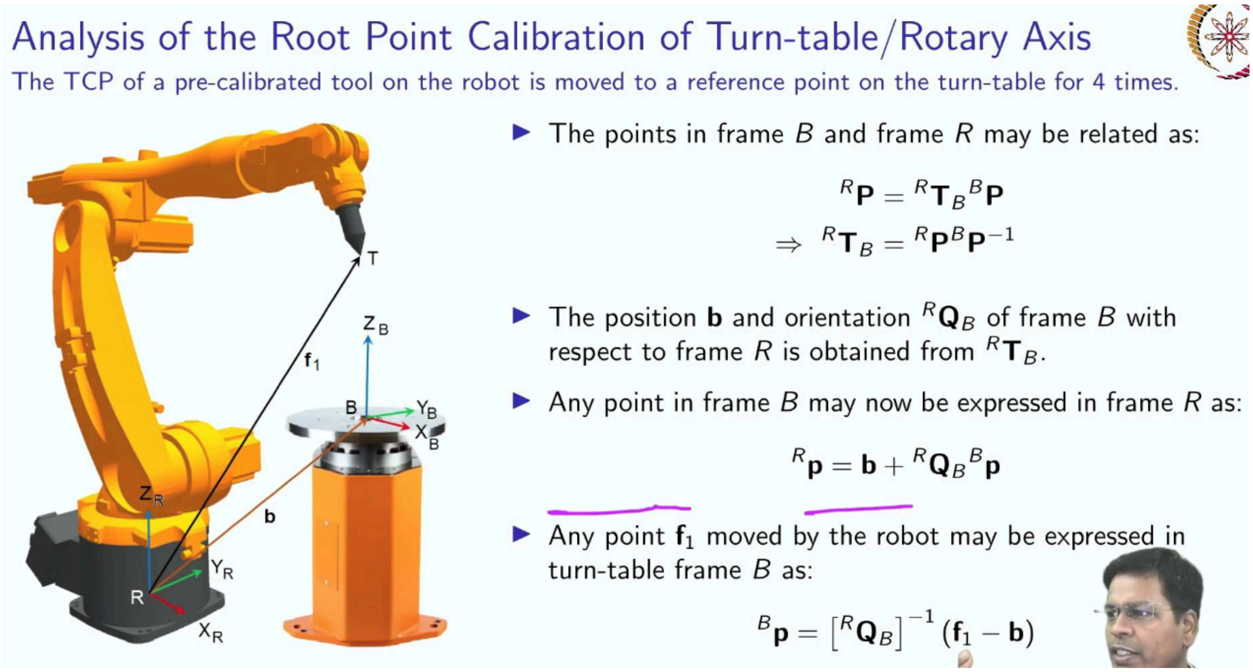
So, calibrating the root point of the turntable, how it is done, and the TCP of a pre-calibrated tool on the robot, again, you have a TCP which is mounted on top of the robot's tool flange, and it is calibrated. TCP calibration is done for that tool and is moved to the reference point on the turntable from four different directions. So, your robot is made to touch over here and over here. Four different locations. So, you make your turntable rotate, make your TCP go there again, rotate, go there again, rotate, go there like that. Do it four times. The position of the reference point is kept different. Touch each time by rotating the turntable axis. They should be very, very distinct. Again, you should be very careful. Your solution should not go singular, so they should be very much different. TCP locations and the rotary axis positions are recorded. That means the angle at which it has turned. You have to record that That is recorded and TCP locations. So, wherever your robot is going, you know if it is a pre-calibrated tool using forward kinematics, you can calculate where it has gone. So, all four locations you will record. You will also record the angle at which it has turned. By going to those four locations The controller can now calculate the root point of the external axis. So, that is what is our job. Let us do it.



Let us analyse how the controller does it. So, analysis of the root point calibration of the turntable, which is also known as the rotary axis. The TCP of the pre-calibrated tool on the robot is moved to the reference point on the turntable four times. You know there are four distinct points. The turntable is recorded in static frame B. So, this is the frame which I have put here and this is a static frame about which your turntable is rotating. So, the turntable rotates about the Z_B axis, and about this axis, your turntable is rotating, Assuming the reference points are at a distance r. So, if at all, you are touching upon the same point at every location of the turntable. So, r is fixed. So, this distance is fixed from the centre. It is moving about the axis Z_B and substance angle theta I from X. Let us say your reference frame is posed like this XB, YB and Z_B, that is, the B frame, which is over here. So, you are measuring theta from the X axis of frame B In plane XB, YB. The points can be written as i is: 1, 2, 3, 4, because there are four instances. So, points may be written as P1, P2, P3 and P4. Those are the points. Each of these points is r cos theta and r sin theta and zero because they are on the plane with respect to this turntable. So, Z is equal to zero for all of them. But because theta is varying, your Pi's are varying. So, you have four different PIs which are here. So, to make it compatible with other homogeneous transformations that I am going to use, I am putting one over here so that I can handle it with a homogeneous transformation matrix. These four points are known in which frame it is, in frame B.



Now, what the robot was doing? So, whenever the robot was going to those four locations, it was also recording those points in its frame. So, in its frame, if you take this point to those four locations, you will get these positions in the R frame as well. Let us just understand what are the unknowns which are here. The first unknown is this: that is, the position vector that connects the r to frame B. That is given by the Bx, By, and Bz vector, which is here: position of frame B in frame R and also this orientation error which is there. That is the orientation matrix of frame B with reference to the robot's base frame R. That is a 3 x 3 orientation matrix. These two are something that we have to find out by this calibration. So, now the robot combined transformation matrix from here to here may be given as this: This is your orientation change. This is your position vector, which takes It from here to here, Got it? So, this is your transformation matrix that takes it from here to here, R to B. Then the same four points are also recorded in frame r. you know, using forward kinematics, you can obtain all these fs. So, f1, f2, f3 and f4 are four locations that you have recorded again, making it compatible with a homogeneous transformation matrix. So, these points are again in frame R. So, earlier, it was in frame B. This time, it is in frame R.



So now, relating both of them, the points in frame B and frame R may be related, as these are points in frame B. This is the point in frame R. So, here goes your unknown transformation matrix, which is to be found out. So you can quickly write it as this.

$$^{R}T_B = {^{R}P_B}\, P^{-1}$$

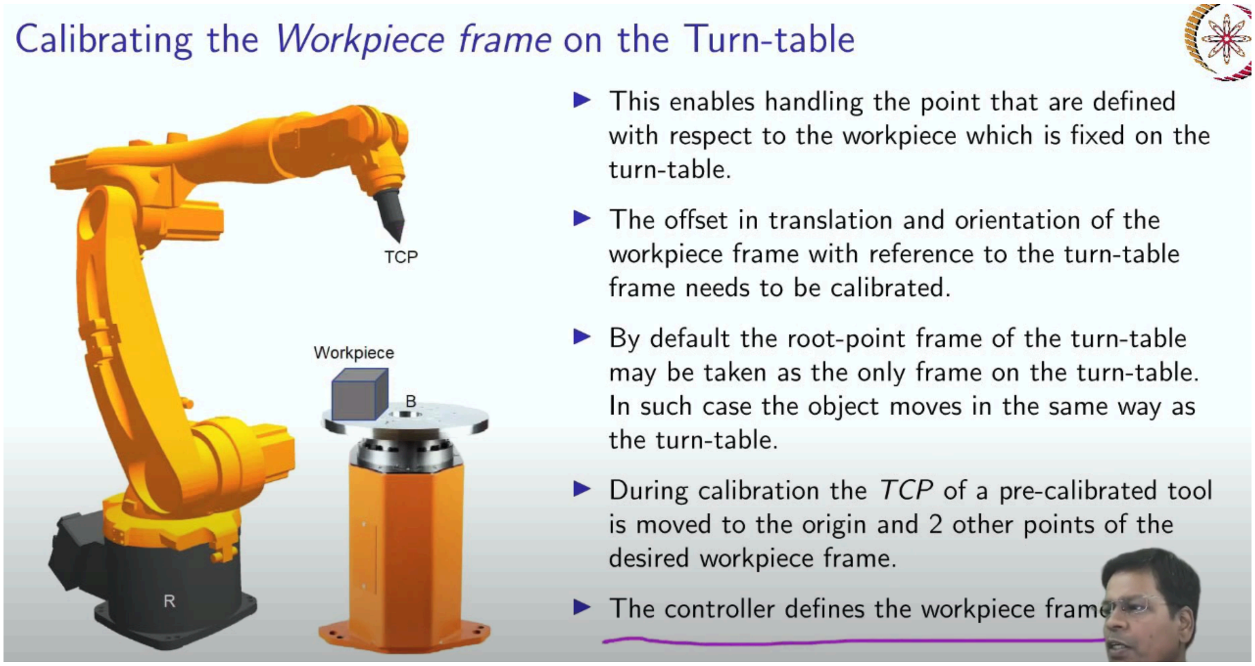
So, this gives you the end result, Got it? So, this is your calibration matrix. So, it has two components in it. It has, first thing, that is, the position of B and orientation of this, of frame B, with respect to frame R. So, that is there in this.

$$^{R}p = b + {^{R}Q_B}\, {^{B}p}$$

So, any point now, if it is, i.n. Frame B can be expressed in frame R. So, if it is with respect to frame B, you can express that using this. So, this is the same old equation that I have used just in the previous slide. So, it is this one. So, this is all you can obtain. So, any point F1 which is moved by the robot can now be expressed in a turntable frame. So, if I am taking my robot somewhere, I want to know where it has reached with respect to B. This can be calculated using this. So, just by rearranging this same equation, I can obtain this. Got it?

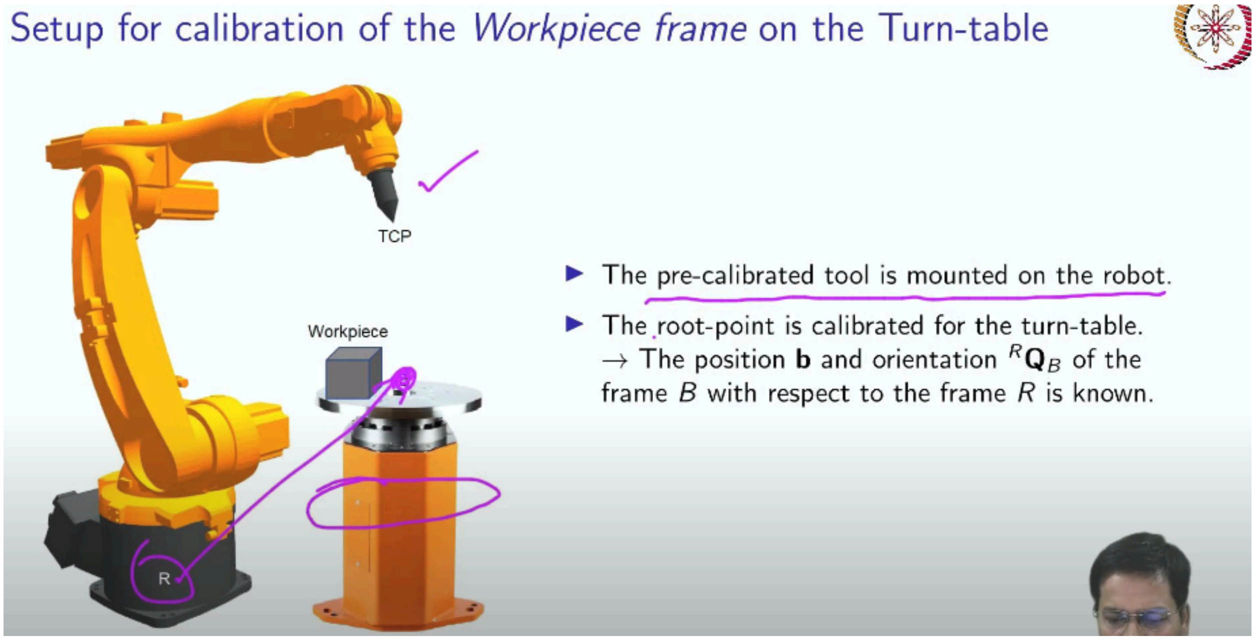
$$^{B}p = [{^{R}Q_B}\,]^{-1}\,(f1 - b)$$

So, the forward kinematics f1 vector is known, and B is known through calibration. These two are known through calibration. So, you can obtain this. Any point where this robot is going, you can express it in B frame, Got it? So this is what our job was.

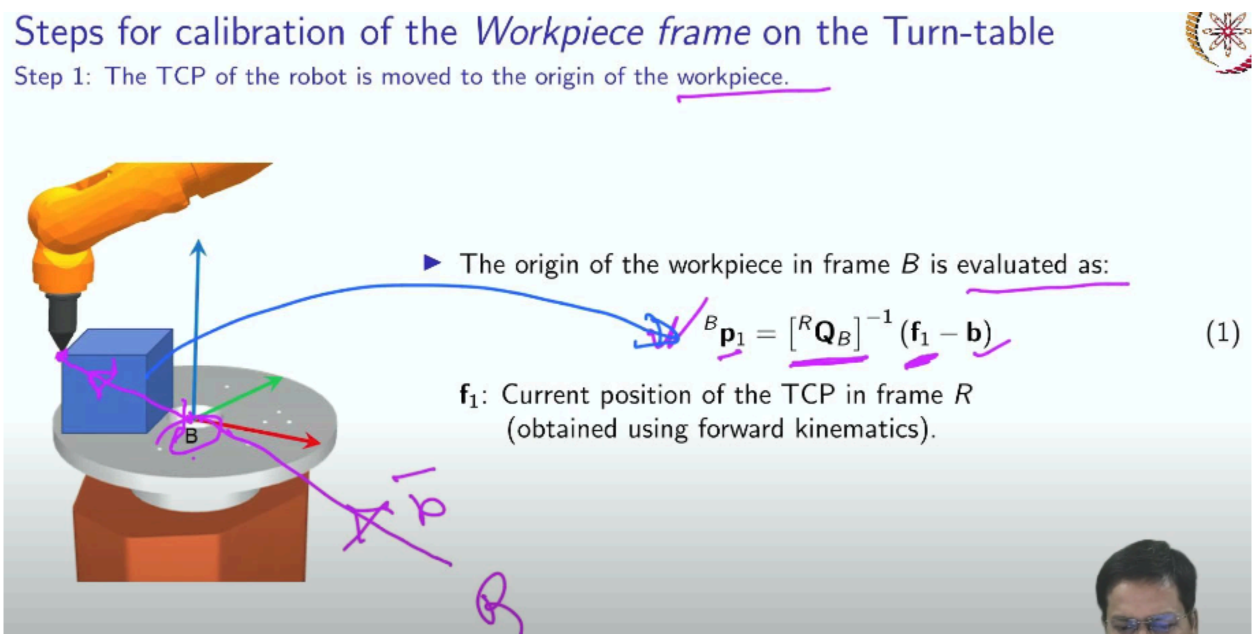


Now, let us put a workpiece on top of this turntable, and whatever I know, I have complete information about that workpiece. So, each and every point on this workpiece is known with respect to its frame but not with respect to the turntable on which I have kept it. But I have to machine it from different sides the way it was doing. So, you see, there must be some offset between the frame D and the workpiece frame. So, that offset now is to be determined. That offset will be in terms of displacement as well as orientation. So, that is what is the workpiece frame calibration. So, this will enable the handling of the points that are defined with respect to the workpiece, which is fixed on the turntable.

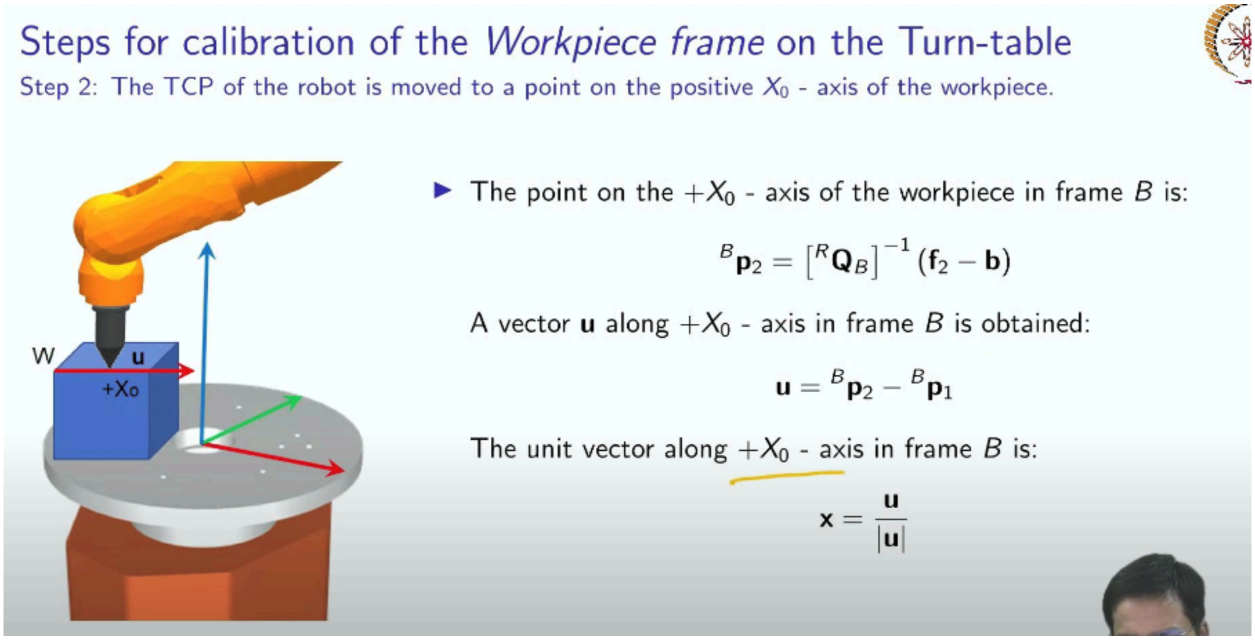
The offset in translation and orientation of the workpiece frame with reference to the turntable frame is to be calibrated. So, that offset needs to be known By default, if at all. You don't calibrate it, and you have to work with the existing frame, which is there, and all the points need to be addressed with respect to B so that you can express it with respect to R, and your robot can reach there. How do we do it? The TCP of the pre-calibrated tool. This is, again, a pre-calibrated tool. The tip is now moved to the origin, wherever you want to make it as an origin of this workpiece. You have to take your robot there and to other points on the desired workpiece frame. So, the controller is now able to define the workpiece frame.



So, let us analyse that. So, the setup for that will be: the pre-calibrated tool is mounted on the robot, and the root point of the turntable is also calibrated. That means this point position, as well as orientation with respect to R, is now known. Now, I will mount the workpiece on top of this.



So, in the first step, the TCP of the robot is moved to the origin of the workpiece. So, the origin of the workpiece in frame B is evaluated using our earlier formula that we have handled. So, this is obtained using forward kinematics wherever my TCP is going because it is a calibrated TCP. So, I know f1. This is a calibrated root point. So, this position of this with respect to the robot's base is known. So, this is known. So, this vector b is also known, This is the calibration matrix for the root point that we have obtained. So, I can now know the position of this point. I call it P1 with respect to B is now known, So I will quickly obtain this vector. Got it? So that goes. This is nothing but this one. Got it, So that is what is obtained. So, the correct position of the TCP in frame R is obtained using forward kinematics. So, in this case, it is f1.



Now, the second step. So, the TCP of the robot is now moved to the point on the positive X0 axis of the workpiece. So, anywhere along the X axis, wherever you want to make it as the X axis of the workpiece, you touch it somewhere on that axis, and again, you calculate your next point, that is this one.

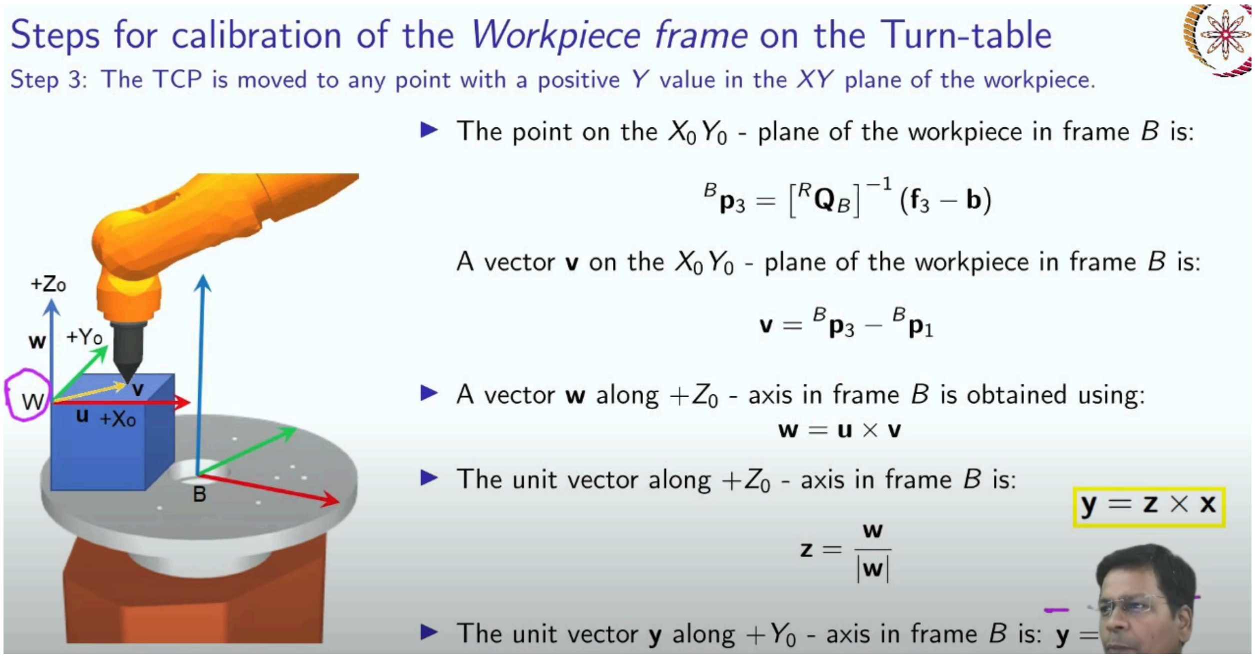
$$^Bp2 = [\ ^RQ_B\ ]^{-1}\ (f2\text{-}b)$$

Where is that? It is this one. So, it is in frame B, which is attached here. So, this is your p2. and what is that? It is nothing, but again, you calculate the TCP point f2 vector, and B is known. You know the B vector for this. It is already known, and the orientation calibration matrix for the base frame is known, the B frame is known, and the timetable and root point are known so that you can obtain this. Again, I will use this further and calculate a vector along the X-axis because you already know this point with respect to B. Also, the point which is here with respect to B. Taking the difference, you get a vector along the X axis, positive X axis. So, this is your u. Mind it.

$$u = {}^Bp_2 - {}^Bp_1$$

This is all in frame B because you have expressed all the vectors in frame B. So, this is your u. So, the unit vector along the X0 axis will be: this is your X0 axis is X, that is u by the magnitude of u.

$$x = \frac{u}{|u|}$$



Again, in step 3, the TCP is moved to any point with a positive Y value on the XY plane. So, this is the top. So, this is already your X-axis. This was your Y-axis. So you have to go somewhere over here. So, a line which connects the origin to this is your v vector. I am taking it as a v vector. This was already known, that is P1 was known in frame B, so now use all this information. So, the point on the X0, Y0 plane on the workpiece in frame B is known using this same old formula. So, again, V can be calculated using P3 minus P1. So, this, the third step, this location is P3 in frame B P3 minus P1 will give you this vector.

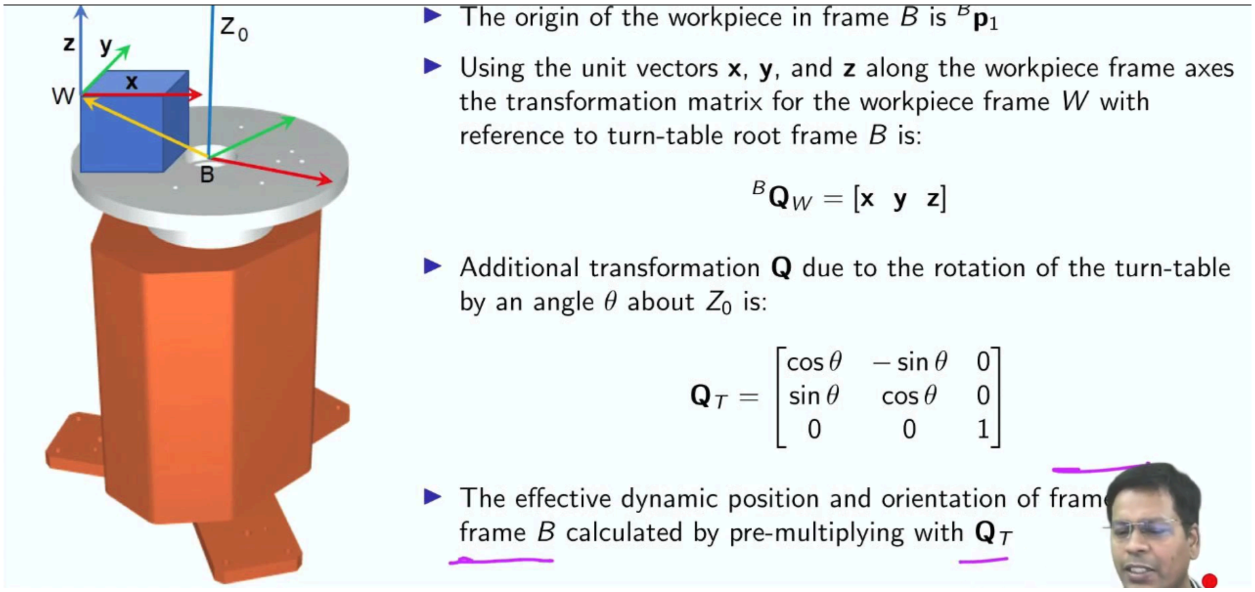
$$v = {}^Bp_3 - {}^Bp_1$$

So, that is what is obtained here. So, now you know. We did it earlier, also similar, while we were calibrating the external reference plane. So, the same thing we are doing here again. So, a vector w is given by w = u × v.

w is nothing, but it is a vector which is aligned to the Z0 axis in frame B. That can quickly be obtained by taking the cross product of u and v. So, they are lying in the plane XY, so it will give you a vector which is along the Z0 axis. I call it w, So I have obtained W. So, unit vector W will be Z. It is a unit vector along the Z0 axis given by Z, is this? So you have obtained unit vector W. You already know unit vector u. So you can obtain a unit vector along the Y axis also. That is given by Y is equal to Z cross X.

$$y = z \times x$$

So, all three unit vectors are now known. That means this frame w is now fully defined with respect to B because all the vectors are in frame B.



So now, the origin of the workpiece in frame B is this one. The first time when you touched, you calculated what You calculated P1 in frame B. Again, the orientation is given by B to W. So, you already have XYZ in hand. You can quickly write it as this.

$$^{B}Q_{w} = [x\ y\ z]$$

So, this is nothing but the orientation of work piece frame with respect to B. This is, by definition, a rotation transformation matrix. You know it.

So, now any additional transformation Q due to rotation of the turntable by an angle theta about Z0 is an additional rotation transformation. This is nothing but rotation about this by an angle theta. So, this is your additional transformation. So, this is to be included. You can now express any point which is on this, any point which is represented in this frame to this frame. If you know it in this frame, you can express it in the robot's frame.

Additionally, if you have further rotated it by an angle theta, the turntable rotates. Then you can additionally include this; that is, you have to pre-multiply the points to this rotation matrix. So, whatever you are doing with this, additionally, you have to do this transformation. So, effective dynamic position, that is, position and orientation of frame w in frame B, is calculated by pre-multiplying with Q_T. So, this is how you can handle any rotating axis if it is there in the work space. So, you have now seen a linear axis on which the robot can be mounted. You have now seen an external axis too, that is, a rotary axis in the workspace.

So that is all for this module. In the next module, we will study robot statics. So, we will deal with link forces, moments, gravity, compensation force, velocity, and ellipsoid. So, that is all for today. Hope you are enjoying it. Thanks a lot.