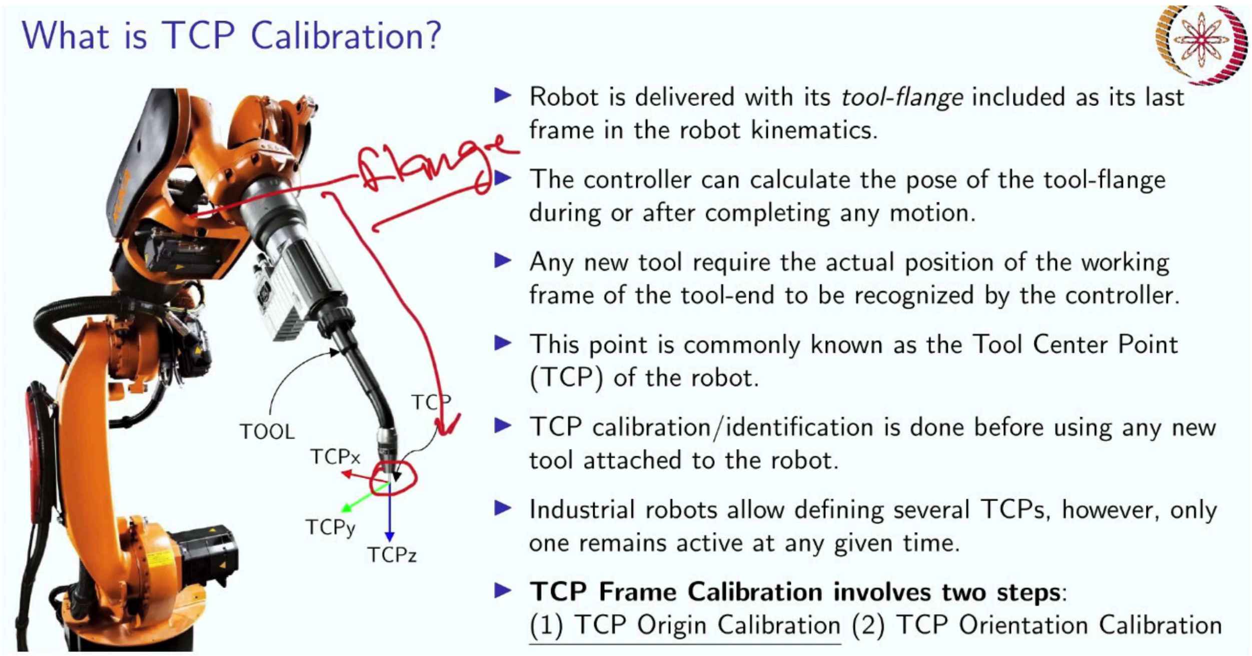
NPTEL Online Certification Courses
Industrial Robotics: Theories for Implementation
Dr Arun Dayal Udai
Department of Mechanical Engineering
Indian Institute of Technology (ISM) Dhanbad
Week: 06
Lecture: 28

TCP Calibration using 4-Point method and External reference method

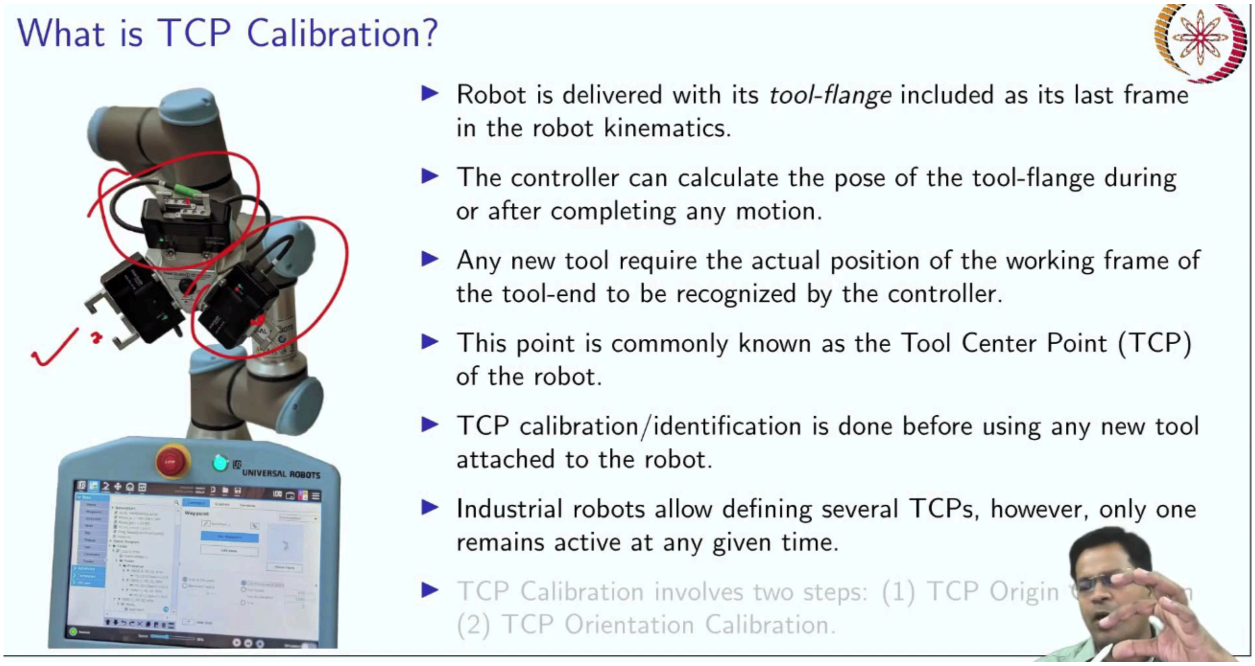
Welcome back to the module installation and commissioning of industrial robots. So, in the first lecture of this module, I showed you a video of installing a robot from the ground up. We built a concrete structure, we put the robot on that, and finally we made it run at the joint level using jogging. In the next lecture, I also showed you how joints are getting calibrated So I could calibrate my joints to give the feedback properly to the controller so that you can run the robot well at the Cartesian level. So, extending that further, today we will do end effector calibration, that is the tool which is attached at the end of the robot, the end of the serial chain which actually does the job. So, let me just start with this module.

I will start with a small video recalling what we have done while installing. So, you see, this was the new robot that we received, and we could install it all around. It looks like this, and here it is. So, you see, this robot came with a bare flange. So, when it comes, it comes with this end effector flange. So, the robot controller knows the center of this flange, that is, as the origin of the last frame. That is what it knows, and it can precisely control that end effector in Cartesian coordinates anywhere in the workspace. So, because it can do forward kinematics using the joint level kinematics that it can do, and it can precisely calculate the end effector position, it can take this end effector anywhere in the workspace. So, now, apart from this, the robot has to do something. So, we have to attach, let us say, a gripper. That is what we also did. I will just show you what we did later on after this. So, once it was fully installed, you see, we attached a gripper.

It was a two-fingered pneumatic gripper. So, now the robot end effector doesn't end up at the flange, which is shown as something like grassy. So, in this case, when it was delivered, now it also has an attachment. That is this tool, that is the two-fingered gripper. Now, it ends up quite further away from the actual position of how the robot was delivered to us. The controller now should know this new position, because now you don't have to control the flange to go to any place. You have to control this new tool, which is attached. This will pick up the object. This is what will go on the conveyor and pick up an object. It will place something, If it is a welding tool that will make contact or maybe a contactless way, it does some welding job; got it. So, that is what is the problem and that is what we want to solve in this particular lecture.

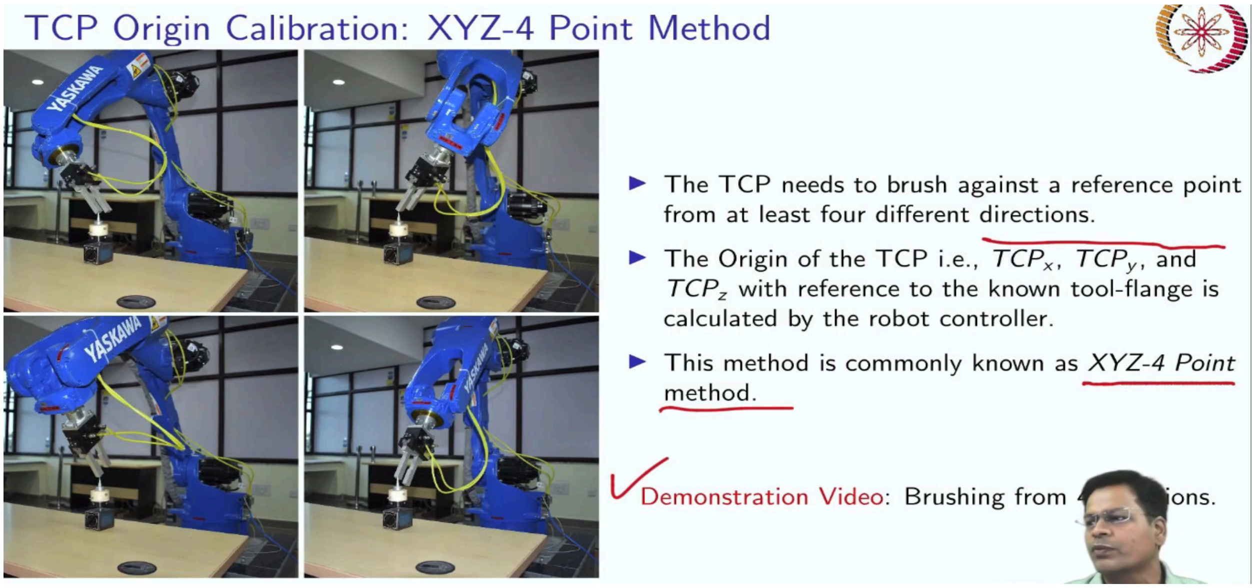


So, continuing further. So, this is how the robot is delivered. It is with the tool flange included as its last frame in the robot kinematics. So, the controller knows till this flange centre point that is the origin. Also, the flange orientation is known to the controller. Where your end effector frame is placed. The controller can calculate the pose, that is position and orientation of the tool flange during or after completion of any motion while moving. Also, it knows where it is. Any tool which is attached after this sets the actual position of that flange to the tip of that tool. So, any new tool requires the actual position of the working frame of the tool end to be recognised by the controller. Once it recognises that, now it can be commanded this new TCP. That is, the tool center point, can be commanded to go at any velocity, go to any position, and take up any orientation if the flange orientation is also calibrated. So, this is our job now. This point is known as the tool center point, and the frame which is attached to this point is known as the TCP frame or tool center point frame. So, TCP calibration and identification are required before using any robot. New tool which is attached to the robot Got it. So, before we actually put our robot to use, we should calibrate this step so that the flange does not go and reach the commanded position, and your tool will actually get into the object.

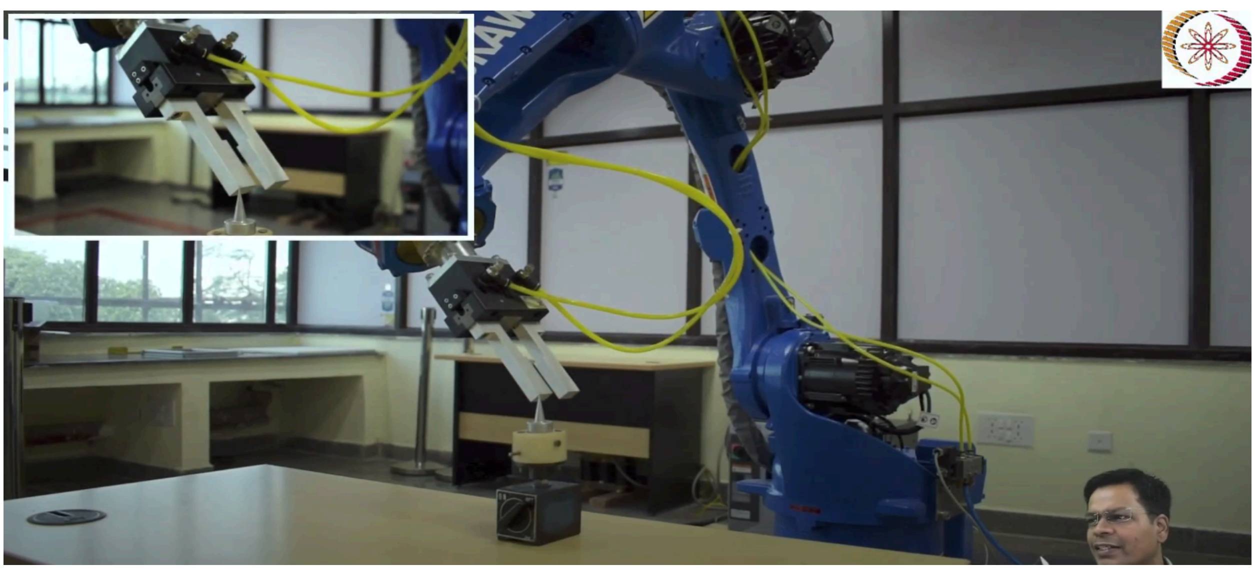


So industrial robots they allow multiple tools like this. You see, you have multiple tools here. One is here, and the other one is here, and another one is here. So, you have three tools. Each one of them will have a different tool center point, and that needs to be taken to the object. So, it allows defining several TCP's. However, at any instant of time, only one remains active. You can take it. You can command, tool number one, go to this point, tool number two. As soon as you say tool number two, it rotates and makes that active. It does all the velocity control, position control, and orientation control for that new tool. So, over here it has three different tools. Even a single tool can have multiple points which can be calibrated. You can command the center of a two-finger gripper to go to certain objects or you can command, let's say, the tip of one of the fingers to go to a certain object. Got it? So a single tool can have multiple TCP or multiple tools can have individual TCP's defined for each one of them got it. However, only one of them will remain active.

So, this problem of TCP definitions includes two particular problems. So, it is a TCP origin calibration. That is the tip of this tool. That is this point. That is what I want to know where it is with respect to the flange that I already know. This is the flange with which this robot was delivered from here to here. What is the transformation which is there? That is an additional transformation. That is what I want to know. OK, so that is the origin calibration and also the orientation of this frame, how the new Z axis, new X axis, and new Y axis of the new tool are oriented with respect to the flange frame, which was there when the robot was delivered. So, there are two problems. In today's lecture, we will be covering TCP origin calibration.



So this is one of the very popular methods which is there with most of the industrial robots that, through teach pendant. It has a pre-built program which supports calibrating a robot using this technique. How this technique works basically. The TCP, that is, the tool, center point, the tip of the finger- one of the fingers may be or center of the finger. The TCP, the point at which you want to make a TCP, needs to press against a reference point. The reference point is nothing but a type of reference point which is fixed in the robot workspace, and it has a sharp pointed tip. OK, so you have to make almost like contact. TCP can make exactly very good and can go very close to that TCP tip, that reference tip, and it can just brush against that from four different directions. So, this figure shows exactly that.

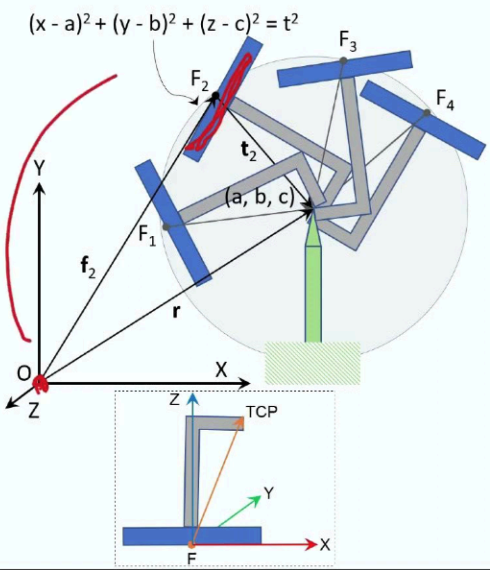


Let me show you through one of the videos once again. You just notice this. So, I am just marking this. I am placing a mark over here so that I want to calibrate for this point. This will be my TCP. So, now, one of the views shows a closer picture of that. So, the robot is being taken from one of the angles. Actually, the process is very, very slow because you have to take it very, very gradually to that point. Otherwise, you may hit the target, and it will move from its location. One of the orientations it has touched. Same point, TCP. If you can remember, this looks like a dial gauge stand. Yes, it is that. We have made it in our lab and this is a mint reference point, and it has a switch so that it can be mounted. Well, it is a heavy tool, and it doesn't move if it gets hit because on a metal table, normally we switch it on, and it can. It has a magnet, and it remains there. Your reference point should be very, very firm. So, now it is approaching from a different angle, so all four angles should be quite distinct. Got it? This is a quick sequence that I have shown to make you understand. Otherwise, it is a very, very slow process because the person has to go at a dead slow velocity when it approaches near to the target. Through jogging, manual jogging, it is done. Now, it has approached from all directions, that's all.

So, you see, it has brushed against the reference point from four different directions. This is what I meant. The origin of the TCP, that is, the TCPx, TCPy and TCPz, with reference to the known tool flange, that is, the flange with which the robot was delivered. Till here, till this one, till this one. So, that is calculated by the robot controller. So, the robot controller, once it takes it to any of the locations, the location of the TCP, that was the flange, that was the initial TCP that is, the tool flange is calculated using the robot forward kinematics by the controller because it knows all the joint angles through the feedback. It knows the robot's dimensions. The whole of the DH parameter is there inside. It does the forward kinematics and calculates the tool flange location for each of those four brushes.

Now, this method is commonly known as XYZ, a four-point method why I will tell you later. So, I have shown you the demonstration video, which showed touching that. So, the tool flange location can be easily controlled, and easily calculated by the controller. Now, the controller also has an inbuilt algorithm with which it can calculate the TCP location with respect to the tool flange. So, how it does it? That is what we are going to do now.

XYZ - 4 Point Method: Analysis using Best Fit Sphere!



- Use forward kinematics to obtain the tool-flange locations $F_1 \cdots F_4$.
- The points $F_1 \cdots F_4$ would lie on a sphere. Assuming its radius as t, the equation of sphere would be: $(x_i - a)^2 + (y_i - b)^2 + (z_i - c)^2 = t^2$, $i \in \{1, 2, 3, 4\}$
- As the points $F_1 \cdots F_4$ satisfy the equation of the sphere. Evaluate the unknowns a, b, c.
- The reference point $\mathbf{r}$ vector is obtained by
$$\mathbf{r} \equiv a\mathbf{i} + b\mathbf{j} + c\mathbf{k}.$$
- Use any positions vectors $\mathbf{f}_i$ (among $F_1 \cdots F_4$) to find the vector $\mathbf{t}_i$ in frame $\#O$ as:
$$\mathbf{t}_i = \mathbf{r} - \mathbf{f}_i$$
- Use forward kinematics rotation matrix ${}^O\mathbf{Q}_i$ to find TCP with respect to the frame F_i as:
$${}^F\mathbf{t}_i = [{}^O\mathbf{Q}_i]^{-1}\mathbf{t}_i$$

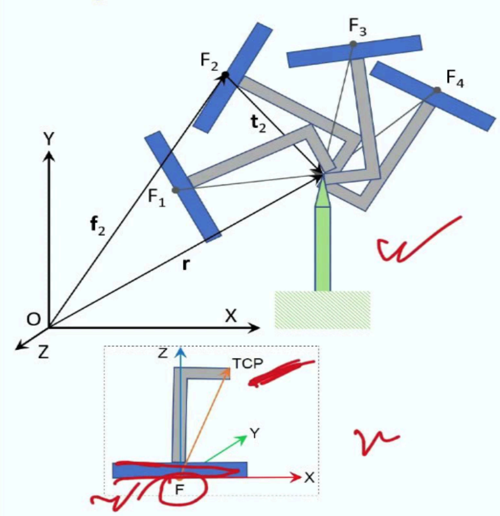
So, this is one of the coordinate geometry approaches that you can think of, that is, an XYZ four-point method. You can think of a way like, let us say, this is your tool, okay, this is your flange. The blue one is your flange, on which you are. You have attached the L-shaped tool. Okay, so this is the tip of the tool that you want to make as a tool center point. Although it is named as a tool center point, it may not be at the center, got it? It is actually the frame center. That is actually the origin. So, this is what I want to make it. Okay, so now it has approached from four different positions. For each of those positions, it has f1, f2, f3 and f4. Four different sides, so all the flange locations are shown in this figure. So, use forward kinematics to obtain the two flange locations f1 to f4. This, this, this, this can easily be calculated using robot forward kinematics. You have already done DH for the GP-12 robot.

You know how to do it, so you can now calculate these locations for all the blessings that it did with this, this point which is there in the robot workspace, okay, so now you know these points as well, okay? Secondly, because it is at a constant distance, this f1 to f4, you know it has brushed against the same point that is situated over here, and it should be at a constant distance. This distance is constant for all those blessings, and this becomes the radius. This becomes the radius of a sphere, because it has approached from multiple directions. So, this is the reason why four blessings are to be done. So, you got it. How many variables does it has? It is a sphere that you want to know. all the points will lie on a sphere, so the minimum number of blessings should be four. So, I am equal to one to four. This is the equation of: that is a sphere, which has four variables. Three of them define the center, that is, the ABC location of this, the reference point tip that is there. That becomes the center of this sphere ABC. A, B, and C go here, and T is the radius of this sphere. So, this is the minimum set that you need to define this sphere completely.

As the points f1 to f4 will satisfy the equation of a sphere. Now, you substitute all the f1 to f4 locations to this equation and solve this to get the unknowns ABC. You can also obtain T, but that may not be useful because it is just the magnitude. It is just the radius. That is fair, so that is not very useful here. So, at least getting this point, ABC is useful. That gives you the location of your reference point with respect to the robot frame, that is here somewhere. So, reference point r, that is, a vector which can be obtained by r, is equal to this. So, you got this r vector. You also had all these locations, f1 to f4 vectors. Those were known, okay, so using that you can calculate the vector T, Ti, T1, T2, T3, T4 you can calculate. So, each one of them will have a different vector, but the magnitude of that will be the same because that is the radius of the sphere. So, Ti is equal to r minus phi. Simple vector triangle you can get.

Now, use the forward kinematic rotation matrix. That is the rotation matrix of this flange with respect to the base frame. So, this is it. Okay, so you already have this because you know forward kinematics, you have done forward kinematics now, so this is now much clearer to you. So, this is the matrix that is known to you to find the TCP with respect to the flange, tuned flange. So, you have a transformation that takes you from here to here. So, now you can do T with respect to the frame. F can be obtained using this inverse of the rotation transformation matrix into Ti, Ti is this? that is in this frame. So, this one is in which frame? This is with respect to the O frame, which is here. Now, I want to express this in this frame-that is, the flange frame, so that I can get exactly the offset along X, Y, and Z with respect to the existing tool frames. So, this is how it is obtained. So, yes, this is one of the approaches that you can follow. But, yes, it is quite difficult to solve this sphere equation and get these variables quickly, so that is the reason there are many elegant ways of doing it.

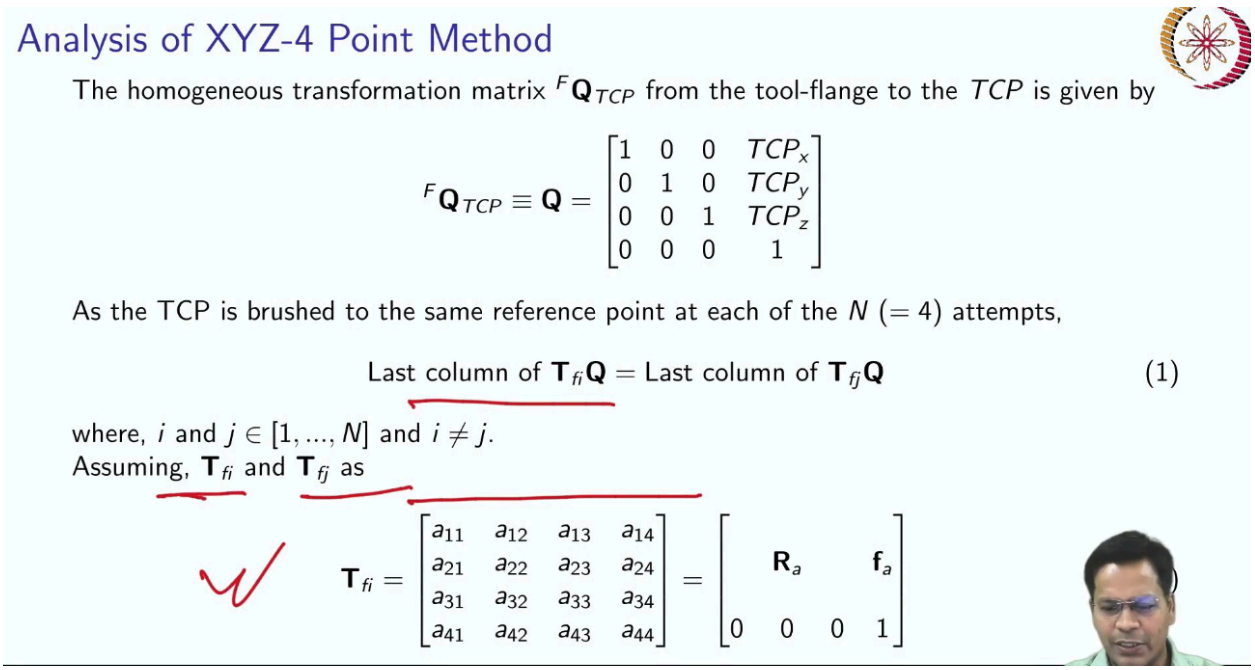
Analysis of XYZ-4 Point Method: Symbols used



Symbol	Meaning
$\mathbf{T}_f$	Homogeneous transformation matrix of the flange with respect to the robot base.
N	Number of different poses the TCP was brushed with the fixed reference point.
$\mathbf{T}_{fi}$	Homogeneous transformation matrix of the flange with respect to the robot base for the i^{th} attempt, where $i \in [1, ..., N]$.
$[TCP_x \quad TCP_y \quad TCP_z]^T$	Translation of the TCP origin with respect to the tool-flange.
$\mathbf{Q}$	Homogeneous transformation matrix of the TCP with respect to the tool flange.

So, let us start with a new approach. This is what we did. This is how your TCP is defined with respect to the flange. You already know this because you know the forward kinematics wherever the robot goes. You can get this immediately. What you want to know is this with respect to this, that is, the offset along X, Y, and Z for this new TCP location. So, these are the symbols that I will be using. I will just go through it.

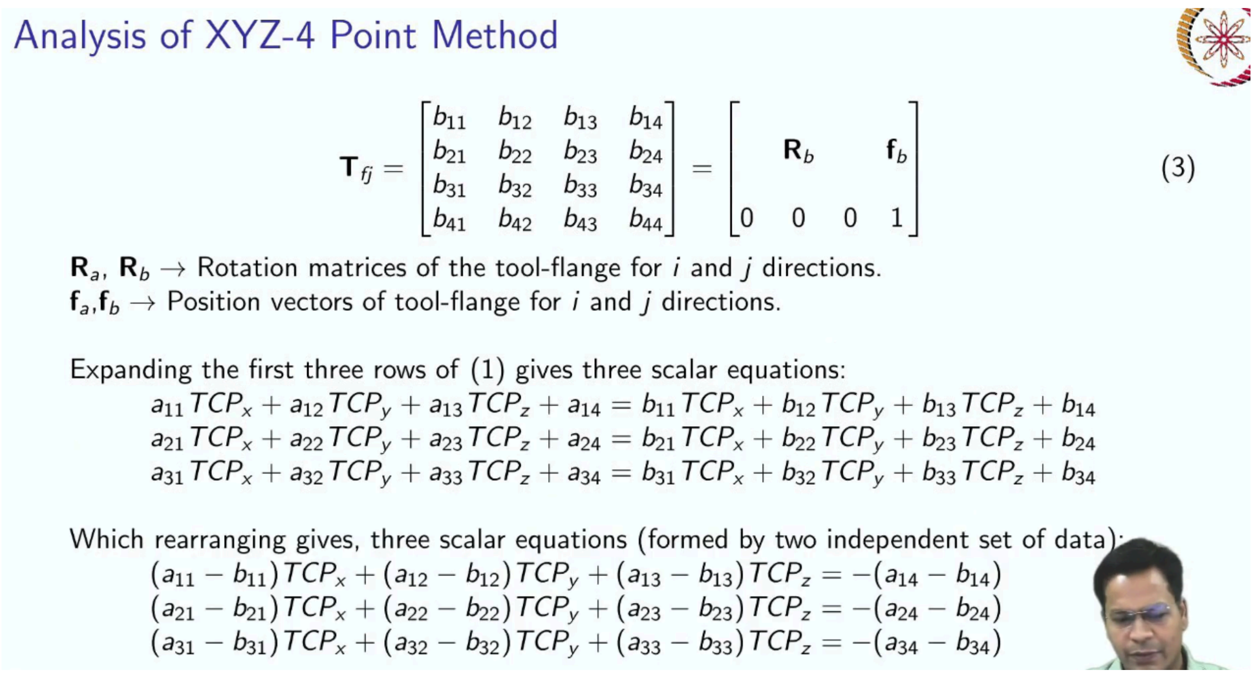
So, T_f is the homogeneous transformation matrix of the flange with respect to the robot base. N is the number of poses that it is taken while brushing from different directions onto the fixed reference point. T_{fi} that is, the flange homogeneous transformation matrix, when it approaches from i, is equal to one to four directions. So, T_{f1}, Tf2, Tf3 and Tf4 are there. TCP X, Y, and Z are the offsets. That is what we want to know. We call it TCP origin with respect to the tool flange. Q is the homogeneous transformation matrix of the TCP with respect to the tool flange, got it? So, this is the homogeneous transformation matrix that will include this displacement. There is no orientation calibration as of now. I assume the TCP also has the exact same frame, which is here shifted to this location. All X, Y and Z- will be oriented in the same way as they are there at F, got it? There is just a translation. There is no rotation. That is what is an assumption. That will be done later using TCP orientation calibration. Now we are just interested in this.



So, let us begin. The homogeneous transformation matrix from the flange to the TCP location is given by this, that is Q. I call it from now on, instead of calling it this way, I will simply call it Q. So, it is nothing but translation along X, along Y, along Z. This is your homogeneous transformation matrix. Quite easy.

But this TCP has brushed to the same reference point for.on four different directions, four attempts. So, Q remains the same on both sides. Q remains the same. What changes it? From the first direction, from the second direction, from the third direction? This frame location, f1, f2, f3 and f4. So, that changes. So, that actually changes this first matrix, that is, the transformation matrix of the frame i with respect to the robot base. So, this is what changes. So, your robot is now at a new location. This is what is known. That is the transformation of frame location with respect to the base with contact that it established. So, this is what you know for every instance from here to this point is your Q. So, in order to reach here, because this is your reference frame, this is the same for every directional approach. Okay, wherever you are approaching, you are reaching the same point. That is this reference point. R got it. So, this is your forward kinematic transformation from the robot base to the flange location for ith contact, and Q is the matrix which is here. That will finally take you to R. R is the same, so that makes the last column of these two directions the same.

Assuming this and this both like this. So, I have just assumed. Once you take the product, you get to a matrix that looks like this: it's a four cross, four homogeneous matrices. Normally, these terms will be 0, and this will be 1. This will tell you the translation. This is the orientation matrix. That's a standard structure of any homogeneous transformation matrix.

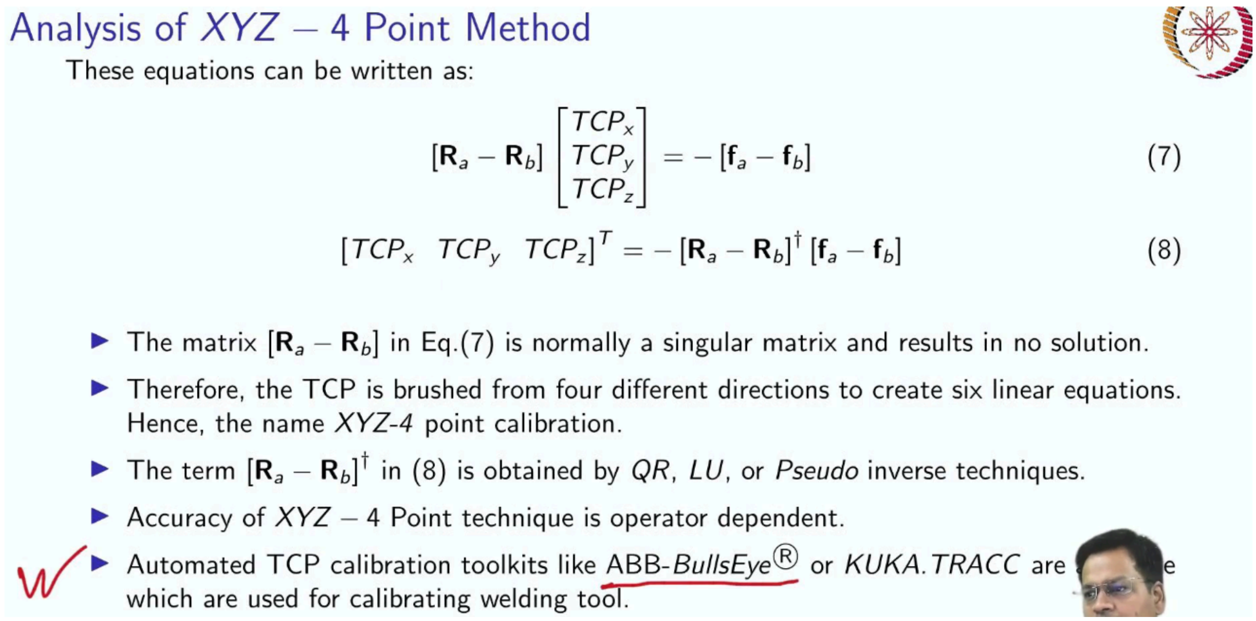


So, let us say, from A direction, the orientation matrix was like this: this was your translation and similarly, from the B direction, you have a matrix that has a structure something like this. So, these two (R_a, R_b) are rotation matrices of the tool flange from the i and j directions. i and j are two different atoms. f_a and f_b are position vectors of the tool flange for the I and J directions. That is the position vector f_a and f_b. got it? So that is basically connecting the base to the tool flange, got it? So this will give you the orientation of that. This will give you. The displacement of that.

So, if you expand this taking product with the Q, Q contains TCP X, all the directions. Let me just show you once again this: it looks like this, got it. So, this is your Q, this is your F, I, that goes before this. So, once you take the product, it looks like this: these are three scalar equations. The top three rows are just written like this, expanded like this: and you get to this because both are the same A and the B part. So, it is written over here like this: these are the three scalar equations. Rearranging you can quickly get to this.

$$(a_{11}-b_{11})TCP_x+(a_{12}-b_{12})TCP_y+(a_{13}-b_{13})TCP_z=-(a_{14}-b_{14})$$
$$(a_{21}-b_{21})TCP_x+(a_{22}-b_{22})TCP_y+(a_{23}-b_{23})TCP_z=-(a_{24}-b_{24})$$
$$(a_{31}-b_{31})TCP_x+(a_{32}-b_{32})TCP_y+(a_{33}-b_{33})TCP_z=-(a_{34}-b_{34})$$

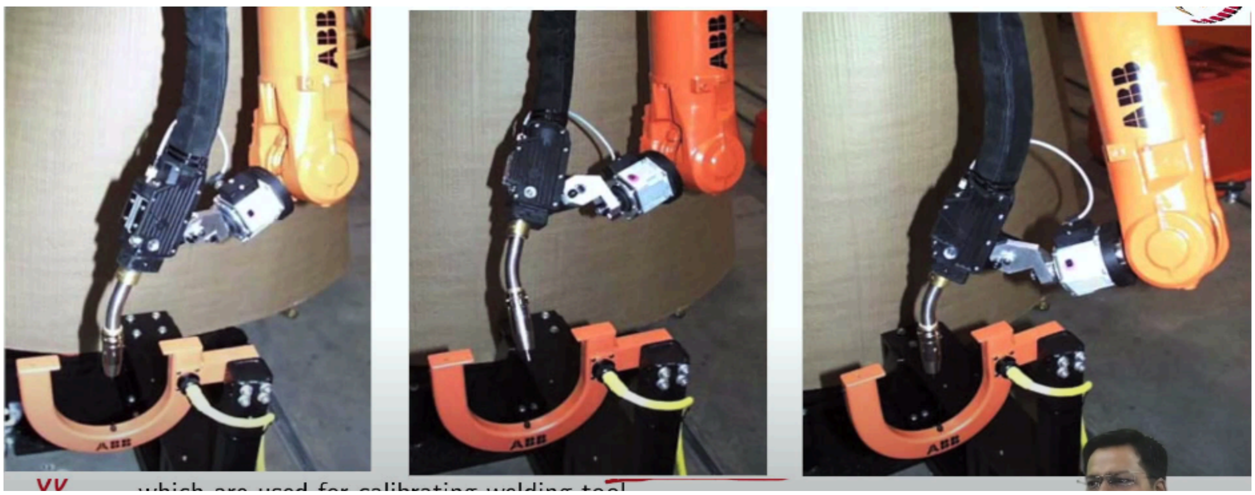
I have just rearranged it. Took some common, that as TCPx, here and here, TCPy here and here, TCPz, here and here, and the same for all the equations that can be in a compact way can be written like this.



So, that is nothing but Ra and Rb. Those are the rotation matrices. This is the column containing the TCP location. These are the flange locations. f_a and f_b are the position vectors of the tool flange for the I and J directions. These are there, and if you take the inverse of this, bring it. Here it becomes like this: got it. So, this is your inverse. It is something like the inverse. I will tell you what it is. It may be a square matrix. Then, it can be quickly calculated by taking the inverse. Otherwise, there are other approaches to do it. I will tell you how. So, yes, this is your equation. This needs to be solved in order to get the TCP location. That is what we are interested in.

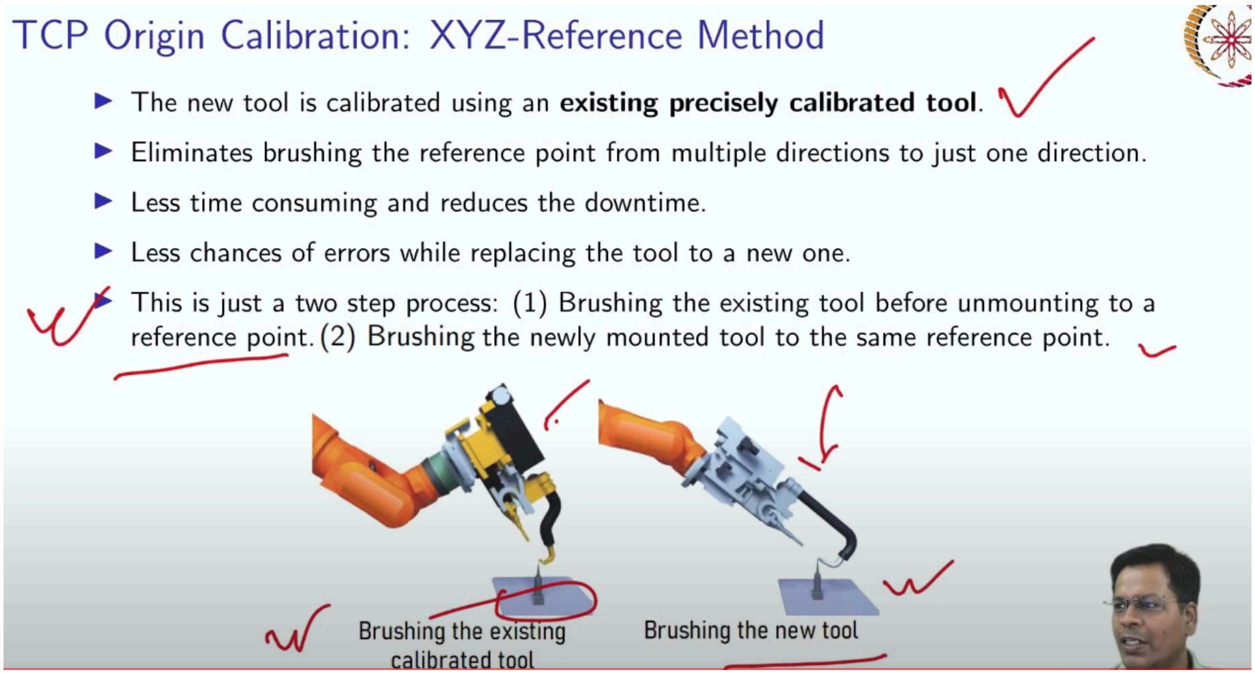
So, the matrix Ra minus Rb, as given in this equation, is normally a singular matrix. If it is just two directions you are making, establishing a contact, and this will result in no solution at all. This is the reason why it is brushed from four different directions, and those two, all the directions, those, all the directions, should be quite distinct to create six linear equations. So, I have taken first and second contacts over here, third and fourth contacts over here, and TCP location is like this and this is here. So, this effectively creates six linear equations. So, now the term this in this equation is obtained by QR decomposition, or it can be a LU decomposition method or pseudo-inverse technique if it is not a square matrix. Otherwise we can simply take

the inverse, but over here, it is not a square matrix. You have to go for one of these techniques to find out the inverse, and finally, we get to this.
The accuracy of the XYZ four-point technique? You see, it is operator-dependent because you manually do the job using your I estimation. You can just see whether it is making almost like contact from four different directions. So, that is what is the problem. So, human precision does matter over here, that is what is making it too much operator dependent, and it is not very, very accurate, but quite good for quite a number of applications like pick and place and this kind of application.
So, what additionally, manufacturers have provided are TCP calibration toolkits, like ABB has provided BullsEye. I'll show you a picture of that. What does it look like?

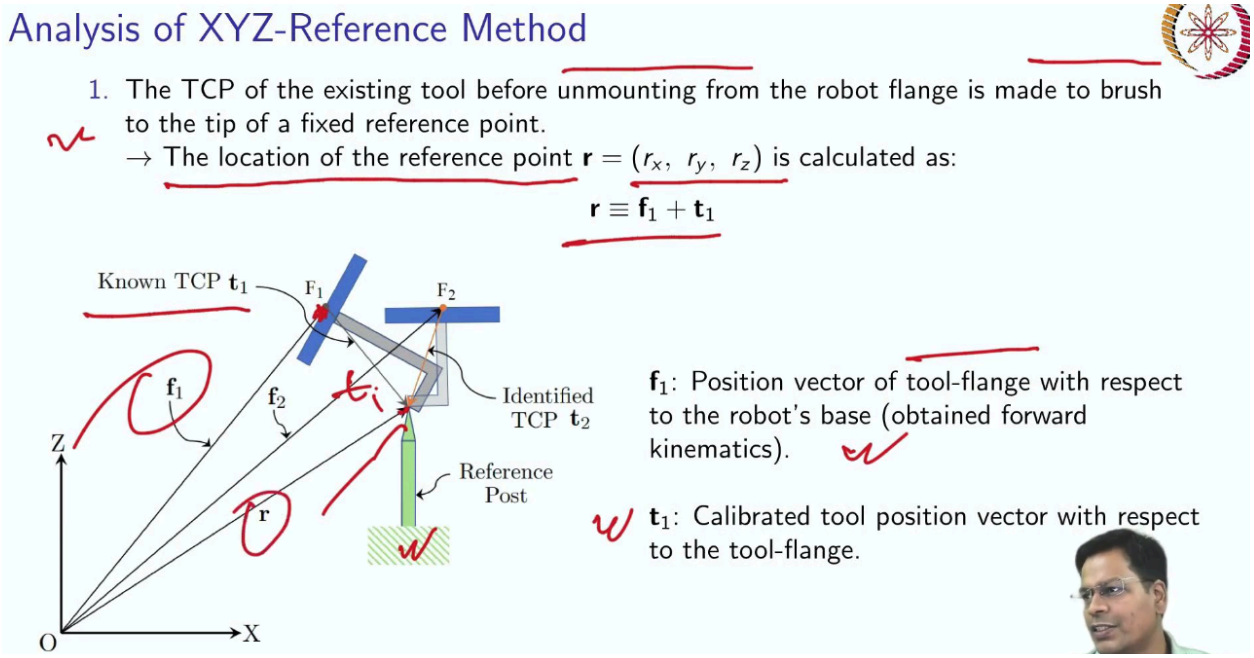


So it looks like something like this. You see, so it is. It is, you see, it is a frame which is in the board, the direction. There are sensors which can precisely check the location of the TCP with respect to the base frame. This calibration tool itself is already calibrated with respect to the base frame. So, yes, that's the way. You just need to take your robot in between that frame, and it can detect the tool TCP. So, this is ABBbullseye.

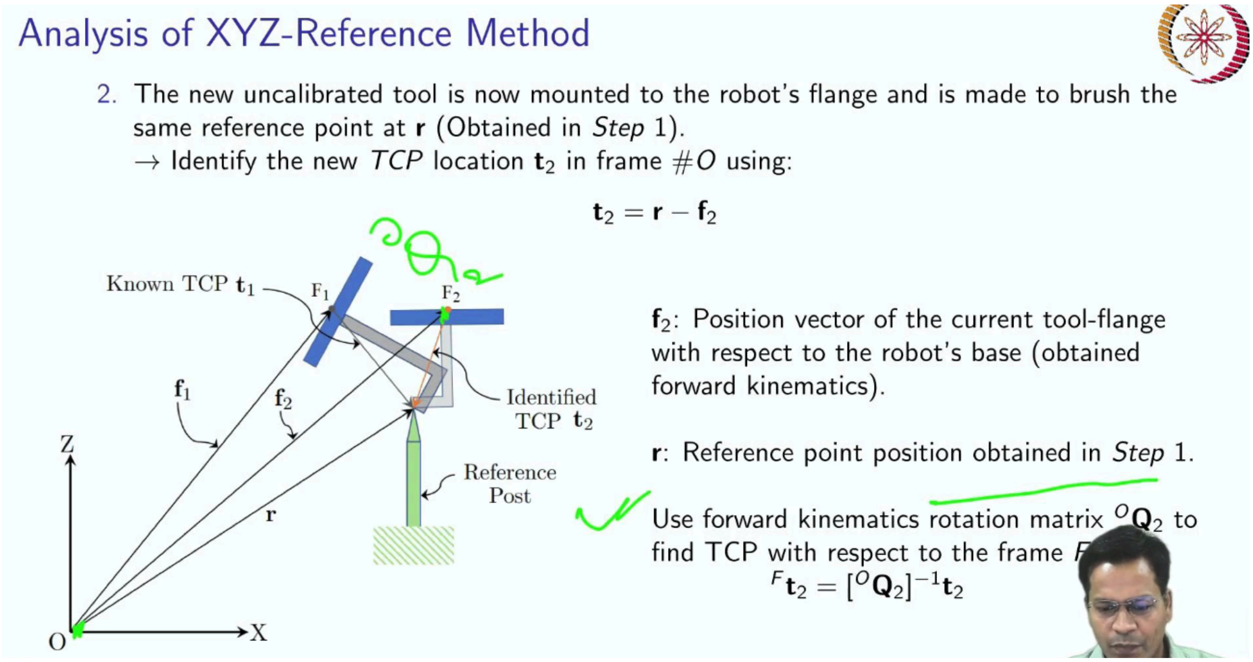
Similar is the Kukan TRACK, the Kukan track it is known as. So, it has, okay, two different stands which are there, which looks like a frame. You have to take the robot tip in between and, using sensors, it can detect the TCP location very, very precisely, and it automatically updates the location in your robot controller. Okay, so using this, you can save a lot of time and make things very, very precise, and this is very good for calibrating any welding tool.



So, there is another technique, which is known as the XYZ reference method. In this method, a new tool is calibrated using an existing precise tool. Okay, so let us say you already have a tool, which is precisely calibrated against the flange. once you have fitted it is calibrated. Do you want to change this with a new one? This eliminates brushing the reference from points from different directions to just one direction. You now don't need to brush it from four different directions, so the probability of error getting carried over to the robot TCP calibration due to the manual error is less. In this case, it is very, very less time-consuming because you, because you-are just brushing it from one direction. So, it also reduces the downtime when you exchange a tool for the other tool. So, fewer chances of error when replacing the tool with the new one. And it is just a two-step process. So, brush the existing tool, which is a calibrated tool, before unmounting to the reference point before unmounting, I will take it to a reference point, and then I will brush the newly mounted tool to the same point, okay? So, this is the brushing of the existing calibrated tool to the reference point, which is there in the workspace and the new tool. now you unmount this put the new tool and again take it to the same location. So, hope you got the idea of what it must be doing. Actually, it is calculating the location of this reference frame. So, once you put the new tool, you are taking it to the same location, which means you are taking it to a new vector. So, we will do a good analysis of this now.



So, how it works is very, very simple. The TCP of the existing tool, before unmounting from the robot plant, is made to brush to the tip of the fixed reference points. So, this is the reference post. This is the point which is here. So, what it actually had done when you did that job. It found the location of the reference point, so it found the location of this point. R is calculated using this. So, what is this? This can be obtained using. This is what this is: the position vector of the tool flag with respect to the robot base, obtained using forward kinematics. So, you have obtained this quite easily because it is a calibrated tool. So, you already know what t1 is. So, t1 is for the norm tool. It is already known. So, you already know t1. Okay, so you just take vector sum and can reach this point. So, you calculate r very easily got it. So, you got till r. Now that you know r, you are ready to take it out from the flange and put your new tool.



So, that is what is the second step. So, the new calibrated tool is now mounted to the robot plan, and it is made to brush to the same reference point at r as we have obtained in Step 1. Okay, so we can identify the new TCP location, t2. So, this is your identified t2, how it is identified. You just see now, again, that F2 is the position vector of the current tool flange with respect to the robot base. This, again, can be calculated even with the new tool flange location. Your knowledge can be calculated using forward kinematics. So, we did this, we found out. The reference point r was obtained in the first step. So, now you know r, you also know f2. Now, you can calculate this t2. So, this t2 is now identified. But now this t2 is in this frame. This is the vector which is expressed in the robot's base frame. But I want to achieve what I want to find out in the robot flange frame, this frame. So, how to obtain that from here to here, this vector? I want to know.

So, how to obtain that? So that is what can be obtained using a forward kinematics rotation matrix, again 0 to Q2, so this is known-using forward kinematics, with respect to the frame f2, TCP with respect to frame f2 can be obtained like this. So, f to t2, this transformation can now be obtained using this. You just take the inverse of this. This can be expressed in the flange frame like this. So, this is how you can quickly obtain the TCPx. What is this? This is nothing but a vector of TCPx, TCPy, and TCPz. So, this is what is obtained here. That's all. So, you see, it is quite easy to do using multiple techniques two of them are supported by some robots, and some other robots may have some different ways, but the idea remains the same. You have to get to this point, that is, the TCP point, with respect to the flange.

So, that's all for today. We have already calibrated our TCP origin. In the next class, I will tell you how to get the frame orientation, that is, the TCP orientation, with respect to the robot end effector flange. So, that is what we will be doing in the next class. That's all. Thanks a lot.