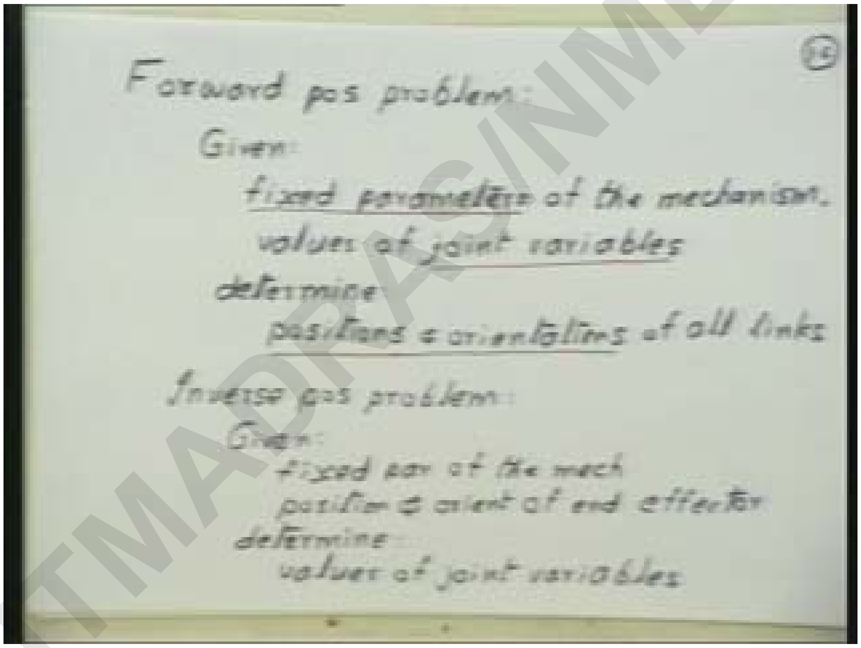
ROBOTICS

Prof. K.Kurien Issac
Mechanical Engineering
IIT Bombay

Lecture No. 15

Trajectory Planning (Time: 1:20)

[???] relate the major point that was discussed yesterday, or rather, the problem we had started addressing, we figured out that two problems in kinematics are very important to tackle and they are the forward and inverse positions problems. Later on we will look at velocities. The way these problems are defined, the forward position problem is to find the positions of the links given the joint variables. Of course, the link parameters have to be known, the mechanism parameters have to be known, and the inverse problem is to find out the joint variables which will take the end effector tools certain specified positions. So these are the two important problems to be tackled in the controller (refer slide time 01:55).

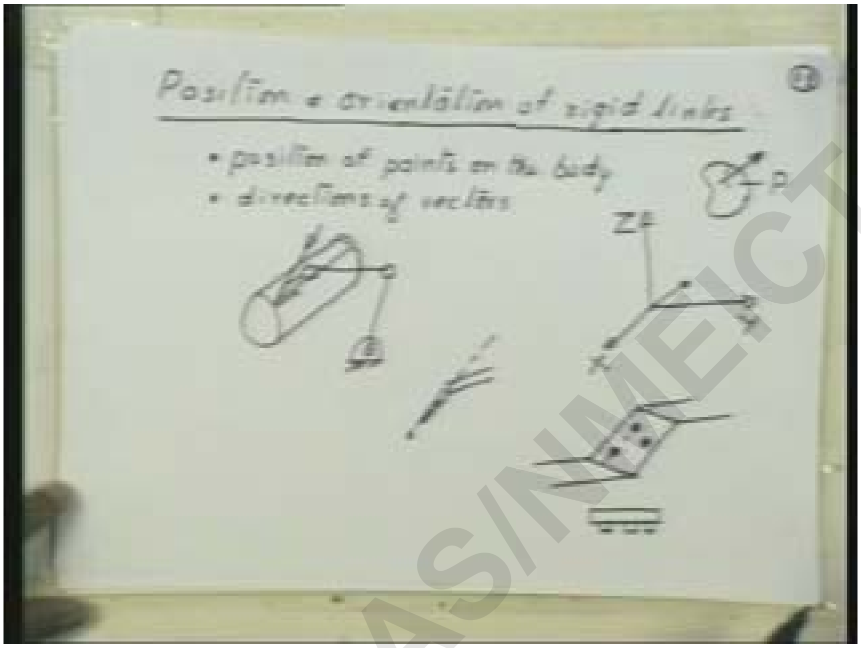


So, in order to, before we solve that, we need to understand the problem a little better, especially terms like positions and orientations in space. It is fairly easy to understand them in plane, [noise] so we need to understand what they mean in space. We also need to understand what is meant by fixed parameters of a mechanism. Again, in a spatial mechanism, you wouldn't have really seen that much; we need to understand this.

So, we had started out trying to understand what is position and orientation, so that is what we will cover today.

So we had started discussing that. So what is important in specifying position and orientation of a rigid body, or not just position and orientation of the rigid body but the other positions of certain points on the rigid body, on the last body, and maybe directions

of certain vectors on that body. If you want to point a gun at somebody, you need to look down the barrel
so that that points towards the person, or the target, and you are not really interested in any other aspects of the gun, it's orientation of position. What you are interested in is that the barrel points towards the target. So, what is important to specify depends on the situation or the particular task the robot is carrying out (refer slide time 04:29).



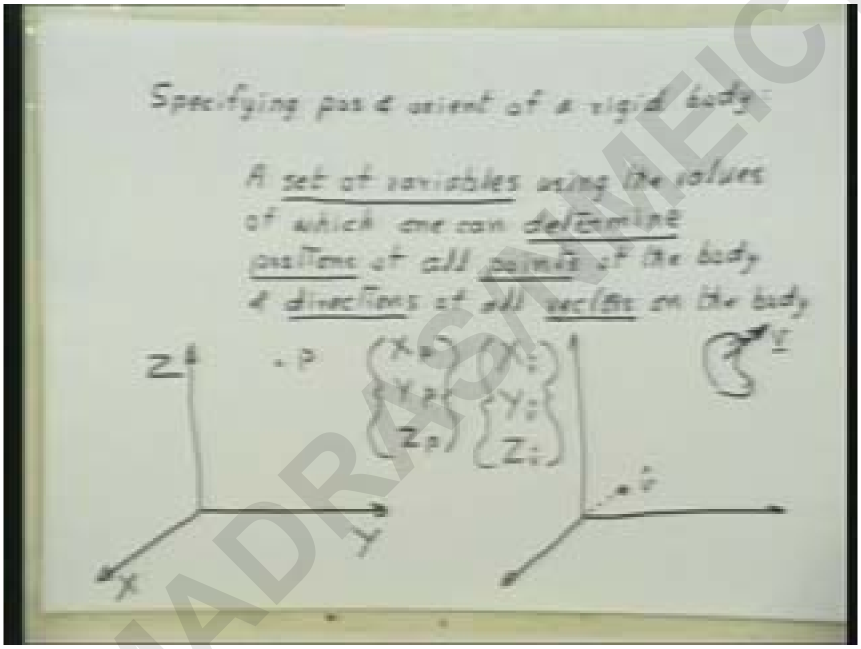
So, we had seen an example where we need to, the robot needs to lift this particular part which is three projections, take it to this point, and assemble that there, push it in, basically, into these three recesses.
This requires positioning this part over this part. Now, how do you specify it in such a way that the positioning and orientation is precise? In this, this is a case where this particular example is the case where you need to position, as well as orient the part fully, fine? We will perhaps see the example in more detail later.
Now let us look at what is really required in positioning and orientation. So what we are looking for is the following. So this is essentially what we are looking for. If we say a set of variables specifies position and orientation of the rigid body, once the values of those variables are given,
we should be able to find out the location of every point of the body as well as the direction of every vector on the body, then we can say that we have defined the position and orientation of the body, fine?
So, we need to find out a set of variables which helps us to do this, and we need to, using them, determine position of points and also directions of vectors, fine?
So, let's take what we, take a system where this we normally use for specifying locations of points. Let's say this is the global reference frame which one assumes is fixed. So, if you take a point p what we do is the position of that point is specified by [noise] the three coordinates Xp Yp Zp.

This is the vector of variables which specifies the position of the body. If you are given a vector on a body (so let me draw the body), a point, a vector which we draw on the body, which is attached to the body, fine? It's direction is given by the direction cosines of the particular vector, ok?

So, whatever be the length of the vector, if the direction is what is important we need to give only the direction vector, or the components of the unit vector in that direction. Direction cosines are nothing but components of the unit vector in that direction.

So, this way, if I call this vector v with an underscore to indicate it is a vector, then v u, the unit vector v cap, the coordinates of that, will give us the direction of this vector, which is, let's say can be written as the X coordinates of that unit vector.

So, these are things we have already learnt, so this is what we meant by finding out position of points and directions of vectors (refer slide time 10:09).



Fine. Now the problem is the following. Suppose there is a body, and we have a global reference frame, the body will be moving around in space. Suppose there is a point on the body. How do we specify this is the point that we mean, because there could be several points on the body, 1 here, 2 here, 3 here, 2 here, which point are we talking about?

Right. So there has to be something with respect to which we define the position of points on the body. Either we mark it, then we can see it while the body moves. Mathematically, that is not possible. We need to have a way of specifying that.

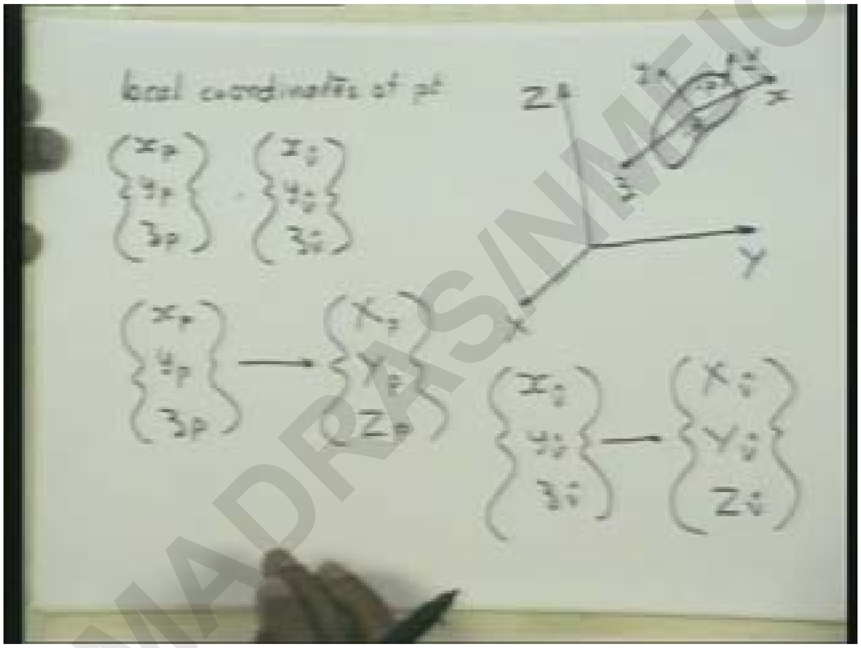
So, what is used is precisely what we had seen. One of the best ways of doing it, there are many ways of doing this, that is, essentially locate point on the body itself .

You set up a reference frame on the body itself; let's say, I indicate it with small x, small y, small z, the coordinates, so this is a Cartesian reference frame fixed on the body. Now it is easy to say which point you are talking about, right?

Now the coordinates of p in the body reference frame, once you give, you know where the coordinates, where the body reference frame is, how it is attached on the body, so points can be specified with respect to that. Same thing, vectors also can be specified with respect to that, right, in slightly the same way we did with respect to the global frame, fine?

These local coordinates or points, so I can give it the small x notation, we will change it a little bit later on, fine? You have position of points, this is the local position of the point, p, and if you take the vector, v, its local direction could be x v cap, small x v cap, small y v cap, and small z v cap, fine? So we need a scheme basically for locating points and direct, finding out direct, specifying direction of vectors on the body itself, or specifying the vector on the body itself.

So now, with this we can define what we are talking about. What we are talking about is basically a set of variables, by which a body can be located and oriented which essentially means that given the values of those variables, we should be able to find out the position of every point on the body and every vector on the body, right? <u>(refer slide time 16:00).</u>



Which means, given the location of a point p on the body, x p, y p, z p, this is the local coordinate, we need to find its global coordinates. That is, given x p, y p, z p, the local coordinates, by some means we should be able to find its global coordinates,

fine? Same thing with respect to the direction of vector. Given the local direction of the vector we need to find out the global directions of that vector,

right? And we are saying that variables which we use in order to help us do this are variables which specifie position and orientation of the body, because, what these values are would depend on those variables, fine? What the global location of the point is, and the local, global direction of the vector is, depends on how the body is oriented and how the body is positioned, fine? So, this is what we are talking about. Now this can be done with the three-dimensional geometry that you have learnt.

Let me first tackle orientation, the, not orientation, but the other, how we will transform the direction vectors from local coordinates to global coordinates. So, take two reference frames, one global, one local.

Now consider a vector in this. No, let me give notations for these. So these are unit vectors whose coordinates in the local frame are known. We need to find out its coordinates in the global reference frame, ok?

So, let's assume that the direction cosines of this are these vectors, these vectors. Can you see clearly? Not very clear? If I draw the local again larger here, look at the three projections along the three local coordinate directions. Those are the direction cosines of this vector, or this being a unit vector, they are nothing but the coordinates of the vector, fine?

So, this is, the length of this particular vector is small x v cap, right? That is how we get the notation, small y v cap, and small z v cap. So we know this, we need to find out the global coordinates of this. That is essentially obtained by transferring this vector here and projecting it along the global axis directions, and those coordinates are the global coordinates of that vector, fine?

Now, we can do the following. These vectors which are drawn here, or rather, which are here, are vectors in this global coordinate frame, right? And their sum gives you the total vector v cap, correct? So, if I am able to express these vectors, this small x, small y, small z, or rather, this x v, the vector with length x v in this direction, y v in this direction, z v in this direction, and sum them up, then I get the global coordinates of that vector, that is x v cap, y v cap, z v cap, ok?

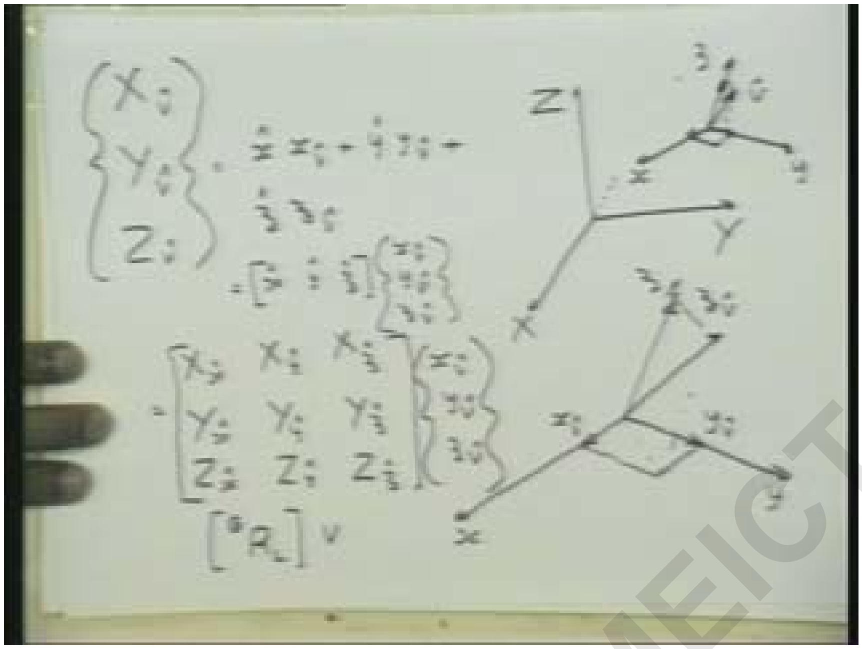
So I can write this in the following way. Look at unit vectors in that direction of the local axis. That multiplied by these lengths, x v cap, y v cap, and z v cap, are the three vectors I am talking about, right? Sum them up. So, let me call the unit vector in the x direction as x cap, fine, multiply it with this length which is x v cap, you get this vector, fine? Similarly, y cap, the unit vector in the local y direction, into y v cap which is the length of the vector, plus (all of you agree with this, right?)

I can put it in simple matrix form to make the notation more compact as follows. This is one column, this is another column, this is the third column. Each of them are three-dimensional vectors. These are the unit vectors expressed in the global frame of the local frame directions, and then I put all this into a vector; I have x v cap, y v cap, z v cap. If you remember, this is nothing but the direction of the vector, v, in the local frame. This vector is nothing but the local frame vector v cap, fine? And this is nothing but the global frame vector v cap,

so we have a matrix which premultiplies it to give you this transformation, and the elements of this matrix are the unit vectors in the directions of the local coordinates expressed in the global frame.

So let me give a notation for that. So this is the vector x cap which is the unit vector in the local x direction, and they are its components in the global x, y and z directions.

So similarly, multiplied by this local vector so this is a matrix which transforms a unit vector in the local reference frame to a unit vector in the global reference frame, and this really [???] the unit vectors, they can be vectors of any size. (refer slide time 25:07).



So they are essentially free vectors, that is what is more important and they are essentially free vectors. Now, this particular matrix is called the rotation matrix, and given the notation R. Because this captures all the information about orientation, why it is called rotation, I will come to later, a little later, we will, for the moment, call it orientation matrix, and this takes a vector in the local frame to vector in the global frame which is given notation like L to G. L as a suffix on the right hand side, and G as a suffix on the superscript on the left hand side. This is a matrix. What is the size of the matrix?

Three by three, right? So this transforms a vector. Now it needn't be a unit vector. Can be any vector, but it's a free vector. So I will say that a this is in the local frame with this subscript l, so you will get the vector in the global frame, the curly brackets typically for vectors, and square brackets typically for matrices,

fine? This is fairly clear. Later we will see that this particular matrix is what is, this particular matrix has been able to give you the directions of vector in the global frame. Once it is given in the local frame, and this turns out to be, this is called the rotation matrix [noise].

So, let us look at the elements of the rotation matrix in a little more detail. Any questions here? It is simply addition of some vectors to give you the global value.

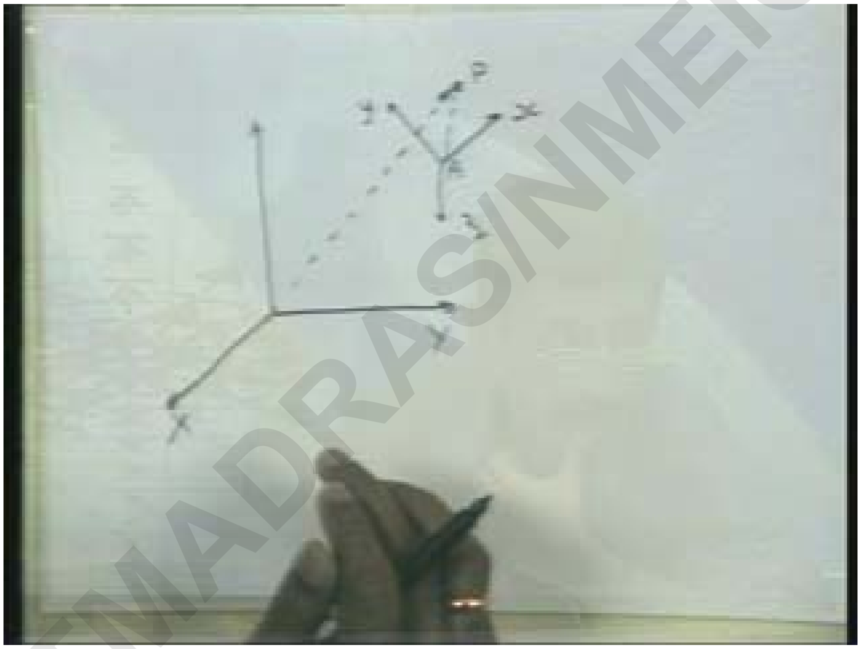
That is how we derive the [???]. So, what is important are these elements. Now, looking at the elements here itself, we should be able to note a few very important facts about these elements.

I said that these elements, these particular vectors, column vectors, are the vectors, unit vectors in the direction to the local axis, so any two such columns are orthogonal,

fine? Let's [???] forward, and also each of them is a unit vector, fine?

So, such a matrix is called an orthonormal matrix. Such a matrix is called an orthonormal matrix. I will, because of some of these things which I said about orthogonality and length being unity, it turns out that this nine elements are not independent, fine? They are more than what you really require to do this transformation, but it is very convenient to use them. We'll see these points a little later.

Now, let's look at typically what numerical values –
yeah we did not, but it it um ya yeah um. Actually, this is related to the concepts of three vectors which are mentioned, so, I will state the question that he asked, and clarify that, because in the camera, it will be captured, what you ask, fine? What [???] is asking is essentially this:
you have a global frame, and you have some body to which you attach some local frame, let's say x y z. Suppose there is a point here and it's vector in the local frame is this. The vector which defines the position of that point in the global frame is this,
right? So, if it, if this turns out to be unit, let me called this point as A. If this vector Ap is the unit vector it will no longer be the unit vector here, right? That is the question that you ask? Or something else? Right, right, yeah. No it is fully reflected actually, Only thing, the problem I am solving is not this. This is what I am going to come to. This is the problem of finding the position of a point in the local frame, the position of that point in the global frame given its position in the local frame. (refer slide time 29:44).



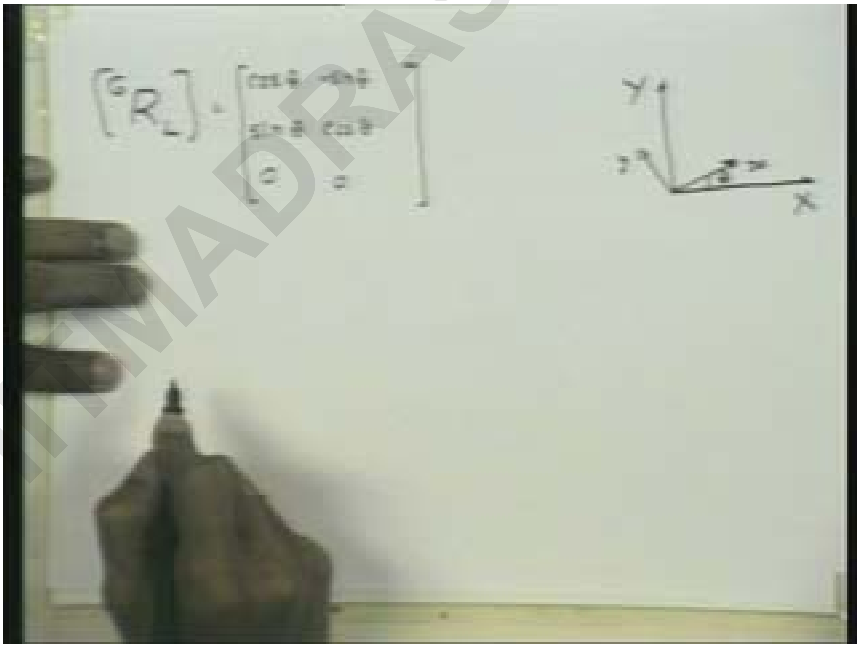
So, imagine that there is some point in space, and you express the position of that point with respect to the local frame. You want to find out the coordinates of that point in the local frame. So, we are looking at coordinates of points. The transformation that I just now talked about, are transformations of free vectors,
fine? So, take this vector which is there in the local frame. I am transposing this to this particular origin, bringing it here. I am looking, I am trying to derive the coordinates of this vector on the global frame, or rather, this particular point, Once you transpose it like this it becomes the coordinates of that point, so these are free vectors which you can translate whatever way you want. What I am expressing is the direction and magnitude of the vector not the location of the point at which the tip of the vector turns out to be, if I place the vector somewhere, fine? So, these are two different things, fine?
So what happens is that if I multiply this with 2, this also gives[???]. So that is captured here, when v is doubled, V G also will be doubled, fine? That point is clear?

Ok, now let us look at a typical transformation. How does it look like? So let this be x, let this be y global, and let this be a local frame, x y, small x, small y. So this has been rotated by some angle, theta. The origins are coinciding, and I have rotated, and initially the local frame is aligned with the global frame. I then rotate the local frame with respect to the global frame about the z axis, fine? So z axis coincides still. They are aligned still. x and y axis are not aligned, fine?

So now let us try to write this matrix L with respect to the global. What do you think would be the element? So, if you remember, if you look at your notes, this particular column is the coordinates of the unit vector along the local x axis. Now, if you look at each of them, these are, you also know that these are direction cosines being unit vectors, so each of them, if you look at this, the cosine of the angles between the global x axis and the local x axis, cosine of the angle between the global y axis and local x axis, cosine of the angle between the local global z axis and the local x axis. That's what it is, fine?

So, having done the rotation the way we just now saw, we can write down the the coordinates very easily. Look at the unit vector along this direction. Along the global x direction it will have cos theta as the component, right? Along the local y axis it will have sine theta is the component, fine? And what about the global z axis? What is the component of 0?

Similarly, the unit vector along y. What are its components in the global reference frame? Minus sine theta, then cos theta, zero, z is out of the plane, right? So the components of the local z with respect to the global frame, no, the entire vector, zero zero 1, right? (refer slide time 34:59).



So this is exactly what a rotation matrix will look like if the local frame is oriented with respect to the global frame in this fashion, ok? We will come to more complex orientations a little later. I just wanted to show you an example.

Now let us look at the next point which is given the position of a point in the local frame. I want it's, we want it's position in the global frame, so let's do that.

Don’t worry about these lines I am drawing. This is to ensures that the lines don’t get mixed up. Later on you can draw. So, I have the local frame here, as I have shown, the global frame here. There is this particular point, Q, on the body, fine? The origin of the local frame is p, so the vector, PQ, is known in the local frame. I want the position of Q in the global frame. The position of Q in the global frame is this vector. Let me call the origin of the global frame as O, right? So, what we want, that is the global coordinates of the vector of the point Q, our notation for that was this I suppose. This is what we want. This is the sum of these two vectors OP and PQ, and these are vectors which are expressed in the global frame, remember, fine?

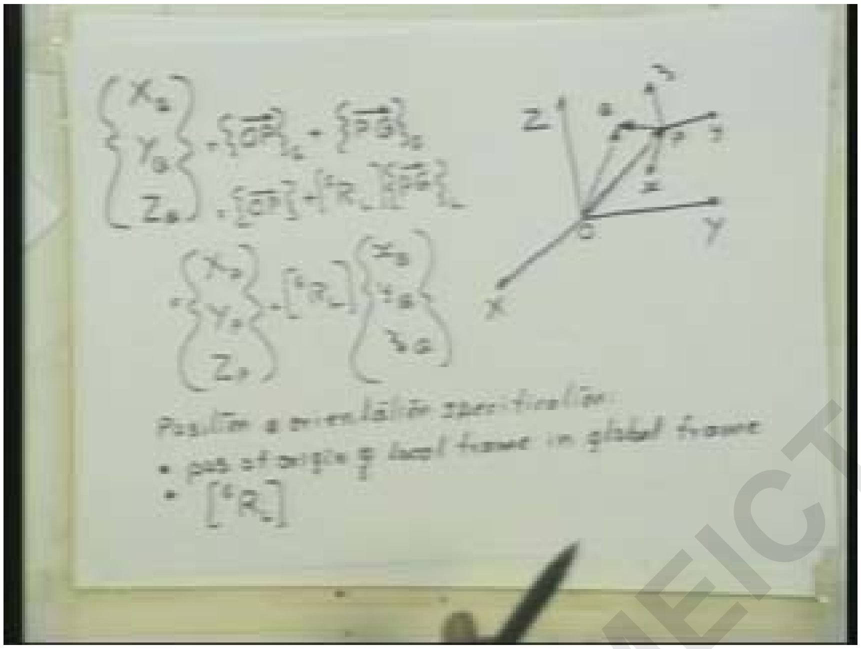
So now, look this vector PQ. If I can get the global, if I know local coordinates of the vector PQ, I can do the transformation which I just showed in order to get its global coordinates, right? So I can write this as OP, plus the rotation matrix from local to global, multiplying the local coordinates of PQ, right?

Fine. So now this can be written as OP is the position of P in the global frame, which our notation for that is capital X p, capital Y p, capital Z p, that is the origin of the local frame in the global reference frame, plus this rotation matrix from local to global into multiplying the local reference local coordinates of the point Q, because P is the origin of the local frame. So this is nothing, this vector is nothing but the local coordinates of the point Q, right? So, that is small x Q, small y Q, small z Q, fine?

This operation transforms points on the local reference frame which is attached to the body, so points on the body to the locations in the global reference frame. So, this is what we set out to do, right? So what are the variables that we used to do this? This is what we, the variables we used to do this, is what we defined, what we said, or those that specify the position and orientation of the body, because when the body moves this variables change, and with these variables you can always find out the location of points on the body in the global frame, fine?

So, what are those variables? Right. Elements of this matrix, nine of them, right, and elements of this vector, three of them. Correct? That is what helps you transform the points on the local frame to the global frame. What about the transformation of free vectors in the local frame to global frame? What is required for that? Just this matrix, right?

So, one way of doing it, this is not the only way, this is a very convenient way, and we will be using this mainly, but one way of specifying position and orientation of rigid bodies is to give these two vectors one position of origin of local frame. So this is in the global frame, and the other is this rotation matrix, local with respect to global. For these two things taken together, specify position and orientation of rigid body, and with these, we are able to do what is important for us, which is, specify directions of vectors and locate positions of points. (refer slide time 44:07).



Ok, now look at this a little bit. Suppose I fix this. Fine?

Imagine that the body is like this, and the global frame is like this, and we have got these matrices and vectors. Now I am going to fix this, and then I am going to vary this vector, allow it to change. What happens to the body? Right? It purely translates with the orientation which is already given to it by this particular specification fixed, ok? What does it mean? It means that this particular matrix table is able to fix the orientation, ok? Now look at the other way. I fix this, and I change the elements of this matrix with the constraint that the columns are unit vectors, and it is orthonormal, with the constraint that it remains orthonormal, fine?

In addition, there has to be one more constraint, but let me say, it remains ortho. What happens to the body? It rotates, but it has a special type of rotation. It rotates about a point, the point P, right? So, when I fix this, and when I change this, it rotates about the point P, right? So, another reason why it is called the rotation matrix.

Now, let as consider the following. What happens to the points on the body and vectors on the body when we do this?

Ok. So, imagine I am fixing this and I am changing this. The body translates. What happens to points on the body? They move with the body, right? They move with the body with respect to the global frame, ok?

What about vectors fixed to the body? They move parallelly, their directions don't change. So the directions of the vectors don't change. That is precisely the definition of translation, as why some of you [???], ok?

Now, what about the other thing. I fix this and change this. What happens to vectors? Their directions change, right? What about points? Except for P everybody else changes, right? So this you have to keep in mind, that when I change the rotation matrix the directions of vectors as well as positions of points change. That is, when I fix this and change this alone the positions of points change but the directions of vectors don't change, ok?

So, [noise] this basically, if you fix this, then body translates when you change this, and if you fix this and change this, the body rotates about the point P, [noise], that is why it is called the rotation matrix.

Now [noise] this is not the only way we can get this particular transformation of global local locations of points to its local global location, and local direction of the vectors to its global direction. This is not the only way we can specify we can get this transformation, or, this is not the only way we can specify position and orientation of rigid bodies. So I will just list down few more alternatives. We can see why they are equivalent. All of them will be able to give you the same thing.

So, one is position of three non-collinear points. So on the body, mark three points; shouldn't be on a straight line, fine? These three points shouldn't be on a straight line.

Now, if you give the location of these three points in the global frame, that, by itself, defines the position and orientation of the rigid body. What it really means is that it fixes the rigid body in a particular position and orientation in a global frame, fine?

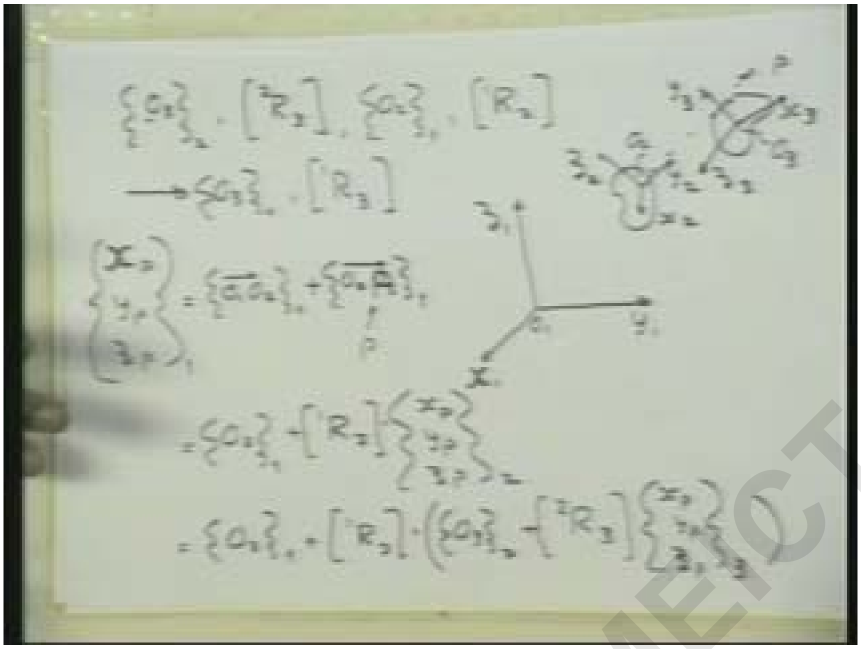
So, how it is possible now to transform those points and vectors in the local frame, or on the body to the global frame, is something we will have to figure out, ok?

Now, another possibility is positions of two points, non-collinear points on the body, [???]. So, [noise] mark two points on the body, mark a vector on the body which is not along parallel to the line between the two points. Yeah; not really. So this question asked was, "Don't you need one more point?" This being a three-dimensional body, three points always lie on a plane, so this appears to specify the location of a rigid body which is planar, right? Don't you need a third, fourth point which is not in this plane? The answer is "no", you don't need it. So you should figure out why it is.

Yeah, but that is already given by these three points. Since you have the plane, you have the normal frame, right? Yeah, so you can still, a plane could have orientation based on the three points that you specify. You can find out the orientation.

So these are definitely question that you should ask. So, in the process of time to figured out how you can transform points, you will be able to figure out why this is possible, why this is, ok? I would like you to go through that exercise. So, treat these as exercises. How you will do the transformation if you have this? Basically, all you have to do is then convert this to what you already know into the picture of the reference frame. If you are able to do that immediately you can do the transformation through that route, ok?

So, this is another alternative. Another is position of one point is all on the body; the vectors and points are all on the body. All these are equally good, equally good in the sense, they will do the job maybe little more difficult to do the job with them. The job is that of transforming vectors and points, fine? And you could figure out other ways of doing this job. (refer slide time 01:01:34).



Now, suppose you have a body here to which we have attached the reference frame, and you have another body on which there is another reference frame as attached, and now you have the global frame. Now I will slightly change the notation and call this x 1, y 1, z 1 because there is more than one body, more than two bodies.

So, x 2, y 2, z 2, let's say z 3, x 3, y 3. Suppose you know the position and orientation of this body with respect to this, and this body with respect to this. Can you find out the position and orientation of this body with respect to this? Fine,

this is an important question, so important that in most of our forward and inverse kinematics we will use this, ok?

So, position of this body with respect to this; what does it mean? It is the position of this point with respect to in this coordinate frame, and the rotation matrix transforming vectors in 3, 3 vectors in 3, to 2, right? So let us write it down, so let's also call this as, let's say o 1, o 2, and this is o 3, fine?

So position and orientation of body 3 with respect to 3 is given by the vector o 3, so let me write it as a vector. So o 3 is a vector expressed in the second reference frame, sorry, o 2, o 3 as a reference frame, so, o 3 is the point o 3 expressed in the second reference frame, and the rotation matrix 3 with respect to 2. This is the position and orientation of this body with respect to this, ok? And we have the position and orientation of this body with respect to this which is o 2 expressed in the first reference frame, and the rotation matrix from 2 to 1, [noise] fine?

And from this, we should be able to calculate, what, o 3 in this reference frame, that is the position and orientation of this body, and this body, that is o 3 in the reference frame 1 and rotation matrix 3 with respect to 1 [noise]. One way this can be done is to look at transformation of points in 3 with respect to 1, right? So take a point here, P.

So try to find out its position with respect to 1. So there is X, sorry, x p, y p, z p in reference frame 1, fine, is equal to [noise],

how do you specify that; so location of this particular point could be made up of a vector from here to here and a vector from here to here, right?

So let me call it o 1, o 2, that segment in 1, fine, plus this particular segment o 2 o 3, again in 1, right, or let me not do that. I will consider this particular vector, o 2 P. So instead of o 3, let me use P, fine, so this is P in 1, ok, o 2 P in 1. So I can write it as follows: o 2 with respect to o 1, fine, is already given, correct, so o 2 in 1 plus o this p with respect to o 2, now we can use this transformation. We can use the position of and orientation of this body with respect to this body to find out the position of p in this reference frame, that is, and then we can, one moment, this is this vector in, if we consider this vector in the second reference frame we can transform it to the first reference frame, right, this considering it as a free reference frame. So, what do we get? We have R 2 in respect to 1, multiplying position of P in this reference frame. That is, x p, y p, z p, in reference frame 2, fine?

So let as expand further. Now look at this [noise] vector position of P in the second reference frame. We can express in terms of its position in the third reference frame, correct?

So, that involves, what, O 3 in the second reference frame, so, that is known, right? That is known, plus rotation matrix 3 with respect to 2, multiplying the position of P in the third reference frame, right? So, x p, y p, z p, ok? What we have done is, we have put all this in terms of what we know, ok? So now, if you expand and simplify.

So, what we have done is understand one way of specifying position and orientation of rigid bodies and see how we can transform points and vectors from one to another, that is essentially what we are doing, so remaining we will see in the next class. Thank you.