

IntroductionToRobotics-Lecture02

Instructor (Oussama Khatib): Okay. Let's get started. So as always, the lecture starts with a video segment, and today's video segment comes from 1991, and from the group at the British Columbia and it deals with Bibet walking so there should be some sound.
[Video playing]

[Inaudible]

Instructor (Oussama Khatib): Well, maybe we need some motors, right. Okay. So today we're going to start covering kinematics and kinematics, as I mentioned last time, kinematics is very, very important, the models that describe the robot position, the robot frames, and links, and joints. So we're going to go over the basics in describing a task, the models that we can use to determine the position and orientation of the end-effector.

Then obviously when we determine the location of a link we need to be able to transform that description to the next link or to describe the position and orientation of the end-effector in our previously link so we need really to handle transformations. Then we need to discuss how we represent the position and orientation. There are many different ways through which we can describe a position or an orientation, and we will discuss a few different representations.

And I'm going also to describe a little bit what is manipulated, what is a robot arm, and then what are these joints, what are the degrees of freedom of a manipulator, how we can represent the position of a manipulator. So a manipulator is defined by a set of links connected through joints. The first one, the first of the thing is fixed. We call it the base. And the last one is actually this gripper. The whole purpose of the manipulator is really to move this gripper and place it in space to do manipulation.

Obviously later, we will see that it is possible to use the body, the links themselves, to do manipulation. We call it whole body manipulation. But for now, we are really interested in locating this end-effector at the same time locating any other links that is moving. So we will see that there are two types of joints that we are going to consider. There will be other possible types of joints, but we can see that any set of joints could be reduced to those two types of joints, the revolute joints and the prismatic joints. A revolute joint allows you to rotate about a fixed axis, and a prismatic joint allows you to translate about a fixed axis.

And this motion is done along or about one axis so it is one degree of freedom motion. So as I said we have links and those links are a number of n . We are going to call n is the number of moving links plus one base link, the fixed link, and we have joints of two types, revolute joints and prismatic joints. So the idea is, we are going to work with one degree of freedom and this is interesting because knowing that we have only one degree of freedom joints then we will be able to connect to this to the last coordinates, as we will see.

And as I said, if we have a joint like a spherical joint, would you know how many degrees of freedom a spherical joint would have?

Student: Two.

Student: Three.

Instructor (Oussama Khatib): Three, yeah. Three. So what we would do is we will then use three revolute joints with zero links length, and then we will introduce these joints and links to represent a spherical joint. Okay. Now, we have this manipulator in this configuration and the question is how can we represent the configuration of the manipulator? What would be a good way to represent the configuration because we need to know where the manipulator is in space with respect to a fixed frame?

So there are many different ways. We can go to each link and try to fix that link. So we can take maybe a given link and say we are going to locate this link with several vectors, lock it there. So if we use like three vectors and three different points the link is defined, and that would give us the configuration of that link. Now we are going to use in that case three vectors, each vector has three parameters in 3D, so we have nine parameters to describe each link, and we have n links, moving links, so we will need nine ends.

A lot of parameters. And this would be one of the representations. So the description of the position using a set of configuration parameters can involve a large number of parameters, and each of them is fine. So any set of parameters that describe fully the configuration is called a set of configuration parameters. So in this case here we have nine parameters per link. Now we are really interested in a particular set of parameters – configuration parameters that has minimal number of parameters involved.

We don't need all these parameters in that three vectors, three points because the points are fixed so there is a contrast between these points, and that is going to show us that these three vectors are not independent. In fact, we will see that we will be able to describe the configuration of the link with much, much less parameters. So that brings us to generalized coordinates. So a generalized coordinate is essentially a set of configuration parameters that brings parameters that are completely independent, and working with those coordinates is very interesting because you can use them to find the dynamics later as we will see.

We can count them, and the number of generalized coordinates gives you directly the number of degrees of freedom of your robot. So let's analyze how many parameters you would need to describe the configuration of a manipulator with a set of generalized coordinates. So we have this manipulator connected through joints, and in order to count how many degrees of freedom we have, what I'm going to do, I'm going to remove the joints.

So let's remove these joints.

Now you have n rigid body in space. Right? If we take one of those rigid bodies, one of the links, how many parameters do you need to describe the position and orientation? Six. Three for the positions and three for the orientation. So with six parameters we have a description of one rigid body. Now we have n moving rigid body. In total, we need $6n$. Now let's think about those constraints introduced by the joints. If we put back the joints, we are going to introduce constraints. Now a joint has one degree of freedom. Right? We said one degree of freedom. So how many constraints, the placement of a joint is going to introduce? Each joint will introduce how many constraints? Hmm?

Student:Five.

Instructor (Oussama Khatib):Five. Exactly. Because it is going to allow only one degree of freedom. So if we think about the number of constraints we will see that we will have five constraints per joint and that leads to $5n$ constraints. Yes?

Student:Would it matter [inaudible] whether it's a position parameter or an orientation parameter that's [inaudible].

Instructor (Oussama Khatib):Well let's look at it. So if we have a joint, it's going to introduce constraints on the rotations and the position, so if we place a revolute joint it's not going to allow any displacement, and it's not going to allow rotations about octagonal axis to the axis of rotations so that's five constraints, three positions constraints and two rotation constraints. In the case of prismatic, a prismatic joint is not going to allow any rotation; it's just translating, and translating about one axis so the two other axes are eliminated.

It doesn't matter. It is still five constraints. So now, let's do the count. So we said we have $6n$ parameters before placing the joints and now we place the joints and we have five constraints. So the question to you, how many degrees of freedom are there? It is going to be, it is going to be the difference, right? So $6n - 5n$, the answer is n . Okay. So indeed, we have just n degrees of freedom, which is really nice if we have one degree of freedom joints and a manipulator in our robot we can be sure that we are going to have n degrees of freedom, the number of joints.

Now I'm talking about a manipulator with fixed base. If I take a humanoid robot and I do the same things – at a given configuration if we lock one of the feet of the humanoid robot, this is correct, but a humanoid robot can move. So in that case the base is moving, and the base has six degrees of freedom. So it will be n plus 6, and fortunately, those last six degrees of freedom of the base are not actuated. There are no motors, and that's what makes the control for humanoid robots very hard.

However, in the case of a manipulator, if the base is fixed we have n degrees of freedom for n jointed robot. Okay? Clear? Okay. So let's go through the end-effector now. The end-effector is this last rigid body in the system so it has all the freedom before to position and orient on the end-effector. So we can think about a point on the end-effector that we are locating, and we can think also about the orientation of the end-effector, how

we orient to this end-effector so there is a sort of frame attached to the end-effector rotating with it, and that allows us to describe the position and orientation of the end-effector.

So if we have just one end-effector in the robot, essentially we have just one rigid body, at most we need, depending on the freedom because some robots can only move in the plane, some robots can move with restricted rotations or orientations or positions so at most we are going to have six degrees of freedom for the end-effector.

But again, we can represent that freedom of the end-effector that is the configuration of the end-effector with many different parameters. So we can talk also about configuration parameters for the end-effector and we can see that some of those parameters can be dependent and when they are independent they form a set of generalized coordinates and then we can have a description of the end-effector using a set of generalized coordinates, task coordinates or what we call operational coordinates.

So if they are just configuration parameters without the condition of independence then we can talk about these end parameters describing the position and orientation of the end-effector. So this is the definition that is we have a set of parameters describing the position and orientation with respect to a fixed frame. Let's see, give me an example.

I need example of a set of parameters that describe the position and orientation of a rigid body end-effector. So to describe the position and orientation of a rigid body, this is a rigid body, here is a rigid body. I would like to describe the position and orientation with respect to this frame. So –

Student: Well, where the orientation sometimes tilts, yaw and roll –

Instructor (Oussama Khatib): Okay, you used three angles. If we use like three [inaudible], or fixed angles, we can find this orientation for the end-effector. And for the position?

Student: X, y and z.

Instructor (Oussama Khatib): X, y and z. So we can take a vector and locate one point, fixed point. So with one vector and three angles, we can describe the position and orientation of the end-effector. Yes?

Student: Isn't that six parameters?

Instructor (Oussama Khatib): Perfect. Six parameters.

Student: But you had said five earlier [inaudible].

Instructor (Oussama Khatib): Oh, I was talking about how a joint introduces five constraints on a rigid body instead of – so imagine the base, imagine the next rigid body,

it was free completely to move, and now I put a joint, and now it has only one degree freedom left.

Student: Oh.

Instructor (Oussama Khatib): So, here we have a very nice example of a representation that is minimal representation. So anyone can give me other presentation that is not minimal?

Student: [Inaudible] three points [Inaudible].

Instructor (Oussama Khatib): Select three vectors or three different points. That would be nine parameters. So that would be a set of configuration parameters. Where we will see also that – have you heard about Euler parameters, how many of them you have? Four. So quaternion or – the reason we will see later is that when we use three angles we have a problem. We have a problem tracking this rotation continuously. There are configuration where the representation becomes singular, and we need a different set of parameters really to keep track of the orientation, and we introduce Euler parameters.

Have you heard about direction cosines? No? Vaguely? Okay. You heard about the rotation matrix, we're going to see this in a few minutes, but you heard about the rotation matrix, correct? So you have a frame, and you're looking at the relationship between the two. Well, the rotation matrix, if you take this rotation matrix between two frames so the description of this frame with respect to a fixed frame, essentially if you take this matrix which is three columns, three vectors, so it's nine parameters.

Well if you take that matrix, these three vectors form the so-called direction cosines of this frame with respect to this frame. And we can use directly that matrix, that description. So we will have nine parameters only for the orientation plus the position three parameters, so we will end up with 12 parameters describing the end-effector. And we will see why we will go that way or why we go to other parameters, and we will examine those singularities of the representation.

So a particular coordinate that we are interested in this set of independent coordinates, we call them operational coordinates or task coordinates. Essentially though, we are looking at the operational point where the robot is acting, where we defined the task. For instance, the task could be if I'm going to grasp, the task is somewhere in the middle between the two jaws as I move this is the point I'm controlling. But if I have a tool the task will be here and an operational point would move depending on where I'm going to do the interaction.

So this is – the first definition is operational points so you'll have three degrees of freedom, and then you add three degrees of freedom like the three angles to form a set of independent parameters and that gives you a set of generalized coordinates or operational coordinates. So this number – so before you remember it was m if it was not independent.

When it is independent we call it m_0 to point out this is independent set of parameters, and that gives us again the number of degrees of freedom of the end effector.

So an end-effector of a robot with six degrees of freedom moving in the three dimensional space, the end-effector itself can be positioned anywhere, and oriented anywhere so it has six degrees of freedom. And that is the most number of degrees of freedom it can have. Now if we go to the plane, if we have a robot that's moving just in the plane, how many degrees of freedom do you expect to see for the end-effector? My plane or robot, it's moving only in this plane. It cannot go out of this plane.

For the position, how many we need in the plane? Two, x and y . And for the orientation?

Student: Just need one.

Instructor (Oussama Khatib): One. Only one. So three. So if we have a plane or robot, then we will be talking about $m_0 = 3$. And you have different robots with different characteristics that ends up to give you m_0 that is equal to 4, or 5 or 6. At most you have six for one end-effector. So now we defined operational coordinates or task coordinates, we defined joint coordinates. And here is an example if we take a plane or robot, so just three revolute joints, $\theta_1, \theta_2, \theta_3$, and this robot is moving in the plane. So we have sort of representation of the joint, so for here you have 80 degrees, 45 degrees and 50 degree representing this configuration of the manipulator.

One way to think about it is to go and represent this whole manipulator as a point in a three-dimensional space, and that space is $\theta_1, \theta_2, \theta_3$ and that would be the joint space where this point, θ , which is the vector $\theta_1, \theta_2, \theta_3$, represents the configuration of this manipulator. So we call this the joint space or the configuration space. And this space plays a very important role in motion planning.

We talk about configuration space, and we talk about planning motions in configuration space. So planning the motion of θ , and we talk also about all these obstacles that we have in this space, in the physical space, that we map to that space that becomes configuration space obstacles or C obstacles that represent the how obstacles in the real world are mapped to that abstract configuration space.

And then we can do the planning around those obstacles. Now for the end-effector, as I said, we locate the end effector with our vector x, y , but that doesn't define completely the position and orientation of the end-effector. We need also to define the orientation or some angle. So we need α , and then x, y and α represent fully the position and orientation of the end-effector, and that defines the three coordinates, operational coordinates for the end-effector.

And obviously we have with that space, the operational space, which is now the combination of x, y for this example, and the orientation α of the end-effector and that is a point. So the robot is reduced to a point, θ , in configuration space and its end-

effector is reduced to a point x, y, α in the operational space, and that represents the manipulator and the end-effector.

Now these two spaces – so the first space is fully describing the configuration of the robot, but imagine that we add one more joint on this robot. The end-effector is fixed, but the robot configuration can vary because of the redundancy we introduced by adding one more joint. So redundancy in this example here you can see we have four joints, and the end-effector still has three degrees of freedom.

So for the same configuration, you have different possible configurations of this structure, of the links, and that means we have redundancy. So we talk about redundancy and we call the robot redundant if the number of degrees of freedom of the robot, n , is greater than the number of degrees of freedom of the end effector m_0 . So if we have this situation, m_0 is equal to 3, and n equal to 4. Then the robot is said to be redundant. And redundancy is very important in order to reach and have accessibility.

You cannot just work with the motion of the robot, for instance three degrees of freedom in the plane; you are going to hit obstacles. So when you have redundancy that helps you to move around obstacles and position the robot in different configurations. And we measure the degree of redundancy by the difference of n and m_0 . Okay? So essentially, we are not going to really discuss redundancy, which is very important, but we are going to focus on non-redundant robot first.

In fact, in the spring, we will cover extensively redundancy. We will talk about the use of redundancy to control the robot, make use of the mechanical advantages, the dynamic reduction that is introduced by redundancy, and also the motion planning, and collision avoidance using redundancy, and how we combine and control redundancy in a way that will allow us to achieve a task while redundancy is maintained to achieve different criteria, and different goals.

So redundancy we will not come back to it in Intro to Robotics, but it is a very important notion that you need to at least know about in terms of its definition. What we are going to do, we are going to go through the basics now, and we're going to start by building the models that will lead to the forward kinematics. I'm going to start with the simple definition of a vector that is defining a point in space, and we are going to go from there to building the models for representing an object in space which requires position and orientation, and then we connect to these objects.

But before that we will talk a little bit about representations of those different parameters I mentioned earlier, so we will talk about Euler parameters, we will talk about Euler angles and direction cosines, and probably this week we will cover all of that. We will cover transformations, how we move between frames and we will be ready next week to build the forward kinematics. So a point in space, a point in space be. How can we define a point in space? And what are the things that really fix that point or define it with respect to some reference?

So what is really important is to think about a point is the fact that the definition of a point, as every one of you probably thinks vectors, right? We can use vectors to define a point. But really is going to determine the vector is another point, a reference point. The origin that you are using to define the point. If you change the origin, you will change the vector. So we will talk about the description of a point with respect to another point to some origin, and this point is going to be represented by a vector P , and this vector P will describe this point with respect to this origin.

If we change origin we will change the vector. Okay? So if we have two points we define again the origin, and then we will have two different vectors representing this point, and these are those vectors built by taking the connection between the origin to the point. Very simple. Now I'm insisting in this because now we are going to introduce this origin, and then we will be able to describe the vector components.

So the vector is independent of this X , Y , and Z which frame we are taking. Now we are going to put a coordinate frame, and then we can express this vector in that frame. If I change the orientation of the frame, the vector is the same. The coordinates, the components, of the vector will change. And we are going to be interested in those transformations between different descriptions that involve different frames. We will work with orthonormal frames X , Y , and Z , and we will go from one frame to another.

So in this case X , Y , and Z , and we did a small rotation, and so about the same origin, and we ended up with a different frame. And we need to know how we can take a description of the coordinate of P in the frame X , Y , and Z , and then go to a description in frame X prime, Y prime, and Z prime. And this is going to happen using transformations. Now this transformation is just rotating the frame about the same point, but we might have a prismatic joint, and then we are going to translate the origin, so there will be a translation.

So we need to not only deal with rotation in the frames, but also translations of the frame. Well, a frame really is related to something beyond that point. The fact that if we're working with points, we just need to really worry about the whole orientation. What is really happening is you have different points on a rigid body, and when we rotate we are keeping the distance, but the orientation is changing. So a frame is really related to the description of the rigid body, so if we take this rigid body, take a fixed point and attach to it a frame.

I'm calling this frame B . So the coordinates are going to be described with respect to both axes X_B , so this is the frame B , we denote X_B , Y_B , and Z_B and the frame itself is B . And the question is how we describe the frame B with respect to a fixed frame, A . So as you know we need to find the relationship between the origins so there is an origin of frame A , an origin of frame B . We need this vector between the two, and that is defined by a vector. And this vector is going to locate the origin of frame B with respect to frame A , and we have its description in frame A , so this is BA .

The orientation of the frame is these vectors, X_B , Y_B , and Z_B . These vectors can have descriptions in different frames. Now we are describing them in frame A, so X_B in A, and this is the rotation. We denote X_B and here we put A to say these are the coordinates of X_B in frame A. So these vectors are going to describe the rotations of B with respect to frame A if it moves.

So essentially, if you think about those vectors and the relationship with frame A, we are really going to find the rotation matrix, which is the first model that we need really to use in order to describe the rotations of one rigid body with respect to another rigid body. So because we are concerned with just the rotations, I'm going to describe this frame and make $BA = 0$. So we will just move to the origin and just think about the rotation.

And in this case we will focus only on the rotation matrix and the rotation of that frame with respect to frame A. So we are now concerned with the rotation of frame B with respect to frame A. This rotation is described by a matrix, it's called the rotation matrix, and it has nine components. We are calling them $R_{1,1}$; $2,1,3,1$, these three columns. Okay. I know some of you know very well the rotation matrix. How many of you knows perfectly the rotation matrix? I remember. How many of you – okay.

All right, this is really important, so pay attention. Those of you seeing this for the first time, but I'm sure everyone has seen it some form or another. So what I'm going to do, I'm going to say state a description of what is the relationship between X_B , this vector X_B , and X_A . X_B defined in frame B is obtained by this rotation matrix, and the resulting vector is X_B in frame A. So the description of X_B in B is now transformed into a description of X_B in A using the rotation matrix. What is X_B in B?

Student:[Inaudible]

Instructor (Oussama Khatib): 1, 0, 0. The X vector in its own frame has one unit along the x direction, so it's 1, 0, 0. So what about Y? It is the same thing, 0, 1, 0. Right? How about Z, 0, 0, 1? These are the unit vector X, Y, and Z in their own frame. Right? And now using the rotation matrix we have the description, the component in the A frame. So the rotation matrix is basically just this. Right? I'm just using the definition, so if you multiply the first column is X_B in A, the second column. So what is the rotation matrix?

The rotation matrix between B and A is simply the component of X_B in A, component of Y in A and component of Z in A. Okay. So always remember this definition. This is very, very important because we are going to find the rotation matrix through many different ways, but sometimes you are looking at a problem, and you are looking at the frames, look at the component of that frame in the other frame, and you will make sense of your result.

And this is always the definition. The rotation matrix, essentially, the columns of the rotation matrix are the component of the axis X, Y, Z of the new frame in the reference frame. Okay? All right, so this is the definition of the rotation matrix from B to A, it is X_B in A, Y_B in A and Z_B in A. So, how do you obtain X_B in A? How do you obtain the

component of \mathbf{XB} in frame A ? You just dot product. So essentially if you do the dot product of \mathbf{XB} with \mathbf{X} , \mathbf{Y} , and \mathbf{Z} you will obtain \mathbf{XB} in A . Right? You agree? Good.

So this means the rotation matrix is essentially the dot product of \mathbf{XB} with \mathbf{X} , \mathbf{Y} , and \mathbf{Z} ; \mathbf{YB} with \mathbf{X} , \mathbf{Y} , and \mathbf{Z} ; \mathbf{ZB} with \mathbf{X} , \mathbf{Y} , and \mathbf{Z} . Right? Okay. Good. So, focus on this and look, look, look, look, look, at this row. Do you see anything like special? So here you have \mathbf{X} , \mathbf{X} , \mathbf{X} , \mathbf{XYZ} . Here you have – anything constant? \mathbf{XA} , \mathbf{XA} , \mathbf{XA} . So it's the dot product of \mathbf{XA} with \mathbf{XB} , \mathbf{YB} , and \mathbf{ZB} . Which is – do you see this? Which is \mathbf{XA} in B written as a row, which means it's transposed.

This is really interesting because if we start looking at these properties we see that this rotation matrix is either the component of B in A or the transpose of the rows of A in B . So B in A , A in B . Inverse relation. Going from B to A ; going from A to B . So going from A to B we are able to see that it's just the transpose, which means that B to A , the rotation from B to A is equal to the transpose of A to B . A to B is the inverse of B to A . So we have this property.

If I'm trying to compute the inverse of the rotational matrix B to A which means it's A to B , it is simply B to A transposed. So the transpose of this matrix is simply – I mean the inverse of this matrix is simply its transpose. And that's a very important property. And this property comes naturally because the rotation matrix formed with these unit vectors that are orthonormal, so the matrix is called orthonormal and this orthonormal matrix is always going to have this property. Its inverse is its transpose. All right?

Example: So can you compute the – I think you have it – don't look at your notes now for this – so could you give me the first column of the rotation matrix from B to A ? What is the first column going to be?

Student:[Inaudible]

Instructor (Oussama Khatib):So it is the component of \mathbf{XB} on A and \mathbf{XB} on A you see \mathbf{XB} and \mathbf{XA} are aligned. So anyone who can say what are the components?

Student:[Inaudible]

Instructor (Oussama Khatib):So 1, 0, 0. What about \mathbf{Y} ? \mathbf{Y} has a component only along the \mathbf{Z} axis, right? \mathbf{Y} is only along the \mathbf{Z} axis and zero above others. What about \mathbf{Z} ? \mathbf{Z} has a component along the minus \mathbf{Y} axis and its four component minus 1, 0, 0. Very simple example, but it illustrates what we have done. Essentially what we are doing, we are doing either dot product between the two vectors or looking at the component of that vector in frame A . At the same time, if you think about this, this is \mathbf{XB} in A , \mathbf{YB} in A and \mathbf{ZB} in A . This is our definition. And in the other direction, if you look it is \mathbf{XA} transposed in B .

So if you take \mathbf{XA} and express it in B , it's going to be 1,0,0 transposed. So these rows represent \mathbf{XA} in B , \mathbf{YA} in B , and \mathbf{ZA} in B . All right. So now we know the rotation

matrix, let's build finally this representation for a rigid body. So we know how to represent the frame, the rotation of the frame. We need the translation of the origin, and by combining X , the description of the rotation that is those vectors, and by locating the origin of the frame B with respect to A we are going to define fully the frame B with respect to frame A.

So the frame B is essentially defined by this rotation matrix B to A, and by the location of the origin with respect to A. Okay. So there is different ways of thinking about the use of this rotation matrices. And in fact, we can think about it as we have done so far which is to say we have a vector P, we have a frame A and a frame B, and essentially we are expressing the component of frame B in frame A and frame B, and we are looking at the relationship between the two. So this is what we call mapping. That is, we are changing the description of a vector from one frame to another frame, but the vector remains the same.

There is another way of thinking about it, which is – so this is the way we are looking at it. You have a Vector P and this Vector P is going to be described by it's dot product with A to give you its component in A, and this is the same thing, I'm just removing the vector outside and putting it as a matrix so it's A transposed, Y and Z transposed. You see this writing? It's the same writing, right? Here I'm doing dot product and just if you move P outside, it means that you are doing that multiplication with the different rows.

So now if you have this relation, you can say P can be expressing any frame. I mean this operation could be represented in any frame which is let's select B, the frame B or C or D, but you have to be consistent. You have to take the same description. And if you do that, essentially you are obtaining that mapping. That is, you are obtaining P in B rotated to give you P in A. So this is changing the description of the same vector P from one frame to another frame.

In the translation we are going to start by considering the one problem but while maintaining the same orientation of the frame. So I'm going to slide this frame and change the location of the origin of the frame. So frame B and frame A has the same orientation, and we're going just to move along the direction of the vector P. So if we have a vector – if we have a point in space, it is located with respect to origin B and described by the vector P. The same point in space is described with respect to another origin A with a different vector PA.

So it is the same point, but described with respect to two different points – two different origins attached to two different frames, and you end up with two different vectors. So this was our initial description in frame B and now we have a new description and this transformation is resulting in two different vectors. You have to realize contrary to the case when we were doing just rotations, now when we do our translation we are going to change the description by changing the vectors involved in the description.

So this operation that involves a translation of a vector P defining this origin of the frame B, it's changing the description from origin B to origin A. And the relationship between

the two, that is the relationship between the green vector and the red vector, is essentially this relation that is giving us the vector with respect to A – origin A, as the sum of the vector with respect to origin B and the translation of the origin. So this fact that we have now two different vectors is going to appear later in the homogenous transformation, and make the transformation a little bit different from rotation matrices because we are introducing a translation, and we are introducing this non-homogenous relation in the model.

So when we come to general transformation, so now I'm saying I'm going to have a description of the same point P, but with respect to a [inaudible] frame B that is rotated with respect to A. Then we need to account for this rotation, and that means that in the description here it's not simply the sum, but I have to do the sum with respect to descriptions in the same frame. That is, I cannot add this vector directly with this vector. I need to rotate this vector to frame A.

And that means we have this general relation, that is we take the description of B, we rotate it to frame A, we have its description of A and we have the origin description of A, and now we can add them together and the result is this vector. We haven't changed anything. We still are talking about this vector plus this vector equals this vector. But what we have to make sure is that this description is rotated correctly to frame A. All right.

Well this is the general transform, and in fact, using this – applying this between links we should be able to compute and propagate, go from this link to the next to the next to the next. But this description is not simple to carry when you have multiple links because you don't – it's not like a rotation matrix where with our rotation matrix if you know the first rotation with the frame with respect between two frames, and you know that the two are rotated with respect to a different frame, all that you need to do is to multiply the rotation matrices.

In here, you have to carry sums and you have to carry those relations. So a better way to handle this transformation is to try to put it homogenous form. How can you do that? This is the sum of two vectors in three dimensional space. You cannot have it in homogenous form. But if you go to four dimensional space, then you can put it in homogenous form. Do you want to see how? Hmm? Yeah. Okay. So I'm just relating the same thing and I'm going to just add one more row for nothing. So this row is saying 1 is equal to 1.

If you multiply this vector by this matrix, you obtain the first relation. Right? And the second part is $1 = 0$ multiplied by the vector $P+1$. So $1 = 1$. But now we have captured the homogenous property that is this vector is transformed into this vector, this description is transformed into this description using the rotation matrix and the translation. Do you see that? And this is what we call the homogenous transformation. It's a 4×4 matrix. You have four components that are doing nothing except help the math to make this transformation homogenous in that when we go from A to B to C to D, then

essentially we are going just to multiply matrices, but we have to handle 4 x 4 matrices instead of 3 x 3 matrices. Okay.

So this is a very important component. Now by the way, sometimes 0, 0, 0, 1, the definition is on the top so sometimes you define 1P instead of – but it is exactly the same computation. So here is an example of the homogenous transformation. Now we have two vectors. I'm taking the frame, you remember the example we saw earlier? The example where we had a rotation matrix with rotating about the X axis. So I'm just translating now the origin of B with this vector 0, 3, 1. So the homogenous transformation so it is the same rotation matrix as before and 0, 3, 1 is the vector describing the origin of frame B in frame A, and your homogenous transformation is here.

And now using this homogenous transformation you can compute the new position of the point B so the point B in frame B is described by 0, 1, 1. This is the point I'm looking at. And to find this vector, all that you need is to take this transformation and multiply this vector by this transformation that you have to add 1. So we take the 0, 1, 1, we add 1 and the multiplication leads to 0, 2, 2, and 1. You drop the 1. The answer is 0, 2, 2. So if you look over there, this was happening in this plane so essentially here we have 1 and this is 2 and 1 and 2. So it's just 2, 2. Okay? Clear? Okay.

Now that you understood everything, I'm going to confuse you. I mean once you understood nothing now we'll have to completely change the intuition. Now instead of mapping we are going to – so a rotation matrix I said allows you to describe the same vector in two different frames. But now I'm going to take the rotation matrix and use it to rotate a vector. So the vector was here; I'm going to rotate it by the rotation matrix. So I said the mapping is changing descriptions. Okay? An operator is moving those points in space. Well that could be useful.

So you have a vector, you have one frame, you have a description. Another frame, you have a description. Now what I'm going to do, I'm going to do a rotation and this rotation will rotate the vector. So in fact a rotation matrix which could describe the relationship between the component of the same vector in two different frames in this way, can be also used as an operator that would operate on the vector P1 to produce a vector P2. And this is really useful later when we compute representations you will find very useful to compute transformation between different frames.

So R here is operating on P1 to produce P2. Okay? So P2 is the rotation applied to the vector P1. And of special interest are those rotations that take place about some specific axis like the X axis or the Y axis or the Z axis with some rotation. So then you can talk about RX with some angle theta, and that will be very useful in some of the operations. So in general we can talk about a rotation about a K vector, not the X,Y,Z, but any arbitrary vector. And it rotates a vector P1 into vector P2.

So here is an example. This is a rotation about the X axis so the X axis is 1, 0, 0 and the rotation about the X axis is a theta, it is cosined theta minus sign, you know this familiar

matrix rotation above the X axis. So if you take P1 which has component on Y and Z so it's 0, 2, 1. If you take this matrix, you will end up with 0, 1, 2 which is over there 1, 2. You see P1 and P2? So this is the vector 0, 2, 1, 2 and 1, and now 0, 1, 2, is 0, 1, and 2. Translations. The same thing.

We can now instead of describing a point with respect to two different frames we are going to translate that point using an operator. So in the mapping we took the vector with respect to B and produced a vector with respect to A in the operator. In the operator this is what is happening. We are changing the point. We are going from P2 to P1 so this point P1 moved so we have two different points and two different vectors. So when you are translating you are thinking about it like in the rotation.

In the rotation we had the vector, we have P1 we rotated to P2 now we have here a vector P1 with this translation is essentially is producing this point.

So I have to remove this, you can see it better. So we are moving P1 to P2 with the translation Q. So P2 is $P1 + Q$. So this is an operator of translation and you have this operator of Q along the X axis, Y axis, Z axis or any arbitrary vector. So a translation through the operator Q would result into a different vector P2 and now you have to describe it.

You can describe it in frame A or any other frame, but through always maintaining the same description for all the vectors when you apply the components for describing the frame, you have to make sure that q is described in the same frame as P1 and the result will be in the same frame. So now we can think about this operator as this operator 1, 1, 1 that there is no rotation at all, but there is only translation QX, QY, QZ, and this is defined by the Q vector. So the translation is done by QX, QY, and QZ. And now you can combine the two.

You can combine the rotation and translation so you could have an operator operating by translating and rotating. So the general operator that you can imagine is this operator rotating about some vector k with some angle theta and translating Q. So the point B which was in P1 now is rotated and translated, and the result is P2. And this is without any definition of the frame then you define the frame in which you want to express all these vectors, but you have to make sure that you are using the same frame for all the operations.

So this is the most general form of transformation to a different a point. It is the same homogenous transformation. It is the interpretation of that transformation whether it is changing the description or operating and changing the points. Okay. Are you enough confused? Good. So two things, we saw the transformation – it is the same homogenous transformation that has two components rotation, translation. Rotation matrix. Translation of the origin of those frames.

And that can give us mapping changing the description, and I just discussed the other interpretation that you could have which is to change the vector itself, and that means

change the points describing those rotations. So when you rotate a vector you are going from one point to another point or when you are translating you are going from one point to another and you can apply both of them.

And now we're going to look at the inverse. Now in the case of rotations, the inverse is very simple. What is the inverse of R ? Rotation matrix from B to A ? Transpose. Now for the homogenous transformation this is not the case. We cannot just say it is transposed that is this matrix is not orthonormal because the presence of this translation.

So the inverse is not exactly the transpose of this matrix, but it is almost because if you look at the inverse of this matrix it involves the transpose – because if you take it by block you will see that you have the inverse here, this is the same, the only thing you have here is the description of the inverse of this vector.

What is this actually? I mean if we were thinking about this inverse it is going from A to B . So this is the origin of A in frame B . So basically you are just writing here the origin of frame A in B . That's it. Okay? So now we now the forward, we know the inverse of the homogenous transformation, and it is going to be essential.

The homogenous transformation we're going to use it to describe the kinematic chain of the manipulator, and you will see that this transformation could be described by those four parameters, the DH parameter so we will be able to describe each relation between successive links just using four parameters, the DH parameters.

So I will not reveal the movie segment for next time, but unless you have any questions we can stop here. Any questions? No. Okay. We'll stop here and we'll see you on Wednesday.

[End of Audio]

Duration: 68 minutes