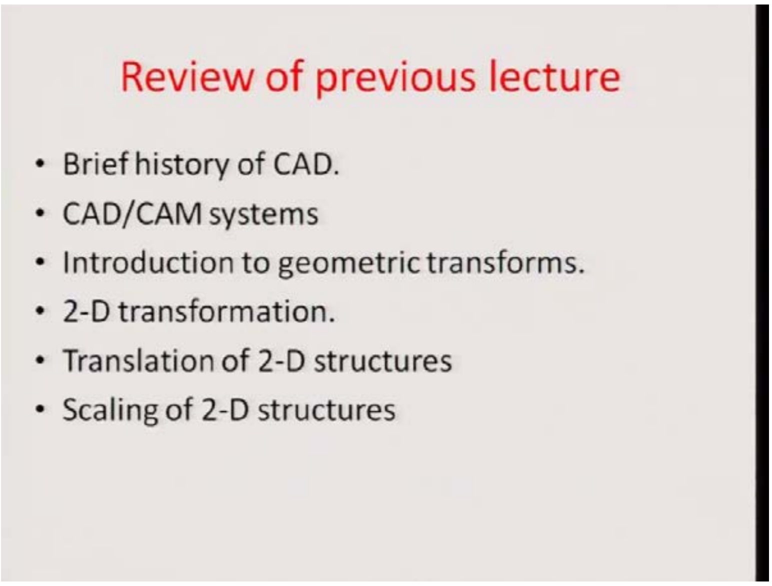
Manufacturing Systems Technology
Prof. Shantanu Bhattacharya
Department of Mechanical Engineering and
Department of Industrial and Production Engineering
Indian Institute of Technology, Kanpur

Module - 01
Lecture - 05

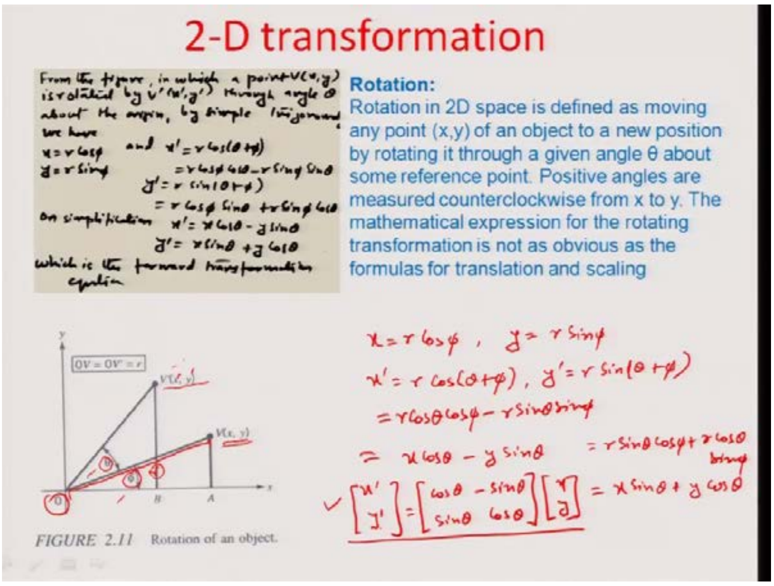
Hello and welcome to this module five on manufacturing systems technology.

(Refer Slide Time: 00:20)



A quick recap of what we did in the last section; we talked about brief history of computer-aided designing, where we started with illustrating what you called the sketch pad coming as a result of Ivan Sutherland thesis. We also discussed about the various CAD-CAM systems; what kind of hardware support would be needed for these CAD-CAM systems. And then finally, started on to geometric transformations, where we described how a two-dimensional object like a triangle can be translated in space using some kind of a transformation matrix addition process. And, we also validated that, while doing this translation, the lengths of the different points, which are interconnected; in our particular case, it was a triangle interconnected – three points were interconnected to each other. So, the lengths after the translation do not vary; and, which shows that, the triangle as an entity gets translated. We also did a similar exercise for a pentagonal shape for which was amplified or magnified along two axes by a certain magnification factor. And, that we called as a scaling.

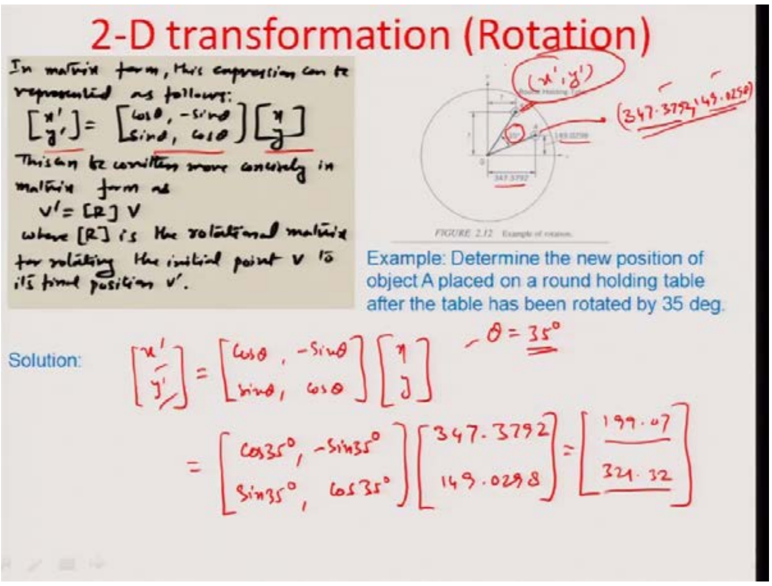
(Refer Slide Time: 01:43)



So, we will look now at a different aspect of the transformation which is rotation. So, let us say in this particular case, we have a vector, which we call OV. You can see this is the vector of radius length r and it is placed at an angle phi from the positive x-axis. And, further we know the location coordinates of the point V as x, y, which is one end of this radius vector. And, supposing we want to rotate this vector by an angle theta in the anti-clockwise direction, so that the point V now becomes or assumes the status of V dash; and, we call the coordinate locations of this x dash y dash. So, we need to find out what is the geometric relationship between the radius r; the different angles which have been translated; and, these different location coordinates, which have been resulting from the angles which are translated. So, we get that.

If we look at – let us say the value of x can be simply interpreted as r cos of phi; and, that of y can be interpreted as r sin of phi. Similarly, when the angle changes from phi to phi plus theta; obviously, x dash can be represented as r cos of phi plus theta; and, y dash can be represented as r sin of phi plus theta. So, we can find out that, if we just expand these expressions, we are left with r cos theta cos phi minus r sin theta sin phi. And similarly, the y dash can be r sin theta cos phi plus r cos theta sin phi. And, that is how you express or write these two equations. So, obviously, r cos phi is earlier expressed as xy can write it as x cos theta minus r sin phi was y; so, y sin theta. And similarly, here we can represent it as x sin theta plus y cos theta. So, in a matrix manner, we can always say that, x dash y dash – the new matrices are formulated out of a matrix. So, the new vectors x dash y dash or x prime y prime can be represented as a matrix now; cos theta minus sin theta and sin theta plus cos theta times of xy. So, that is how you can actually represent rotation. And, mind you – this rotation is about the point O along the z-axis; and, that is, you can translate the points x dash y dash to xy through this set of equations.

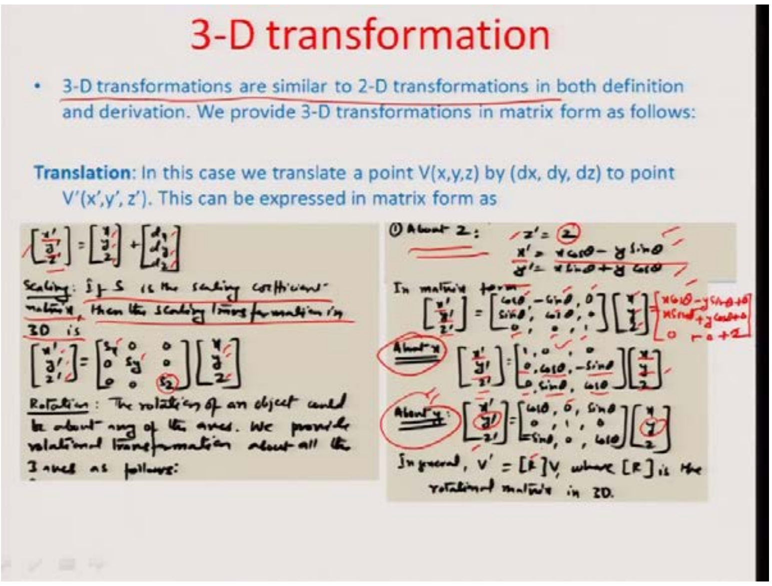
(Refer Slide Time: 05:04)



So, let us now apply this to a practical problem, which is represented here. So, you can see that, in this particular problem, there is a radius OA. And, the coordinates – location coordinates of this A are mentioned here through the lengths, which are indicated on both the x and the y-axis. This can be represented as 347.3792 comma 149.0298. So, that is how you are representing the x and y of OA. And, the final coordinates have to be found out of A dash provided you have rotated the A to A dash through an angle of 35 degrees anti-clockwise. And, we need to now calculate what are the x prime, y prime using the same formulation as done earlier provided these coordinates have been given. So, we know that, the way that we can change or transform xy to xy – x dash y dash actually through the relationship done earlier.

So, cos theta minus sin theta, sin theta plus cos theta times of xy; theta in our case is about 35 degrees and you know that all anti-clockwise rotations are treated as positive rotations whenever we talk about trigonometrically how angle varies and see that the rotation actually through the right-hand group rule lets the z-axis to come out in the positive z-direction actually in this particular case. So, theta is 35 degrees. And, xy coordinates are already well-defined here as 347.3792 and 149.0298. So, we just simply write the transformation equation as cos of 35 degrees minus sin 35 degrees, sin of 35 degrees, cos of 35 degrees times of the position vectors of xy – 347.3792 and 149.0298. So, from calculations… From calculations, we can get this to be equal to 199.07, 321.32 respectively. So, these are the new positions – x dash y dash of the point A, which has been rotated in the anti-clockwise direction by 35 degrees angle. So, you can actually find out in this manner, if the rotation varies along the x-axis and the y-axis, what is going to be the variation in the way that the vectors would be changing positions or locations based on that.

(Refer Slide Time: 08:11)



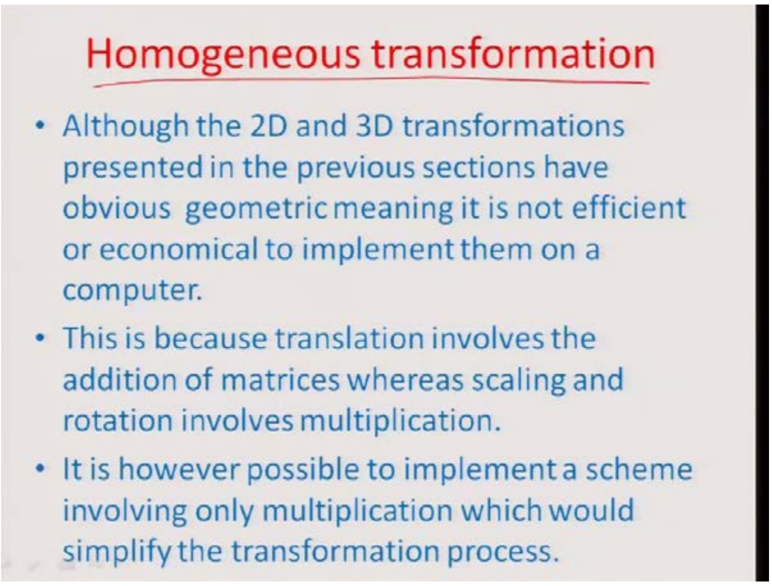
So, I can say now that from the transformation matrix, if we convert from 2-D to 3-D case; so far, we have been seeing on the x and y. The only change that we need to do is in terms of one more coordinate, that is, the z-coordinate addition. And so, therefore, the transformation matrices in 3-D would really work out to be… In case of translation, it is x dash, y dash and z dash related to the xyz and related to the distance matrix. In this case, you have distances all along the x and y and z directions. So, you have dx, dy, dz as those matrices. If scaling has… If the scaling coefficient matrix and then the scaling transformation is converted into 3-D, the only other addition which would result is in the z-coordinate. So, you have sz, which is the magnification factor or the scaling factor along the z-axis, which gets added to this particular 3-D transformation – to create this 3-D transformation. And then finally, for the rotation, you obviously would have an additional coordinate about which the rotation is taking place. So, if the rotation were about z-axis, which was the case, which we discussed earlier; then, if we look at what is z dash, that does not really get changed, because it is the axis of rotation; it is the center of rotation. So, z dash remains as is as z. And then, x dash you found out from your earlier transformation can be represented as x cos theta minus y sin theta. And similarly, y dash can be represented as x sin theta plus y cos theta.

So, if I were to just write all this in matrix form, you can convert this through the transformation matrix x dash, y dash, z dash equal to this rotation matrix, which is cos theta minus sin theta 0, sin theta cos theta 0; 0, 0, 1. And, because the z does not change, so the x, y, z would be the multiplication matrix through which it is converted. So, if you just look at how the final product or the final output would be in this case; so, the first matrix can be or the first element of this matrix x dash can be equal to x cos of theta minus y sin of theta plus 0 z, which is immaterial. The second – y dash would be equal to y or x sin of theta – x sin of theta plus y cos of theta plus 0. And, the third would be equal to 0 plus 0 plus z. So, that is how the transformations x dash, y dash, z dash would happen if you look at all these three equations along the z-axis.

There also similarly, exists rotation about x and about y. And, rotation about x can be represented by changing the rotation matrix little differently, so that the x does not change. So, x dash and x are same here. And, obviously, y and y… y dash and z dash are changing because you are adding… you are having a rotational effect on the yz plane in this particular case. And so, you have the matrices as 1, 0, 0, 0 cos theta minus sin theta and 0 sin theta cos theta on one hand, and the multiplier matrix as x, y, z on the other hand. And similarly, about y, you can have the rotation matrix change into cos theta 0, sin theta 0, 1, 0; sin theta 0, cos theta times of the multiplier matrix x, y, z. In this case, you can see that, y dash is equal to y, because the rotation is about the y; and, x dash and z dash are changing because of the rotation; or, they are enhancing the experience in the rotational front because the rotation is about the plane xz in general. So, that is how you construct the rotation vectors or rotation matrices. When we talk about the rotation along the various axes of the three-dimensional space.

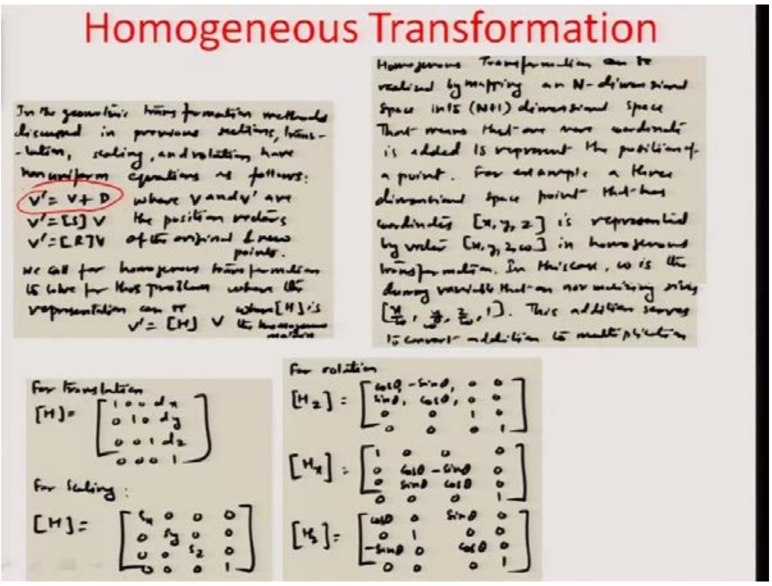
So, we now… two-dimensional as well as the three-dimensional aspect of rotation. And, we also looked into what are the backend computations, which are happening in terms of change of the coordinates, so that if you translate or scale or rotate an object, you can have new coordinate system generated because of that. I would like to point to something very important here is that, so far, if you look at how scaling was being performed or how rotation was being performed, it is really through matrix multiplication that things were being done. But, if you look at only the translation part, it is through matrix addition that we are doing. So, computationally, it becomes very exhaustive if you have both addition as well as matrix multiplication being done by the same processor. And therefore, a goal of a good programmer really is to sort of unify everything in terms of matrices multiplication, which is a much easier and much more flexible process, which can represent all the three translation scaling and rotation transformations into one particular rule or methodology. And, for that, we have to perform something, where the translation effect, which is in otherwise – which is otherwise seen as an addition of matrices can be converted into a matrix multiplier effect. And, for this, we use a process called homogenous transformation.

(Refer Slide Time: 13:42)



And obviously, the homogenous transformation is important because it is economical to implement this kind of a transformation in the computer. So, everything would be in terms of matrix multiplication rather than matrix additions.

(Refer Slide Time: 14:00)



And, how do we do it is a little tricky issue. So, if I were to look at the translation – transformation, which talks about V dash equal to V plus D; where, D is the distance matrix. I can probably generate a methodology. So, we can do this homogenous transformation by mapping a n-dimensional space into n plus 1 dimensional space; which means that, you add one more coordinate basically. And so, basically I am going to take this homogenous transformation in the next lecture in the interest of time. And, I close on this module here.

Thank you.