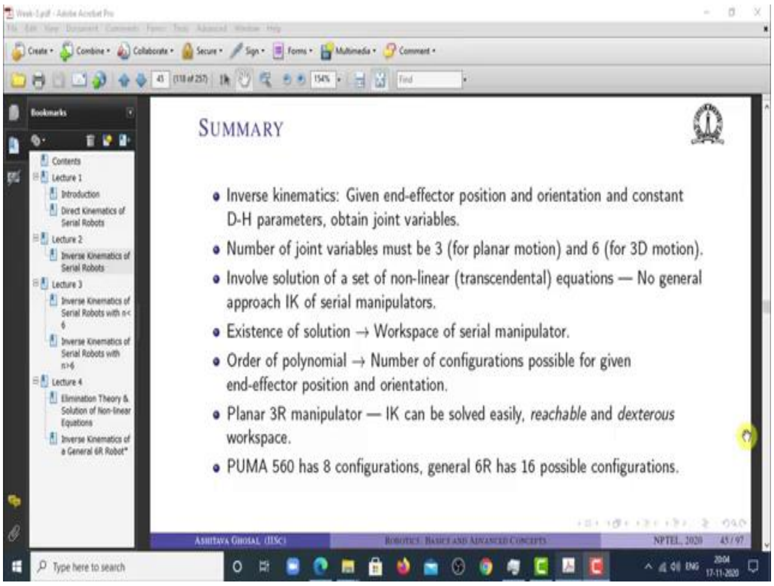
Robotics: Basics and Selected Advanced Concepts
Prof. Ashitava Ghosal
Department of Mechanical Engineering
Indian Institute of Science, Bengaluru

Lecture – 03
Inverse Kinematics of Serial Robots n 6

(Refer Slide Time: 00:20)



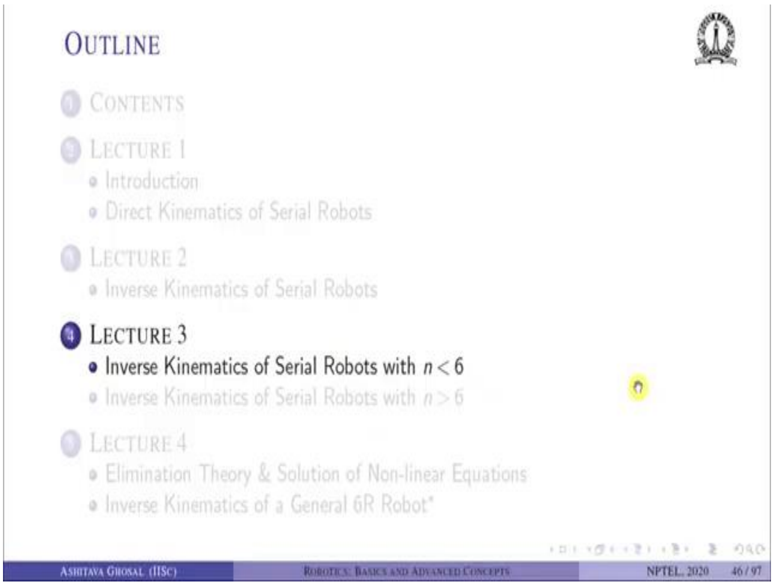
Welcome to this NPTEL lectures on Robotics Basics and Advanced Concepts. In the last lecture, we had looked at Inverse Kinematics of Serial Robots. Just to recapitulate, the inverse kinematic problem is stated as follows; given the end effector position and orientation and the constant DH parameters, obtain the joint variables. We had looked at the cases when the number of joint variables were 3 for planar motion and 6 for 3D motion.

I showed you that the inverse kinematics problem involve solution of a set of non-linear transcendental equations and then there are no general approaches to solve the inverse kinematics of arbitrary serial manipulators. Once we solve the inverse kinematics problem, we had this notion of existence of a solution, which in turn led to the very important concept of workspace of a serial manipulator.

One of the way to solve the inverse kinematics problem was to obtain a single polynomial, a monomial of in one joint angle. The order of the polynomial gave the number of configurations possible for a given end effector position and orientation. For the planar 3 R manipulator, I showed you that the inverse kinematics could be solved very easily using simple trigonometric identities and tricks and we obtained this notion of a reachable and dexterous workspace.

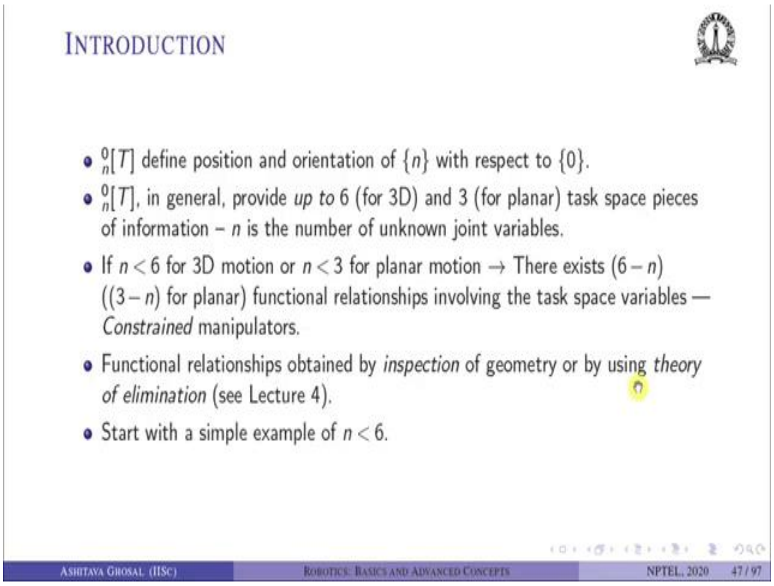
The PUMA 560 which was a 3D spatial 6 degree of freedom manipulator had 8 possible configurations. And I have just briefly mentioned, which we will see later that the general 6 degree of freedom robot with rotary joints has 16 possible configurations.

(Refer Slide Time: 02:21)



So, in this lecture, we will look at the inverse kinematics of serial robots, when the number of joints is less than 6 when it is moving in 3D space or less than 3 when it is moving in a plane.

(Refer Slide Time: 02:37)



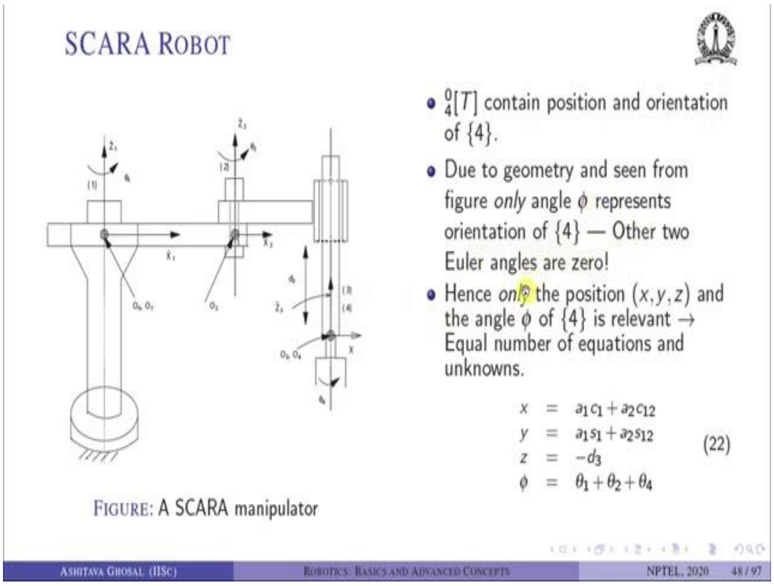
So, $^0_n[T]$ which is the link transformation matrix for the last link end effector, defines the position and orientation of the link n with respect to $\{0\}$ coordinate system, with respect to the fixed base. The $^0_n[T]$ in general, provide up to 6 for 3D and 3 for planar task space pieces of information, ok. So, what do we mean by 6 or 3 task space pieces of information? We have 6 independent equation sorry, 6 independent parameters when it is moving in 3D space.

So, in 3D space, we have x, y, z and 3 from orientation. Similarly, for planar we have x and y and the orientation of the last length. And n in this discussion is the number of unknown joint variables. So, if n is less than 6 for 3D motion or n is less than 3 for planar motion, there must be 6 minus n or 3 minus n for planar functional relationships involving the task space variables. So, basically these are constrained manipulators.

So, you can think of it that, I have a robot which is moving in 3D space; but there are only 4 joints, n is 4. So, out of those x, y, z and 3 pieces of orientation, which is given for the end effector; I cannot have 6 independent equations, ok. So, 2 of those parameters must be related to the other 4 somehow, ok. So, we are looking for functional relationships obtained by inspection or of geometry or by using theory of elimination, ok.

So, most of the time a robot designer would make a robot with let us say 4 joints for a particular task, ok. So, we know why the 2 degrees of freedom in the task space, how they are related ok; why they have been removed for say for some reason.

(Refer Slide Time: 05:00)



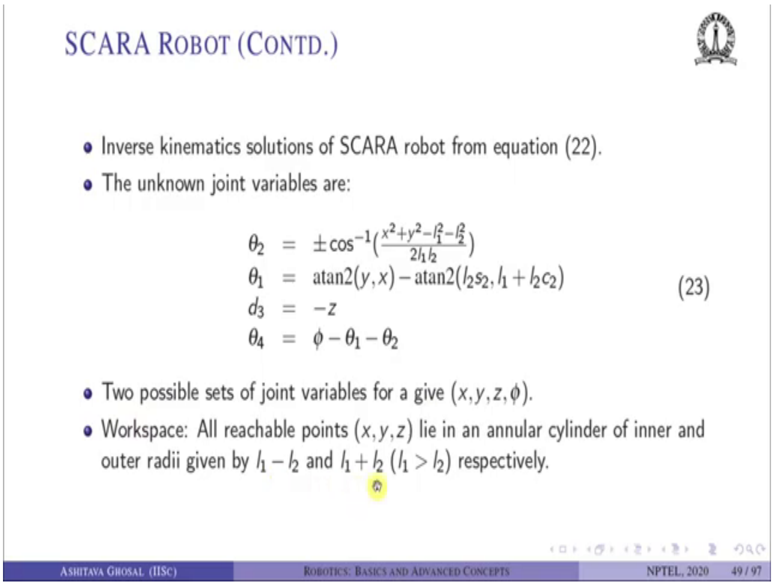
So, let us look at the simple example of n less than 6 and we have looked at this before; this is the SCARA robot on the left, it is a 4 degree of freedom robot. So, there is a joint which is along Z_1, there is another rotary joint along Z_2, there is a translatory joint which is along Z_3 and then there is a rotation about θ_4 about the last Z_3 again.

So, this $^0_4[T]$ which is the transformation matrix of the 4th link with respect to the 0th link contains the position and orientation of the link 4, ok. So, due to geometry and seen from the figure, only the last angle ϕ represent orientation of 4 ok.

So the other two Euler angles are zero. So, we can only rotate about the Z axis; the end effector cannot rotate about the X and Y axis, that is the way the robot designer made this robot. So, hence only the position x, y and z and the angle ϕ rotation about the z axis of the link 4 is relevant, ok.

The other two angles are not relevant, they are constrained, in fact they are zero. So, we now have an equal number of equations and unknowns. So, we have x as $a_1c_1 + a_2c_{12}$; y as $a_1s_1 + a_2s_{12}$; z as minus $-d_3$; and ϕ as θ_1 θ_2 θ_4. So, $\theta_1$is the rotation hereθ_2 is the rotation at the second rotary joint, and then θ_4 is the rotation as the last rotary joint. So, although its n is less than 6 here; but we have been able to derive 4 equations in 4 unknowns. So, its a consistent set of equations.

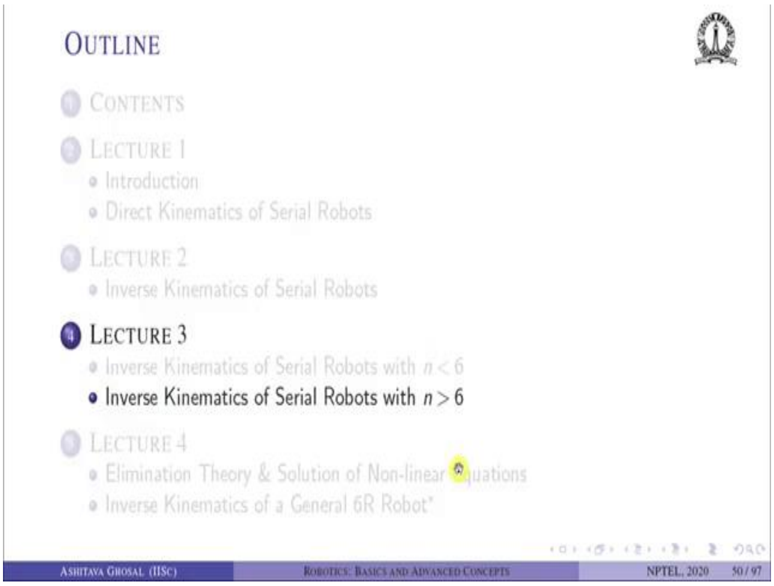
(Refer Slide Time: 06:51)



And this we can easily solve and obtain the inverse kinematics solution of this SCARA robot. So, very straightforward the first θ_1 and θ_2 are very similar to the 2 R planar robot. So, $\theta_2 = \pm\cos^{-1}\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}$. θ_1 is atan2 (y, x)- atan2 some function of θ_2; d_3 is z, directly we can see from this equation d_3 is z.

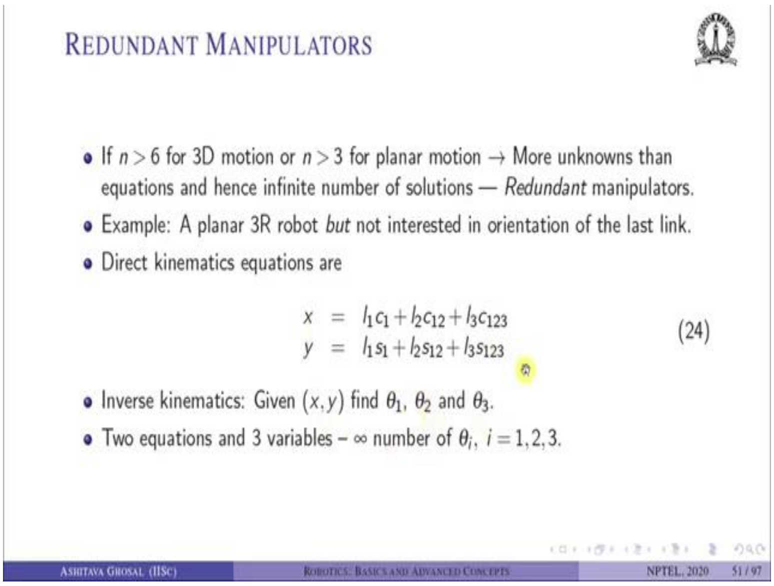
And once θ_1 and θ_2 are solved from these first two equation x and y; we can find out θ_4, which is $\phi - \theta_1 - \theta_2$. So, in this case of the SCARA robot, there are two possible sets of joint variables for a given x, y, z and ϕ. And what is the workspace? Again it is intuitively clear, this is basically nothing, but an annular cylinder of inner and outer radii given by $l_1 - l_2$ and $l_1 + l_2$; we are assuming $l_1 > l_2$.

(Refer Slide Time: 08:06)



So, the case of n less than 6, basically we will assume that there are inherent constraints in the task space variables. In the space of SCARA, there are two Euler angles other than the one rotation about z axis at zero. We will next look at inverse kinematics of serial robot with n greater than 6, ok.

(Refer Slide Time: 08:30)

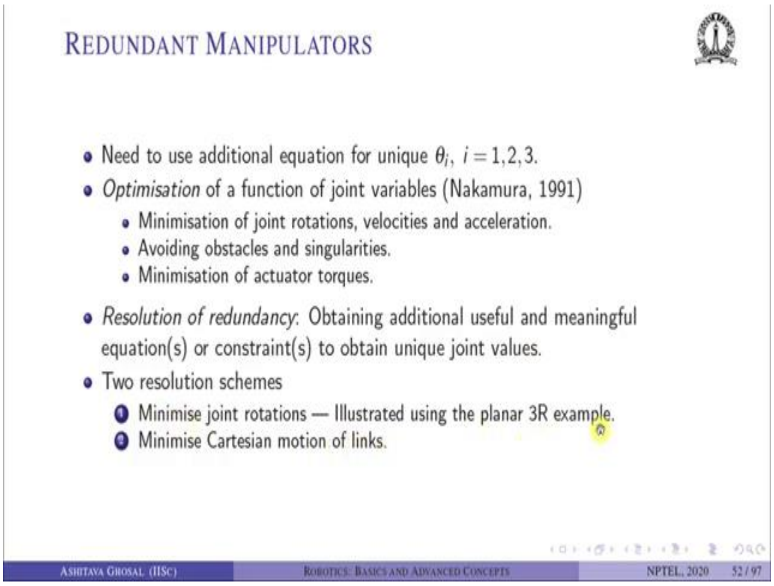


So, if n is greater than 6 for 3D motion or greater than 3 for planar motion, there are more unknowns, ok. So, the number of joints are more than the number of equations, and hence an infinite number of solutions are possible. So, these are called redundant manipulators.

So, for example, a simplest case is a planar 3R robot, but we are not interested in the orientation of the last link. So, the direct kinematics equations are nothing, but $x = l_1 c_1 + l_2 c_{12} + l_3 c_{312}$, and $y = l_1 s_1 + l_2 s_{12} + l_3 s_{312}$. So, again s_1 means $\sin\theta_1$; s 1 2 3 means sin of θ_1 + θ_2 +θ_3.

So, what is the inverse kinematics problem? We are given the left hand side x and y and we have to find θ_1, θ_2, θ_3. So, there are two equations in 3 unknowns, 3 variables. So, clearly there are infinite number of θ_i, 1, 2 and 3, which can satisfy these two equations given any x and y.

(Refer Slide Time: 09:44)



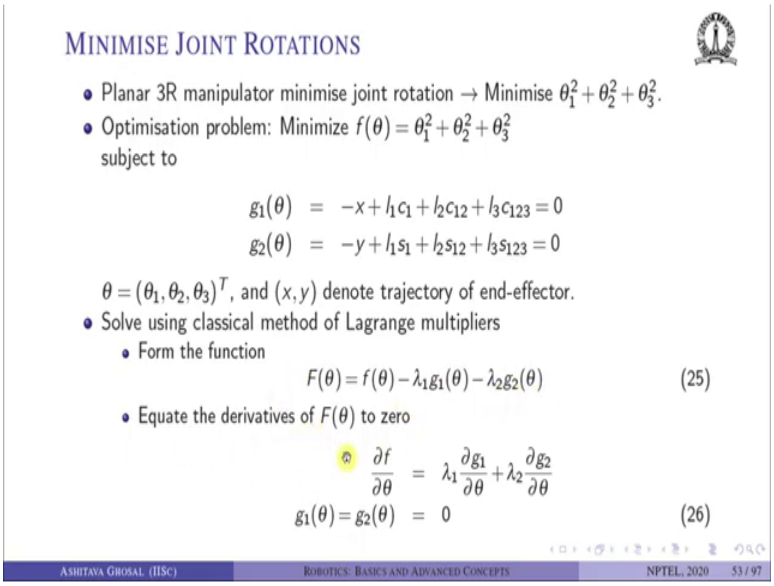
So, if you want to solve inverse kinematics of a robot with similar problems or similar situation; we need to use additional equation for unique θ_i, ok. So, and this is not a cooked up problem; there are robots which have been constructed, where the number of joints. So, number of theta's are more than the number of task based variables, ok. So, what do we do? So, one natural thing is to do some optimization, ok. So, we can find the function of the joint variables and we can use optimization.

So, one obvious thing is we want to minimize the joint rotations; we can also minimize joint velocities and accelerations. Researchers have also suggested that we can use this extra degree of freedom or extra joint to avoid obstacle and singularities, ok. We can also use this extra degree of freedom or extra joint to minimize the torques ok, actuator torques in some least square sense.

So, this notion of obtaining additional useful and meaningful solution or constraints to obtain unique joint values is also called as resolution of redundancy. So, what is the additional equation? What does it mean? Why do we use that additional equation to obtain what?

That is called as the problem of resolution of redundancy. So, I am going to show you two resolution schemes; one is minimize joint rotation and I will illustrate this by using the planar 3R example and the second is minimize Cartesian motion of the links, ok. So, the first one is minimizing joint rotation, second is minimizing Cartesian motion of the links.

(Refer Slide Time: 11:37)



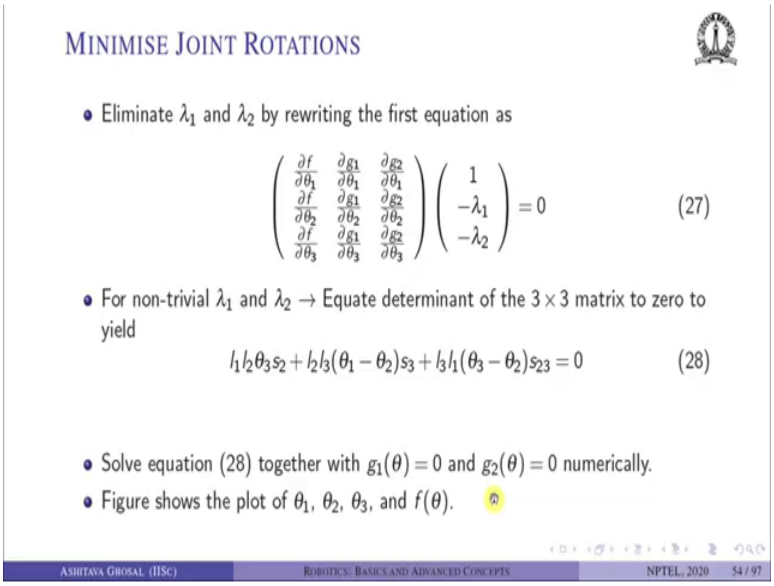
So, let us start with minimizing joint rotations. So, for the planar 3R robot ok, minimizing joint rotation means, something like minimize $\theta_1^2 + \theta_2^2 + \theta_3^2$, ok. So, this is like the l 2 norm of the joint variables θ_1, θ_2, θ_3, ok. So, what is the optimization problem? Minimize $F(\theta)$ which is square of the joint angles; subject to constraints, we still need to make sure that it satisfies the given direct kinematics equation.

So, we are still given x and y, which is related to θ_1, θ_2, θ_3 in this form and we need to make sure that these two constraints $g_1(\theta)$ and $g_2(\theta)$ which is given in these equations ok, they are basically the direct kinematics equation are satisfied.

So, we have θ which is θ_1, θ_2, θ_3. it is a column vector and x and y which denotes the end effector trajectory. So, we can solve this optimization problem and it turns out that, we can solve this optimization problem using classical method of Lagrange multipliers.

So, what is the classical method of Lagrange multipliers; we form another function $F(\theta)$, which is small $f(\theta)$, which is this objective function minus $\lambda_1 g_1(\theta) - \lambda_2 g_2(\theta)$. And the solution procedure is well known; we take the derivative of this $F(\theta)$ to 0 by setting $\frac{\partial f}{\partial \theta}$= equal to this $g_1(\theta)$ and $g_2(\theta)$, equals to 0 .

(Refer Slide Time: 13:28)



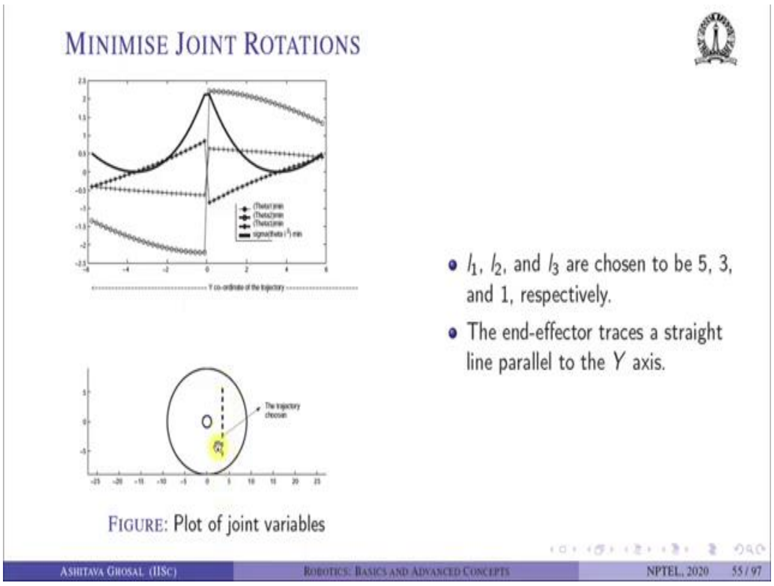
So, we can eliminate this λ_1, λ_2 which are called also as the Lagrange multipliers by writing these three equations in this form. So, $\frac{\partial f}{\partial \theta}$, $\frac{\partial g_1}{\partial \theta_1}$, $\frac{\partial g_2}{\partial \theta_2}$ into 1, - λ_1,$-\lambda_2$,.

And similarly the second row is $\frac{\partial f}{\partial \theta_2}$and so on; third row is $\frac{\partial f}{\partial \theta_3}$ and $\frac{\partial g_1}{\partial \theta_3}$, and so on equal to 0, ok. So, this is a equation, linear set of equation of the form AX equal to 0. And for non trivial λ_1, and λ_2, the determinant of this matrix 3 by 3 matrix must be zero ok, this is from linear algebra.

So, we can obtain the determinant of this matrix and it turns out to be an expression of this form it $l_1 l_2 \theta_3 s_2 + l_2 l_3 (\theta_1 - \theta_2) s_3 + l_3 l_1 (\theta_3 .- \theta_2) s_{23} = 0$, ok. So, we need to solve this equation together with $g_1(\theta)$equal to 0 and $g_2(\theta)$. equal to 0, ok.

So, this cannot be solved analytically, but we can always do it numerically. So, the next slide shows a plot of θ_1, θ_2, θ_3. and $f(\theta)$, the minimization the objective function, which we are trying to minimize as for a given x and y, ok.

(Refer Slide Time: 15:10)



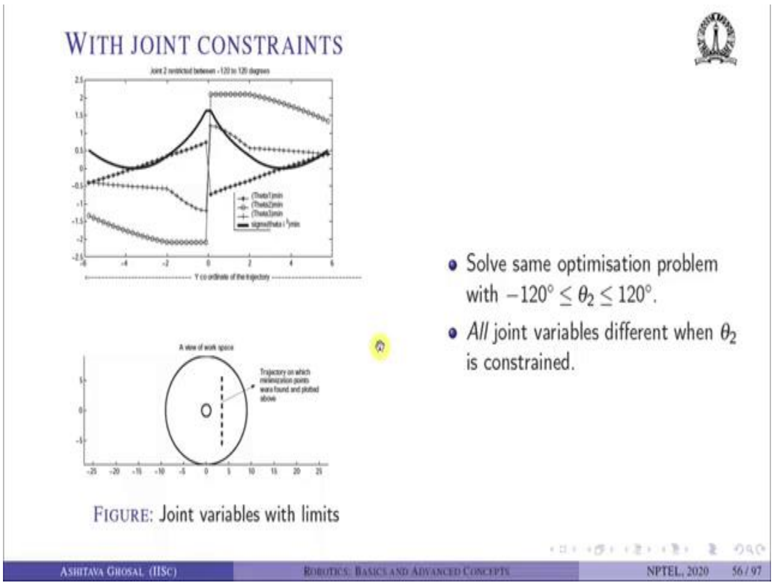
So, we will start with this bottom figure. So, the bottom figure shows a plot of the workspace of this 3R robot. So, basically there is a outer circle which is $l_1 + l_2 + l_3$ and a inner circle which $l_1 - l_2 - l_3$. And we want to trace a trajectory along the Y axis. So, this dotted line shows a chosen trajectory; this has been chosen arbitrarily, we could have chosen any other trajectory ok, which is in the workspace, of course the whole trajectory must be in the workspace.

For numerical purposes, we have chosen l_1, l_2, l_3 as 5, 3 and 1 arbitrarily; we could have chosen any others. And this end effector trajectory is along this Y axis. So, we solve this optimization problem and in this top plot, it shows the variation of θ_1, θ_2, θ_3 and the objective function, ok.

So, what you can see is, we will get some values of θ_1. ok; then we will get some values of θ_2, and we will get some values of θ_3. So, this plus signs are θ_3 , this one is θ_1 minimum, and this one is rather $\theta_2$minimum, this circles and this dark solid line is the value of the objective function.

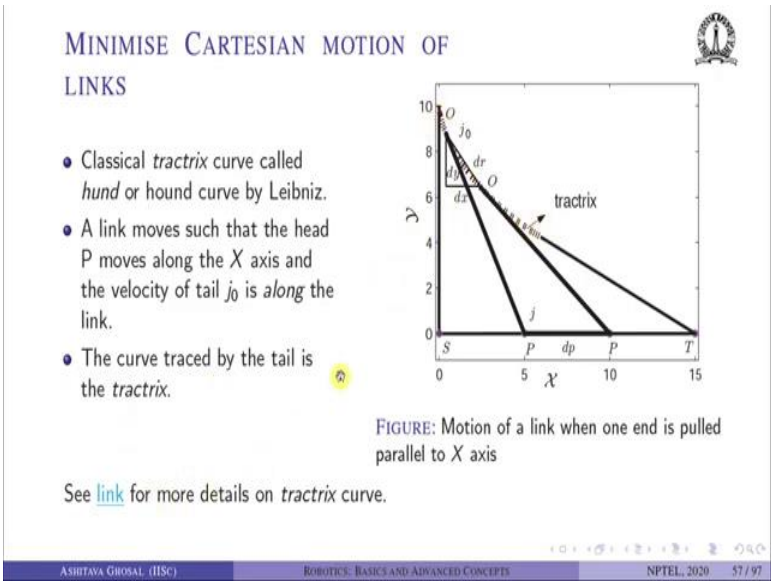
So, what have we done? So, what we have done is, we had a redundant system; we had two equations in 3 unknowns, we are chosen to minimize the sum of the squares of the joint rotation, ok. And then we have solved it as an optimization problem.

(Refer Slide Time: 16:53)



The interesting part of this optimization problem is, we can even impose additional constraints. So, for example, if you say now that in addition to minimizing the square of the joint rotations, sum of the squares of the joint rotation; we say that θ_2 should not cross plus minus 120 degrees, ok. So, again we can solve this optimization problem for the same trajectory and you can see that this plots of θ_1, θ_2, θ_3 are slightly different ok; they do not cross, θ_2 does not cross 120 degrees.

(Refer Slide Time: 17:34)



Now, let us continue with as I said, we can also minimize the Cartesian motion of the links. So, this is a very well known problem; we are going to draw on a very well known problem called the classical tractrix curve ok, also sometimes called the hund or the hound curve and it was invented by this famous mathematician called Leibniz, ok. Leibniz also invented calculus, the way we do it nowadays.

So, what is the basic idea that, consider a link which is lying along the Y axis, so somewhere 0 to 10. What I want to do is, I want to move the head of the link which is lying, which is at 0 along the X axis, ok. With the constraint that the velocity of this tail which is at this end of the Y axis is always along the link, ok.

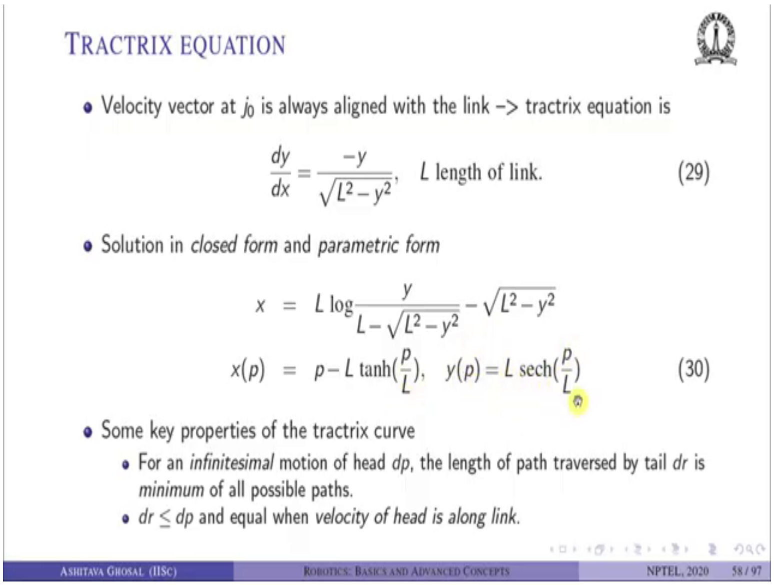
Why do I put this constraint? Because if you just move this head along the Y axis without any constraint, the tail can move arbitrarily in all possible directions, ok. So, the constraint posed by Leibniz was that, we want the velocity of the tail to be always along the link, ok.

So, this problem has a solution and he showed that the curve traced by this link is called the tractrix, ok. So, you can see more details about this tractrix curve, some of the very nice properties in Google, in some Wikipedia link is there.

So, in so basically what are we doing; we have a link a planar case right now and it is being moved, the head of the link is moved along the X axis or parallel to the X axis and at every instant, the velocity of the tail is along the link. So, the curve traced by the tail is this dotted line and this is what we call tractrix.

So, you can see here two quantities which is marked. So, dy is the motion along the Y axis, dx is the motion along the X axis for the tail, and dr is this you know the hypotenuse of dy and dx, small triangle.

(Refer Slide Time: 20:04)



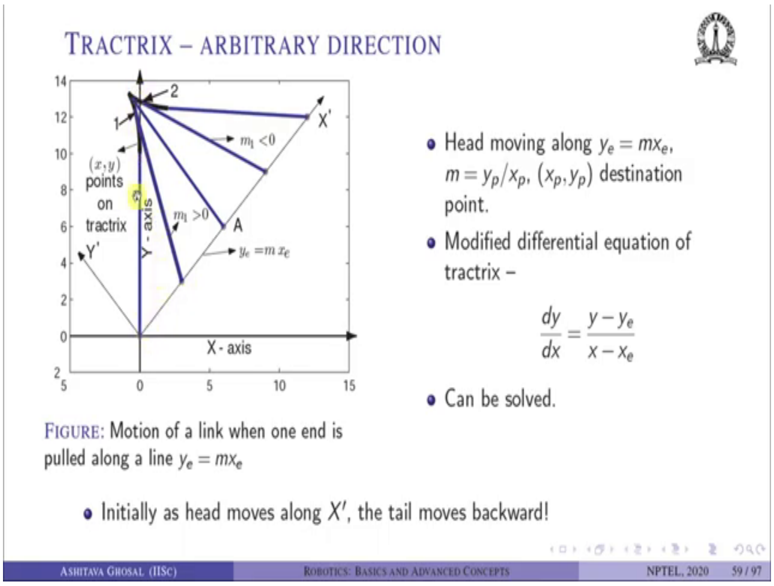
So, since the velocity vector at the end or at the tail or at the j_0 is always aligned with the link; the equation of the tractrix is given by $\frac{dy}{dx} = \frac{-y}{\sqrt{L^2 - y^2}}$. So, the length of the link is L. So, it turns out that the there is a closed form solution for this differential equation, ok.

And it is a interesting solution; because we can solve this differential equation x as a function of y; most of the time we solve y as a function of x, ok. So, in this case, it is $x = L \log \frac{y}{L - \sqrt{L^2 - y^2}} - \sqrt{L^2 - y^2}$, ok. Or if you go back to this picture and if you consider this small motion as dp, like p is a parameter along this X axis motion; so we can write this closed form solution in a parametric form, which is $x(p) = p - L \tanh(\frac{p}{L})$ and $y(p) = L \operatorname{sech}(\frac{p}{L})$, ok. So, this is a closed form solution for this curve traced by the tail, when the head is moved parallel to the X axis.

So, let us look at a few very important properties of the tractrix curve. So, one important property is, for an infinitesimal motion of the head given by dp; the length of the path traveled by the tail dr is minimum of all possible paths, ok. So, there is a small infinitesimal motion of the tail of the head which is happening along the X axis; the tail is moving such that it is always, velocity is always along the link and the infinitesimal motion of the tail given by d r is minimum of all possible motions of the tail.

More importantly, dr is less than or equal to dp and it is equal when the velocity of the head is along the link, ok. So, in this figure dr this quantity is less than or equal to dp; if the head is moving along the Y axis, then dp will be equal to dr, ok. So, this is a rigid link, ok. So, hence this distance will always be conserved between 0 and the head and the tail.

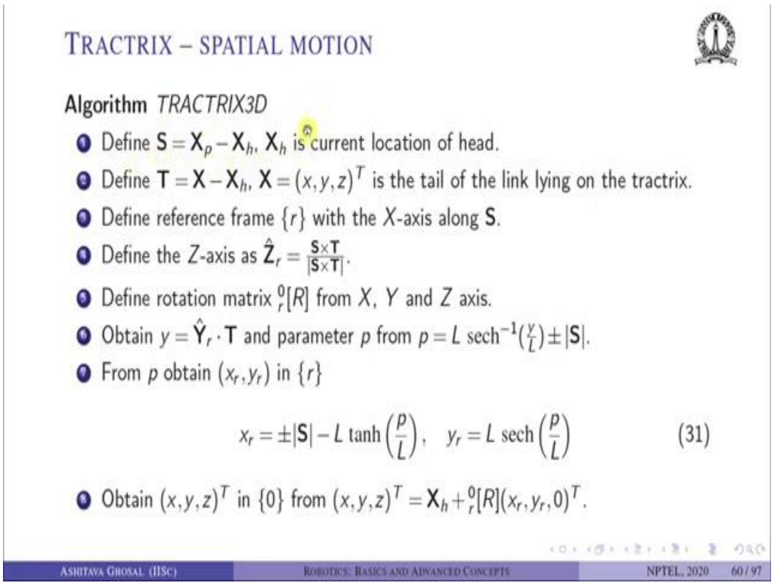
(Refer Slide Time: 22:52)



So, we could extend this idea of a tractrix not moving along the X axis, but in an arbitrary direction. So, along a line which is given by y_e is equal to mx_e. So, basically it is in some along a angle which is given by tan of the angle is m. So, if the head is moving along $y_e$equals mx_e, where m is xp / yp. So, xp and yp are the destination points; we can have a, modified differential equation of the tractrix which is given by $\frac{dy}{dx} = \frac{y-y_e}{x-x_e}$, ok.

So, this can also be solved and a plot of how it, how the tail moves is shown in this picture. So, initially the link is along the Y axis; we make a motion along the X'direction ok, in this direction. As you can see, initially the tail will move backwards, ok. So, it goes backwards and then comes forward and then again eventually it will go like this. So, it goes backwards and then comes backward forward.

(Refer Slide Time: 24:13)



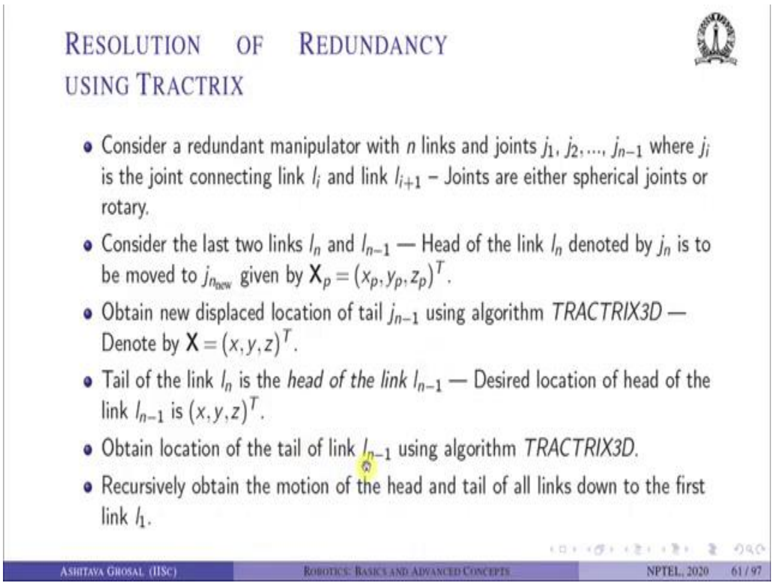
So, let us define a simple algorithm. So, first define vector S, which is $X_p - X_h$. So, X_h is the current location of the head, X_p is the desired location of the head. So, we define a vector T which is $X - X_h$, where X is the x, y, z is the tail of the link lying on the tractrix, ok. So, we are trying to extend it to 3D. So, first define a reference coordinate frame r with the X axis lying along S; z axis lying along S x T, of course we will make it as a unit vector divided by magnitude of S x T.

And then we define a rotation matrix ${}^0_rR$. So, the rotation matrix of this reference frame with respect to the fixed global reference frame; the first column is the X axis, second column is the Y axis, and third column is X axis standard definition of a rotation matrix. Then we obtain y which is $\widehat{Y_r}$. T. And then in this terms of this parameter p, which is $L \; \mathrm{sech}^{-1}\frac{y}{L} \pm |S|$; we from this p we obtain $x_r \; and \; y_r$ in this reference coordinate system r.

So, we can find what is x_r just by the solution of the tractrix equation in terms of this parameter p, which is given by $\pm|S| - \mathrm{L}\tanh\frac{p}{L}$, and y_r is L $\mathrm{sech}\frac{p}{L}$. Then we obtain x, y, z in $\{0\}$ coordinate system by transform it back to the 0th coordinate system by pre multiplying x_r , y_r , ${}^0_r[R]$ with and addition of this X head, the current location of the head. So, what have we done? So, if I want to move not in the plane, but in some arbitrary direction.

So, initial point is X_h and I want to move to x, y, z some other points in 3 D space. By following this algorithm, I can find out in closed form ok, of course discretized closed form in terms of this parameter p; where is the tail and where is the head ok, following the tractrix curve.

(Refer Slide Time: 26:45)

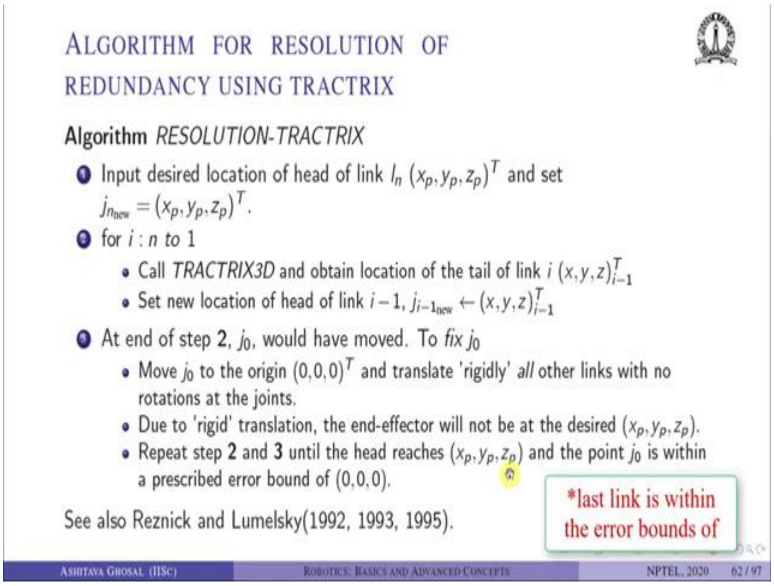


So, now consider a redundant manipulator with n links and joints j_1 j_2 all the way till j_{n-1}, where j_i is the joint connecting link l_i and l_{i+1}. So, the joints could either be spherical joints or rotary joints, ok. Consider the last two links l_n and l_{n-1}. So, the head of the link l_n it is denoted by j_n is to be moved to a new location, $j_{n_{new}}$ and this new location is given by X_p., which is vector which is x_p , y_p, z_p ok.

So, obtain the new displaced location of the tail j_{n-1} using the previous tractrix 3 D algorithm and let us call that point x, y, z. So, now, the tail of link l_n is the head of the link l_{n-1} and desired location of the head of the link l_{n-1} is x, y, z.

So, basically the last link has moved to one position, the tail has moved along the tractrix to some position; then the link before that last link, now we know what is the desired motion of the head, ok. And then we find that this motion of the tail again according to the tractrix algorithm. So, we recursively do this till we go from the head all the way to the tail of the first link l_1, ok.

(Refer Slide Time: 28:21)



So, this is given in terms of a algorithm that, input the desired location of the head of the last link which is x_p , y_p, z_p and set $j_{n_{new}}$ as x_p , y_p, z_p column vector. Then from i equals n to 1, you call this tractrix 3D and obtain the location of the tail of the link i, ok.

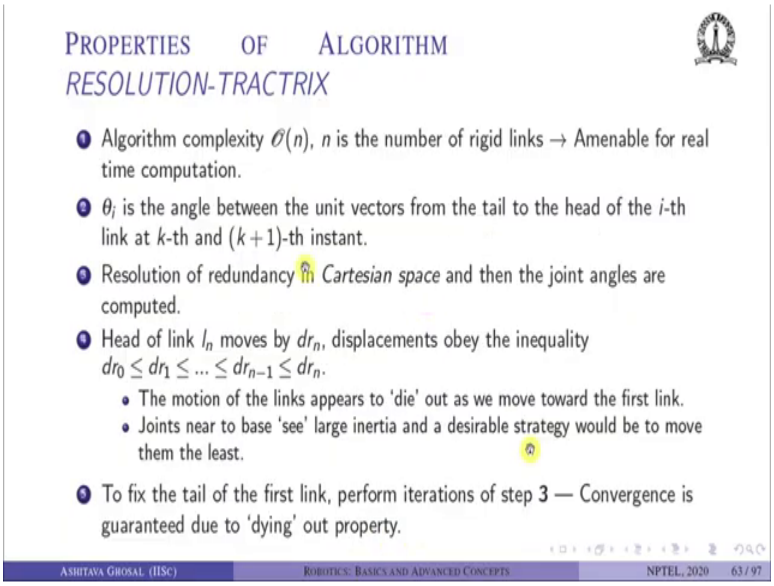
So, n then tail of the link, then the previous link, I know what is the location of the head, new location of the head; find the tail and we keep on going backwards, ok. So, you set the new location of the head of link i - 1 to $j_{i-1_{new}}$ which is x, y, z transpose i- 1, which you have calculated in this previous step.

So, at the end of step 2, j_0 would have moved. To fix j_0 , move j_0 to the origin, ok. So, because every link is moving, the end will also move a little bit. So, we fix the first joint; because normally in a robot, the first joint is fixed, ok.

So, if you want to keep the last joint fixed, we translate rigidly all the links with no rotations at the joint. So, the last first joint has moved a little bit, you translate it back to the origin (0, 0, 0). And due to the rigid translation, the end effector will not be at the desired x_p , y_p, z_p ok.

So, then you repeat steps 2 and 3 till the head reaches x_p , y_p, z_p. So, you have to do a little bit of a arbitration and then you stop when the last link is within the error bound is (0, 0, 0), ok. So, this was an algorithm which was first developed by Reznick and Lumelsky in 92, 93 and 95; they wrote several papers.

(Refer Slide Time: 30:22)



So, let us look at some of the properties of this algorithm, ok. So, let we are I am calling this resolution tractrix. So, the algorithm complexity is O(n), ok. What do we mean by O(n)? It scales linearly with the number of rigid links, ok. So, if you have five links and you apply this tractrix; if you make it double, if you have now ten links, then the effort will become twice only, ok. There are other algorithms, where it can go square of the number n or even worse to the power of n, ok.

So, this is a linear complexity. How do I find the θ_i, which is the rotation at the joints? Because eventually we have to rotate by means of a motor; it is very simple, we find the angle between the unit vectors from the tail to the head of the i th link, at the k th and k +1th instant, ok.

So, at one instant I draw a vector, in the next instant I draw the same vector from the tail to the head and I find the angle between these two vectors, ok. And this as the link is moving, we can constantly find at every instant.

So, this resolution of redundancy is in Cartesian space and then the joint angles are computed. So, remember we are trying to minimize the linear velocity of the tail, ok. So, the velocity dr, which is in some sense related to the linear velocity of the tail; it is always along the link and dr is what is less than dp, ok.

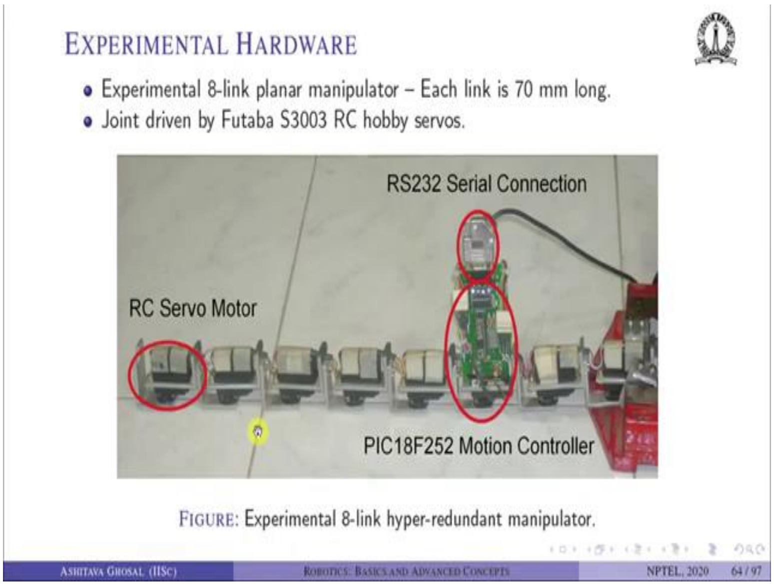
So, the head of the link moves by dr_n, the displacement obeys the inequality dr_0, dr_1 all the way till dr_n. So, remember dr was less than dp and what is first one is dr; the second for the second link, the dp will become like dr for the previous link.

So, at every instant dr is less than dp; which means that dr_0 is less than dr_1 all the way till dr_n, ok. So, what is happening? The motion of the link appears to die out as we move towards the first link. So, I have moved the head, calculate the motion of the tail; then I use the second link, the motion of the head is the motion of the tail of the next link and so on and since dr is less than equal to dp, that motion will always die down, ok.

This is a very good idea, because joints near the base sees large inertia and a desirable strategy would be to move them the least, ok. So, if you have a robot, like let us say the Puma robot and we have motors at the base which are very big and heavy we; because it needs to move the outer links also, ok. So, if you move the first joint the smallest, then it is a good idea.

To fix the tail of the first link, perform iterations of step 3; convergence is guaranteed, ok. Remember if the head is moved, the tail moves little bit; then the head that becomes the head of the previous link and so on. But eventually the first link will also move a little bit; because of this going dying down property; but we can rigidly move it back to the origin and then again repeat the step. So, this will converge, because always dr is less than or equal to dp and there is a dying out property of the links.

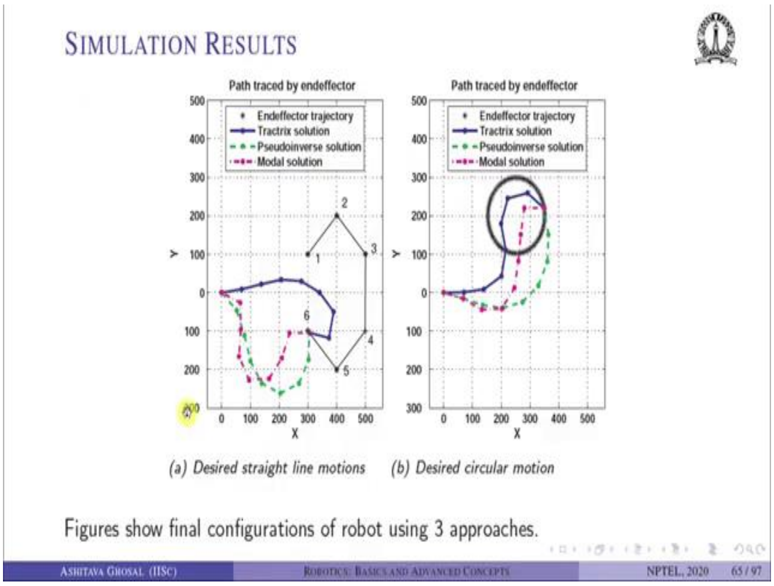
(Refer Slide Time: 34:15)



So, this is an algorithm which was developed by long time back; but we have also worked on it and we implemented it on an experimental hardware. So, we have an experimental 8 link planar robot; each link is 70 millimeters long. So, this is one link ok. And each joint of the robot is some motor, which is an RC hobby servos motor, Futaba S 3003.

So, this motor can rotate, ok. So, we can give some theta rotation of each link and then we have this controller, which can pass from some laptop or somewhere and then there is a driver circuit, which can power all these motors. So, we build this 8 link hyper redundant robot to move on a plane, ok. So, in a plane, anything more than 3 degrees of freedom will make it redundant, ok. So, 8 means, that is large number of joints and ok. So, we have infinitely many possible solutions.

(Refer Slide Time: 35:29)



So, if we implemented this tractrix based solution and we did some simulations. So, here are two simulations. So, I want the end of this 8 link robot to move from 1 to 2, 2 to 3, 3 to 4, 4 to 5, and 5 to 6 and then again 6 to 1. So, it should trace a hexagon, ok. So, these are straight line motions in a plane. So, we can solve this tractrix equation. So, initial location is given here, ok. So, 1 to 2, and 3, 4, 5, and 6.

See this blue dotted line shows how the robot looks like when it has come back to 6, ok. I will show you later how each one of these joints are moving; but this is a picture of the robot, when it has completed this task. There are other ways to resolve redundancy ok, we will go to, we will see this later; many researchers have worked on it, it is called the pseudo inversed solution, ok.

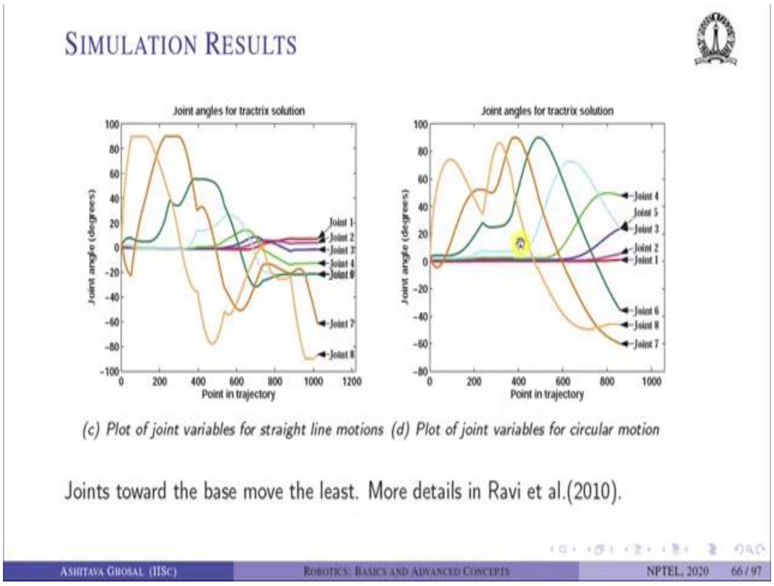
So, for the moment let us assume, you accept my word that there is something called as a pseudo inverse solution; there is something called as a modal solution. The pseudo inverse is nothing, but something similar to minimizing the joint velocities, ok.

This modal solution basically means that, we have a backbone curve and we move the backbone curve, ok. So, if you were to use this pseudoinverse solution for this task, then the green dotted line shows how the robot will look like after it has completed the task.

So, they all start from the same configuration which is 1 traces that are straight lines and come to this point 6. The tractrix solution looks like this; the pseudoinverse solution looks like this, and the modal solution looks like this, this pink dotted line. We also tried another task, which is tracing a circle; here also you can see that the blue dotted line or piecewise line is the tractrix solution, ok.

So, what you can see is that, the first joint here, the second joint here or even the third joint, they are moving very little; the last joints are moving a lot ok, they are rotating a lot. Whereas, in the pseudoinverse and modal solution, the first joints and the first links they also rotate; the same thing can be seen when it is trying to trace a circle, ok.

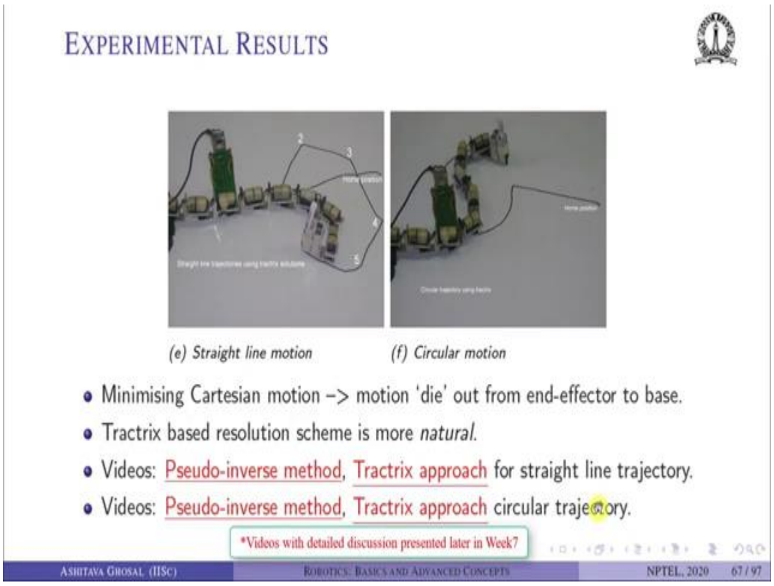
(Refer Slide Time: 38:15)



We can solve for the tractrix by finding the joint angles at each instant of time. And we can see that the joint 1 which is this pink line, it has moved very little; whereas the joint 8 which is this yellow line, it has moved a lot, ok. So, that is the same story when you are trying to trace a circular trajectory. So, what is the basic idea? That, in a tractrix solution, we mean move the farthest link at the furthest joint as much as possible and then we move the previous joint, then we move the previous to that and so on.

So, because this dr is less than or equal to dp; because of this dying motion, the first few joints move very little. So, joint 1, joint 2 moves very little; joint 3 is moving a little bit ok, joint 4 is little bit more, joint 5 and 6 and 7 they will move much more, ok. This work was published in a paper in 2010 by our students.

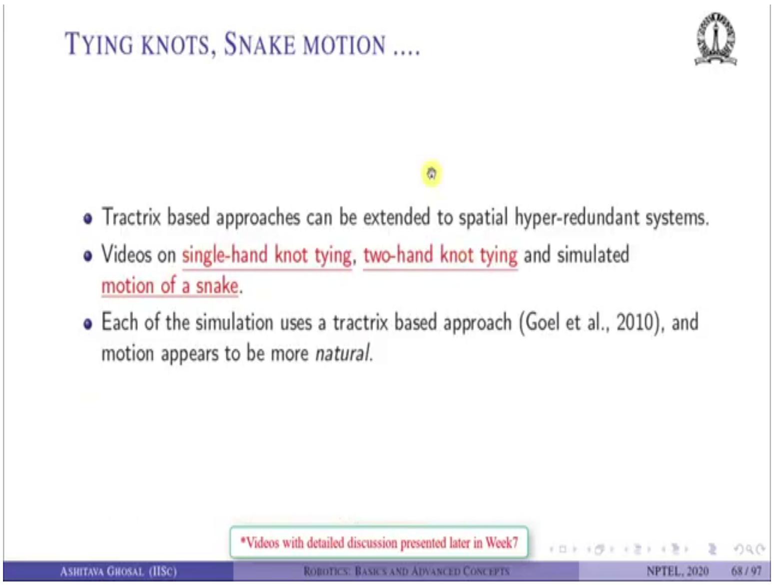
(Refer Slide Time: 39:35)



And I am going to show some experimental results later on, after this lecture is over; I will show you videos of this 8 link robot tracing this hexagon or tracing this circle, ok. So, basic thing is we will see later that, this minimizing Cartesian motion, the motion dies out from the end effector to the base. The tractrix base solution scheme is more natural ok; the joints far away from the base move more and then slowly the motion dies down as you come towards the base.

We will also see in the videos that, the trajectory traced while tracing this hexagon or the circle seems much more smoother and natural as compared to the pseudoinverse solution or as compared to the modal solution.

(Refer Slide Time: 40:37)

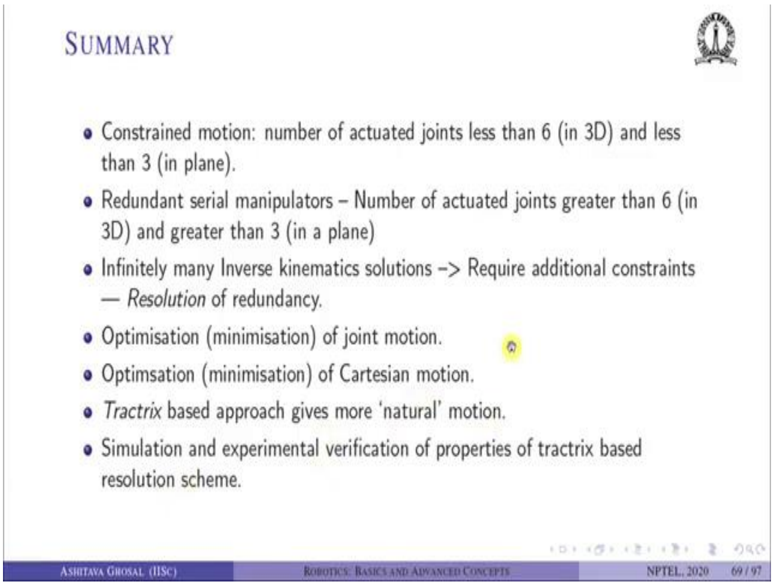


The tractrix based approach can be extended to any spatial hyper redundant system; it need not be a robot. So, we tried this idea of moving according to tractrix for simulating the motion of a thread. So, basically a thread can be discretized into small small circles connected by joints, in this case spherical joints and then we showed that it could move the tip of the thread to tie a knot, ok.

So, we can have a knot which is what is called as a single handed knot. So, only one end of the thread is moving; we can also simulate the motion of a thread when it is tying a two handed knot. So, two handed knot means, both ends of the thread are moving, ok. We also showed that you could simulate the motion of a snake ok.

So, each of the simulations used in this video which I will show later, uses a tractrix based approach, ok. And we can see later or we can see from this videos that, this motion appears to be quite natural looking.

(Refer Slide Time: 41:54)

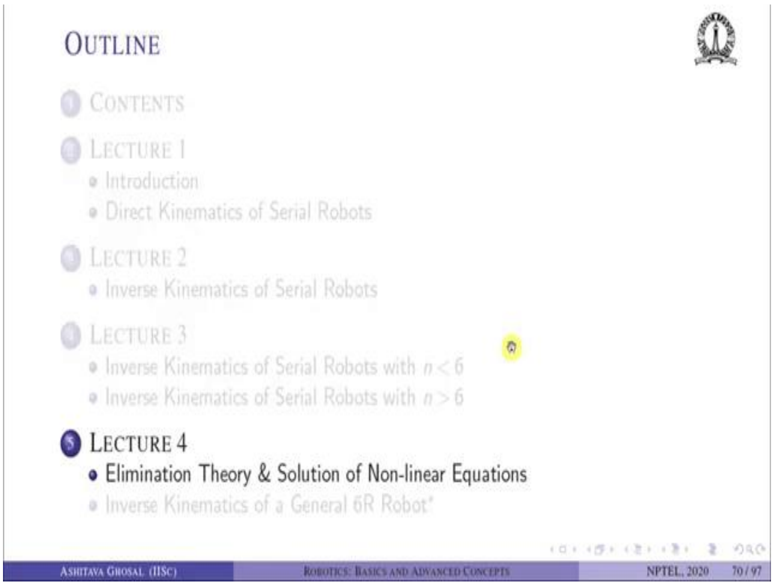


So, in summary, we can have constraint motion when the number of actuated joints is less than 6 or less than 3 in a plane. So, in the case of the SCARA robot, the last two rotations of the end effector are not allowed, those are the constraints. If you have redundant serial manipulators, basically the number of joints is greater than 6 in a 3D and greater than 3 in a plane.

In a redundant system or in a redundant manipulator, we have infinitely many inverse kinematic solutions. So, naturally we need to enforce or require additional constraints. So, this notion of requiring or generating additional constraint is called resolution of redundancy. So, I have showed you one where we can optimize or minimize the joint motion; so we can minimize $\theta_1^2 + \theta_2^2$ and so on, ok.

So, this is minimizing the joint space motions. We can also minimize the Cartesian motion of the links of a robot, ok. And one such approach is this Cartesian based approach which turns out to give a more natural looking motion, ok. And this tractrix based approach can be shown or has been shown using both simulation and experimentally, ok.

(Refer Slide Time: 43:36)



And so, with this I am going to stop and in the next lecture, we will look at with more abstract concepts on elimination theory and solution of non-linear equations, and how we can apply this elimination theory and solution of non-linear equations to find the inverse kinematics of a general 6 degree of freedom or 6R robot.