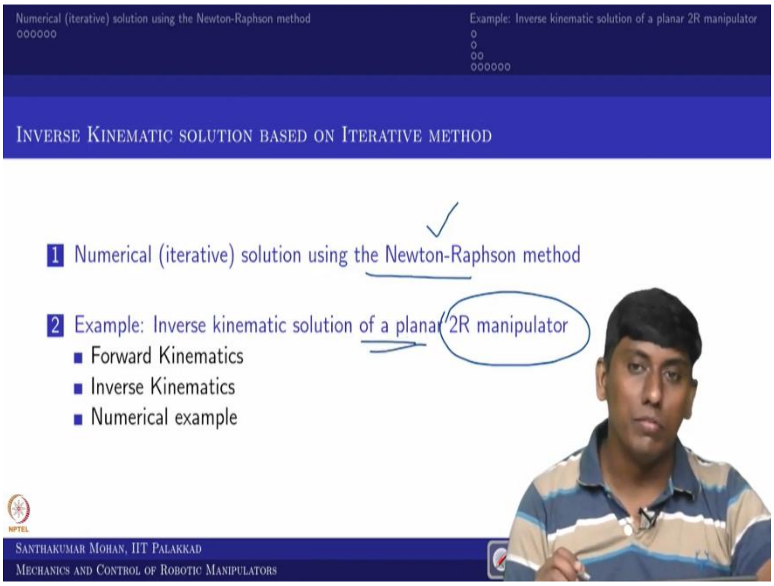
Mechanics and Control of Robotic Manipulators
Professor. Santhakumar Mohan
Indian Institute of Technology Palakkad
Lecture 16
Inverse Kinematic Solution Based On Numerical Methods

Welcome back to Mechanics and Control of Robotic Manipulator. Last class what we have seen was inverse kinematic, especially inverse kinematics offers some serial manipulator we have seen, few examples that is what we have seen. So, this class we are going to add little more. So, what that means?

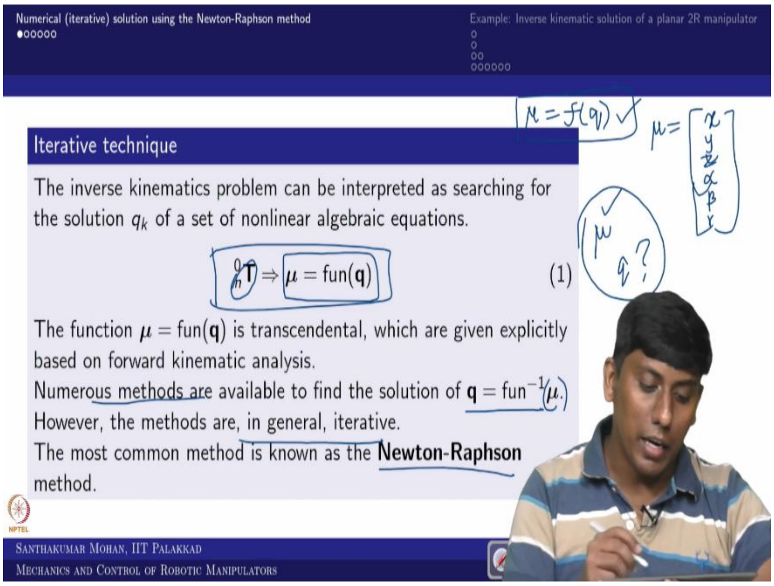
So, you know like there are two methods of solutions for finding inverse kinematics. The last class we have seen few numerical examples which will give the closed form solution. But today we are going to take a few examples which is going to give a numerical method of solution. So, in the sense we are going to use iterative method. So, we have actually like seen there are several iterative methods are possible, but we are going to see one specific method called Newton-Raphson method.

(Refer Slide Time: 00:57)



So, I will be giving an example with 2R planar serial manipulator, but it is not that restricted to such manipulator, it can be used anything. So, I have chosen this only because I can show it in the pen and paper. So, that is the whole idea I was taking 2R serial manipulator. So, let us come first to what is Newton-Raphson method, then we will be going the example which we are going to focus. So, if that is the case, we can start.

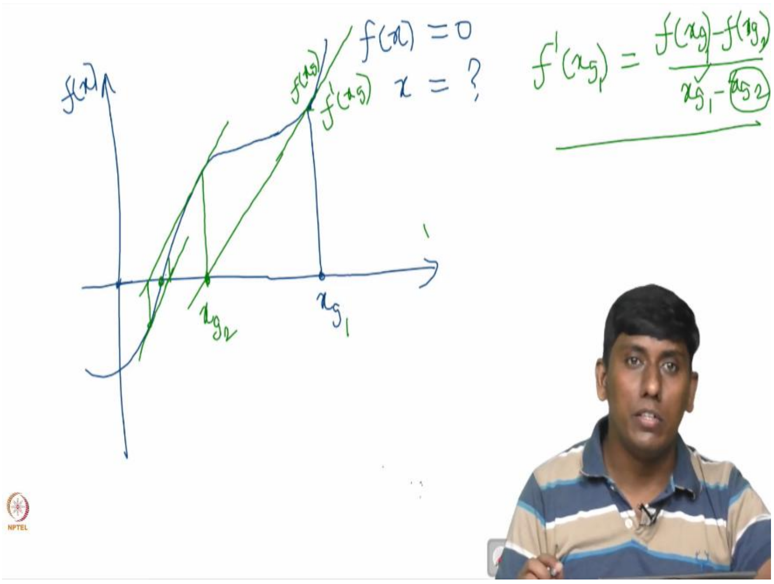
(Refer Slide Time: 01:29)



So, you know like what we have. So, we have mu as so function of q. So, this is given so this is what we have found. So, especially from you can say transformation matrix of nth with respect to 0 you can found finally what is your mu. So, mu is nothing but, so your end effector positions. So, then you have end effector orientations. So, this is what mu, but we can generalise and all. So, now if you have this function, can I use any iterative method where I can find if I give mu, can I find q? So, that is the question.

So, if I have this big question, numerically there are several ways. So, that is what numerous methods are available to find this q is function of, function inverse of mu but we are taking the iterative because that is more general, and it is applicable to any end of serial or parallel manipulator. In that one of the popular methods, we called Newton-Raphson method. Even some people shorty calls it Newton's method. So, this is what we are trying to see.

(Refer Slide Time: 02:43)



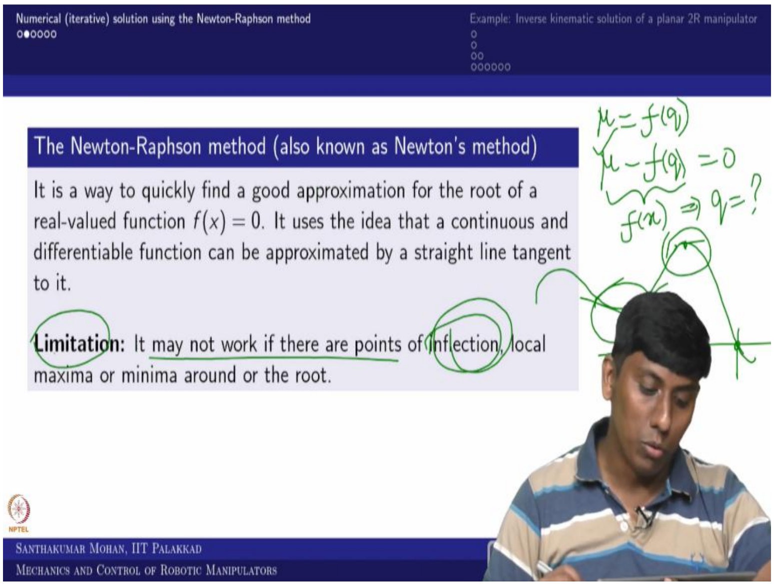
So, for seeing that, I am trying to show one thing. So, this Newton-Raphson method is mainly come with this. So, if function of x is equal to 0, can I find the solution x? So, for that we are taking a plot, x versus, you can say f of x. So, now I am trying to show this, so you can see that this is somewhere like this. So, this is your function of x, and this is with respect to x. So, now you can see like I am taking a random point like I will just complete this.

So, I am taking a random point which I say is the initial guess. So, this is I call x guess. So now what I can do, I can make a slope. So, I will just make it, this is little more. So, something like this. So, I am trying to plot, so I will just because I did not properly make it. So, now I am taking a guess something like this, this is what my initial guess which I call x g. So, now what we are seeing that the slope I can find and the function of x g I can find. The slope I can find, the slope is coming like this. So, this f dash of x g.

So, now what you can see this is ending on x somewhere, I call something like this as, so this is g1 and this g2. So, now what I can find, I can find this is going to be 0 and this I know. So, in the sense that f dash of x g how I can write? I can write this as f of x g 1 minus f of, so x g 2 divided by x g 1 minus x g 2. So, this way I find. From this what I know? I know x g 1, I want to find out x g 2.

So now if I see here, so what I can see this is another slope will come. So, you can see, so now this is giving another iterative point. Then I can make a slope, so then that is giving. So, like that you keep on going, finally you end up with this point. So, this is the basic principle which what you call the Newton-Raphson method is giving.

(Refer Slide Time: 05:22)

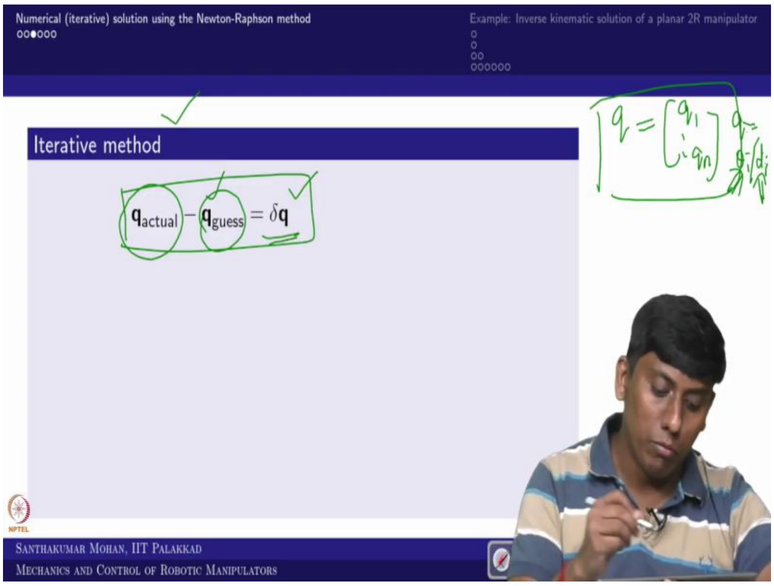


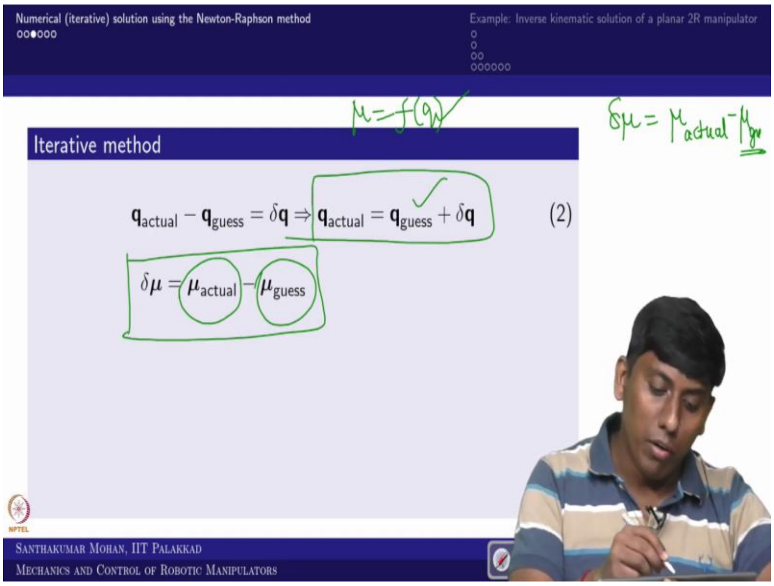
So, the same Newton-Raphson method, we are going to use for a different benefit. What? So, we are going to this what you call mu is function of q. So, this is we are going to rewrite as this way equal to 0. This is what we are going to call as a function of x, where mu is given and we are trying to find out q, what is this with mu is known? That is what we are trying to find.

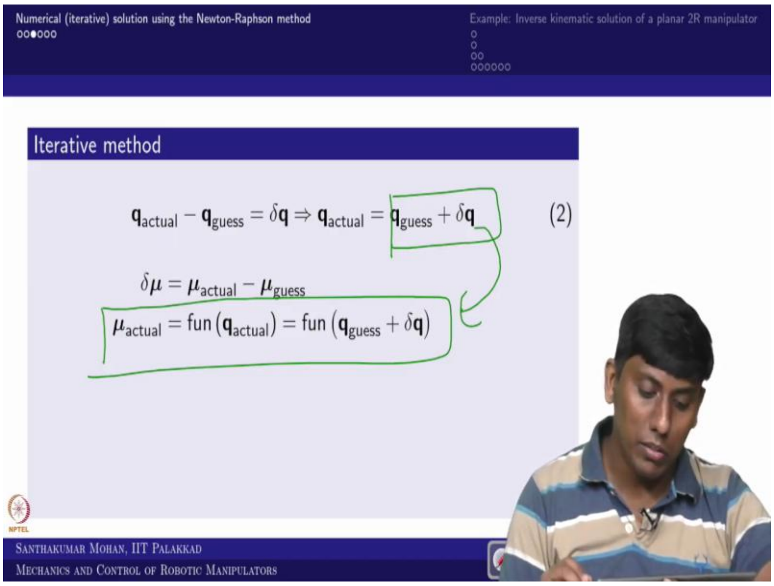
So, if that is the case then what is the limitation? The limitation is, so it may not work if there are points of inflection. So, what that means? So, if it is inflection, so you have something like this, you take a point here. This may not agree with another point, it is come closer here. So, now you can see that the original solution, so probably you can see the original solution somewhere here, so this is what you can see, this is the solution and you have taken a point here, so this point ends up somewhere close here.

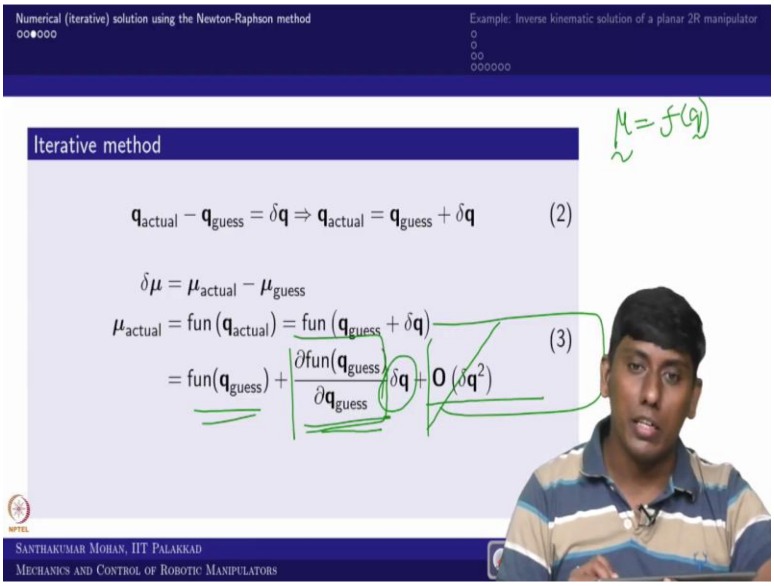
That is what we are saying that the inflection point, but the original solution is here, so which even some people call, this is maxima, and this is the minima. This is the localised, so if any point you choose in your initial guess somewhere here, it would be rotating here itself. So, that is one of the limitations in the Newton-Raphson method, but still, we can try to avoid by providing something, you can say, some additional restriction we can do it. So, if that is the case, so we can see the same Newton-Raphson method what you have come across in several you can say numerical methods courses.

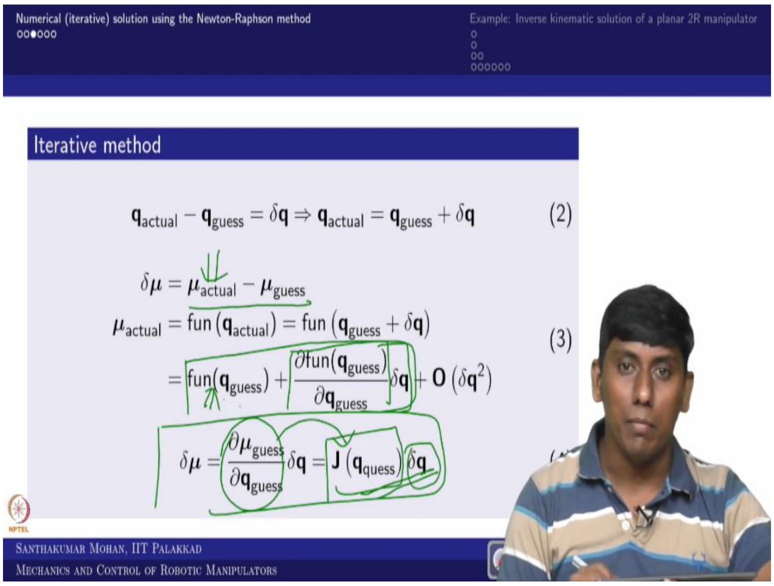
(Refer Slide Time: 07:06)











So, we are going to modify that in this course. So, what we are trying to say that it is iterative method, where we are saying that the actual is somewhere, we are taking a guess, so the error between these two I am calling delta q, so where q is your joint space variable, so q 1 to q n.

This q i can be either so theta I or d i, so if it rotary joined theta i, if it is a translation joint it will be joint distance. So, in that sense you have a q vector, and that q vector would be in this form. So, now we are trying to find out the q vector, but we are taking a guess and actual is something, so we do not know the error, but we will assume that this is the error. Now we want to rectify or go away from this error, for that what we are seeing, so this way you can rewrite this actual as q guess plus delta q.

But what we know? We can write delta mu, so delta mu again you can write this is actual minus mu guess in the sense so you know mu is function of q, so you can substitute this guess and then whatever value is coming that is what mu guess and the mu actual is given. So, now what you will get? You will get another error called delta mu. So, can we relate anything? Yes, so for that what we are trying to do?

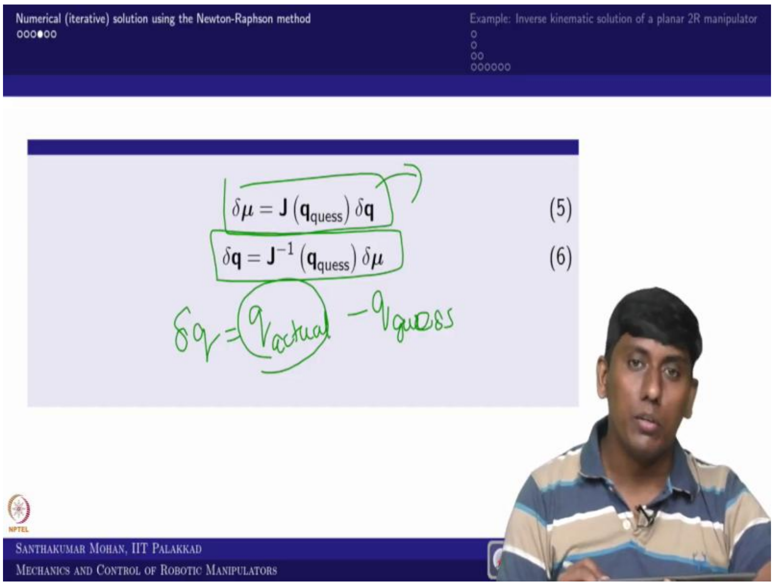
So, we take the mu actual, so mu actual what? So, it is function of q actual but the q actual is we have written this form. So now you substitute this. So, what you will get? q guess plus delta q, so now you can say mu actual is in this form and you use a Taylor series of expansion. So, what that will give?

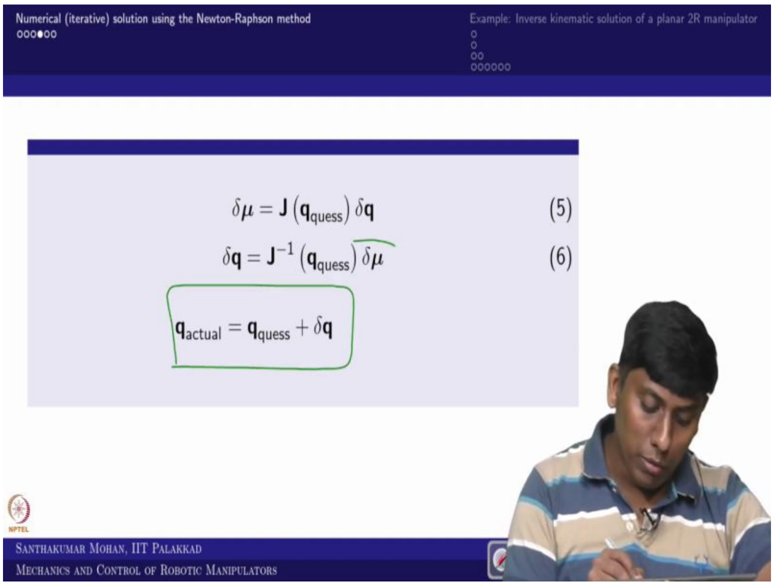
That will give function of q guess plus the partial derivative with respect to delta q and the higher order partial derivative multiplied with the delta q squared and other things. So, we are assuming that this delta q is very, very small, so this all we are making it as 0. So, now what we actually see? This is what left but this is what it is, so you have function of q. So, this is also vector and this is also a vector. So, you are taking differentiation with respect to, you can say vector differentiation with respect to another vector.

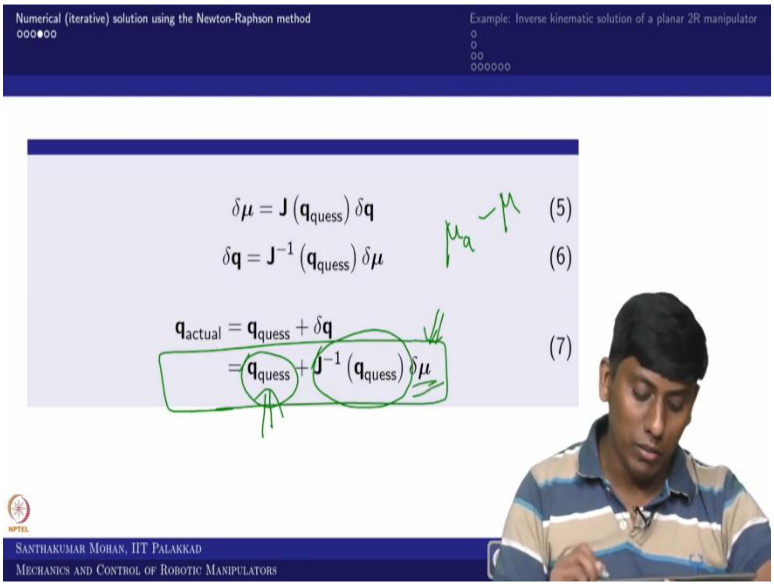
So, obviously that would be end up with what you call partial derivative. So, that partial derivative we are getting. So, that is what we are finding. So, in that sense, what you can write. So, you can write, this is your case and now you can substitute mu actual in this form. So, this will give you mu guess and you have mu actual as mu guess plus this.

So, if you substitute this two will go and only this would be remained. So, this partial derivative would be in the form of matrix, so that matrix I am calling J of q guess which is nothing but a Jacobian matrix. So now in that sense, what we have end up? We have end up with a matrix multiply with an error. But in this error, we do not know.

(Refer Slide Time: 10:19)



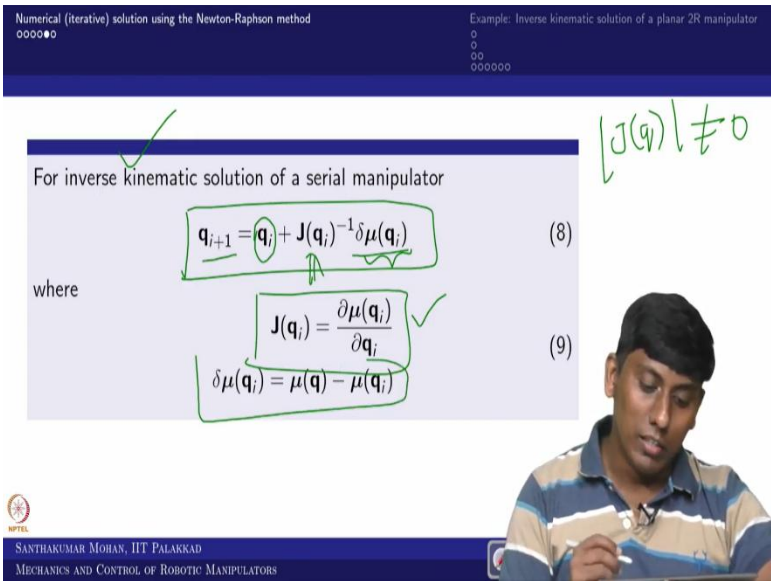




So, we are rewriting this equation and we are rewriting this delta q in the form of this. So now what you have this. So, now this delta q is what? This is q actual minus q guess. So, in the sense, what you can do? You can make q actual one side and put it the q guess the other side. So, what you will get? So, you will get the final solution in this form. So now you substitute this here, so what you will get?

This is the final equation which we are going to use in the Newton-Raphson method. So, now what you can see? Once you choose the guess, so you have found this given, you can find J and then J inverse you can get, and this mu actual is given and mu guess you can find because this function is known to you. So, you can get the delta mu and you can solve q actual one by one, in the sense iterative.

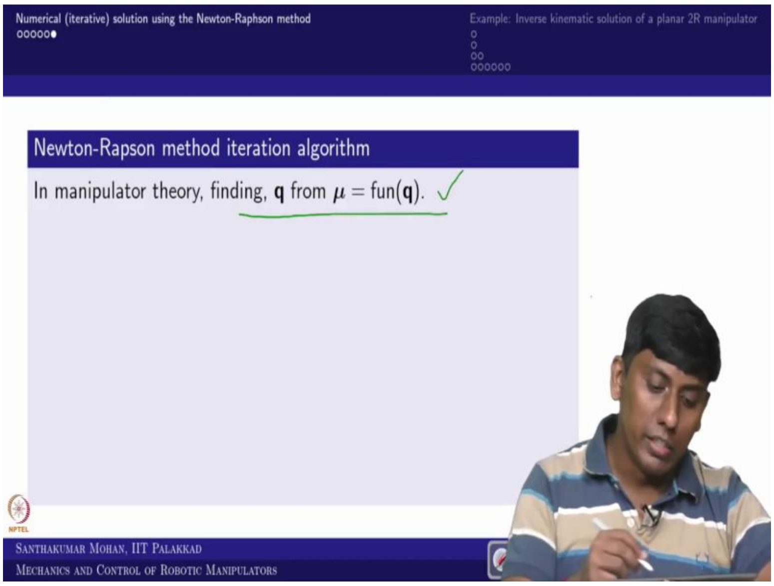
(Refer Slide Time: 11:24)

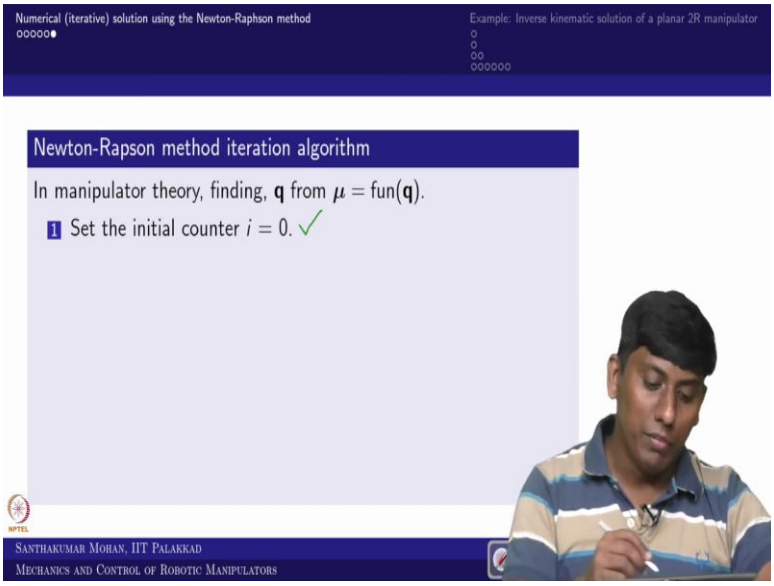


So, that is what we are trying to write. So now you can see that the, if the forward kinematics solution is available. We can find the inverse kinematic in iterative manner, where q i is the initial guess and J of q i is actually like based on the initial guess, what is the Jacobian matrix and this is actual minus, your function of q.

So, this delta mu is obtained and once you get you will get i plus 1, then the next iterative that i plus 1 become i and i plus 1 would be new value. Like that you can go iterative and that would be obtainable. So here, only condition is, you should have this J, this J supposed to be non-singular, in the sense J of q determinant should be non-zero. Otherwise, what happens the J q inverse would be not possible. So, that is what we are writing. So, we will take one, one simple example in the next slide.

(Refer Slide Time: 12:22)





Newton-Raphson method iteration algorithm

In manipulator theory, finding, $\mathbf{q}$ from $\mu = \text{fun}(\mathbf{q})$.

1. Set the initial counter $i = 0$.
2. Evaluate an estimate solution $\mathbf{q} = \mathbf{q}_i$.

Numerical (iterative) solution using the Newton-Raphson method
Example: Inverse kinematic solution of a planar 2R manipulator

Newton-Raphson method iteration algorithm

In manipulator theory, finding, $\mathbf{q}$ from $\mu = \text{fun}(\mathbf{q})$.

1. Set the initial counter $i = 0$.
2. Evaluate an estimate solution $\mathbf{q} = \mathbf{q}_i$.
3. Calculate the Jacobian matrix, $\mathbf{J}(\mathbf{q}_i)$.

Santhakumar Mohan, IIT Palakkad
Mechanics and Control of Robotic Manipulators

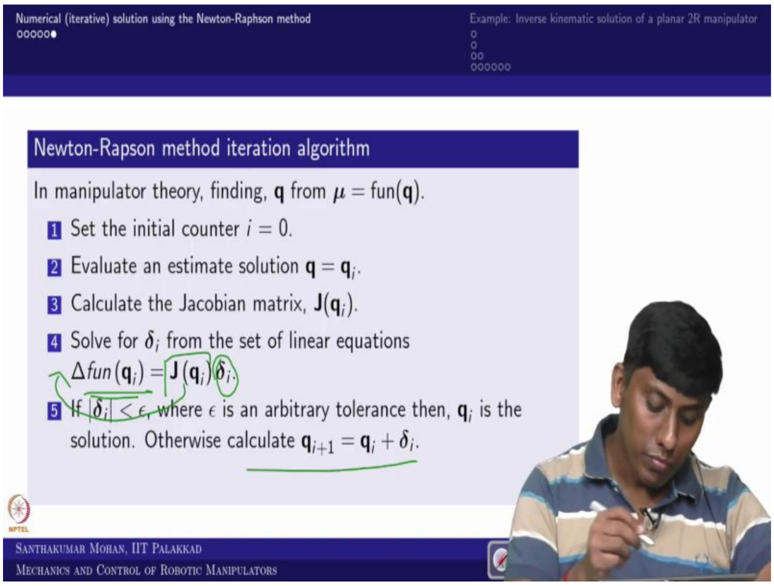
Numerical (iterative) solution using the Newton-Raphson method
Example: Inverse kinematic solution of a planar 2R manipulator

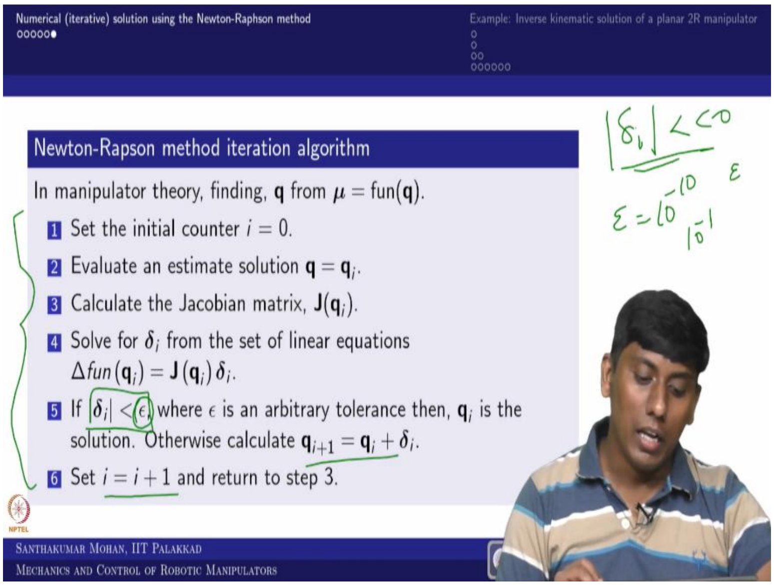
Newton-Raphson method iteration algorithm

In manipulator theory, finding, $\mathbf{q}$ from $\mu = \text{fun}(\mathbf{q})$.

1. Set the initial counter $i = 0$.
2. Evaluate an estimate solution $\mathbf{q} = \mathbf{q}_i$.
3. Calculate the Jacobian matrix, $\mathbf{J}(\mathbf{q}_i)$.
4. Solve for δ_i from the set of linear equations $\Delta fun(\mathbf{q}_i) = \mathbf{J}(\mathbf{q}_i)\,\delta_i$.

Santhakumar Mohan, IIT Palakkad
Mechanics and Control of Robotic Manipulators





So, for making this as algorithm. So, what we can do we can start this is the function, this is available. So, we will start with the you can say, iterative algorithm, so we will start the counter as 0 and we start with the initial guess. So, that is what we call it as q i that we start. Then we are going with the next step, where we calculate Jacobian and Jacobian inverse we calculate and then we will find this.

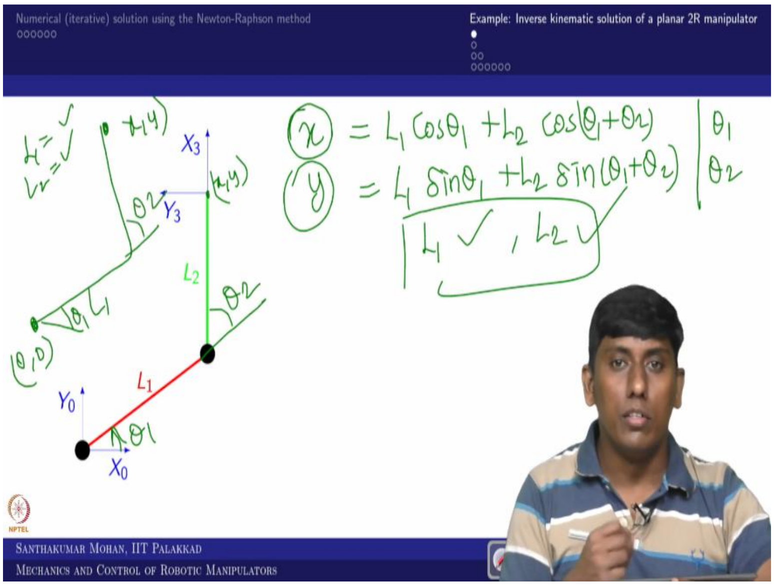
So, once you find this, then we can keep getting it further and you can find this particular aspect. So, in the sense this delta i can be taken as this function, you can say the difference in the function inverse obtainable that you can substitute. So, in the sense whatever we have used in the previous slide that I have rewritten in the form of algorithm.

So, once you have this, you can stop. For stopping we are putting some you can say tolerance. So, for example, your delta i is very close to 0, then you can stop. Need not to be 0.

So, that is what so how close, so that you need to make your own arrangement, so for that we are taking as an epsilon. So, epsilon is your tolerance values, so this epsilon value you can choose based on that you can restrict this algorithm.

For example, you like running in online. So, you no need to go the epsilon as 10 power minus 10 and all. You can even restrict 10 power minus 1 is sufficient, you can stop it. So, that is the criteria here. So, now you can see the same thing what we have written in the equation form, that is written in the form of what you call algorithm. The same algorithm, we will try it in a numerical example. So, then you will get much more clarity how this algorithm work. In fact, the next lecture I will be showing this in the MATLAB simulation or you can say MATLAB algorithm based. So, we will see.

(Refer Slide Time: 14:27)

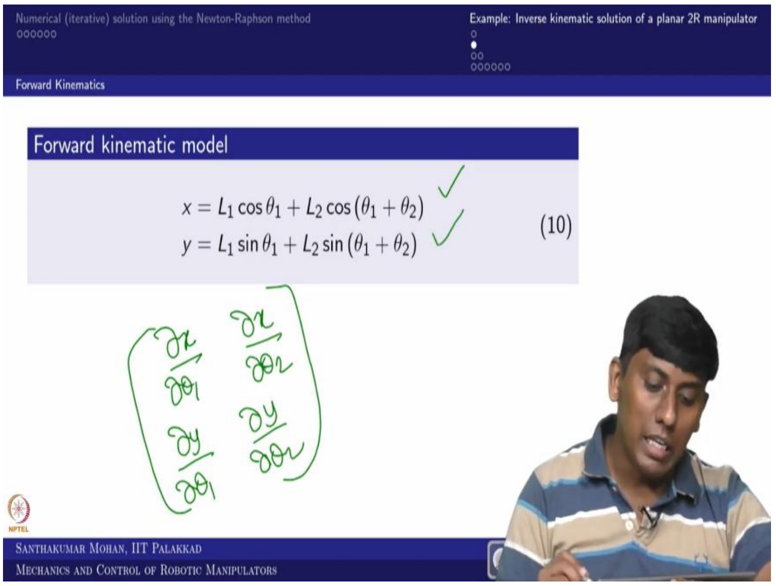


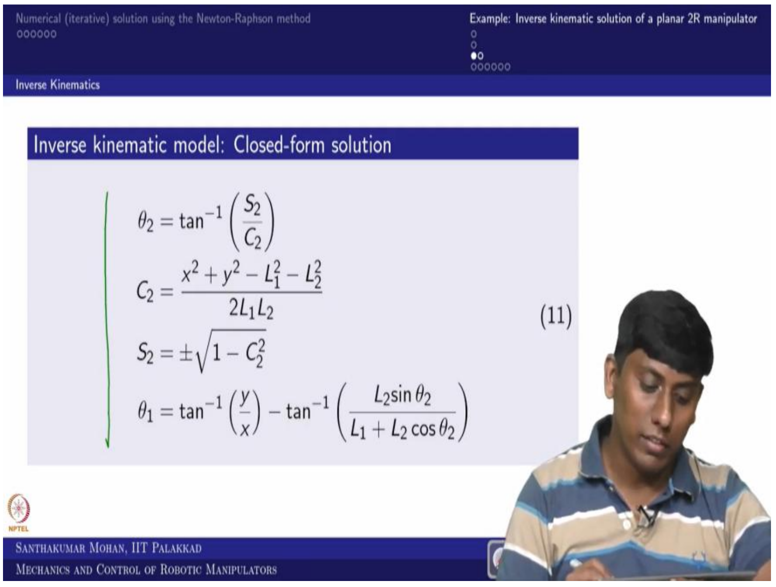
So now let us move the numerical example, for that we are taking a 2R serial manipulator. So, we know like this. So, what would be the x position of this, this is x and y. So, this x would be L 1 cos theta 1 plus L 2 cos theta 1 plus theta 2. So, theta 1 is this angle and theta 2 is this angle, so we can get this.

So, similarly, y is L 1 sine theta 1 plus L 2 sine theta 1 plus theta 2. So, now what we are saying. So, if we know these two. Can I find theta 1 and theta 2 numerically? Yes. So, for that what are the things supposed to be known? This L 1 should be known and L 2 should be known. These are geometrical parameter which are supposed to be given.

So, in addition to that, what actually will be given to you. So, it would be like, 0, 0 and your x, y would be given, and the L 1 value will be given, L 2 value will be given. So, you are trying to fix this in the form, so that you can find theta 1 and theta 2. So, this is what they are trying to do.

(Refer Slide Time: 15:44)

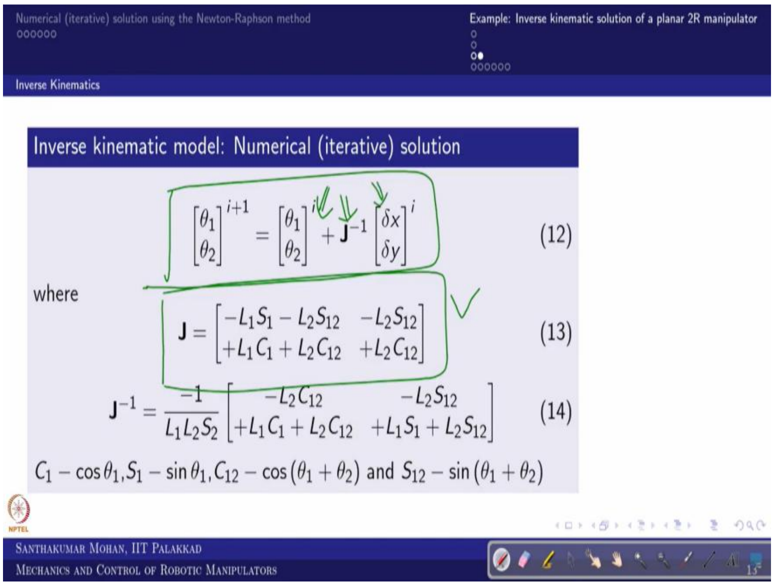




So, for that what one can do? We can go the Forward Kinematic model. So, this is what I have written. So, now you can take the partial derivative, what that means. So, doe x by doe theta 1. So, doe x by doe theta 2 and doe y by doe theta 1 and doe y by doe theta 2. So, this is what you call the Jacobian matrix.

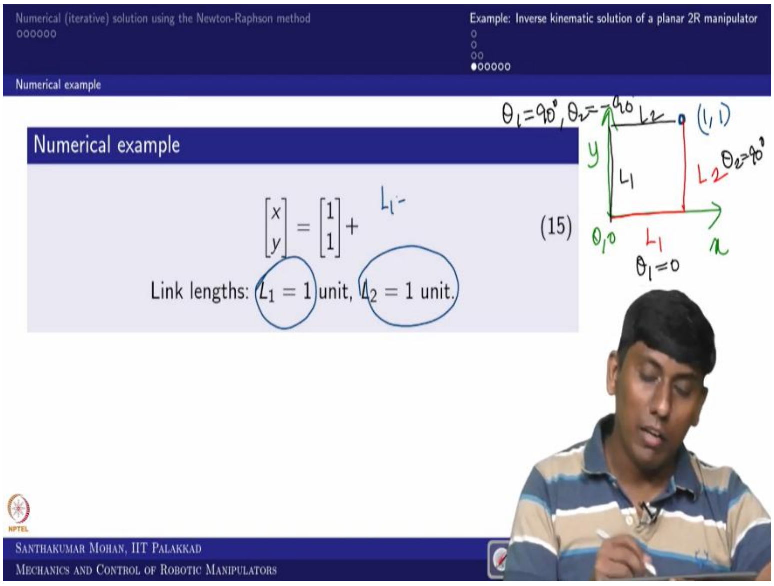
So, you can find. Before that, what one can see, so this is Forward Kinematics solution is known and we have already the closed form solution which are already available. So that is what this just for comparison I am taking this. So, this is what we have already solved in the last class. So, same solution I have put it here.

(Refer Slide Time: 16:29)



So, now you can see like what we need to know this Jacobian matrix. So, this is we have obtained, and this is the algorithm which is we have formed. So, now we are starting with the initial guess and based on the initial guess, we can find the Jacobian inverse and this delta x and delta y, then we can go ahead.

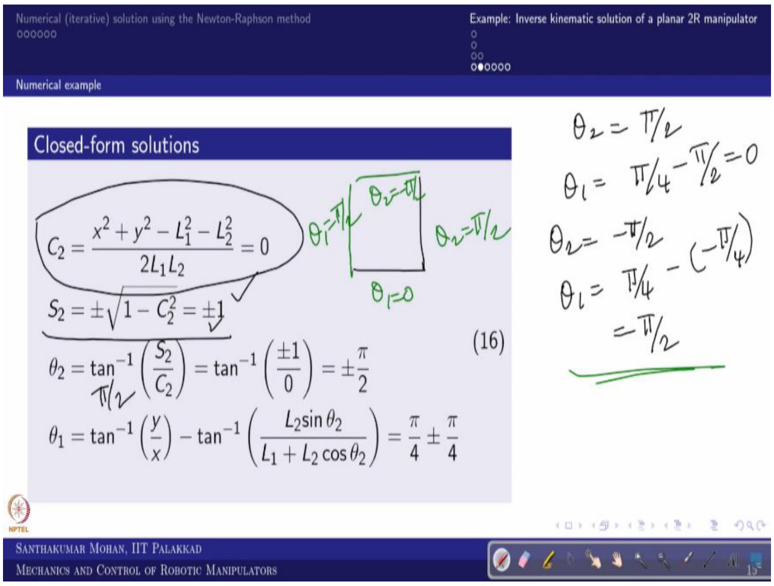
(Refer Slide Time: 16:52)

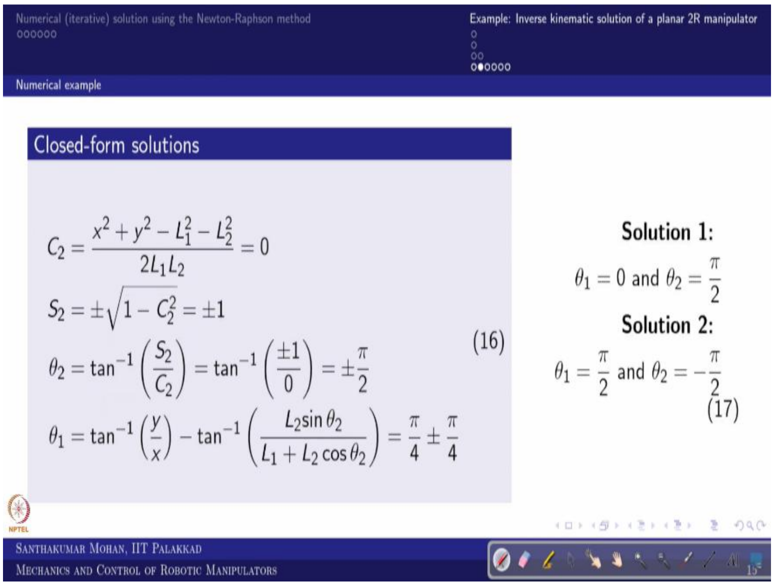


So, then you can see we are taking one of the numerical examples. So, where x and y are 1 and 1, in the sense you take this is actual your x axis and y axis and this is 0, 0. So what you have obtained? So, this is 1, 1. In addition to that what L 1 is 1 unit and L 2 also 1 unit. So, in that sense, what is the best possible solution?

So, you can easily see the possible solution is, so this is 1 unit, and this is 1 unit. So, L 1 and L 2 you can fix it or what you can do, so you can fix it this way. In the sense, theta 1 is 0 and theta 2 is 90 degrees is one solution and the other solution is, theta 1 is 90 degree and theta 2 is minus 90 degree is second solution. So, these are the two solutions, which we have obtained. So, in that sense you can see like, can we find the same solution in numerical example.

(Refer Slide Time: 17:57)





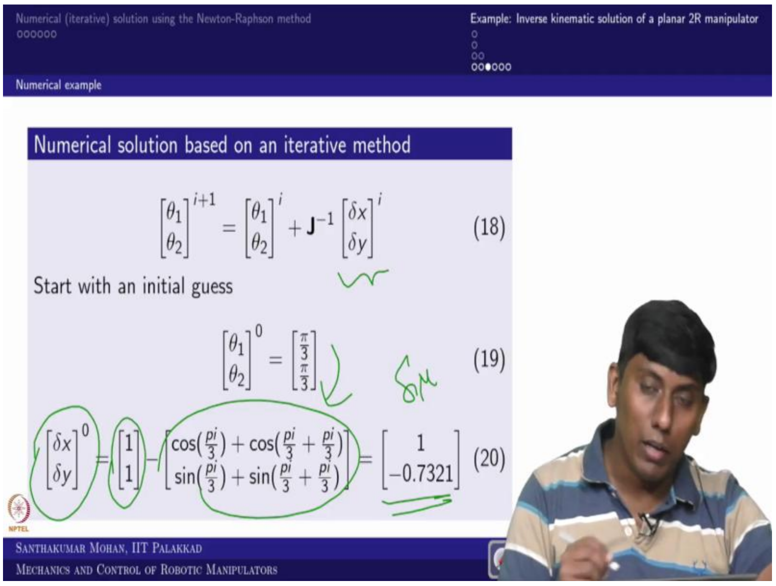
So, for that, first we are substituting this into the closed form solution, which we know. So, you substitute L 1 and L 2 and x and y, all are one. So, in the sense this would give 0. So, what is S 2? S 2 like is plus or minus square root of one minus C 2 square, so that would give plus or minus 1.

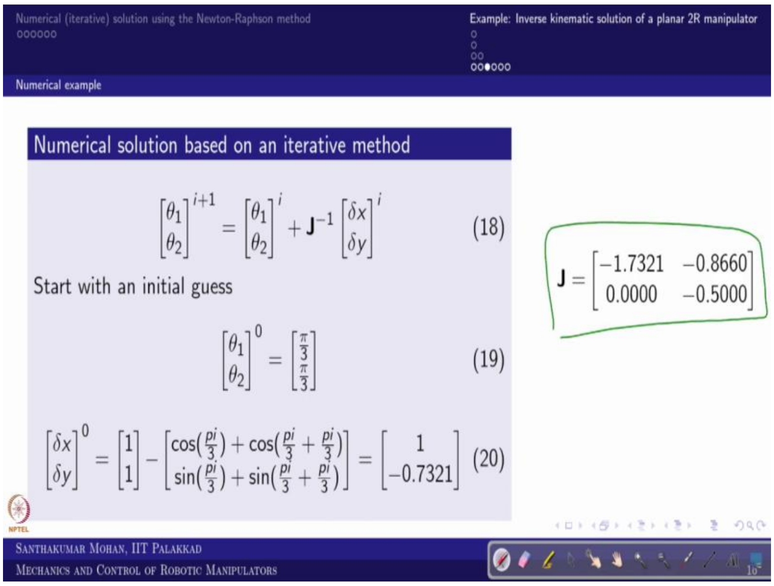
If you take plus 1, that will give you one solution. If you take minus 1, that will give another solution. If you take plus 1, what happens this is 1 by 0, so in the sense that it is infinity, so this will give pi by 2. So, what it is, you take plus 1, so theta 2 would be pi by 2, so then theta 1 would be, so tan inverse of 1, this is pi by 4 and this what happens, this become 90, so L 2 would be there and here L 1 would be there and this would be 0.

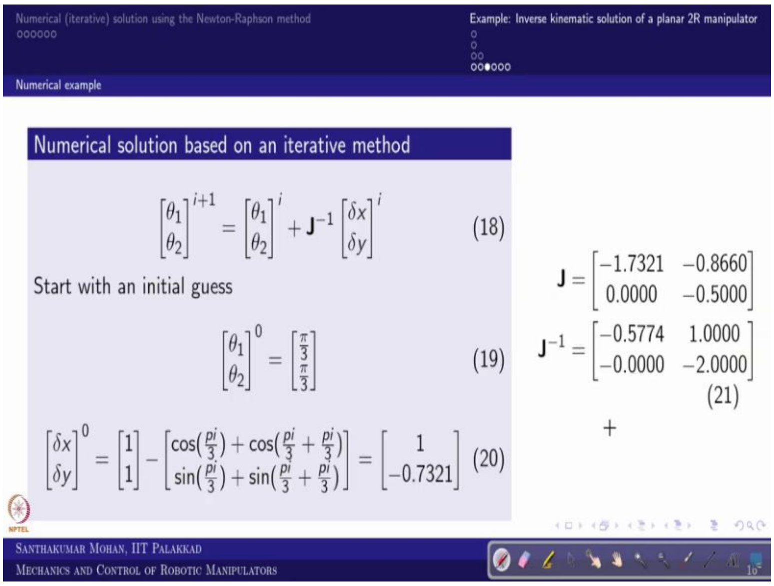
So, in the sense what we get, 1 by 1, so in the sense minus pi by 4. In the sense theta 1 is 0. If we take it as minus 1, so then theta 2 would be minus pi by 2. If you take this, this is minus 1. So, in the sense it is actual like minus pi by 4 and theta 1 is, so pi by 4 minus of minus pi by 4, so in the sense it is giving, so pi by 2. So, the same solution what we have obtained, this is one solution. So, in the sense, theta 1 is 0 and theta 2 is pi by 2. So, the other solution is this.

So, theta 1 is pi by 2 and theta 2 is minus pi by 2. So, both the solution we have obtained. So now we can see the same thing can be obtainable in the numerical basis. So, what that means? So, you can try to apply the algorithm which we have done, so that we are trying to see. So, these are the two solutions.

(Refer Slide Time: 19:53)



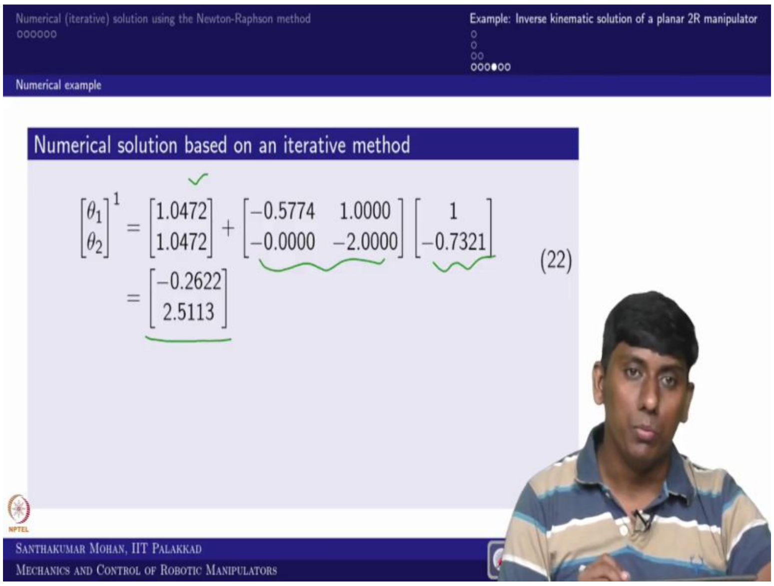


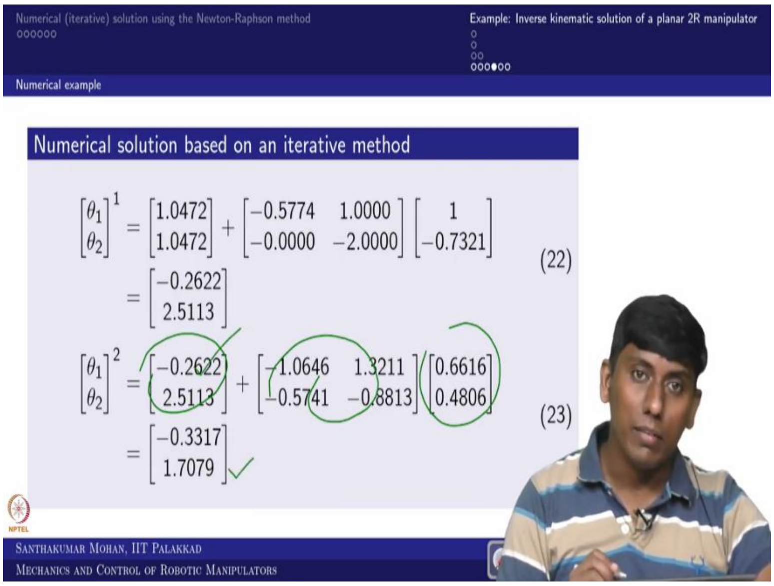


So, we are taking into this case. So now what you can see? So, you are taking an initial guess. Initial guess can be anything, so what can be initial guess? Initial guess I am taking as 60 degree each, so if I substitute this, what I will get? So, I will get delta x something because, so 1 and 1 is given, this is the mu actual based on this initial guess.

So, now this is the difference in the delta mu, this is equal to delta mu, this is the value and J inverse I need to find. So, I substitute that value into you can say here, so the J value is obtained. So, then I can do J inverse. So, then what I can do, so since J and J inverse are there, I can go again with the algorithm.

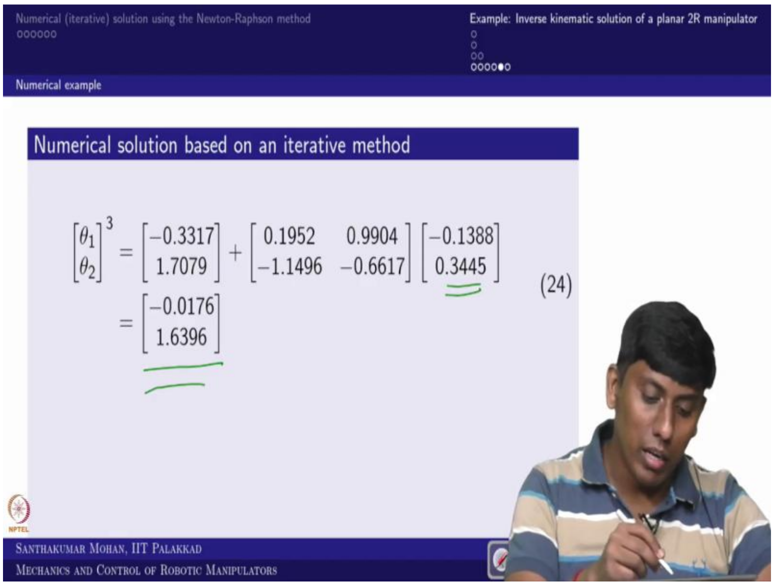
(Refer Slide Time: 20:42)

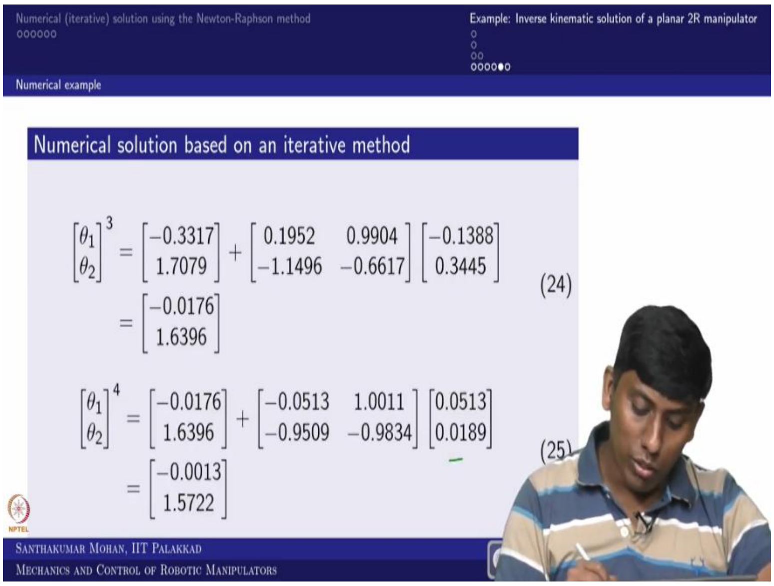




So, I can substitute the J inverse and the delta mu is obtained, initial guess is there. So, I will get mu you can say q i. So, now based on this again I will calculate delta mu, the J inverse, this delta mu and the initial guess and then I will get.

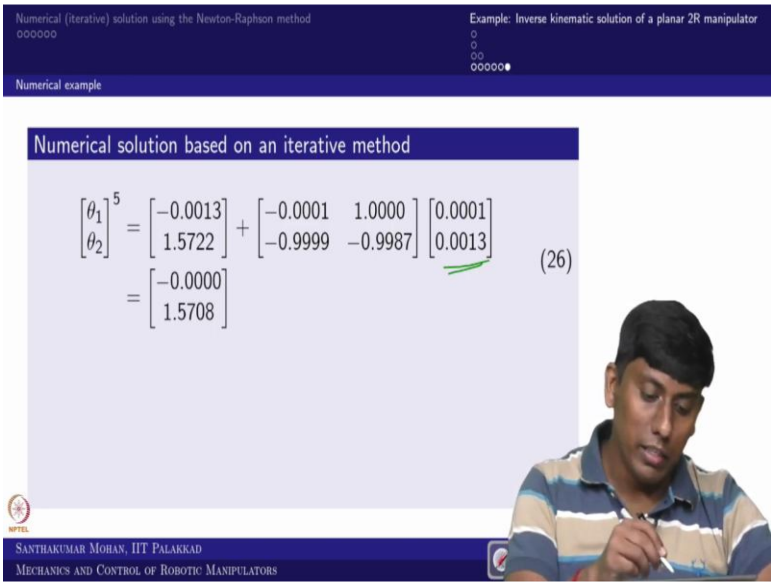
(Refer Slide Time: 21:03)

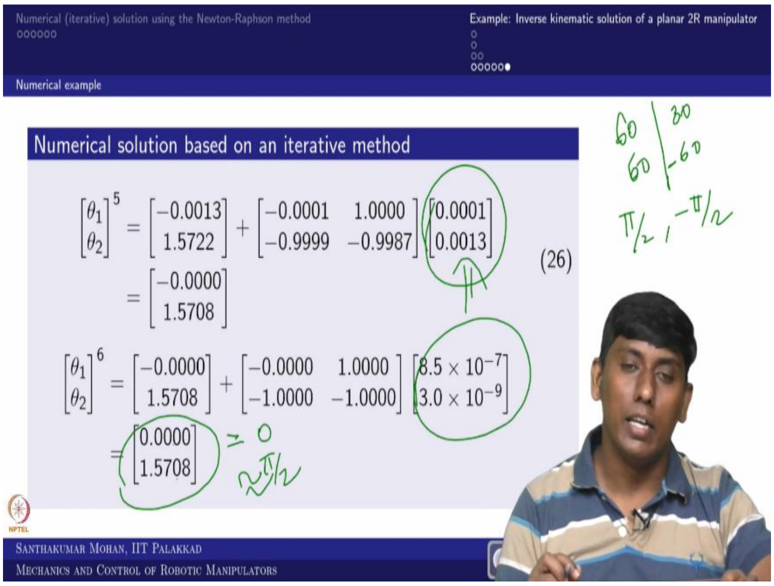




So, like that if keep on going it, so what you can see it is like coming closer and closer. You can see these values getting closer and closer in the sense getting reduced.

(Refer Slide Time: 21:13)





And you can see one point of time it is actually coming, so 1 e power minus 3 ratio and mu you can see this is coming, so 10 power minus 9 in the order. So now you can see this is equal, so this is 0 and pi by 2. So, pi by 2 in radians if you write it is 1.57 something. So, in the sense this is equal.

So, now we have obtained whatever we have derived. So, now if you are taking instead of initial guess as 60 degree and 60 degrees, if you are taking probably something like 30 and 60, the iterative may be taking time. If you are taking 30 and minus 60, it may be, it may be ended with the next solution. What is the other solution? It would be pi by 2 and minus pi by 2.

So, only one condition I am putting here, so whenever you are substituting, you are substituting the common unit. So, you substitute in the radians, because your Jacobian matrix would be in the order different and you are giving your position values in metre, better you use in radians everywhere. So now these values are in radians, so you can convert that to degrees whenever you require.

So, with that we will end up with the Newton-Raphson method. So, now the same method can be applied to any serial or parallel manipulator, but this course is restricted to serial, so you can apply for the serial manipulator. So, even you can go for 6 DoF system or 3 DoF a system, it can be RPR or RRR or RPP, whatever manipulator you have, you can apply.

So, that is one benefit but one disadvantage again and again saying that, if you are taking a point which is inflection, that may end up with the local region itself, I will show you in the numerical example. So, with that I am ending this particular lecture, so see you with numerical example with MATLAB in upcoming classes. So, until then, see you. Thank you. Bye.