Mechanics and Control of Robotic Manipulators
Professor. Santhakumar Mohan
Indian Institute of Technology Palakkad
Lecture 17
Forward Kinematic Solution Using MATLAB

Welcome back to Mechanics and Control of Robotic Manipulator. So, last few classes we have seen forward kinematics and inverse kinematics. We have seen several examples on the you can say slide. So, but I told like these things we can do it even in MATLAB base, so this particular lecture is going to talk about forward kinematic solution, finding using MATLAB symbolic toolbox.

So, I am not promoting MATLAB, but we can even do it several other softwares but this particular course, MATLAB we are going to use. So, where the symbolic math toolbox we would be adapting, so where we can write you can say the DH matrix, what you call arm matrix and you give the DH table, then it automatically generates with the help of our own code it is not like robotic toolbox.

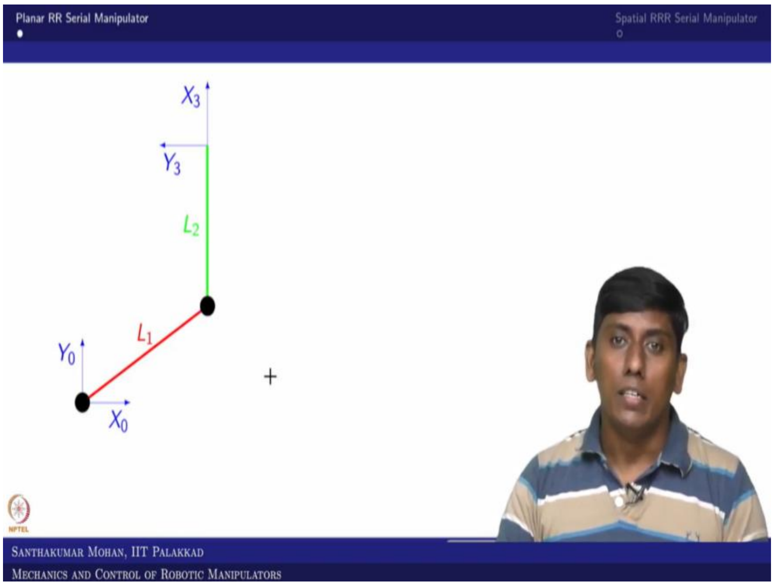
So, robotic toolbox will also do the similar sense but I want you to understand the behavior so you would be using basically like simple symbolic math code, and we would be writing our own MATLAB code. So, in the sense that if you give a DH table and the arm matrix, it would be finding the entire you can say forward kinematic solution along with what you call position vector and individual normal sliding and approach vectors and all.

So, this is the whole idea, if you get this then even, we can use the same MATLAB for solving the equation. Then we can even find the inverse kinematic solution as long the sufficient and necessary conditions all, fulfilling. So, in that sense, what one can see?

(Refer Slide Time: 01:49)

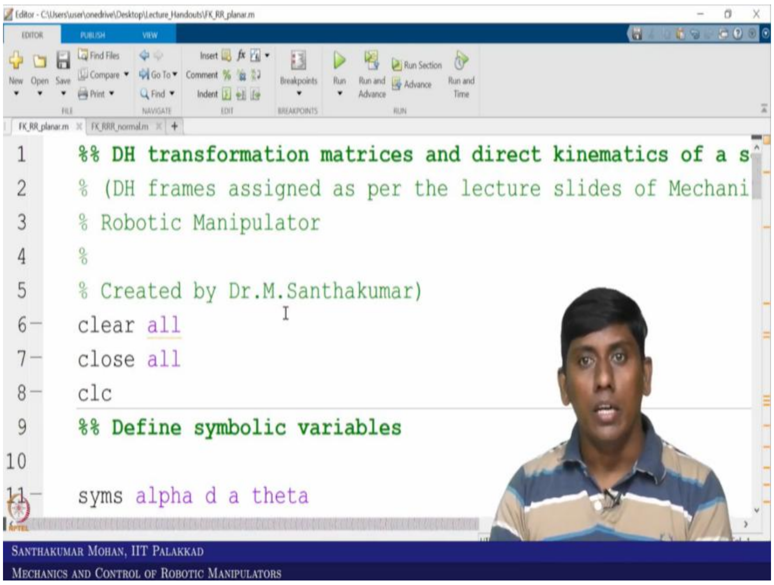
So, we would be taking two such example. So, initially we will start with the planar RR serial manipulator. Then the other one is spatial RRR serial manipulator we will be taking, but as I already mentioned we are going to use MATLAB, symbolic MATLAB toolbox. It is a very small set of you can say tools only we are going to use. Only symbol and you can say substitutions. So, these two only we are going to use.

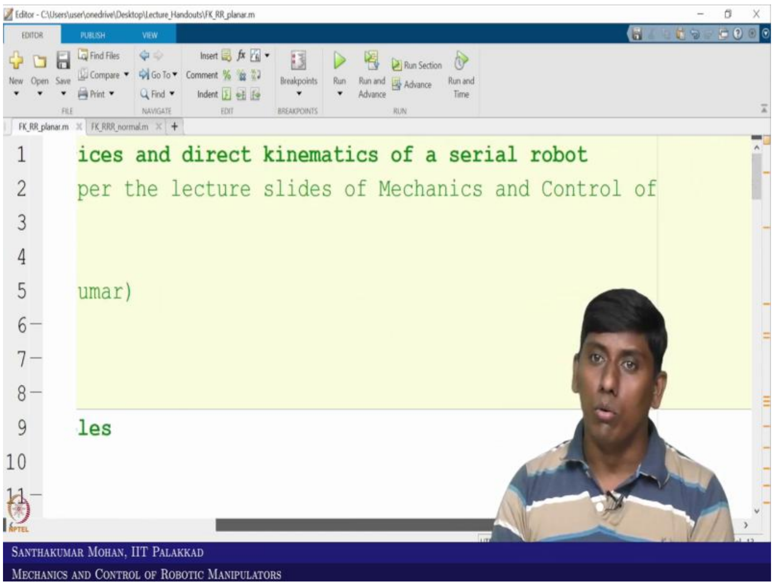
(Refer Slide Time: 02:17)



So, for that what we are taking? We are taking first case. So, we are taking 2R serial manipulator. I hope you remember what is the DH parameters. So, if I recall this, so this would be having alpha 0 is 0 and a 0 is again 0, but when you come to alpha 1, that is also 0 and a 1 would be as equal to what you call L 1. So, similar way you can come to alpha 2, alpha 2 0 and a 2 is L 2 and the other side if you come, so this theta 1 and this is theta 2. So, that is what we get. So, we will go to MATLAB window and then we will see.

(Refer Slide Time: 02:58)

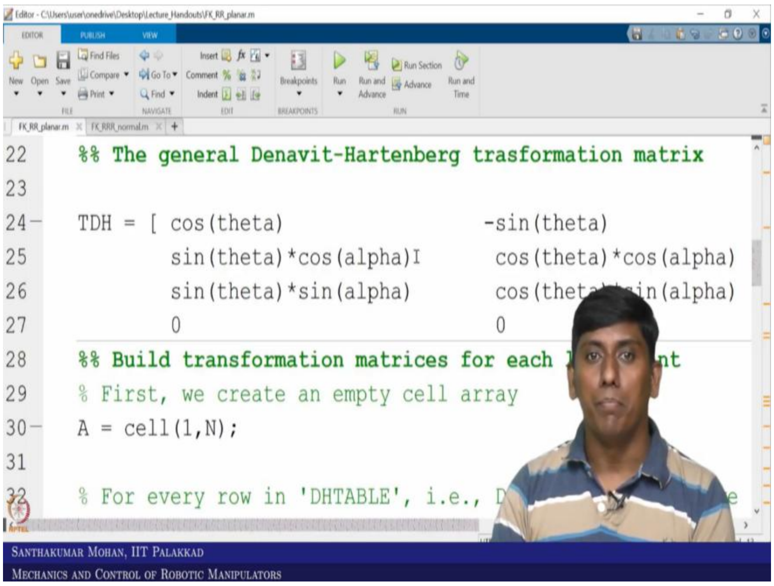


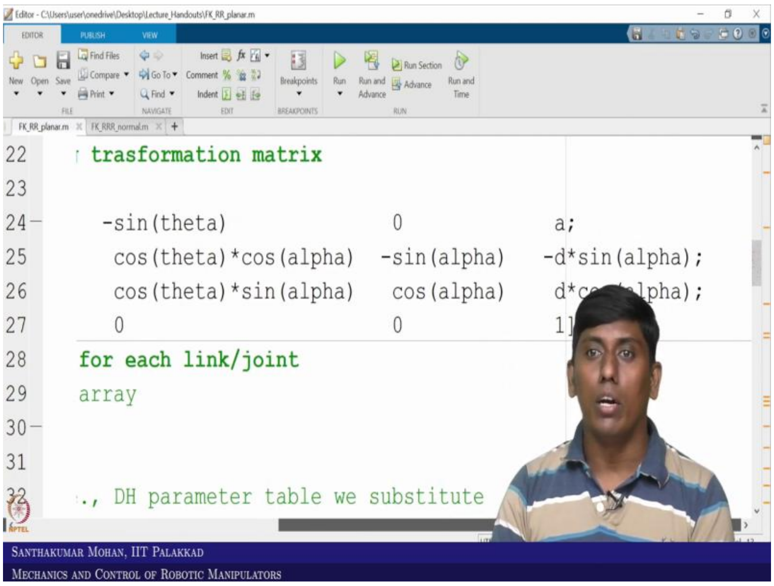


I have already created a; you can say; a file for you. I will be explaining how I have generated. So, this is DH transformation matrices and direct kinematics of a serial robot. This is the general code. The same code can be used for any serial, robotic manipulator. That is the whole idea.

So, here, the notation we are going to use are the frame assignment, all we have used based on the course which we have covered like mechanics and control of robotic manipulator. What that specific class I am putting here? Because this particular matrix what we are going to use as a DH matrix is the non-standard one.

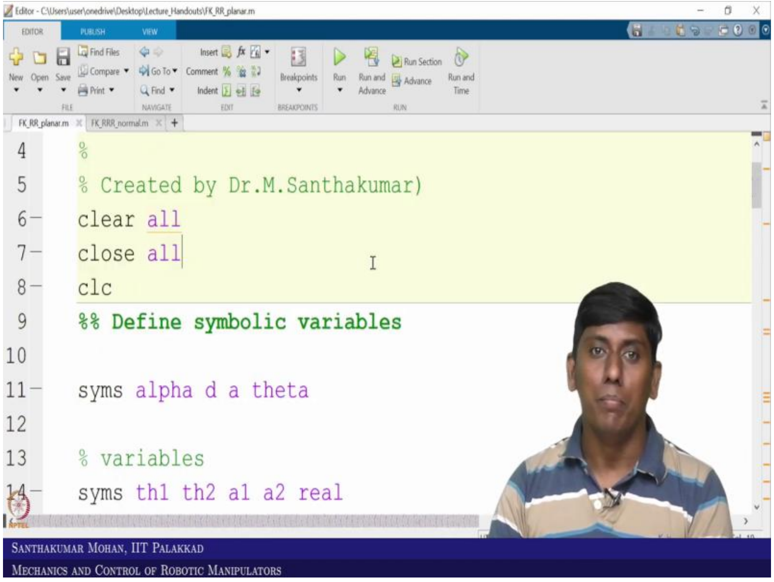
(Refer Slide Time: 03:40)

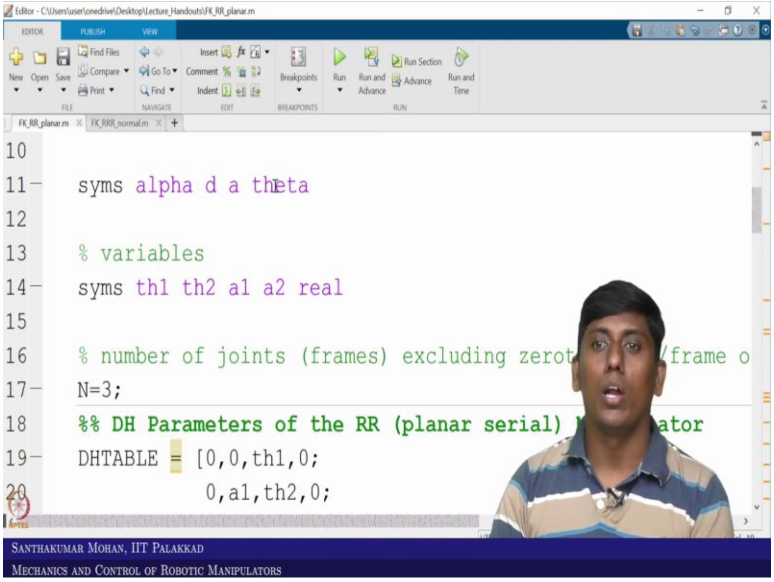




So, of you want to use the same thing for the standard one, the, the arm matrix what I have written here, so this need to be modified. So, what that means? So, here I have written as mix of alpha i minus 1, a i minus 1 and theta i and d i although I did not label it here, but that is the idea which change. So, the general Denavit-Hartenberg transformation matrix here we use as the non-standard one.

(Refer Slide Time: 04:04)

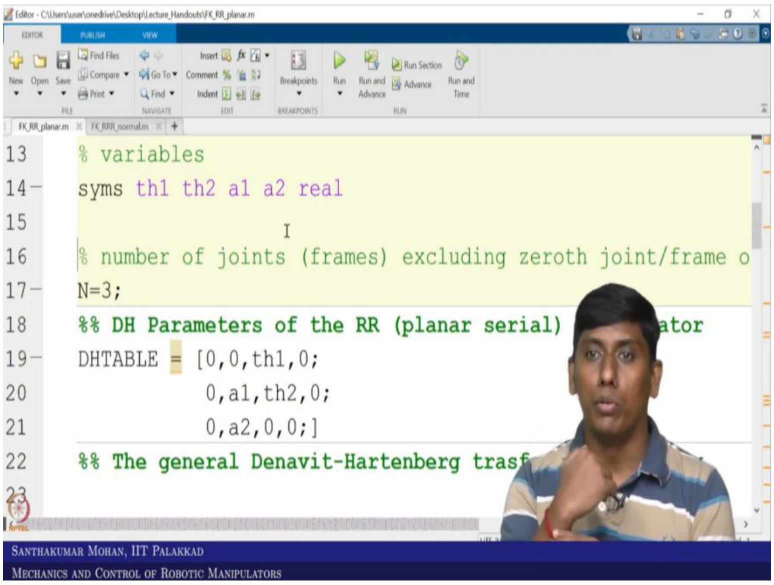




Let us go to the code what exactly it is written, so the initial code is very same. So, clearing the memory or you can say clearing the workspace and closing all the; you can say window which is figure window is opened and clearing the command history. Then we are creating four symbols mainly for denoting the kinematic parameters, what you call alpha i minus 1, a i minus 1, d i and theta i, so which are I am writing without any you call suffix, in the sense without any frame arrangement.

We are writing a, alpha, theta and d, that is what we have written here. You can see alpha, d, a, theta. So, in addition to that you know, so based on the DH table you know like there would be a 1, a 2, theta 1 and theta 2 would be coming in this case, but what we know additionally these all are real variables. So, initially I did not mention it is a real or, you can say complex.

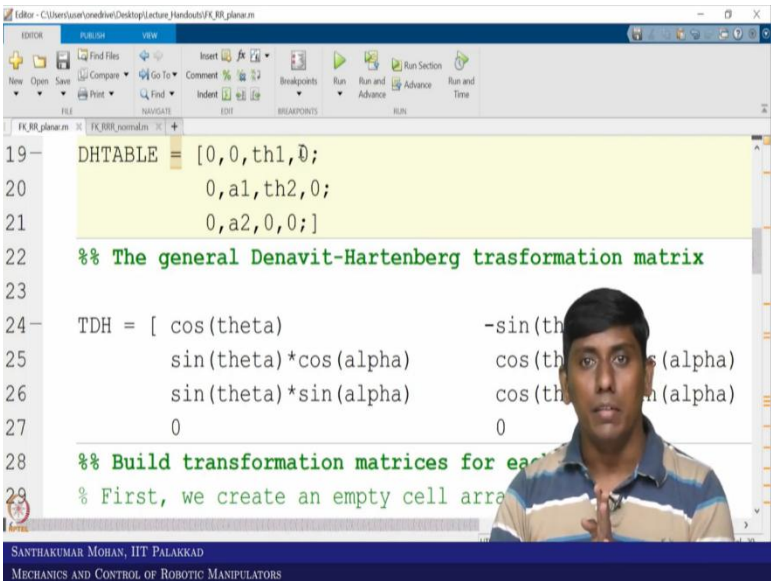
(Refer Slide Time: 04:59)



But here so whatever you have got, so theta 1, theta 2 and a1 and a2 all I am writing as what you call, so the symbols which are real symbols. So, now in that case, what we can see? So, how many frames we have used? So, there are four frames because it is a two axes robot, so one end-effector frame and zeroth frame.

So, when we are writing the number of transformation matrix, we usually ignore the 0, because we do not have any transformation before that, in the sense transformation matrix of 0 with respect to minus 1, we normally will not do it. So, that is why you can see number of joints, in the sense number of frames excluding zeroth joint or frame is number here is 3. Because 2 axes and 1 end-effector point. So, in the sense N equal to 3. So, now what we are writing?

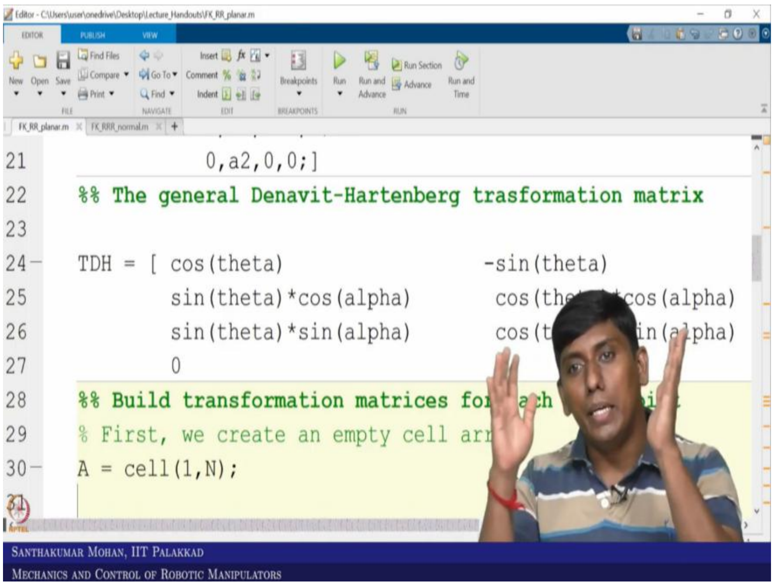
(Refer Slide Time: 05:48)

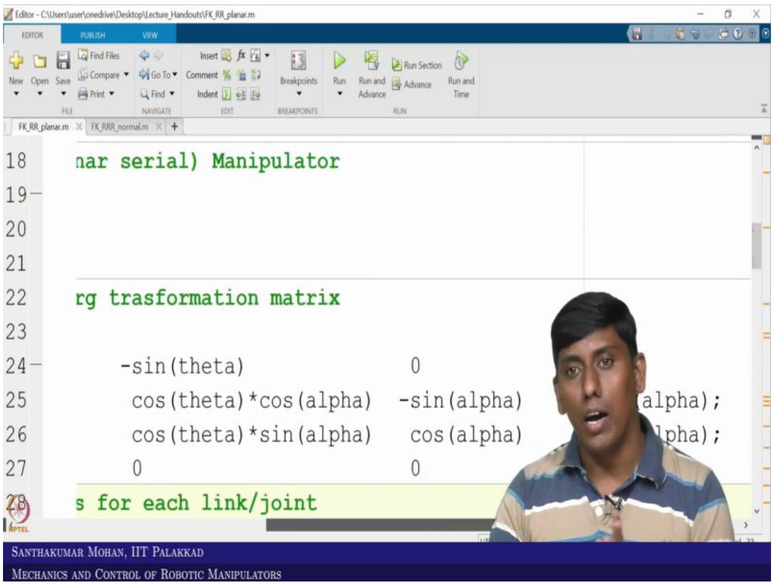


So, DH table we are writing, so the DH table would be having four you can say columns, in the case N is 3, so three rows. So, in the sense what you can see, like this is alpha a i minus 1, so you can alpha i minus 1 in the sense all alpha 0, because it is in planar and a 0 is 0 and a 1 is as a 1 and a 2 as a 2 and theta 1 is there which is active variable, theta 2 is another active variable and theta 3 is, it is passive, so it is 0 in this case.

So, similarly, when you come to what you call the final d, so there is no translation along Z 1, Z 2, Z 3 so all these are zeroes. So, this is what, this is equivalent to alpha, this is equivalent to a, this is equivalent to theta, this is equivalent d. So, in this DH table, it is everything we have written. So, now what we are doing? We are bringing the non-standard what you call transformation matrix in the sense arm matrix we are bringing it.

(Refer Slide Time: 06:48)

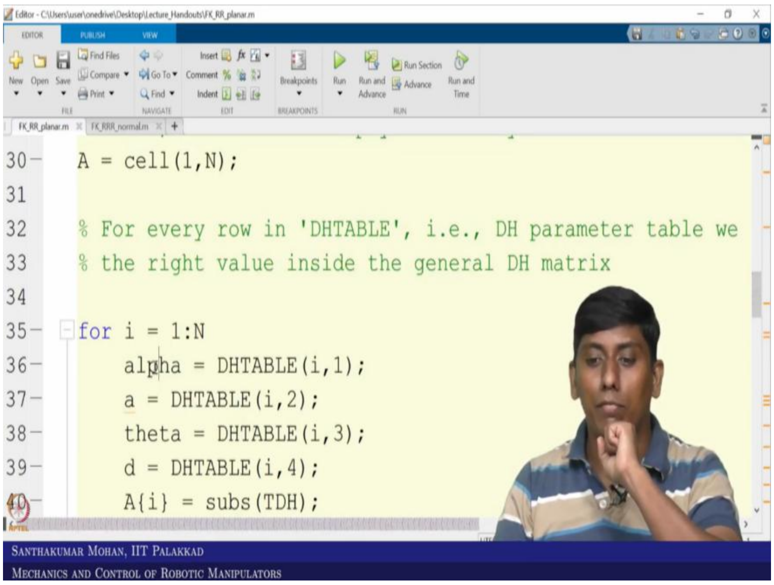


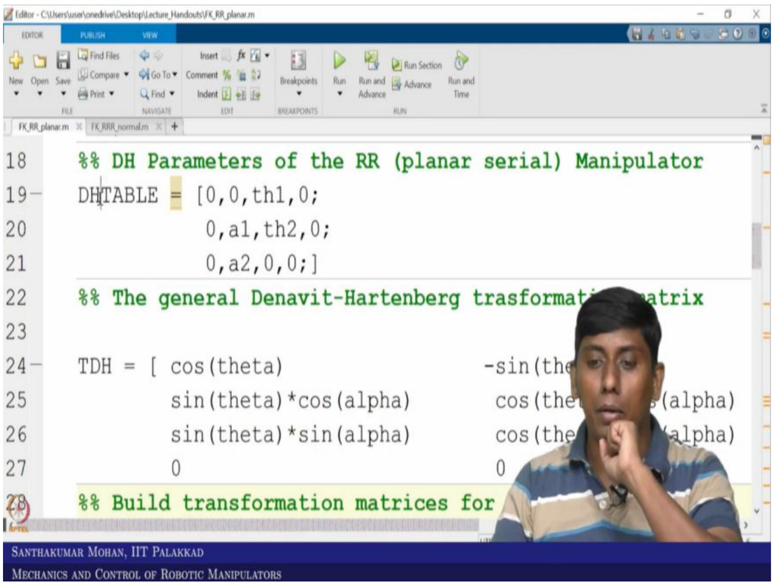


So, now this is the non-transformation matrix. So, you can see it here, so this is non-standard matrix. So, we have used this, so now what we can do it? So, we are getting the one simple briefcase. What that mean? So, we are creating a bigger stack or rack, so this rack would be consisted of N cells, so here N is 3. So, each cell would represent individual transformation matrix. Now in that sense what we are trying to create?

We are trying to create a cell of A, which is having 3 cells inside. So, what that means? So, the first cell corresponding to transformation matrix 1 with respect to 0, second cell would be corresponding to transformation matrix of 2 with respect to 1 and the third cell will be corresponded to transformation matrix of 3 with respect to 2. So, in the sense we are making a rack, so that rack formation only we are calling as a cell 1, N here N is 3. So, we are making it, so number of columns is 3, in the sense it is horizontal the rack 1, 2, 3 like that it is going to be there.

(Refer Slide Time: 07:59)





So, now, once you create a cell, what we are trying to do? We are trying to substitute the; you call DH parameter inside the arm matrix, so for that we are taking for loop because there would be three rows, so each row would be corresponding to you can say substitution which end up with corresponding transformation matrix. So, for that we are substituting first alpha, then a, then theta and d.

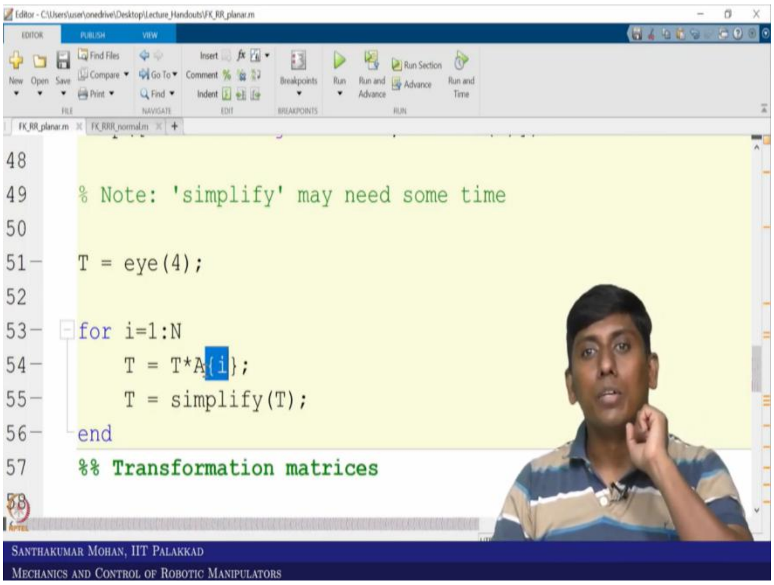
We are substituting into the DH table, so you can recall this is the DH table. So, we are taking the DH table value corresponding to a, corresponding to alpha, corresponding to theta and d.

So, this we are going to substitute into TDH. So, that is what we are going to do. We will see initially we have you can say equate alpha, a, theta and d corresponding to the rows.

So, corresponding row in the one in this case first row, first column, second column, third column, four would be denote alpha a theta and d. Once you get this alpha to d, then you can substitute these variables into TDH, in the sense your arm matrix. Now what it will come? The first cell will fill with the first DH table, first row of DH table, that will give T 0, 1, in the sense T1 of 0. The second iteration it would be second cell that would be T2 of 1. The third cell that would T3 of 2 that we would be getting.

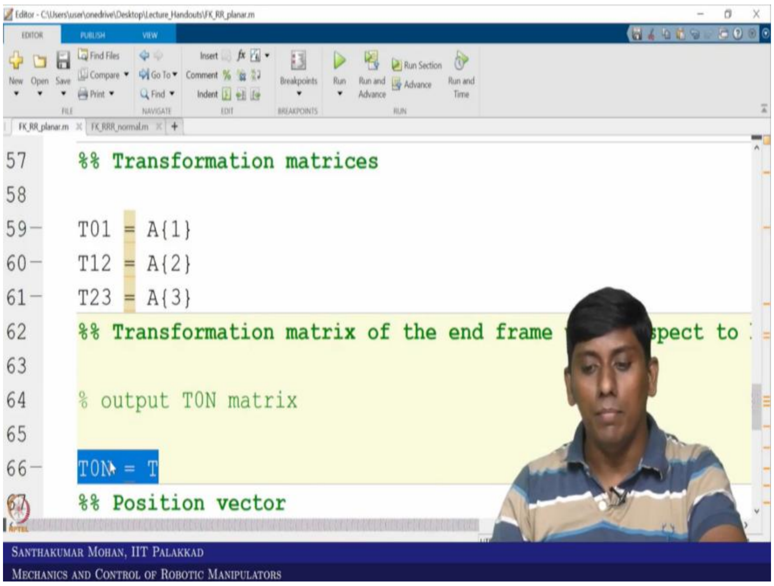
So, once you get what you wanted, so individual transformation matrix is there, then you can post multiply and get the final solution. So, for that we are displaying it, I will show you till the output.

(Refer Slide Time: 09:40)



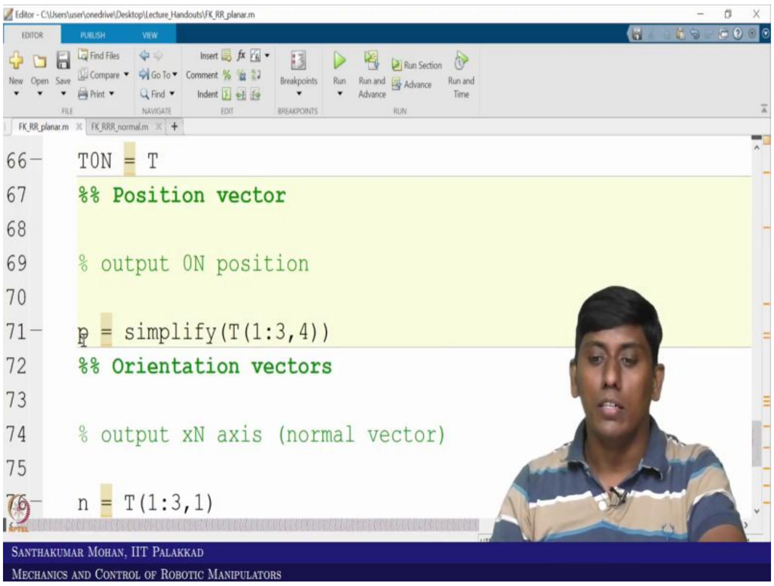
So, we are trying to first start the iteration again, so for that we are taking because we have N steps, so we do not know how it start. So, we are starting T as a simple identity matrix. Now I am post multiplying 1 by 1. So, what I am doing? The first initial I take the identity matrix; the first cell is post multiplied. That I am simplifying, then the second iteration. The T would become A of 1, then that is post multiplied with the A of 2. Then the third iteration that would be post multiplied with the A 3, but that pre multiply with, you can say T 0 to 2, that is what the equation.

(Refer Slide Time: 10:22)



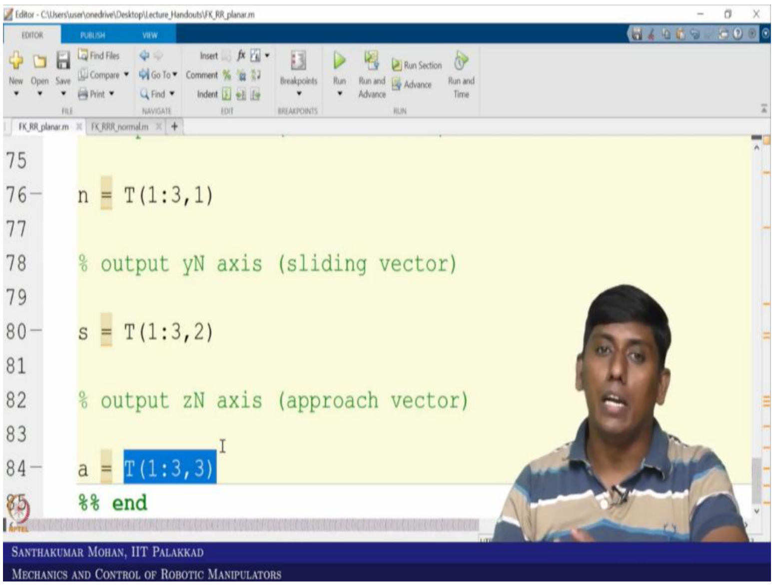
So, in the sense what you can see, so this is the individual matrix and the output is final, you can say 0 to N, the N is here 3.

(Refer Slide Time: 10:32)



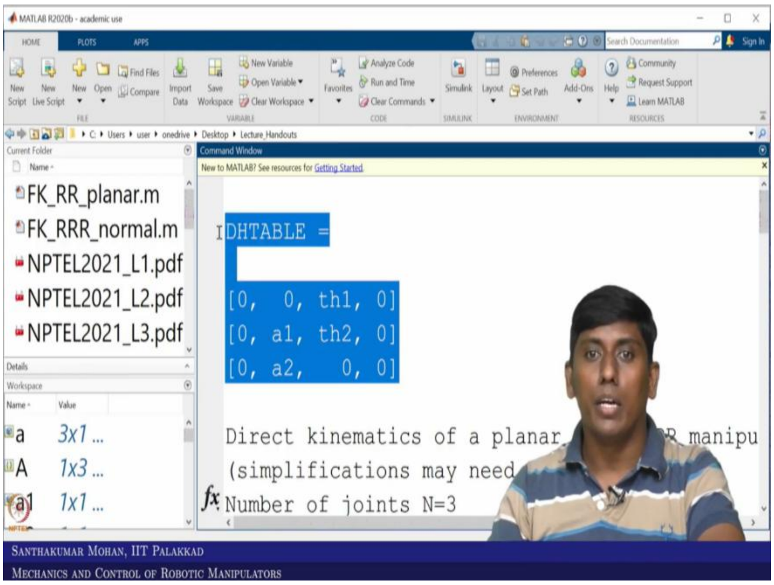
And the last column, first three rows is actually position vector.

(Refer Slide Time: 10:36)



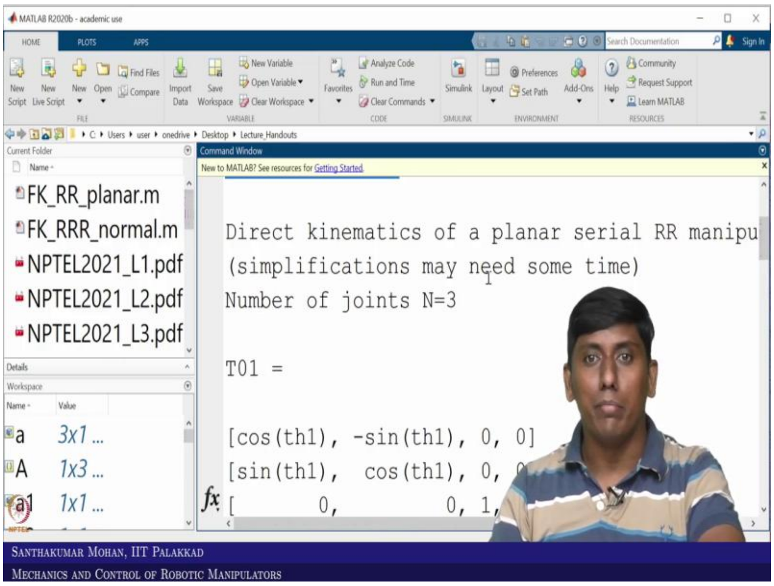
And the first, you can say column would be related to normal vector. Then the second, you can say column related to sliding vector, the third column would be related to approach. So, now if I run this, so what you can get it? So, you will get the output.

(Refer Slide Time: 10:55)



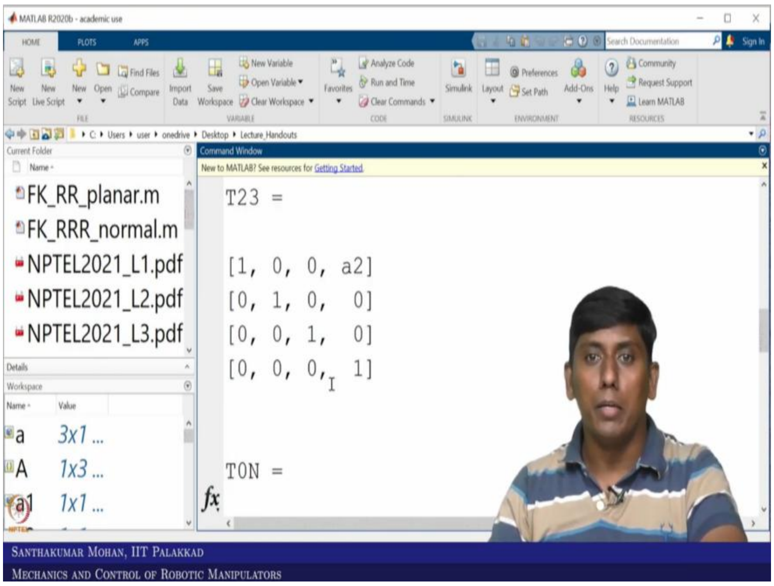
So, you can see here. So, now you can see like what it is given. So, first it gives the DH table.

(Refer Slide Time: 11:04)



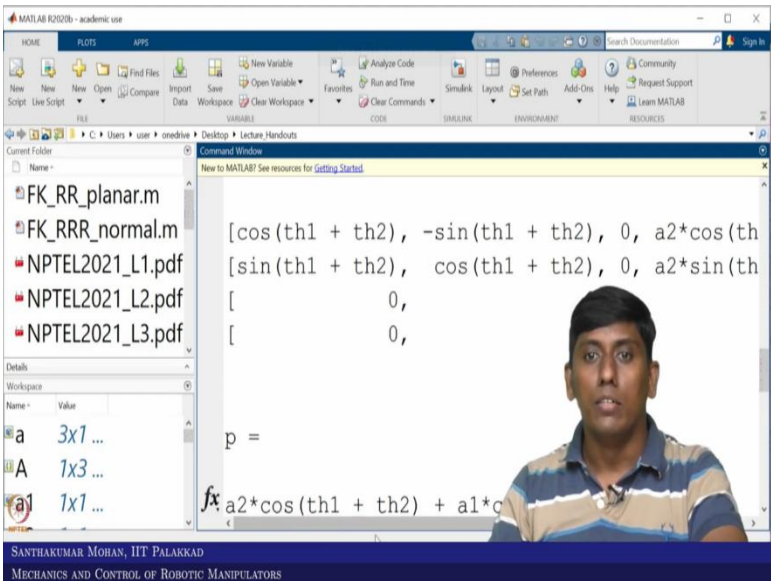
So, then it says that the direct kinematics of a planar serial RR manipulator is happening, and simplification may need some time. The number of joints is here 3.

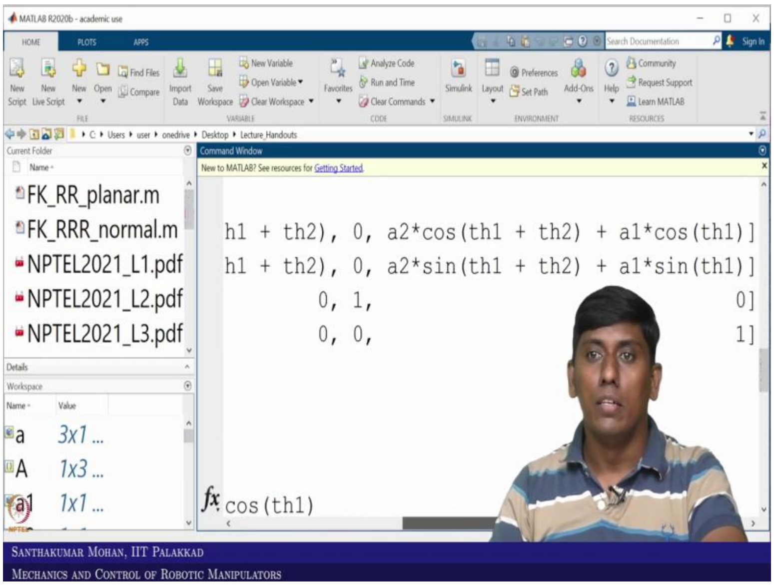
(Refer Slide Time: 11:13)



So, in this sense, you would be expecting three, you can see transformation matrix T 0, 1 then T 1, 2 and T 2, 3.

(Refer Slide Time: 11:26)

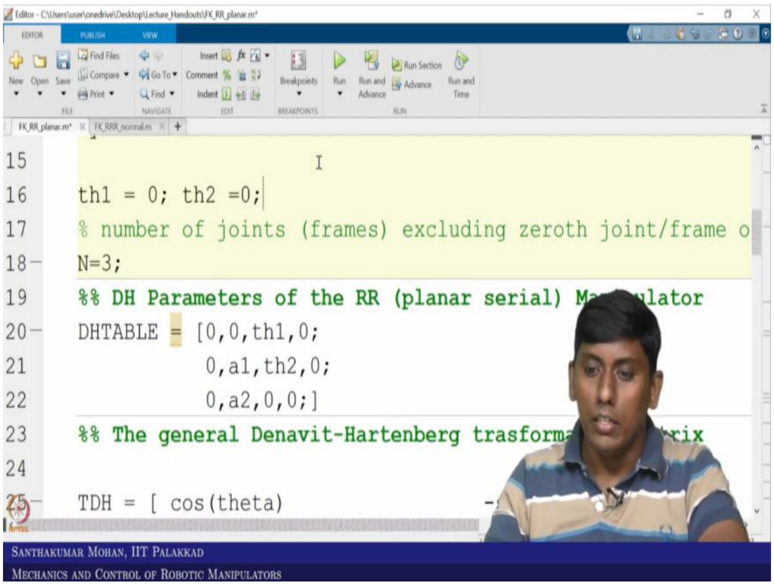




So, now if you post to multiply finally you will get T 0 N. So, which is in this form. So, you can see this is the form. So, you can take individual, so in the sense the position vector. So, the position vector what we used to get, so a 1 cos theta 1 plus a 2 cos theta 1 plus theta 2. So, it is the order is changed, but the output is same.

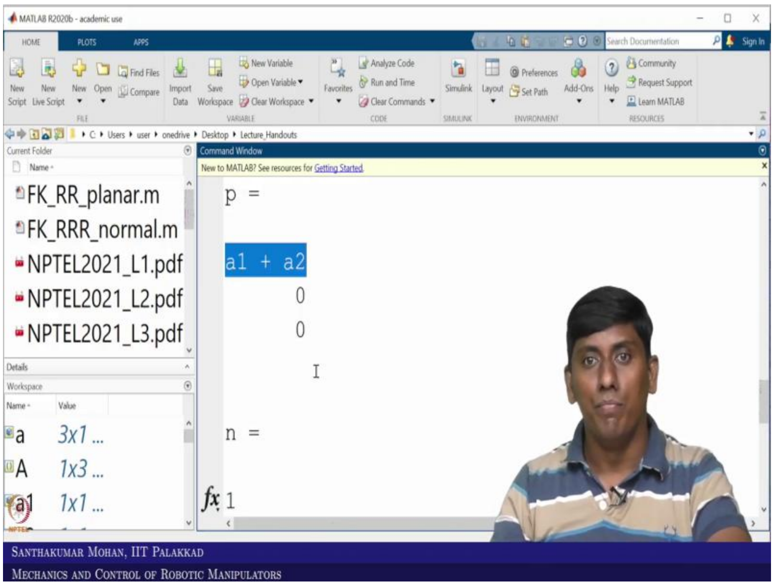
So, in the sense this sine case, that would be you are what you call the Y vector or Y value, this is the X value, this is the Y value end of vector position and this is your normal and this is sliding and this is your approach. Now once you obtain what one supposed to do, so you are to cross check whether the home position is matching. So, for that we take here.

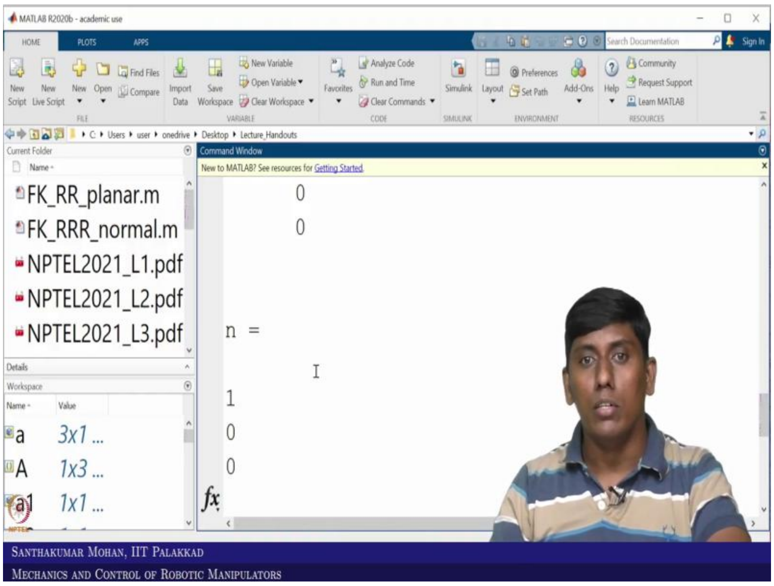
(Refer Slide Time: 12:09)

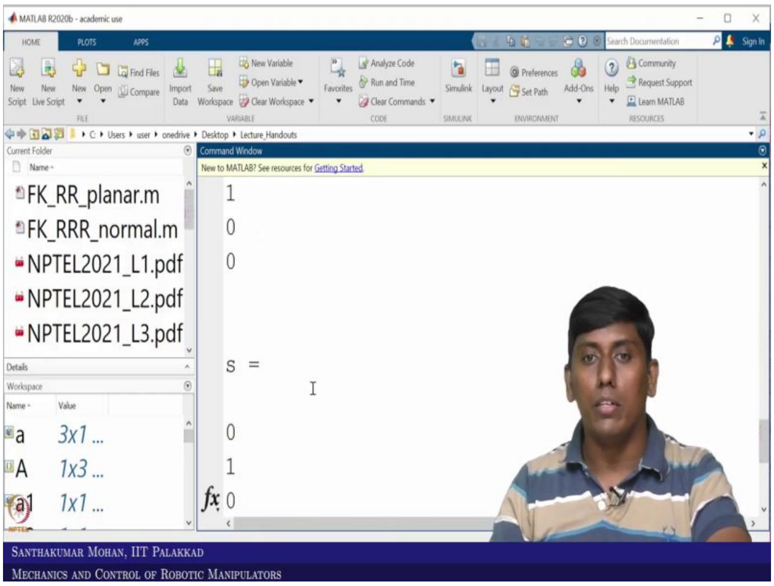


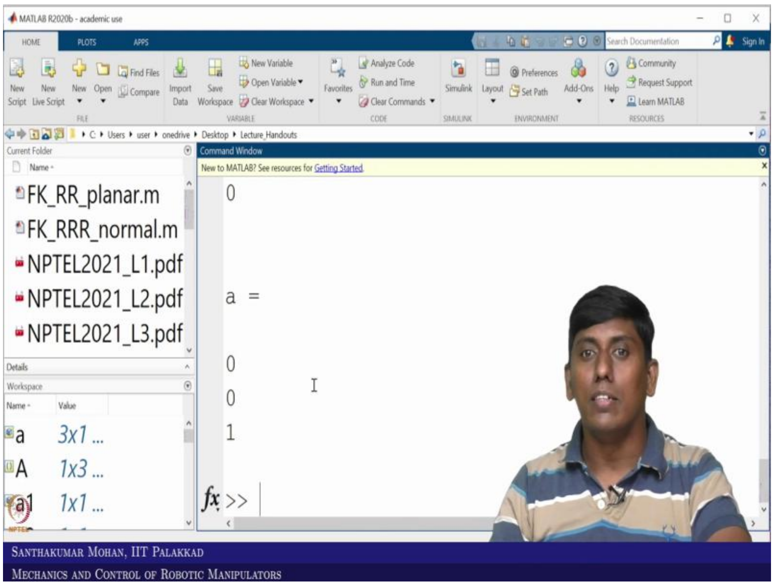
So, what we are trying to do the home position in the sense, the theta 1 and theta 2 we substituted into 0, so in the sense, what we are trying to do? So, I am putting theta 1 equal to 0 and theta 2 equal to 0. So, I am substituting that. So, now if I substitute what you will get? So, it will give the home position, you can find it.

(Refer Slide Time: 12:30)



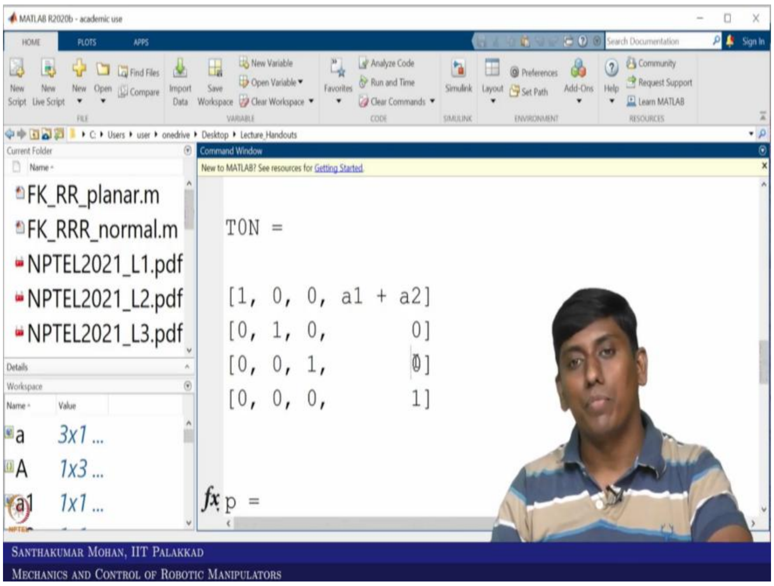

MATLAB R2020b - academic use
Current Folder
FK_RR_planar.m
FK_RRR_normal.m
NPTEL2021_L1.pdf
NPTEL2021_L2.pdf
NPTEL2021_L3.pdf
Workspace
a 3x1 ...
A 1x3 ...
a1 1x1 ...
Command Window
New to MATLAB? See resources for Getting Started.
0
0
n =
1
0
0
Santhakumar Mohan, IIT Palakkad
Mechanics and Control of Robotic Manipulators


MATLAB R2020b - academic use
Current Folder
FK_RR_planar.m
FK_RRR_normal.m
NPTEL2021_L1.pdf
NPTEL2021_L2.pdf
NPTEL2021_L3.pdf
Workspace
a 3x1 ...
A 1x3 ...
a1 1x1 ...
Command Window
New to MATLAB? See resources for Getting Started.
1
0
0
s =
0
1
0
Santhakumar Mohan, IIT Palakkad
Mechanics and Control of Robotic Manipulators


MATLAB R2020b - academic use
Current Folder
FK_RR_planar.m
FK_RRR_normal.m
NPTEL2021_L1.pdf
NPTEL2021_L2.pdf
NPTEL2021_L3.pdf
Workspace
a 3x1 ...
A 1x3 ...
a1 1x1 ...
Command Window
New to MATLAB? See resources for Getting Started.
0
a =
0
0
1
>>
Santhakumar Mohan, IIT Palakkad
Mechanics and Control of Robotic Manipulators

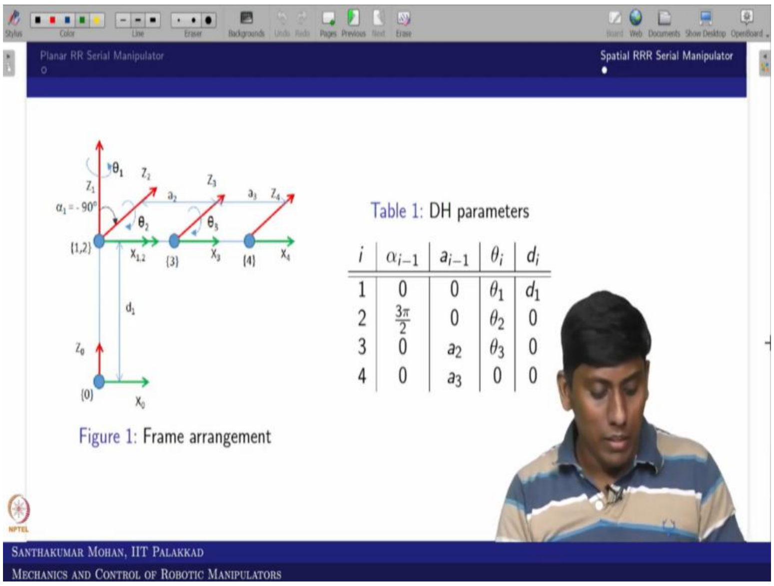
So, now you can see that the X displacement is a 1 plus a 2, that is matching with our previous attempt with straightforward kinematic relation, without you can say MATLAB and the normal would be 1, 0, 0 and slide would 0, 1, 0 and approach in the sense, so you would be getting identity matrix. So, because your Z axis is parallel to your Z 3 axis is parallel to Z 0, your x3 is parallel to X 0, so in the sense you will get the orientation matrix as identity. So, that is what you are getting.

(Refer Slide Time: 13:04)



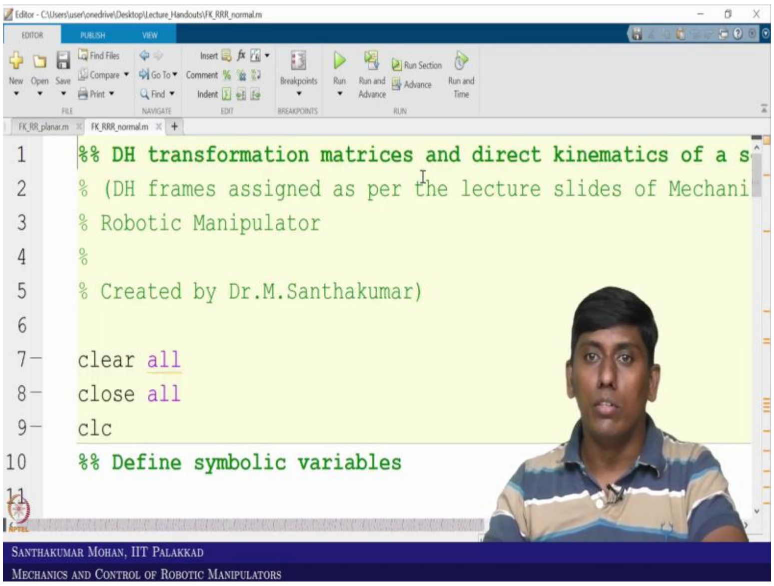
So, for example, you can see, so it is getting identity matrix and the X displacement only there, Y is 0 and Z is again 0. So, now you can see that this is the way you can actually use the MATLAB as one of the aids and we have obtained. So, now the same thing can be used even for other benefits. So, for example, I have already written a code for RRR manipulator. So, before that, I will come back to this slide set. So, you can see like what we can see.

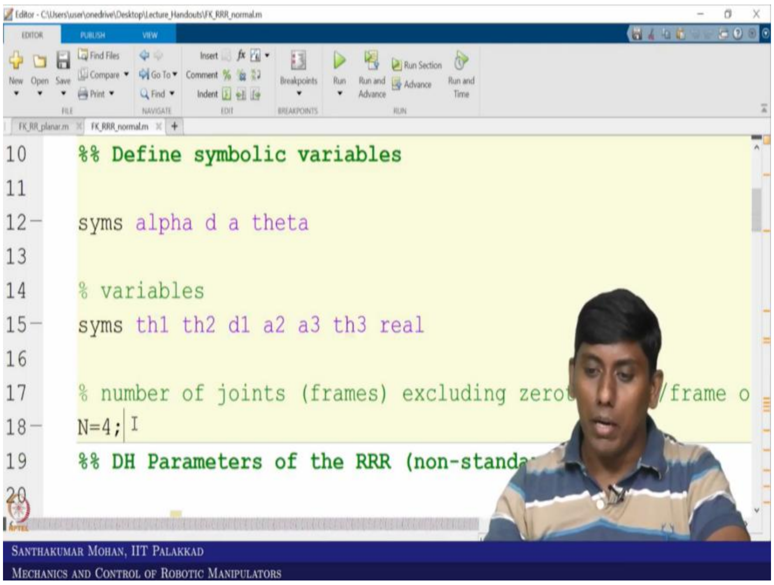
(Refer Slide Time: 13:34)



So, you can actually like find this as the RRR manipulator. So, we have already found the DH table. So, this DH table if I put it inside the, what you call the new code which I have written. So, I will modify the N access 4 and I replaced this DH table. So, I will get the forward kinematics solution. Can I check that? Yes, I will check this.

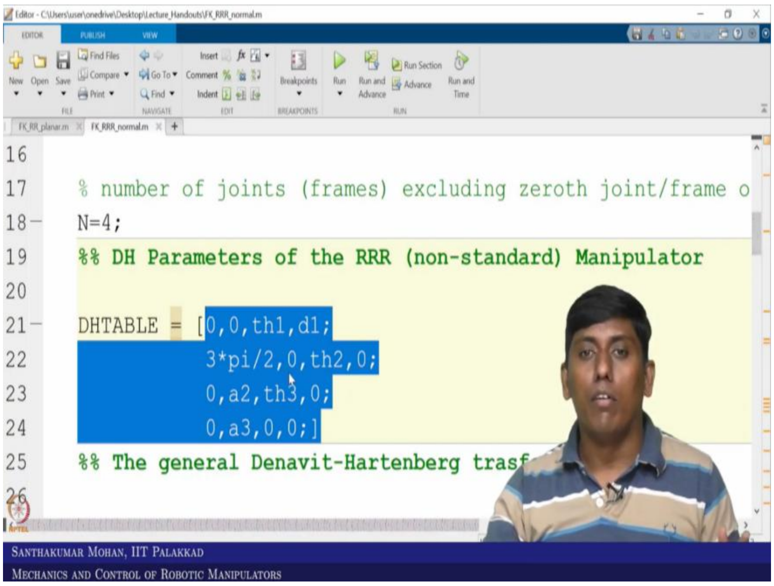
(Refer Slide Time: 14:04)

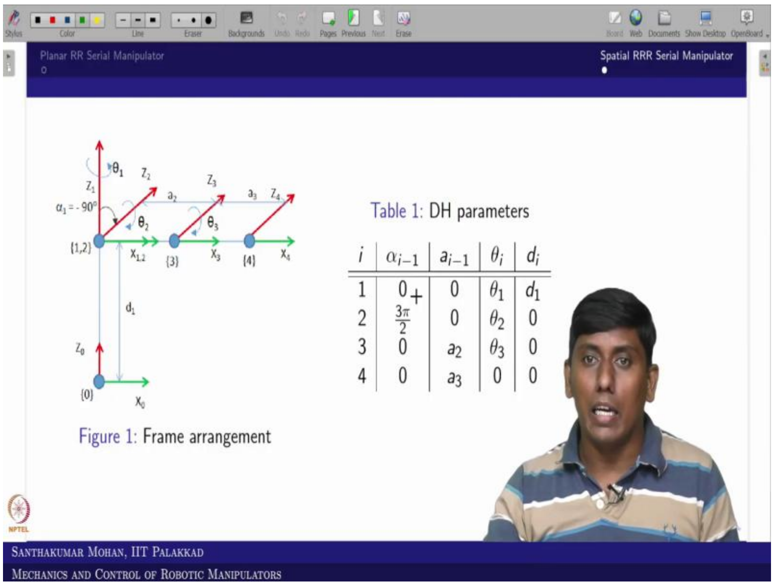




So, you can see now what I did, I have modified this. So, now this is written the same thing, but what I am trying to do? So, here you can see like, there is a 1 a 2 is replaced with a 2 and a 3 because your DH table does not have a 1 but a 3 is there. Similarly, you can see that there is displacement on the Z axis. So, d1 is there and here again, it is 3 axis system. So, theta 3 also there. So, these all are real variable. So, now the number of joints is 4.

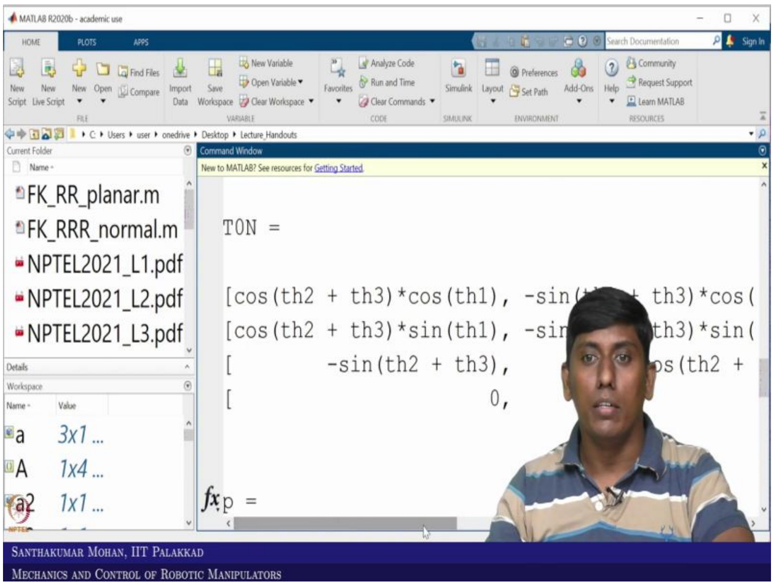
(Refer Slide Time: 14:34)

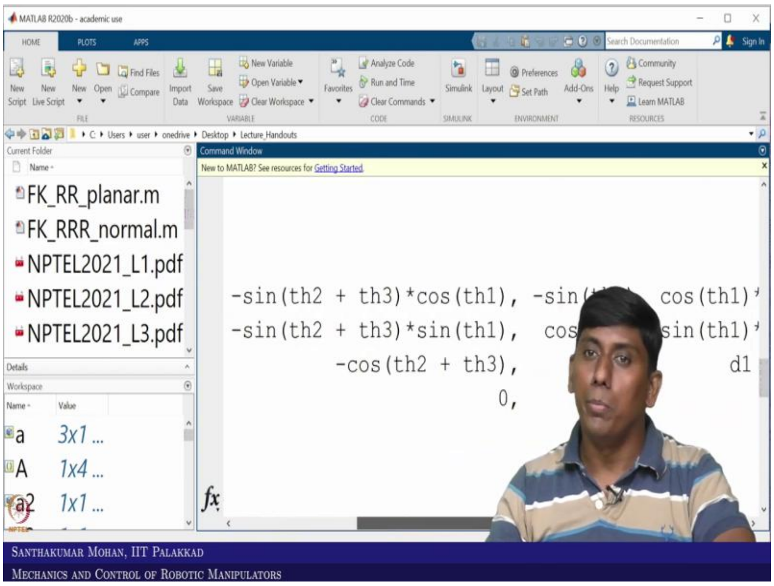


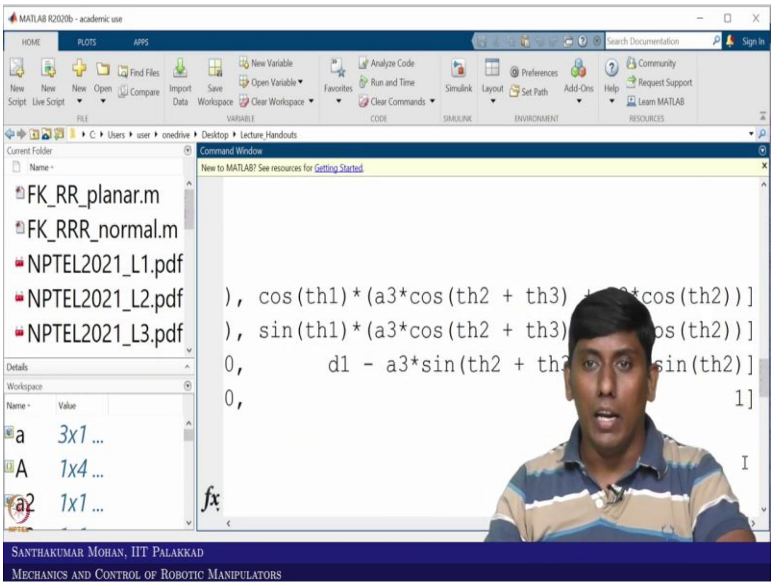


So, now the DH table whatever we have seen in the previous slide. So, this is directly I substituted here. So, now if I run this, what I will get? So, I will get transformation matrix individual and then finally I will get the final position vector, orientation matrix. So, can I run this? So, I can run now.

(Refer Slide Time: 15:04)

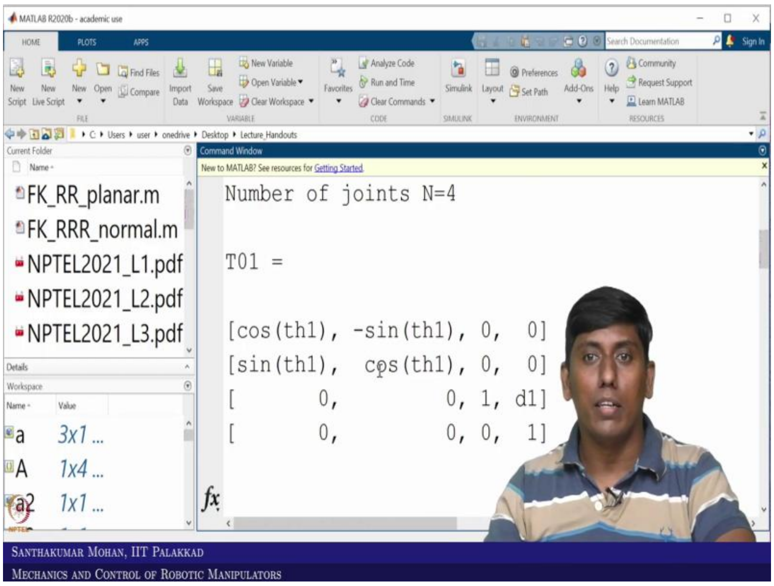


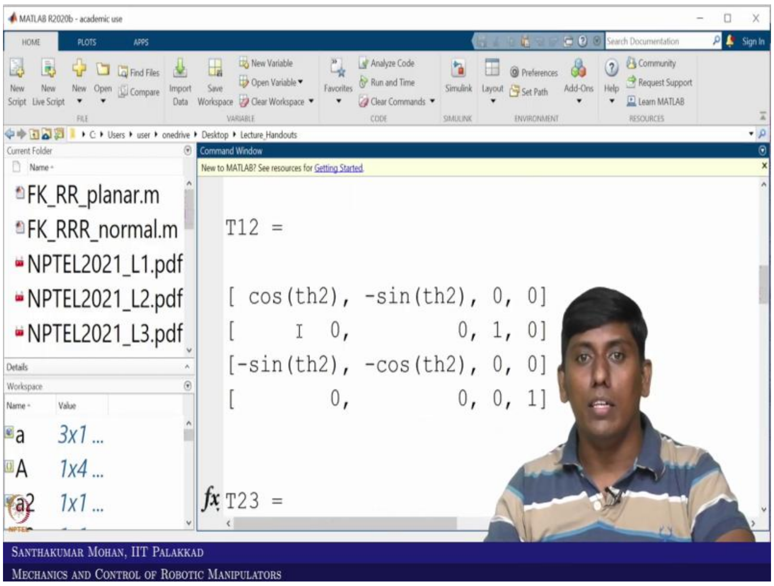


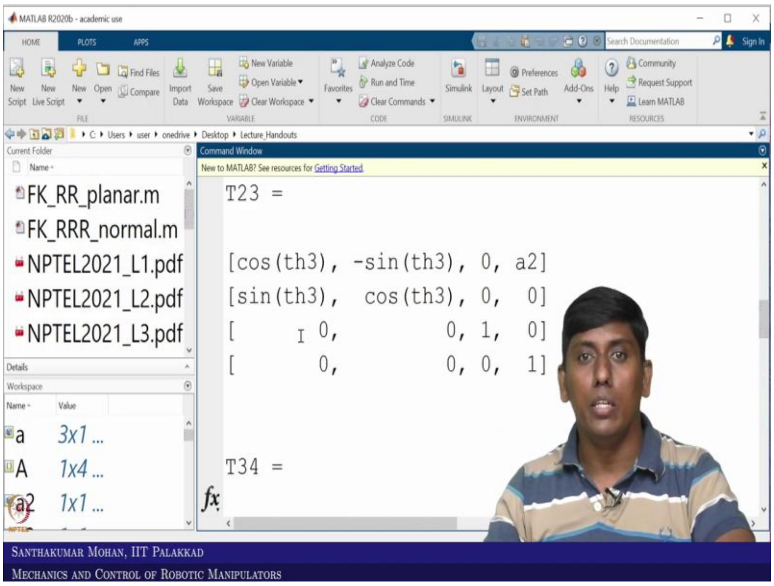


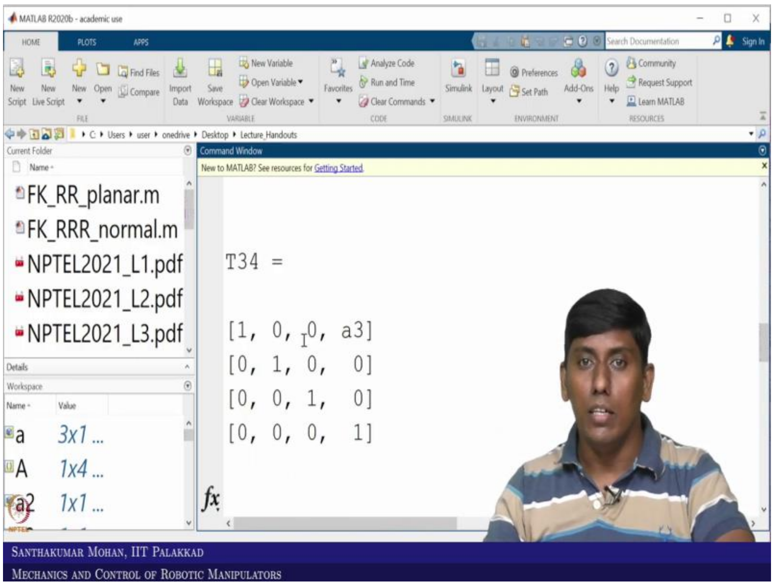
So, you can see, so it takes time. So, in the sense I will just run this. So, it is ran now. So, you can see this is the final you can say output. So, this the final output of T 0, 1. Anyhow, I would be sharing the same code with you in the lecture material, you can also run and cross verify.

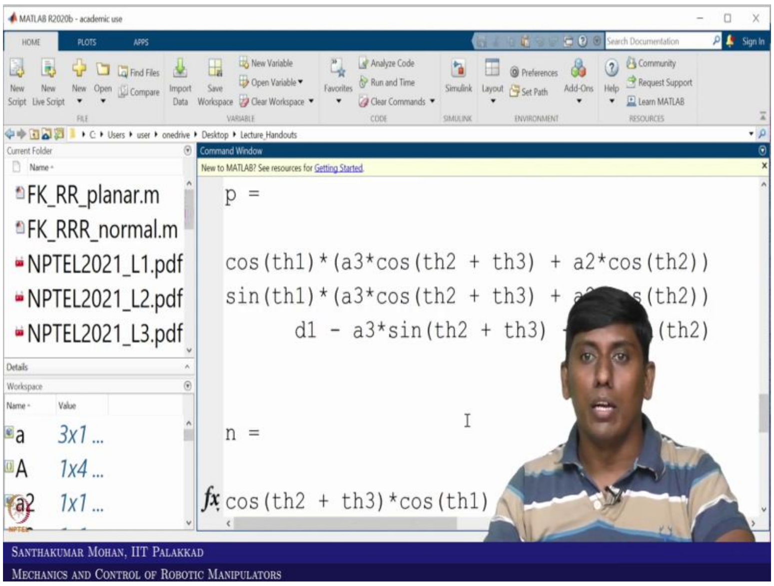
(Refer Slide Time: 15:31)





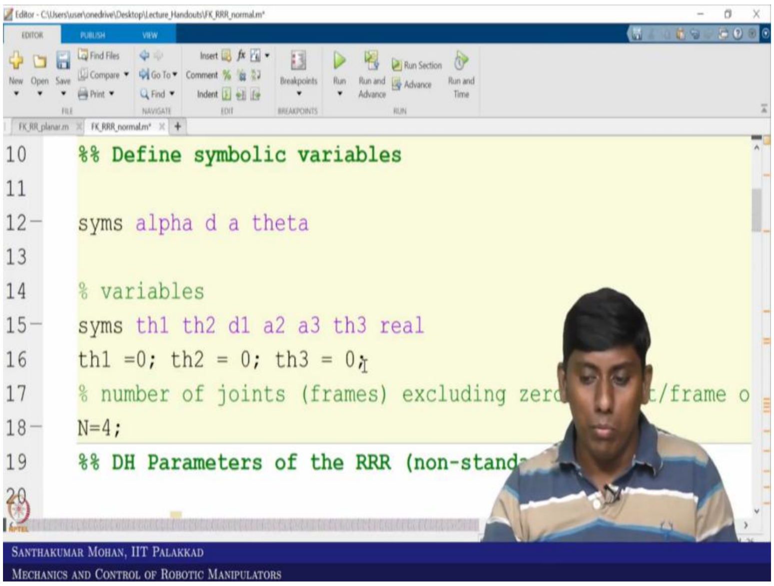






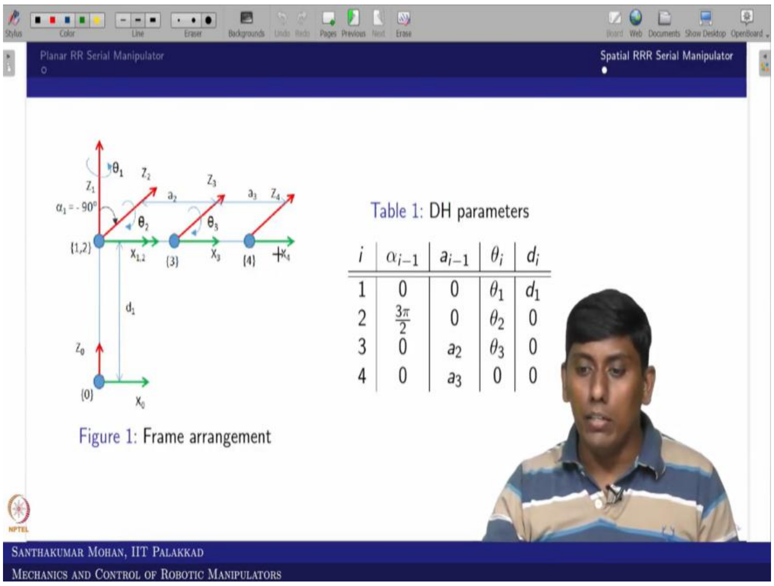
But now you can see that this is the individual transformation matrices. T 0, 1 T 1, 2 T 2, 3 T 3, 4 and you can see this is the position vector. So, you can see the position vector is exactly matching what you have obtained in your you can say straight forward relation which you obtained in previous lectures, and this is the normal and this is the sliding and this is the approach vector. So, now, as usual we need to cross check whether the solution is actually like matching or not.

(Refer Slide Time: 16:01)



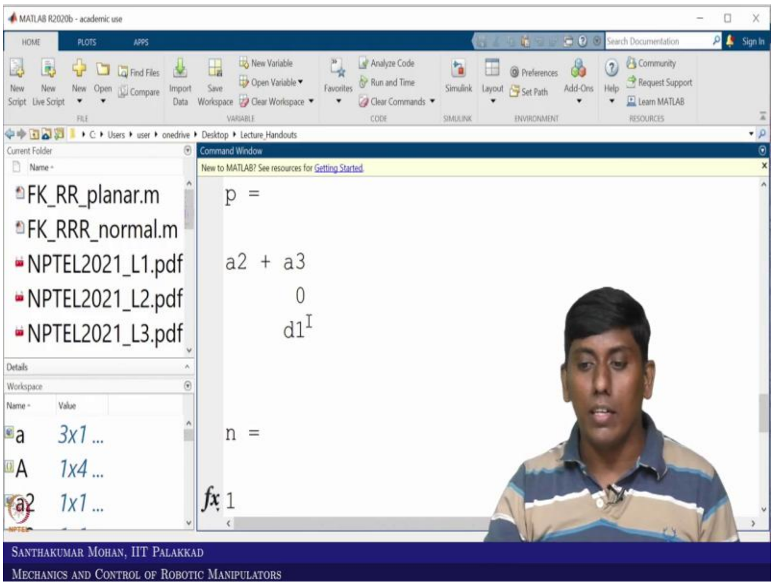
So, for that what you can do, you can take the theta 1 equal to 0 and, so theta 1 equal to 0 and theta 2 equal to 0 and theta 3 equal to 0. So, now in that sense, what would be the home position?

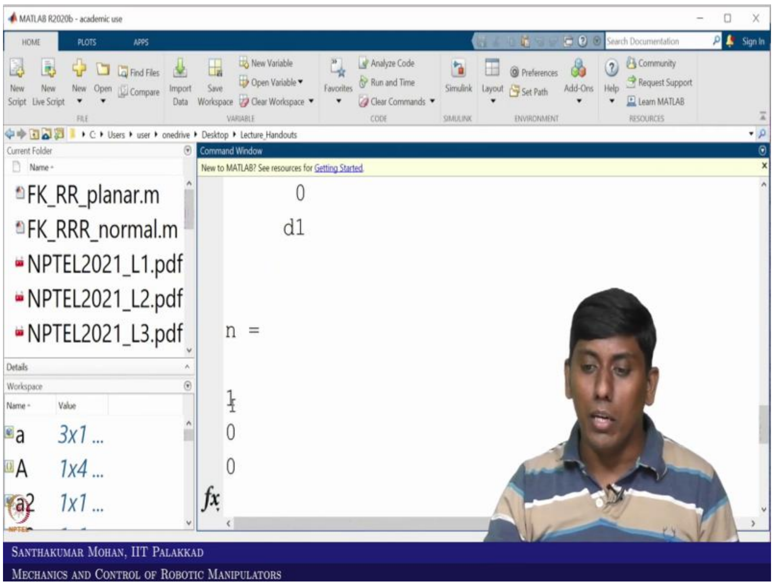
(Refer Slide Time: 16:24)

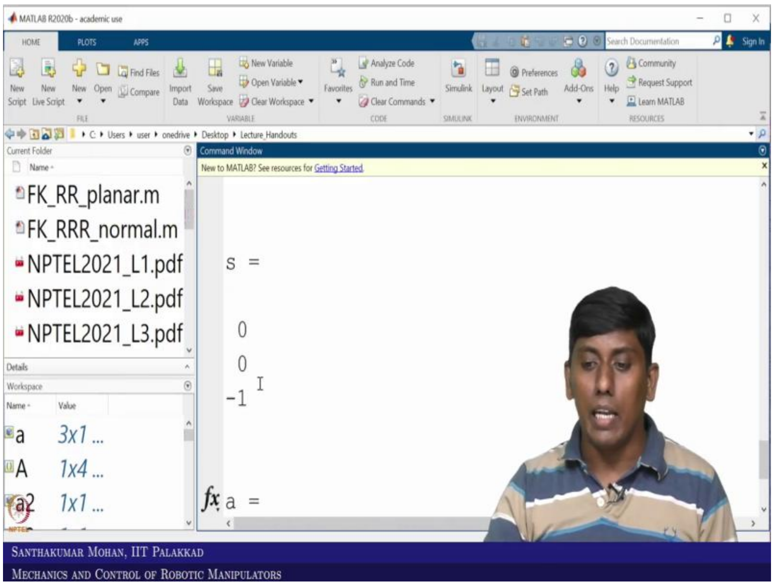


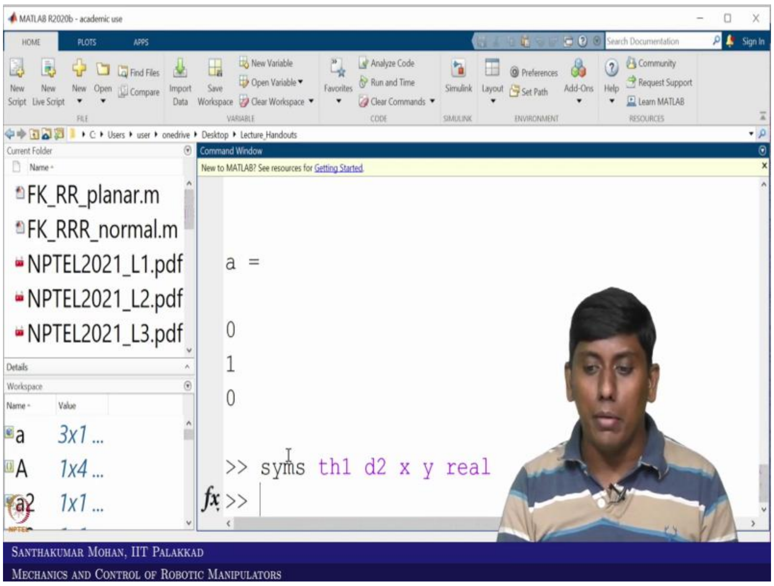
So, you can actually like see home position says that in the X axis, a 2 plus a 3, Y axis 0 and Z axis d1 and the orientation you can see X 4 and X 0 is parallel, so in the sense of 1, 0, 0, but Z 4 is parallel to Y 4 in the sense, the third vector would be 0, 1, 0, but the second vector which is actually like Y 4 is coming downward, in the sense 0, 0 minus 1. So, that is what we can expect, so we can run that.

(Refer Slide Time: 16:54)





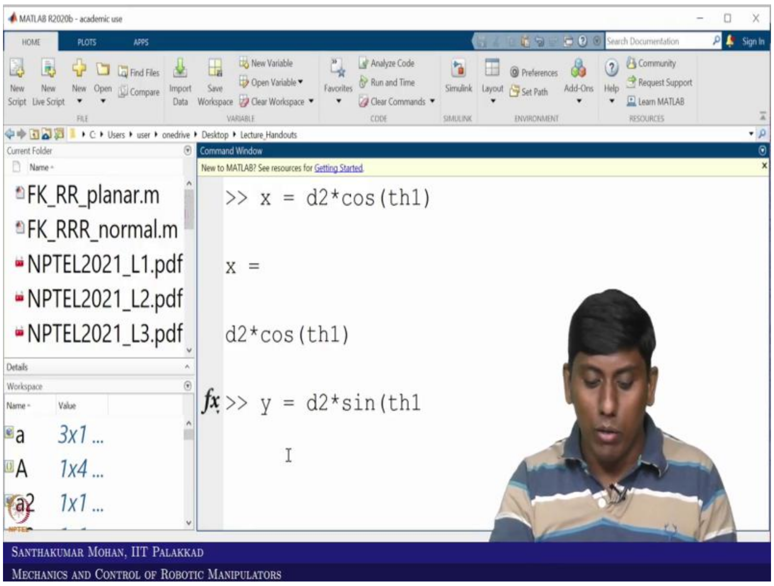


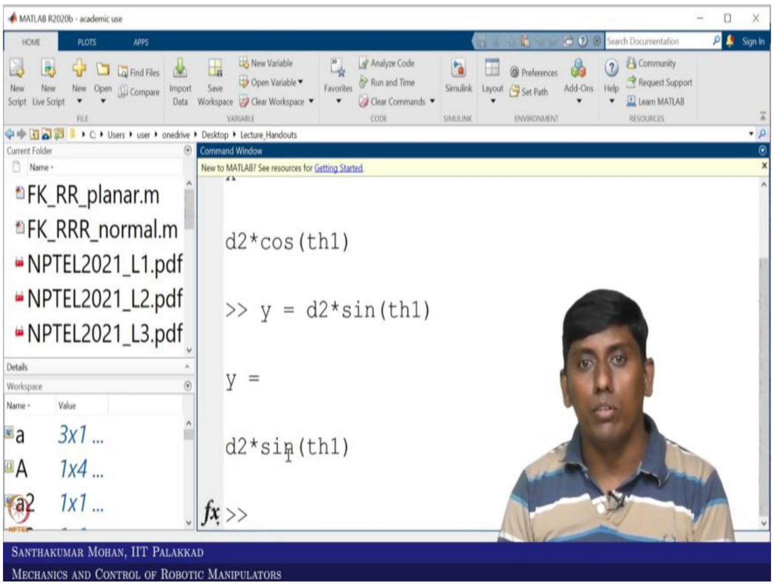


So, if I run this, what you can expect? So, in the X axis, it will be a 2 plus a 3. So, we can check, and Y axis is 0 and Z axis d1 and normal is 1, 0, 0 what we have also checked and the sliding is 0, 0 minus 1 that is also cross verified and approach is 0, 1, 0 that is also cross verified. In the sense, your, you can say home position is matching with your frame arrangement, in the sense your DH parameters are right and whatever you obtain the forward kinematics solution is right.

So, as I already mentioned the inverse kinematic direct closed form solution using MATLAB may be possible. In fact, it is possible. Only thing the complexity is very, very high. I do not want you to end up with that way, very simple example I will show you here. So, I can write symbols. So, I can write symbols are theta 1 and D2 which are real. So, even I am writing as X and Y also real.

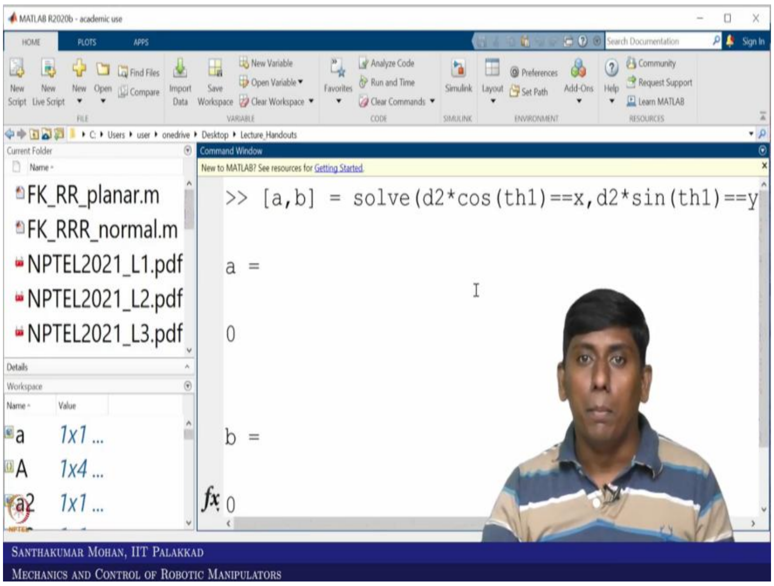
(Refer Slide Time: 18:18)

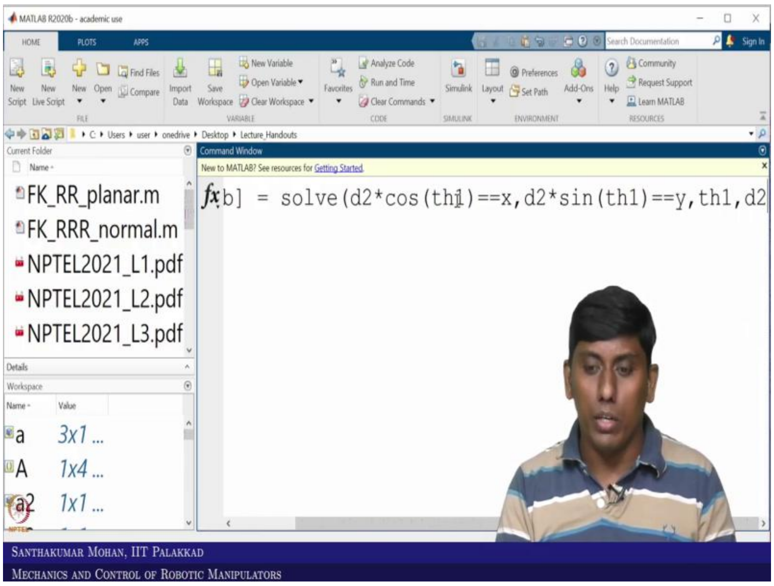


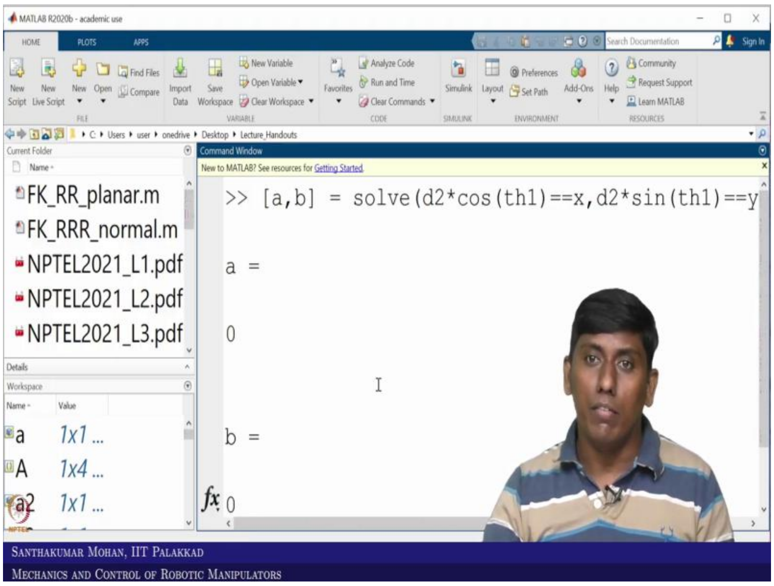


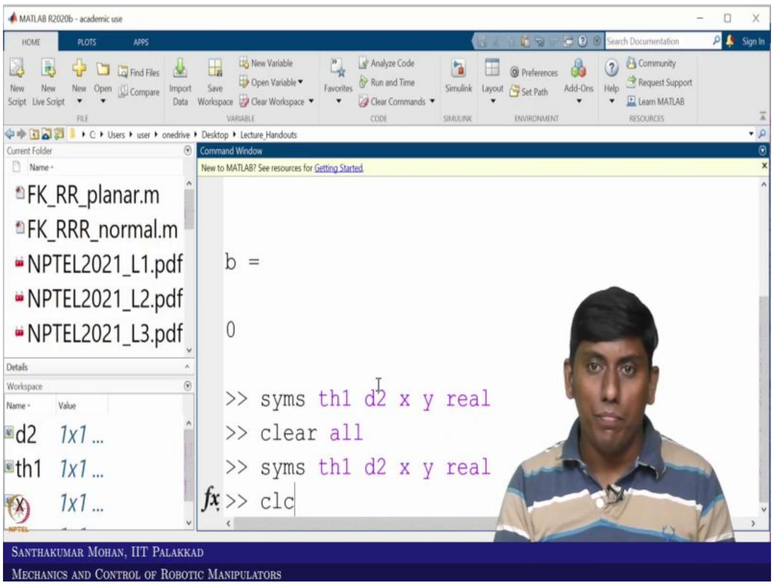
So, now you can see that I am writing it. I will write it as X, you can write, so what is that d 2, so cos theta 1, so this is the way Y is d 2 you can say sine theta 1, you can recall the RP manipulator we, have run. So, now this is you know already forward kinematics solution. Now this kinematic solution is given. Can I find inverse kinematics solution? So, for that what we are trying to use, so we are trying to use the solve command.

(Refer Slide Time: 18:52)





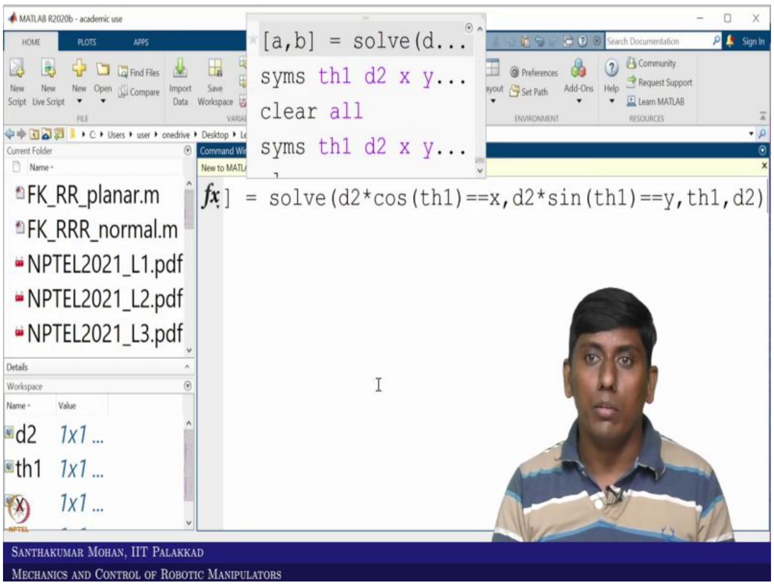


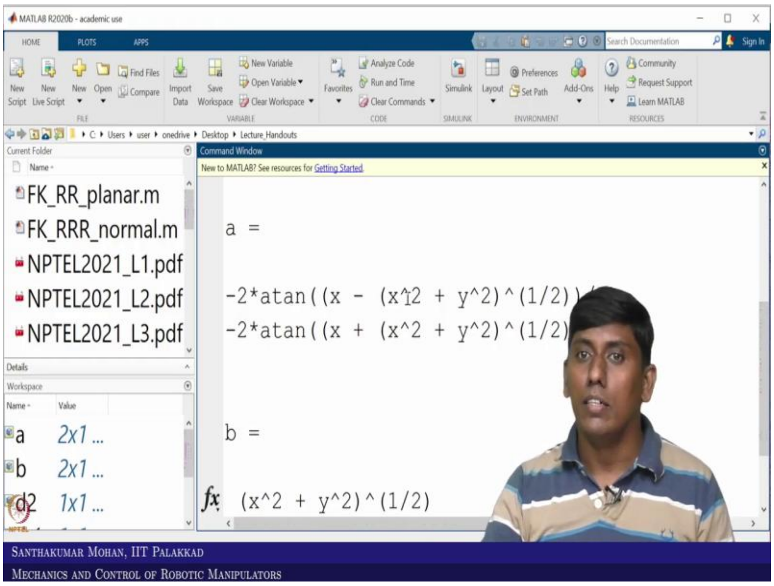


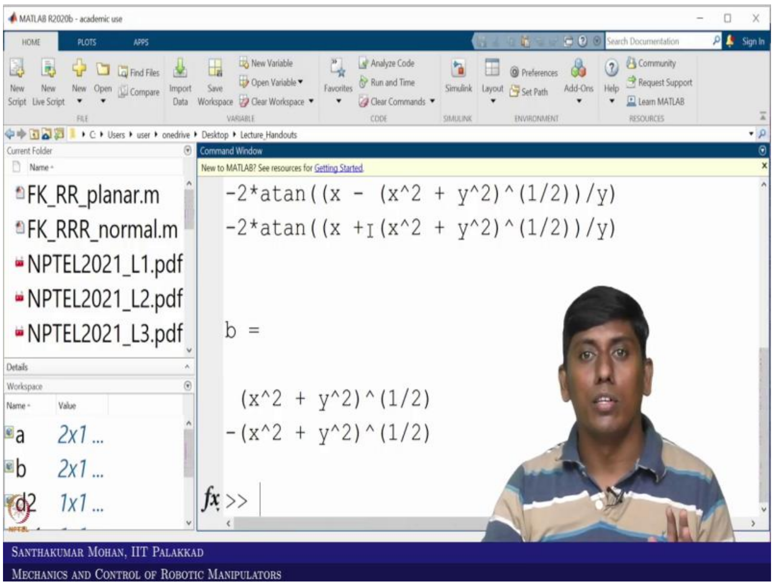
So, what we are trying to say so a and b are the final result, which is equal to theta 1 and d 2, I am just using a random variable. I am solving two equations. What equation? So, d 2 sine theta 1 equal to X. So, since I am writing an equation, so equal to, like this. So, then d 2 sine. So, this is sine that has cos, so I will correct.

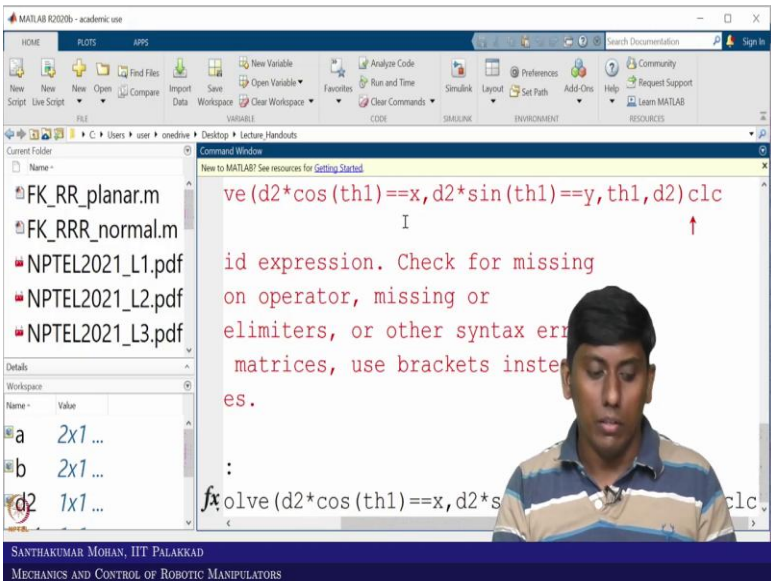
So, this is equal to Y. So, this I will correct it as cos. So, I have written as these two equations and what I am trying to solve? I am trying to solve theta 1 and d 2. So, now if I put this, so it will give a solution. So, why it is giving 0, 0, because we have already found that theta 1 is 0. So, now I am substituting that. So, first I am clearing everything.

(Refer Slide Time: 20:10)




MATLAB R2020b - academic use
FK_RR_planar.m
FK_RRR_normal.m
NPTEL2021_L1.pdf
NPTEL2021_L2.pdf
NPTEL2021_L3.pdf
a =
-2*atan((x - (x^2 + y^2)^(1/2))/
-2*atan((x + (x^2 + y^2)^(1/2)
b =
(x^2 + y^2)^(1/2)
a 2x1 ...
b 2x1 ...
d2 1x1 ...
Santhakumar Mohan, IIT Palakkad
Mechanics and Control of Robotic Manipulators


MATLAB R2020b - academic use
FK_RR_planar.m
FK_RRR_normal.m
NPTEL2021_L1.pdf
NPTEL2021_L2.pdf
NPTEL2021_L3.pdf
-2*atan((x - (x^2 + y^2)^(1/2))/y)
-2*atan((x + (x^2 + y^2)^(1/2))/y)
b =
(x^2 + y^2)^(1/2)
-(x^2 + y^2)^(1/2)
>>
a 2x1 ...
b 2x1 ...
d2 1x1 ...
Santhakumar Mohan, IIT Palakkad
Mechanics and Control of Robotic Manipulators


MATLAB R2020b - academic use
FK_RR_planar.m
FK_RRR_normal.m
NPTEL2021_L1.pdf
NPTEL2021_L2.pdf
NPTEL2021_L3.pdf
ve(d2*cos(th1)==x,d2*sin(th1)==y,th1,d2)clc
id expression. Check for missing
on operator, missing or
elimiters, or other syntax err
matrices, use brackets inste
es.
olve(d2*cos(th1)==x,d2*s
a 2x1 ...
b 2x1 ...
d2 1x1 ...
Santhakumar Mohan, IIT Palakkad
Mechanics and Control of Robotic Manipulators

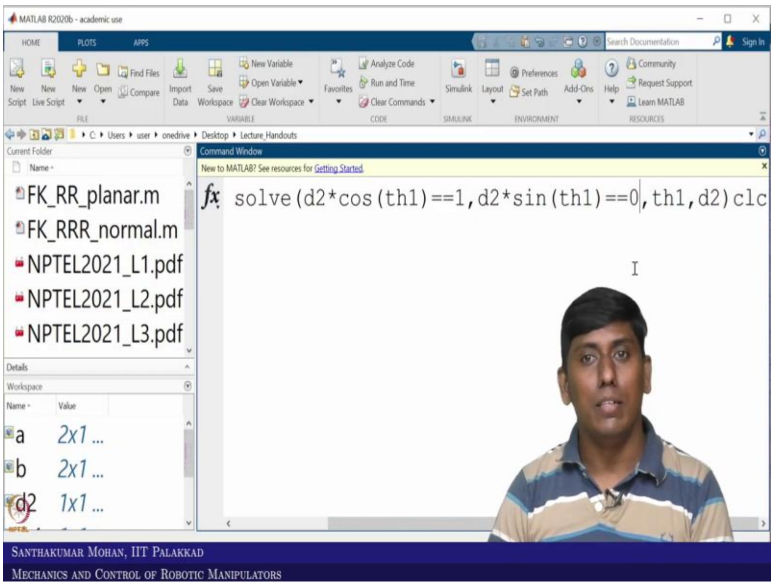
So, now I am, this one and so now I am using this relation, so now I am finding it. So, now what you can see I got the final solution. So, the final solution says that the theta 1 is, so 2 times have a tan inverse, this is tan inverse of X minus. So, this is X squared plus Y squared, which is square root divided by Y, this is one solution.

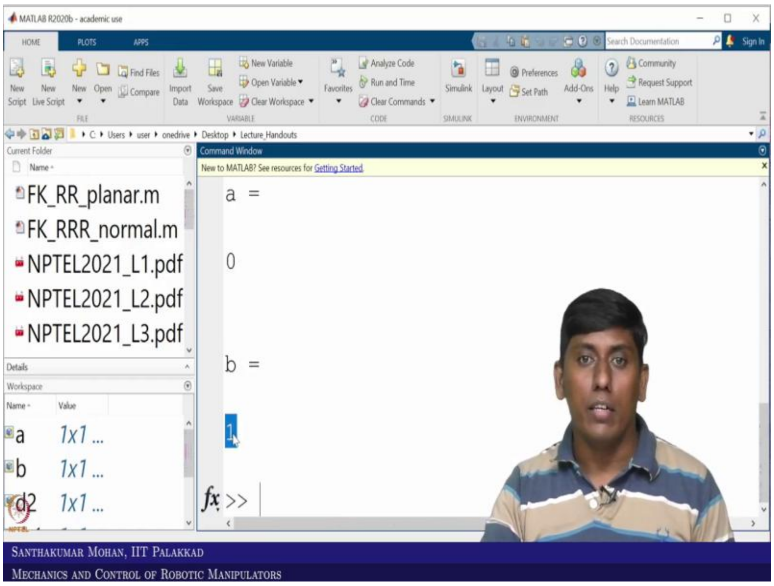
If I put minus, that is another solution. So, how this is solved? So, it is solved with the help of what you can see, so X is actual like d 2 cos theta 1, Y is d 2 sine theta 1. So, it has solved square and add, so it would give you d squared. Then you can say that X minus in the sense, it used half angle quantities, in the sense it used sine theta in the form of tan theta by 2 and cos theta also, tan theta by 2 that relation of quantity and then get it.

So, in the sense, the same thing can be used, but what we have used, we have straightforward written as, so theta 1 is tan inverse of, so Y by X. So, the other one is simple square root of X squared plus Y squared, whole square root, you can take it. That is what d, d2. So, that too we will take plus solution, because the minus solution will not be possible because it cannot go beyond.

So, that is what the whole idea. So, in that sense, you can see that the inverse kinematic solution also can be found using MATLAB, but it is slightly tricky, but if you give a numerical solution. So, numerical solution in the sense for example, I am using it, so instead of this I am saying that. I will just make this clear all. So, I am putting clear all.

(Refer Slide Time: 22:10)





So, now the same equation, I am some value. So, this is one and this is probably 0. So, in the sense what you can expect the d 2 would be 1 and theta 1 would be 0. So, we will get it or not. So, now, it is still defined as a symbol.

So, it is you can see that the theta 1 is 0 degree and d 2 is 1. Like that you are getting a solution. So, like that you can use even MATLAB for solving the inverse kinematic solution, but I do not recommend that. So, if you want to solve the inverse kinematics solution. So, better you can go with numerical method. So, the next lecture would be talking about numerical method. So, until then. Thank you and see you. Bye take care