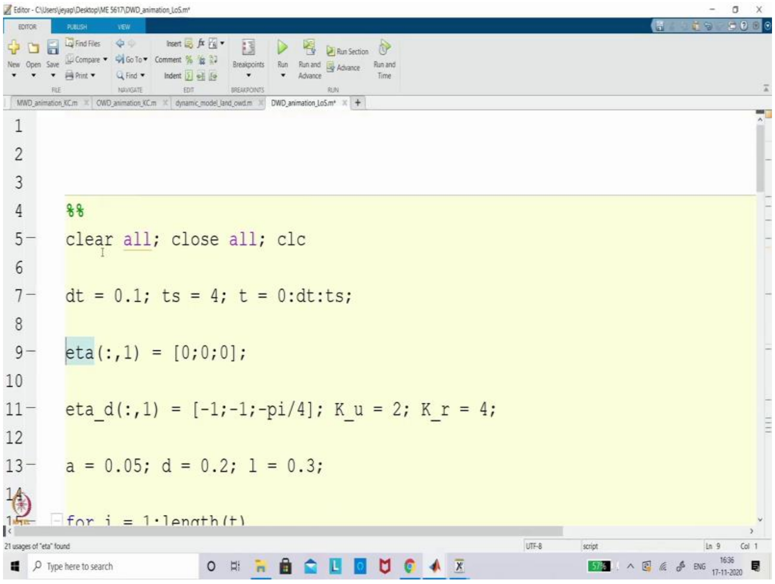
Wheeled Mobile Robots
Prof. Santhakumar Mohan
Department of Mechanical Engineering
Indian Institute of Technology, Palakkad

Lecture - 38
Simulation of Land-based Mobile Robots along with Kinematic Control Part 3

So, in the last lecture what we have seen actually like how the differential wheel drive can be incorporated with the line of sight and polar coordinate. So, now in this lecture we will see that what we have done in the last lecture, that we will try to incorporate in MATLAB and try to simulate how that will work ok.

So, in the sense we will take a differential wheel drive and we do both set point and as well as tracking with position tracking or we can see like how to do the orientation tracking. So, with that we will move to the MATLAB window now and we will take the earlier model and we will try to understand how to incorporate the you call polar coordinate or line of sight.

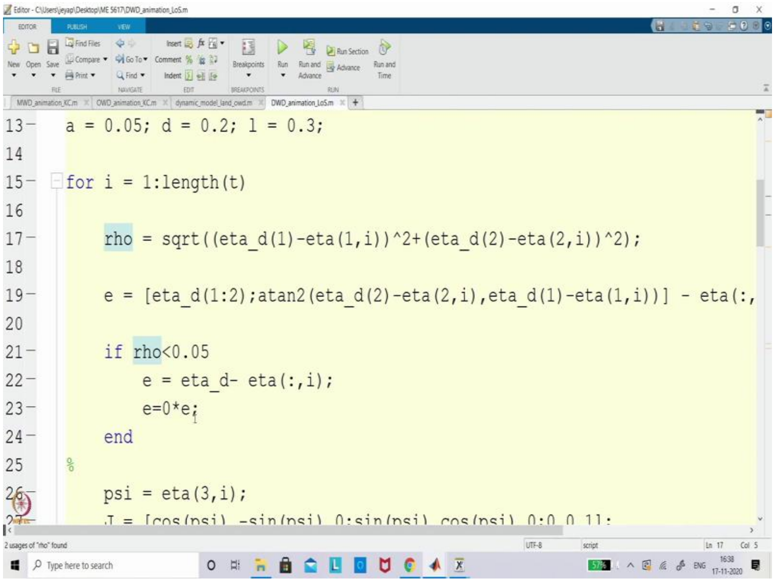
(Refer Slide Time: 01:01)



```



%%
clear all; close all; clc

dt = 0.1; ts = 4; t = 0:dt:ts;

eta(:,1) = [0;0;0];

eta_d(:,1) = [-1;-1;-pi/4]; K_u = 2; K_r = 4;

a = 0.05; d = 0.2; l = 0.3;

for i = 1:length(t)
```

So, now we can actually like come back to the you can see the MATLAB window where I have already written the code for line of sight for a differential wheel drive.

(Refer Slide Time: 01:11)



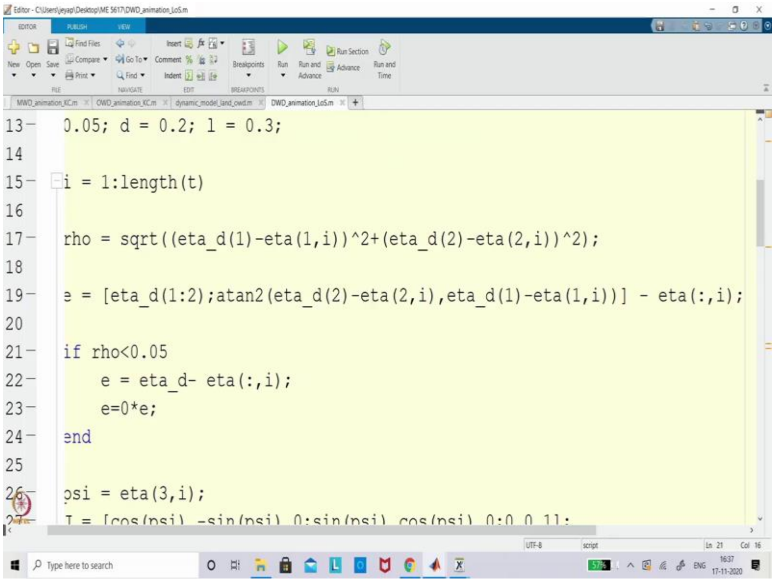
```
a = 0.05; d = 0.2; l = 0.3;

for i = 1:length(t)

    rho = sqrt((eta_d(1)-eta(1,i))^2+(eta_d(2)-eta(2,i))^2);

    e = [eta_d(1:2);atan2(eta_d(2)-eta(2,i),eta_d(1)-eta(1,i))] - eta(:,

    if rho<0.05
        e = eta_d- eta(:,i);
        e=0*e;
    end
%
    psi = eta(3,i);
    J = [cos(psi) -sin(psi) 0;sin(psi) cos(psi) 0;0 0 1];
```

2 usages of "rho" found

So, you can see that I will actually like explain. So, we have actually like taken K_u and K_r for the polar coordinate. Otherwise, we will directly take as a K and these are the what you call the vehicle parameters where the a is the radius of the wheel and you can see that this is the initial condition and these are the simulation parameter.

So, let us move what we have done. So, what we have done is actually like there are two things we have done. So, one is actually like we have calculated the ρ ok. So, the other one is we are calculating the error.

(Refer Slide Time: 01:43)

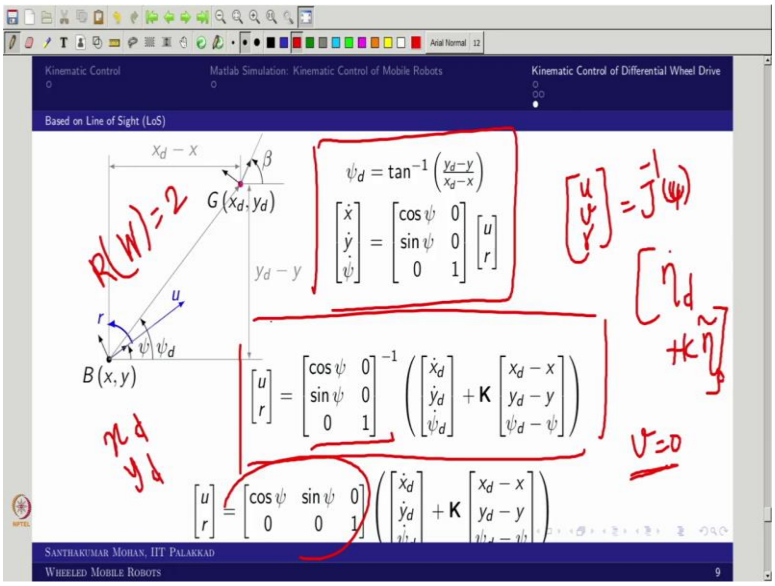


```
).05; d = 0.2; l = 0.3;

i = 1:length(t)

rho = sqrt((eta_d(1)-eta(1,i))^2+(eta_d(2)-eta(2,i))^2);

e = [eta_d(1:2);atan2(eta_d(2)-eta(2,i),eta_d(1)-eta(1,i))] - eta(:,i);

if rho<0.05
    e = eta_d- eta(:,i);
    e=0*e;
end

psi = eta(3,i);
J = [cos(psi) -sin(psi) 0;sin(psi) cos(psi) 0;0 0 1];
```

So, the error is actually like based on the line of sight. So, first we will see the line of sight directly. So, where we will incorporate only e. So, e is actually like $\tilde{\eta}$. So, the $\tilde{\eta}$ would be actually like calculating based on the line of sight.

So, where the η_d would be compared with η, but the angle which is what you call desired angle is actually like modified as new change. So, that angle is actually like tan^{-1} of that if you recall here.

(Refer Slide Time: 02:09)



Kinematic Control

Matlab Simulation: Kinematic Control of Mobile Robots

Kinematic Control of Differential Wheel Drive

Based on Line of Sight (LoS)

$$\psi_d = \tan^{-1}\left(\frac{y_d - y}{x_d - x}\right)$$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}\cos\psi & 0\\ \sin\psi & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}u\\ r\end{bmatrix}$$

$$\begin{bmatrix}u\\ r\end{bmatrix} = \begin{bmatrix}\cos\psi & 0\\ \sin\psi & 0\\ 0 & 1\end{bmatrix}^{-1}\left(\begin{bmatrix}\dot{x}_d\\ \dot{y}_d\\ \dot{\psi}_d\end{bmatrix} + \mathbf{K}\begin{bmatrix}x_d - x\\ y_d - y\\ \psi_d - \psi\end{bmatrix}\right)$$

$$\begin{bmatrix}u\\ r\end{bmatrix} = \begin{bmatrix}\cos\psi & \sin\psi & 0\\ 0 & 0 & 1\end{bmatrix}\left(\begin{bmatrix}\dot{x}_d\\ \dot{y}_d\\ \dot{\psi}_d\end{bmatrix} + \mathbf{K}\begin{bmatrix}x_d - x\\ y_d - y\\ \psi_d - \psi\end{bmatrix}\right)$$

$$\begin{bmatrix}u\\ \psi\\ r\end{bmatrix} = J^{-1}(\psi)$$

$$\left[\dot{\eta}_d + K\tilde{\eta}\right]$$

$v = 0$

Santhakumar Mohan, IIT Palakkad

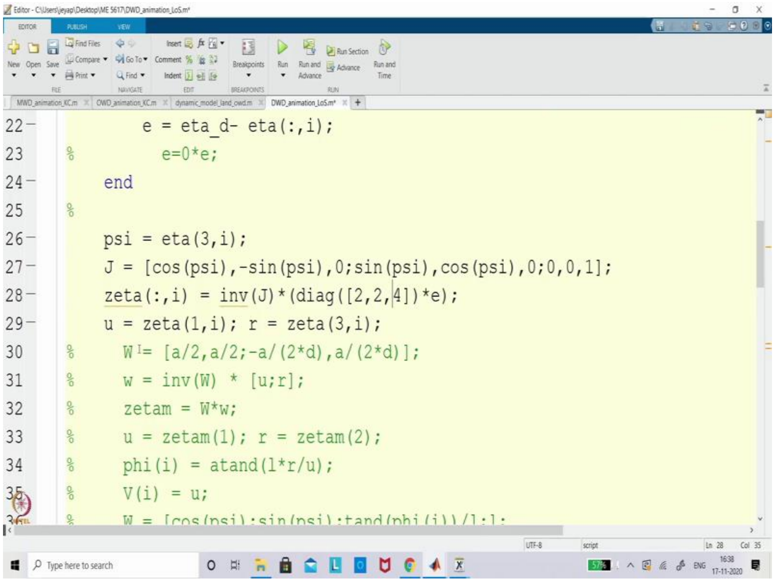
Wheeled Mobile Robots

9

So, here if you recall this is what the Ψ_d. So, the same angle we are actually like incorporating here. So, that is what we have taken and why it is actually like two argument a tan inverse you know already based on the quadrant choice. So, now this is what the error. So, now this error is actually like it is a smaller range in the sense I am calculating the ρ.

So, from the starting to the end point if the ρ is keep on reducing, if the ρ is actually like within 5 centimeter circle. So, then I am moving to the what you call the orientation follow; otherwise I will actually like keep it as simple position following or position tracking. So, that is what the whole idea. So, I will actually like come back. So, now I will actually like make it this a pass command.

(Refer Slide Time: 02:52)

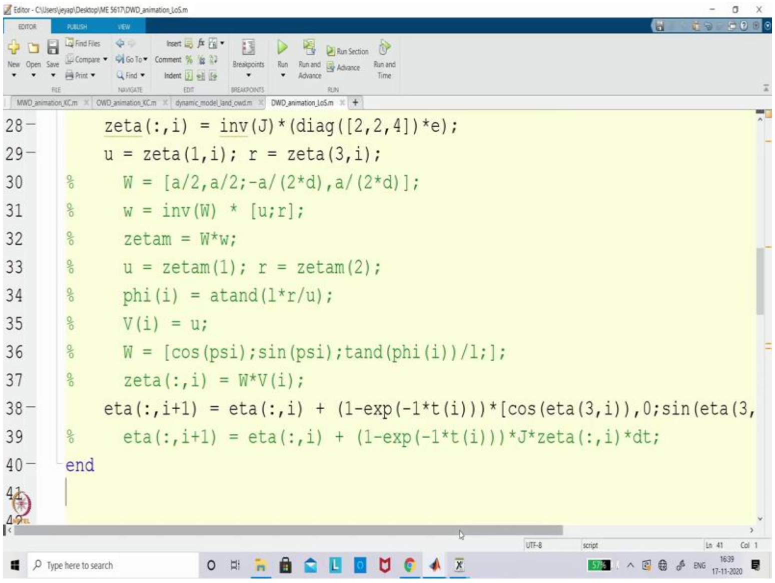


```
        e = eta_d- eta(:,i);
%          e=0*e;
    end
%

    psi = eta(3,i);
    J = [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1];
    zeta(:,i) = inv(J)*(diag([2,2,4])*e);
    u = zeta(1,i); r = zeta(3,i);
%     W = [a/2,a/2;-a/(2*d),a/(2*d)];
%     w = inv(W) * [u;r];
%     zetam = W*w;
%     u = zetam(1); r = zetam(2);
%     phi(i) = atand(l*r/u);
%     V(i) = u;
%     W = [cos(psi);sin(psi);tand(phi(i))/l;];
```

So, now what we are doing the same thing. So, we are taking as a ξ, ξ is actually like inverse of this is actually like position set point tracking. So, you would not be having $\dot{\eta}_d$.

So, that is why directly we have written here. So, and we are assuming that the K_1 and K_2 is actually like 2 each in the sense K_x and K_y or λ_x λ_y is 2 each and λ_Ψ is 4. And now I have actually like taken I already told right the v is 0 right. So, that is why v, u and r only there.

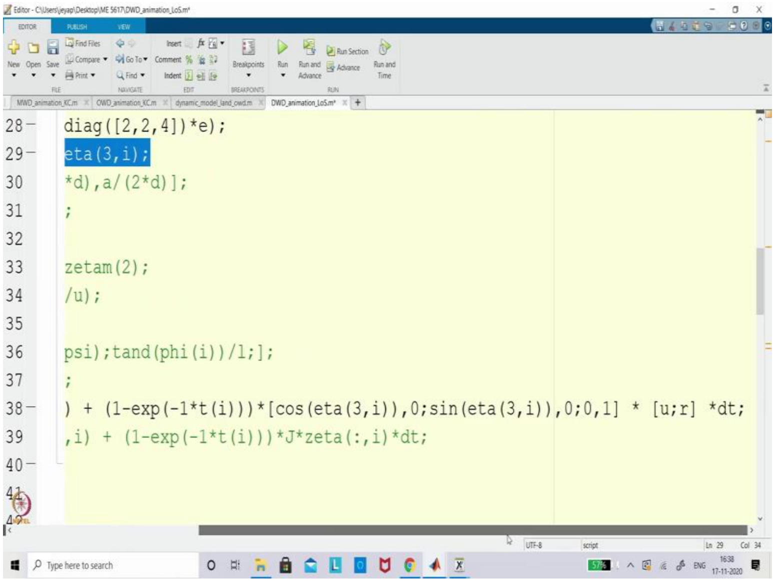
(Refer Slide Time: 03:28)



```
    zeta(:,i) = inv(J)*(diag([2,2,4])*e);
    u = zeta(1,i); r = zeta(3,i);
%     W = [a/2,a/2;-a/(2*d),a/(2*d)];
%     w = inv(W) * [u;r];
%     zetam = W*w;
%     u = zetam(1); r = zetam(2);
%     phi(i) = atand(l*r/u);
%     V(i) = u;
%     W = [cos(psi);sin(psi);tand(phi(i))/l;];
%     zeta(:,i) = W*V(i);
    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,
%     eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*J*zeta(:,i)*dt;
end
```

And then we are actually like incorporating the wheel configuration and then we are actually like doing it.

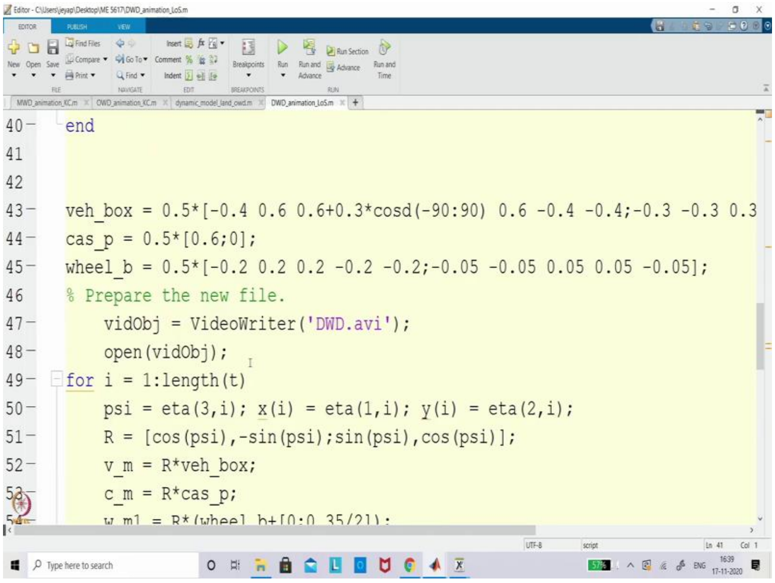
(Refer Slide Time: 03:33)



```
diag([2,2,4])*e);
eta(3,i);
*d),a/(2*d)];
;

zetam(2);
/u);

psi);tand(phi(i))/l;];
;
) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,i)),0;0,1] * [u;r] *dt;
,i) + (1-exp(-1*t(i)))*J*zeta(:,i)*dt;
```

So, the wheel configuration I have written already in the matrix format ok. So, this matrix what you have obtained here right. So, you can recall this is cosΨ sinΨ 0 0 0 1 right. So, the same thing I have actually like incorporated here you can see. So, cosΨ you can see sinΨ 0 and 0 0 and 1. So, that is what we have actually like incorporated.

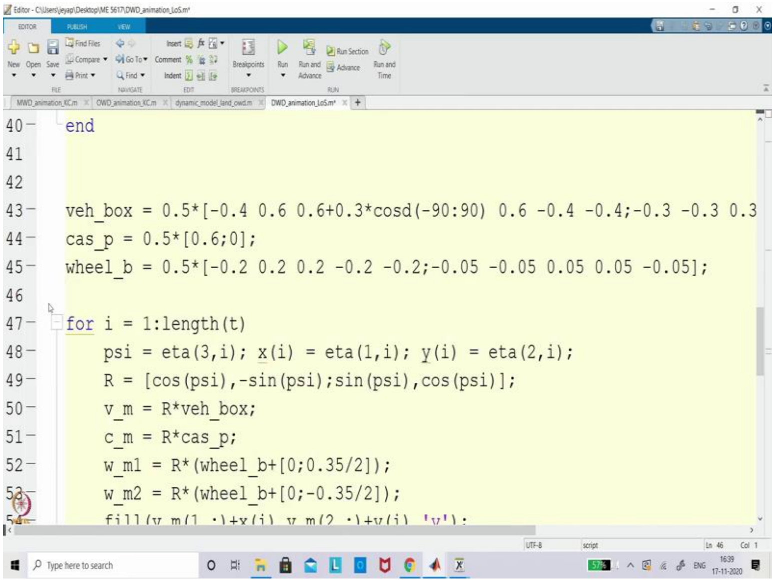
(Refer Slide Time: 04:00)



```
end


veh_box = 0.5*[-0.4 0.6 0.6+0.3*cosd(-90:90) 0.6 -0.4 -0.4;-0.3 -0.3 0.3
cas_p = 0.5*[0.6;0];
wheel_b = 0.5*[-0.2 0.2 0.2 -0.2 -0.2;-0.05 -0.05 0.05 0.05 -0.05];
% Prepare the new file.
    vidObj = VideoWriter('DWD.avi');
    open(vidObj);
for i = 1:length(t)
    psi = eta(3,i); x(i) = eta(1,i); y(i) = eta(2,i);
    R = [cos(psi),-sin(psi);sin(psi),cos(psi)];
    v_m = R*veh_box;
    c_m = R*cas_p;
    w_m1 = R*(wheel_b+[0;0.35/2]);
```

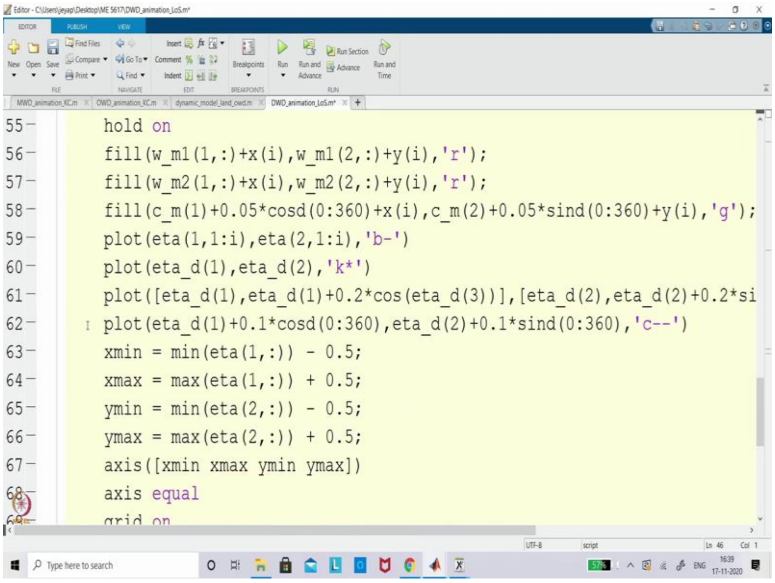
So, now if I run this. So, what you can expect? So, this will actually like follow something.

(Refer Slide Time: 04:04)



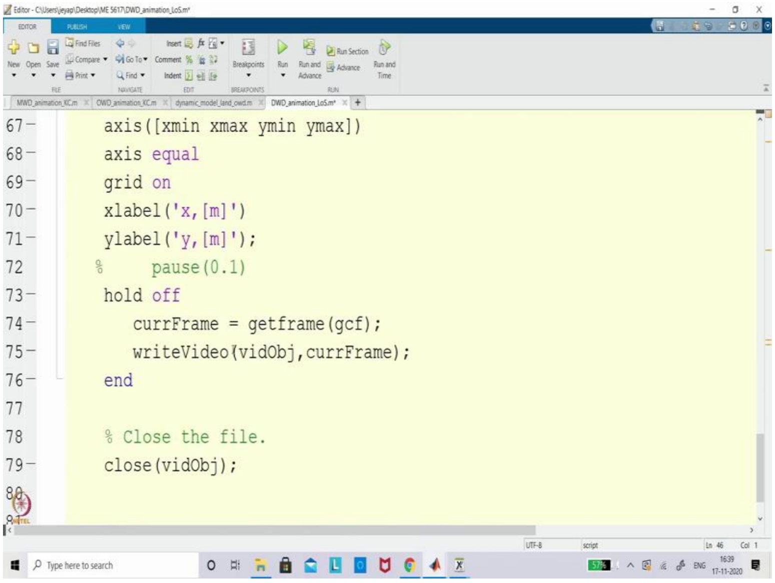
```
end


veh_box = 0.5*[-0.4 0.6 0.6+0.3*cosd(-90:90) 0.6 -0.4 -0.4;-0.3 -0.3 0.3
cas_p = 0.5*[0.6;0];
wheel_b = 0.5*[-0.2 0.2 0.2 -0.2 -0.2;-0.05 -0.05 0.05 0.05 -0.05];

for i = 1:length(t)
    psi = eta(3,i); x(i) = eta(1,i); y(i) = eta(2,i);
    R = [cos(psi),-sin(psi);sin(psi),cos(psi)];
    v_m = R*veh_box;
    c_m = R*cas_p;
    w_m1 = R*(wheel_b+[0;0.35/2]);
    w_m2 = R*(wheel_b+[0;-0.35/2]);
    fill(v_m(1,:)+x(i),v_m(2,:)+y(i),'y');
```

(Refer Slide Time: 04:05)



```
    hold on
    fill(w_m1(1,:)+x(i),w_m1(2,:)+y(i),'r');
    fill(w_m2(1,:)+x(i),w_m2(2,:)+y(i),'r');
    fill(c_m(1)+0.05*cosd(0:360)+x(i),c_m(2)+0.05*sind(0:360)+y(i),'g');
    plot(eta(1,1:i),eta(2,1:i),'b-')
    plot(eta_d(1),eta_d(2),'k*')
    plot([eta_d(1),eta_d(1)+0.2*cos(eta_d(3))],[eta_d(2),eta_d(2)+0.2*si
    plot(eta_d(1)+0.1*cosd(0:360),eta_d(2)+0.1*sind(0:360),'c--')
    xmin = min(eta(1,:)) - 0.5;
    xmax = max(eta(1,:)) + 0.5;
    ymin = min(eta(2,:)) - 0.5;
    ymax = max(eta(2,:)) + 0.5;
    axis([xmin xmax ymin ymax])
    axis equal
    grid on
```

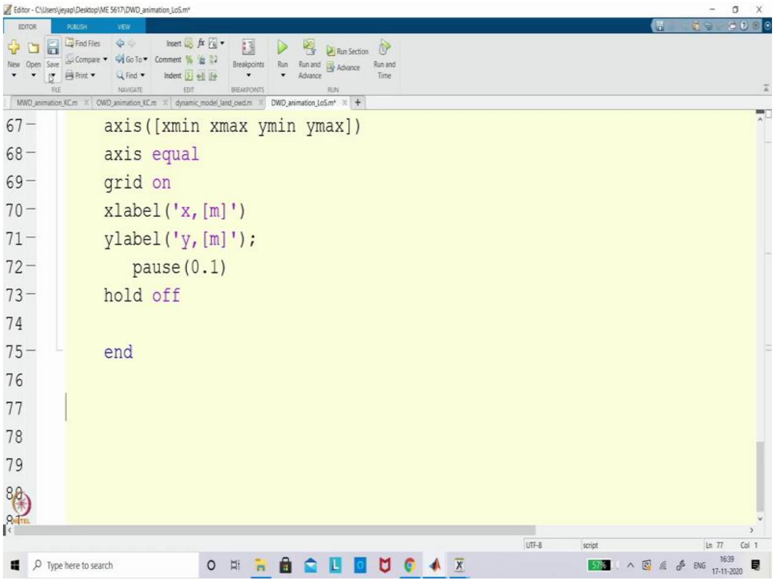
(Refer Slide Time: 04:05)



```
    axis([xmin xmax ymin ymax])
    axis equal
    grid on
    xlabel('x,[m]')
    ylabel('y,[m]');
%       pause(0.1)
    hold off
       currFrame = getframe(gcf);
       writeVideo(vidObj,currFrame);
    end

    % Close the file.
    close(vidObj);
```

So, this particularly we have try to record it. So, now we no need to record that.

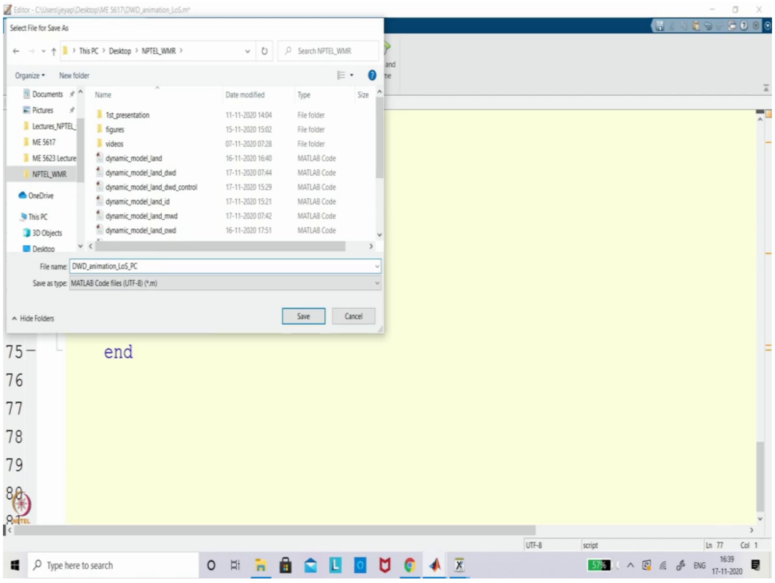
(Refer Slide Time: 04:08)



```
    axis([xmin xmax ymin ymax])
    axis equal
    grid on
    xlabel('x,[m]')
    ylabel('y,[m]');
       pause(0.1)
    hold off

    end
```

So, I am just making it that. So, then there is no recording at all.

(Refer Slide Time: 04:15)



Editor - C:\Users\jeyap\Desktop\ME 5617\DWD_animation_LoS.m*

Select File for Save As

This PC › Desktop › NPTEL_WMR ›

Name	Date modified	Type	Size
1st_presentation	11-11-2020 14:04	File folder	
figures	15-11-2020 15:02	File folder	
videos	07-11-2020 07:28	File folder	
dynamic_model_land	16-11-2020 16:40	MATLAB Code	
dynamic_model_land_dwd	17-11-2020 07:44	MATLAB Code	
dynamic_model_land_dwd_control	17-11-2020 15:29	MATLAB Code	
dynamic_model_land_id	17-11-2020 15:21	MATLAB Code	
dynamic_model_land_mwd	17-11-2020 07:42	MATLAB Code	
dynamic_model_land_owd	16-11-2020 17:51	MATLAB Code	

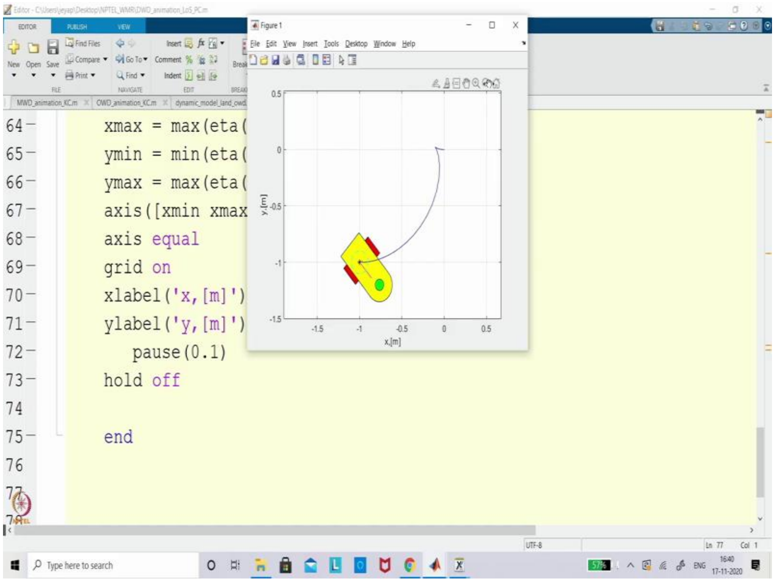
File name: DWD_animation_LoS_PC

Save as type: MATLAB Code files (UTF-8) (*.m)

```
    end
```

So, I am just making it this as I will write a position control.

(Refer Slide Time: 04:27)

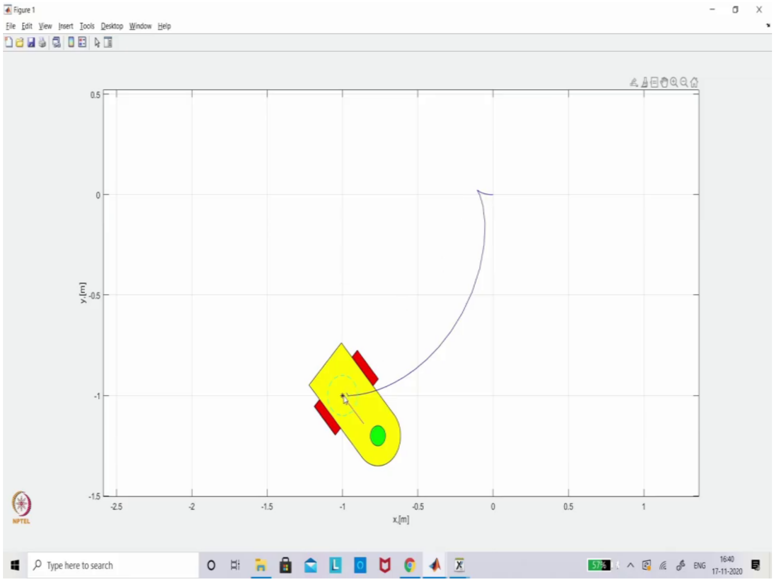


Editor - C:\Users\jeyap\Desktop\NPTEL_WMR\DWD_animation_LoS_PC.m

```
    xmax = max(eta(
    ymin = min(eta(
    ymax = max(eta(
    axis([xmin xmax
    axis equal
    grid on
    xlabel('x,[m]')
    ylabel('y,[m]')
       pause(0.1)
    hold off

    end
```

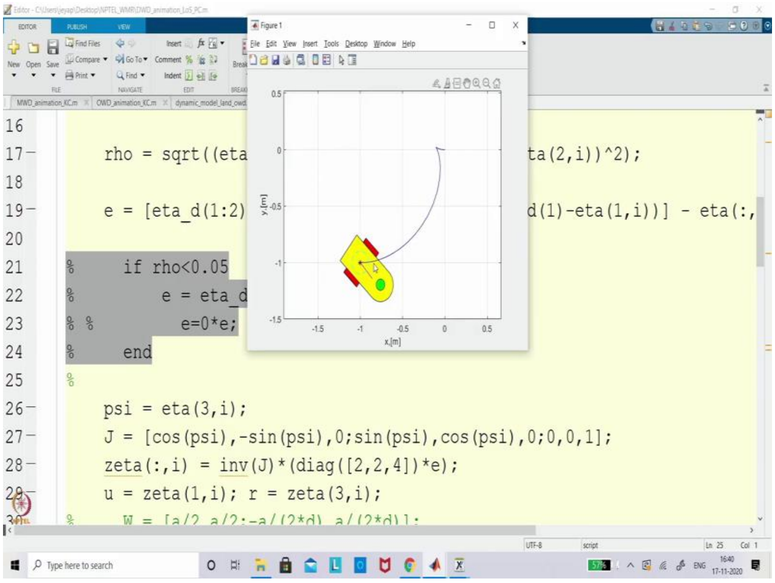
So, now what you can actually expect? This is the differentiable drive that is actually trying to follow the position which is the goal point and after this in, you can say within that circle if it is enter within the 5 centimeter circle it is trying to follow the orientation. So, that is what you would have actually seen again I am running that. So, I will actually like show the maximize.

(Refer Slide Time: 04:46)



You can see this is the starting point and it is trying to follow and it suppose to come here it is not a you can say position tracking it is just a position following. So, this is a given point and this is a final point and you can see that within that circle it reach it is trying to follow the orientation also ok. So, that is what I have actually like given as a code. So, now, you do not want that you can actually like make it that also like a comment.

(Refer Slide Time: 05:15)

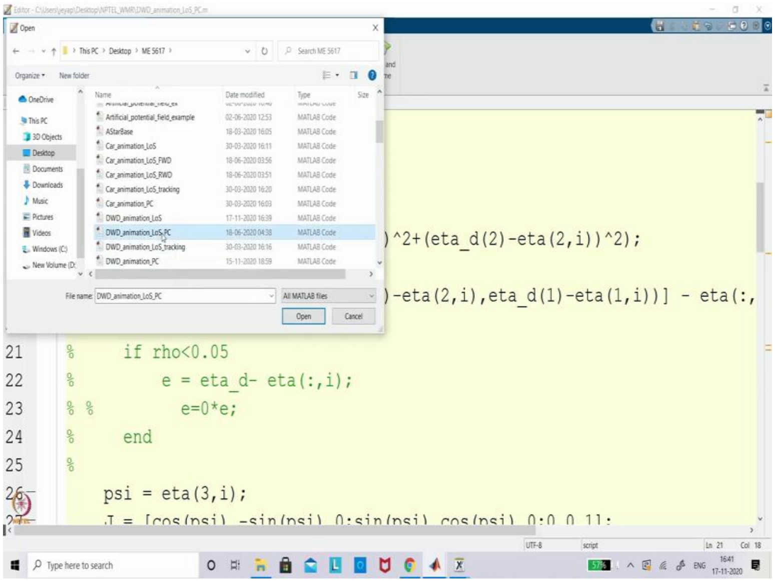


```
    rho = sqrt((eta                                   ta(2,i))^2);

    e = [eta_d(1:2)                                   d(1)-eta(1,i))] - eta(:,

%     if rho<0.05
%         e = eta_d
% %       e=0*e;
%     end
%
    psi = eta(3,i);
    J = [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1];
    zeta(:,i) = inv(J)*(diag([2,2,4])*e);
    u = zeta(1,i); r = zeta(3,i);
%   W = [a/2 a/2;-a/(2*d) a/(2*d)];
```

So, I am just making it that as a comment. So, now what you can actually see that it would try to converge to that given point. So, without following the orientation you can see the difference right. So, by what one can see? Once it is coming to one particular profile you can see that the vehicle is start over you can see rotating, why it is so?

Because you are you can say the control activity is actually like giving preference to both ok. So, because your Ψ_d is actually like become almost 0, but you are trying to follow that and as well as you are trying to follow your position profile. So, this is one you have to actually like understand.

(Refer Slide Time: 05:58)


Editor - C:\Users\jeyap\Desktop\NPTEL_WMR\DWD_animation_LoS_PC.m

Open

This PC › Desktop › ME 5617 ›

Name	Date modified	Type
Artificial_potential_field_example	02-06-2020 12:53	MATLAB Code
AStarBase	18-03-2020 16:05	MATLAB Code
Car_animation_LoS	30-03-2020 16:11	MATLAB Code
Car_animation_LoS_FWD	18-06-2020 03:56	MATLAB Code
Car_animation_LoS_RWD	18-06-2020 03:51	MATLAB Code
Car_animation_LoS_tracking	30-03-2020 16:20	MATLAB Code
Car_animation_PC	30-03-2020 16:03	MATLAB Code
DWD_animation_LoS	17-11-2020 16:39	MATLAB Code
DWD_animation_LoS_PC	18-06-2020 04:38	MATLAB Code
DWD_animation_LoS_tracking	30-03-2020 16:16	MATLAB Code
DWD_animation_PC	15-11-2020 18:59	MATLAB Code

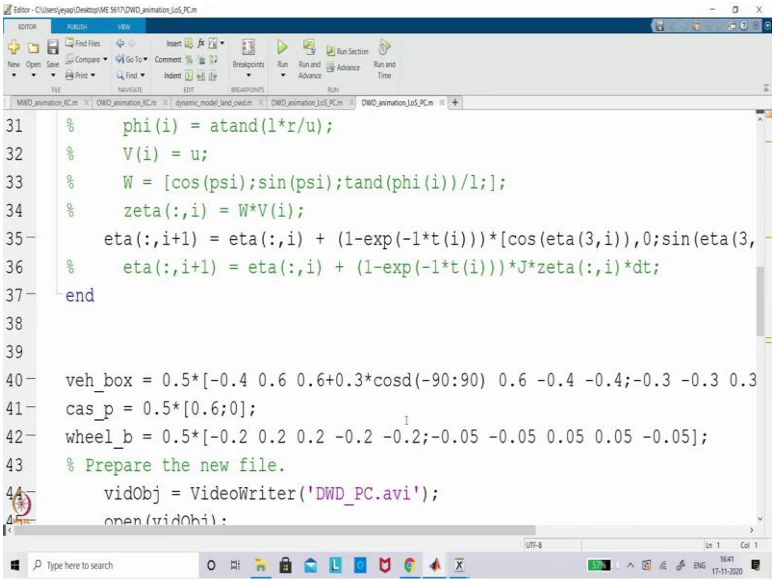
File name: DWD_animation_LoS_PC

All MATLAB files

```
)^2+(eta_d(2)-eta(2,i))^2;

)-eta(2,i),eta_d(1)-eta(1,i))] - eta(:,

%     if rho<0.05
%         e = eta_d- eta(:,i);
% %         e=0*e;
%     end
%
    psi = eta(3,i);
    J = [cos(psi) -sin(psi) 0;sin(psi) cos(psi) 0;0 0 1];
```

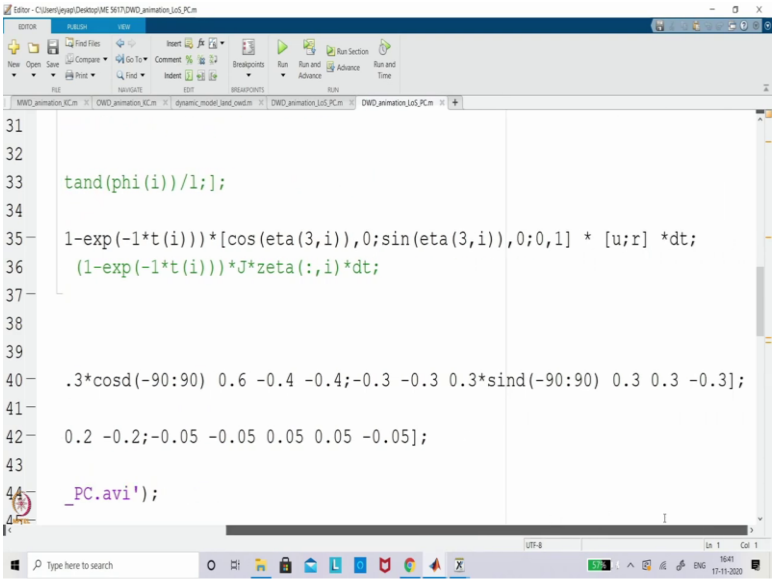
So, now if I give probably a positional coordinate as a polar coordinate how that would work? So, for that also I will actually like take another file which I have already written. So, I will actually like take that ok.

(Refer Slide Time: 06:10)



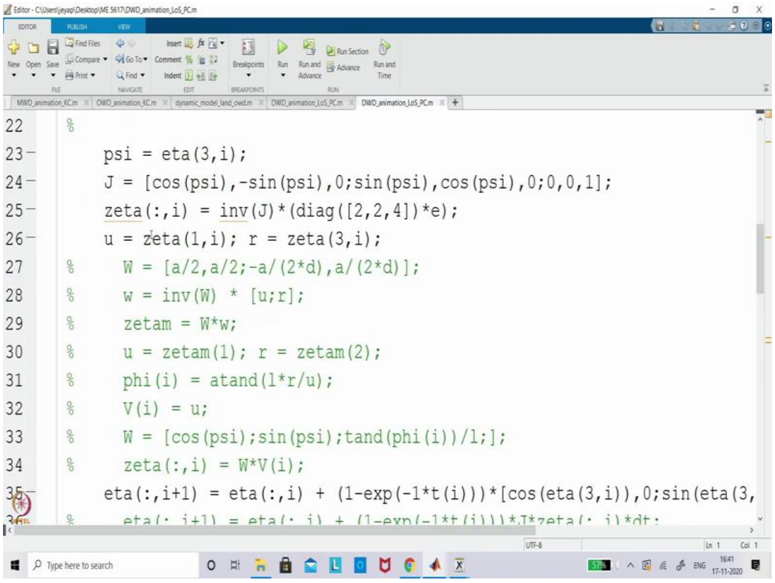
```
%     phi(i) = atand(l*r/u);
%     V(i) = u;
%     W = [cos(psi);sin(psi);tand(phi(i))/l;];
%     zeta(:,i) = W*V(i);
    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,
%     eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*J*zeta(:,i)*dt;
end


veh_box = 0.5*[-0.4 0.6 0.6+0.3*cosd(-90:90) 0.6 -0.4 -0.4;-0.3 -0.3 0.3
cas_p = 0.5*[0.6;0];
wheel_b = 0.5*[-0.2 0.2 0.2 -0.2 -0.2;-0.05 -0.05 0.05 0.05 -0.05];
% Prepare the new file.
    vidObj = VideoWriter('DWD_PC.avi');
    open(vidObj);
```

(Refer Slide Time: 06:11)



```


tand(phi(i))/l;];

1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,i)),0;0,1] * [u;r] *dt;
 (1-exp(-1*t(i)))*J*zeta(:,i)*dt;




.3*cosd(-90:90) 0.6 -0.4 -0.4;-0.3 -0.3 0.3*sind(-90:90) 0.3 0.3 -0.3];

0.2 -0.2;-0.05 -0.05 0.05 0.05 -0.05];

_PC.avi');
```

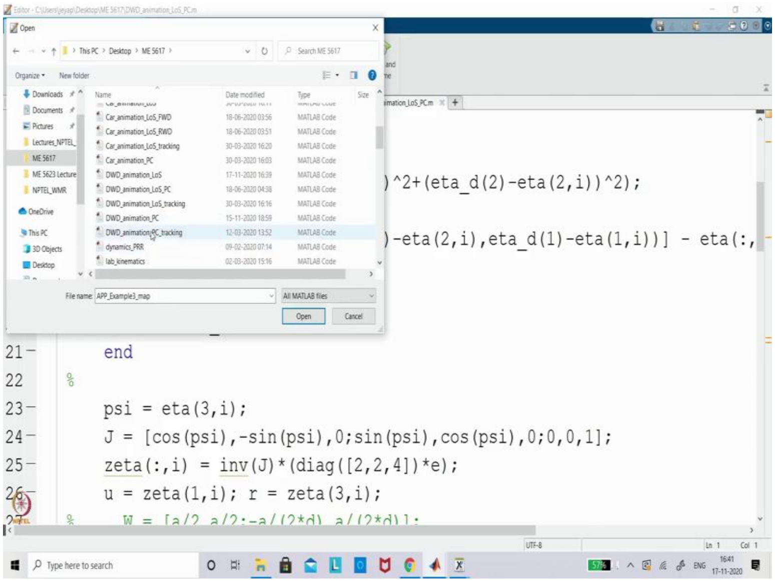
(Refer Slide Time: 06:14)



```
%
    psi = eta(3,i);
    J = [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1];
    zeta(:,i) = inv(J)*(diag([2,2,4])*e);
    u = zeta(1,i); r = zeta(3,i);
%     W = [a/2,a/2;-a/(2*d),a/(2*d)];
%     w = inv(W) * [u;r];
%     zetam = W*w;
%     u = zetam(1); r = zetam(2);
%     phi(i) = atand(l*r/u);
%     V(i) = u;
%     W = [cos(psi);sin(psi);tand(phi(i))/l;];
%     zeta(:,i) = W*V(i);
    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,
%     eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*J*zeta(:,i)*dt;
```

So, here you can see that the u and r. So, I am actually like taking as a polar coordinate. So, and then I am actually like trying to control that. So, and this is also like same thing.

(Refer Slide Time: 06:25)



This PC › Desktop › ME 5617 ›

Name	Date modified	Type
Car_animation_LoS_FWD	18-06-2020 03:56	MATLAB Code
Car_animation_LoS_RWD	18-06-2020 03:51	MATLAB Code
Car_animation_LoS_tracking	30-03-2020 16:20	MATLAB Code
Car_animation_PC	30-03-2020 16:03	MATLAB Code
DWD_animation_LoS	17-11-2020 16:39	MATLAB Code
DWD_animation_LoS_PC	18-06-2020 04:38	MATLAB Code
DWD_animation_LoS_tracking	30-03-2020 16:16	MATLAB Code
DWD_animation_PC	15-11-2020 18:59	MATLAB Code
DWD_animation_PC_tracking	12-03-2020 13:52	MATLAB Code
dynamics_PRR	09-02-2020 07:14	MATLAB Code
lab_kinematics	02-03-2020 15:16	MATLAB Code

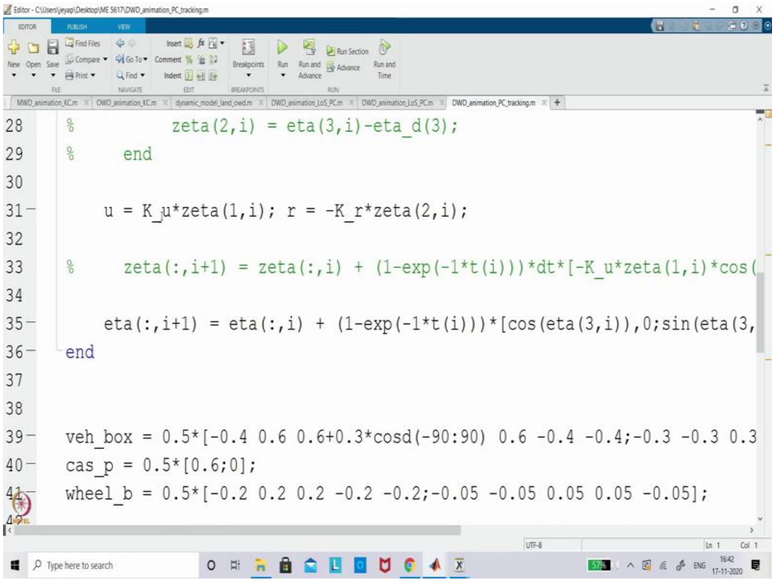
File name: APP_Example3_map

All MATLAB files

```
)^2+(eta_d(2)-eta(2,i))^2);

)-eta(2,i),eta_d(1)-eta(1,i))] - eta(:,

    end
%
    psi = eta(3,i);
    J = [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1];
    zeta(:,i) = inv(J)*(diag([2,2,4])*e);
    u = zeta(1,i); r = zeta(3,i);
%     W = [a/2,a/2;-a/(2*d),a/(2*d)];
```

So, it is a also line of sight I just see that particular file yeah.

(Refer Slide Time: 06:38)

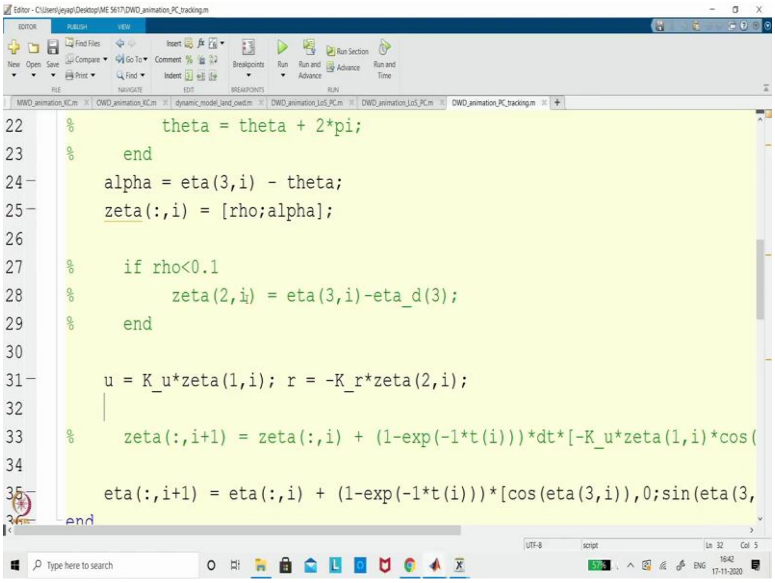


Editor - C:\Users\jeyap\Desktop\ME 5617\DWD_animation_PC_tracking.m

```
%           zeta(2,i) = eta(3,i)-eta_d(3);
%       end

    u = K_u*zeta(1,i); r = -K_r*zeta(2,i);

%       zeta(:,i+1) = zeta(:,i) + (1-exp(-1*t(i)))*dt*[-K_u*zeta(1,i)*cos(

    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,
end


veh_box = 0.5*[-0.4 0.6 0.6+0.3*cosd(-90:90) 0.6 -0.4 -0.4;-0.3 -0.3 0.3
cas_p = 0.5*[0.6;0];
wheel_b = 0.5*[-0.2 0.2 0.2 -0.2 -0.2;-0.05 -0.05 0.05 0.05 -0.05];
```

So, now you can see that this is what we have given right. So, only thing here the sign is actually like interchange the K_u sign why it is so? Because you want actually like reduce it. Now the error is actually like ρ. You are trying to reduce the error. So, in the sense you have to rewrite the entire equation in the other form.

(Refer Slide Time: 06:57)

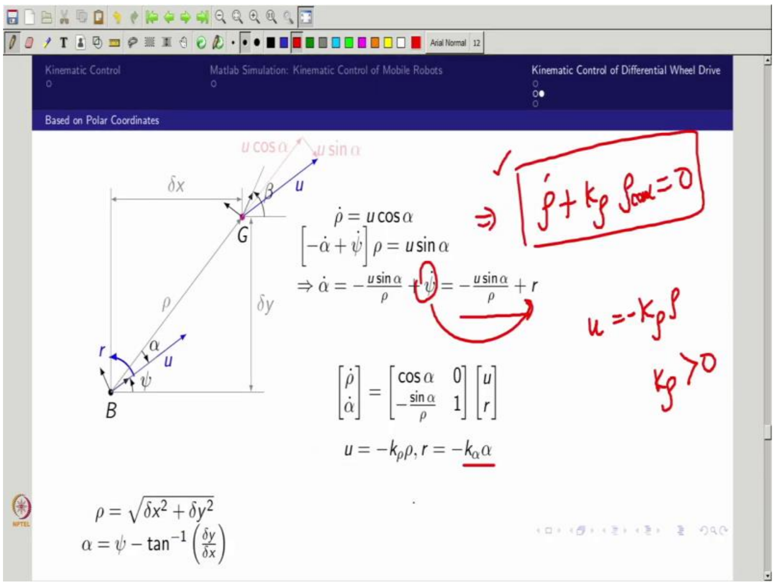


Editor - C:\Users\jeyap\Desktop\ME 5617\DWD_animation_PC_tracking.m

```
%           theta = theta + 2*pi;
%       end
    alpha = eta(3,i) - theta;
    zeta(:,i) = [rho;alpha];

%       if rho<0.1
%           zeta(2,i) = eta(3,i)-eta_d(3);
%       end

    u = K_u*zeta(1,i); r = -K_r*zeta(2,i);

%       zeta(:,i+1) = zeta(:,i) + (1-exp(-1*t(i)))*dt*[-K_u*zeta(1,i)*cos(

    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,
end
```

So, then the K_u would be positive. So, if you recall this particular you can see equation.

(Refer Slide Time: 07:05)



Kinematic Control

Matlab Simulation: Kinematic Control of Mobile Robots

Kinematic Control of Differential Wheel Drive

Based on Polar Coordinates

$$\dot{\rho} = u\cos\alpha$$

$$\left[-\dot{\alpha}+\dot{\psi}\right]\rho = u\sin\alpha$$

$$\Rightarrow \dot{\alpha} = -\frac{u\sin\alpha}{\rho} + \dot{\psi} = -\frac{u\sin\alpha}{\rho} + r$$

$$\Rightarrow \boxed{\dot{\rho} + k_\rho \rho_{\cos} = 0}$$

$$u = -k_\rho \rho$$

$$k_\rho > 0$$

$$\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\end{bmatrix} = \begin{bmatrix}\cos\alpha & 0\\ -\frac{\sin\alpha}{\rho} & 1\end{bmatrix}\begin{bmatrix}u\\ r\end{bmatrix}$$

$$u = -k_\rho \rho,\ r = -\underline{k_\alpha \alpha}$$

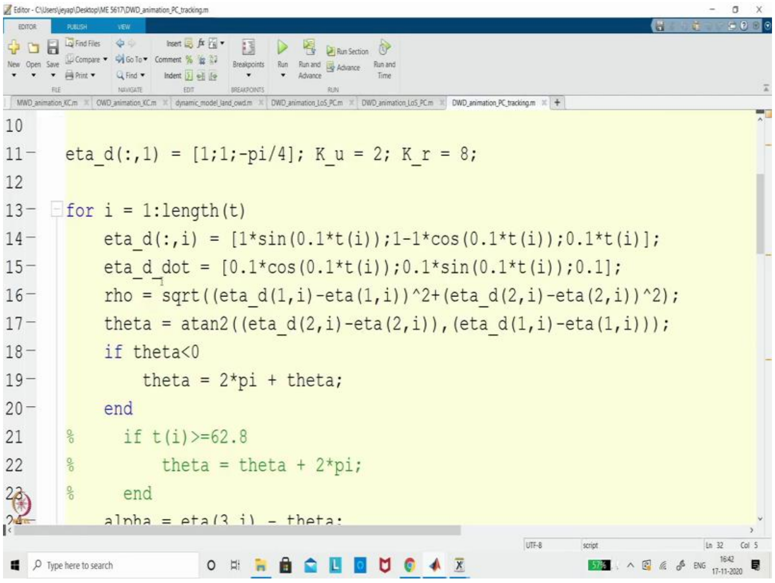
$$\rho = \sqrt{\delta x^2 + \delta y^2}$$

$$\alpha = \psi - \tan^{-1}\left(\frac{\delta y}{\delta x}\right)$$

So, you can see this equation. So, this equation u is actually like we have written as - K_ρ × ρ, but what we are actually like seeing? This ρ is actually like a positive, but what we are interested? So, this point actually like supposed to reach from here so, it is in the opposite direction.

So, that is why there is a negative sign coming into here ok. So, in the sense u I have written as - K_u × ξ, but right now I have put K_u × ξ. You can actually like ran the simulation and you will get your own idea.

(Refer Slide Time: 07:34)

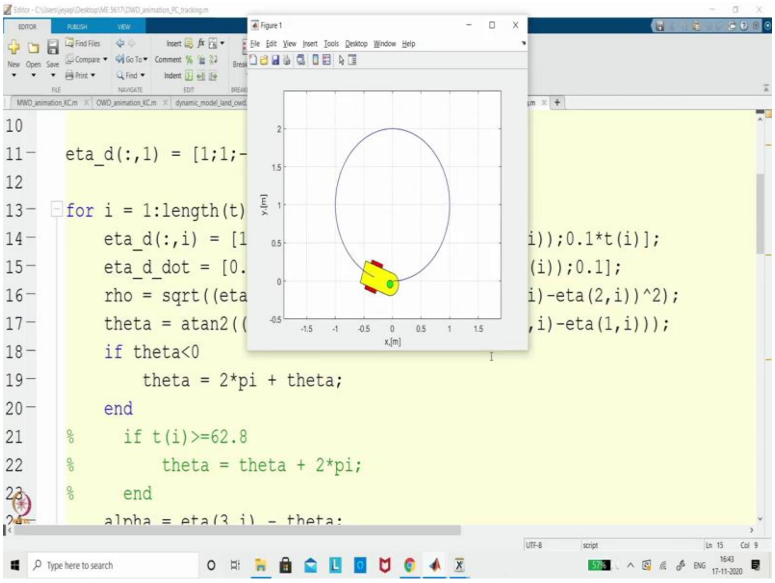


Editor - C:\Users\jeyap\Desktop\ME 5617\DWD_animation_PC_tracking.m

```
eta_d(:,1) = [1;1;-pi/4]; K_u = 2; K_r = 8;

for i = 1:length(t)
    eta_d(:,i) = [1*sin(0.1*t(i));1-1*cos(0.1*t(i));0.1*t(i)];
    eta_d_dot = [0.1*cos(0.1*t(i));0.1*sin(0.1*t(i));0.1];
    rho = sqrt((eta_d(1,i)-eta(1,i))^2+(eta_d(2,i)-eta(2,i))^2);
    theta = atan2((eta_d(2,i)-eta(2,i)),(eta_d(1,i)-eta(1,i)));
    if theta<0
        theta = 2*pi + theta;
    end
%     if t(i)>=62.8
%         theta = theta + 2*pi;
%     end
    alpha = eta(3,i) - theta;
```

Now I am actually like giving the eta desired also like some kind of circular path. So, in the sense $\dot{\eta}_d$ also coming. So, if I run this so what one can actually like see it? It is actually trying to follow a circular path you can see right.

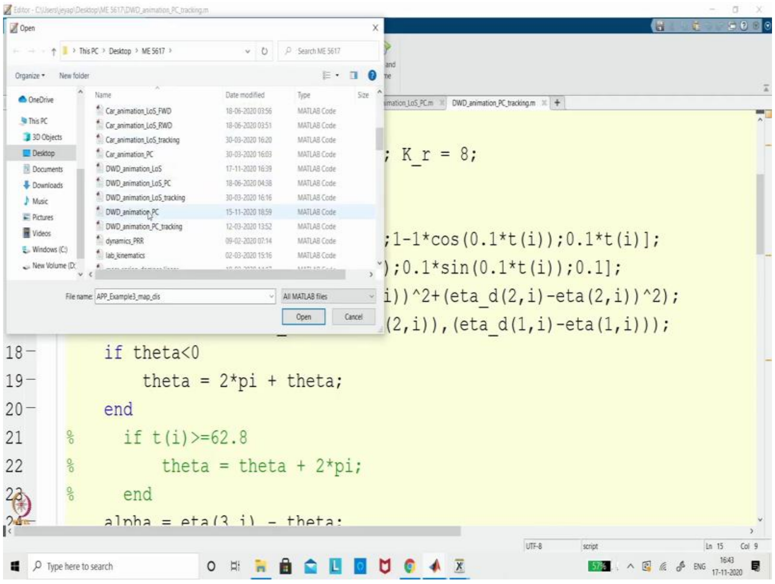
(Refer Slide Time: 07:45)



```
eta_d(:,1) = [1;1;-
for i = 1:length(t)
    eta_d(:,i) = [1                              i));0.1*t(i)];
    eta_d_dot = [0.                              (i));0.1];
    rho = sqrt((eta                              i)-eta(2,i))^2);
    theta = atan2((                              ,i)-eta(1,i)));
    if theta<0
        theta = 2*pi + theta;
    end
%     if t(i)>=62.8
%         theta = theta + 2*pi;
%     end
    alpha = eta(3,i) - theta;
```

But if I give a orientation component it may not follow. So, for that you can actually like see. So, although I have given 0.1 times of t as like this, but if I give probably $\frac{pi}{4}$ and all it may not actually like follow.

(Refer Slide Time: 08:04)



Name	Date modified	Type	Size
Car_animation_LoS_FWD	18-06-2020 03:56	MATLAB Code	
Car_animation_LoS_RWD	18-06-2020 03:51	MATLAB Code	
Car_animation_LoS_tracking	30-03-2020 16:20	MATLAB Code	
Car_animation_PC	30-03-2020 16:03	MATLAB Code	
DWD_animation_LoS	17-11-2020 16:39	MATLAB Code	
DWD_animation_LoS_PC	18-06-2020 04:38	MATLAB Code	
DWD_animation_LoS_tracking	30-03-2020 16:16	MATLAB Code	
DWD_animation_PC	15-11-2020 18:59	MATLAB Code	
DWD_animation_PC_tracking	12-03-2020 13:52	MATLAB Code	
dynamics_PRR	09-02-2020 07:14	MATLAB Code	
lab_kinematics	02-03-2020 15:16	MATLAB Code	

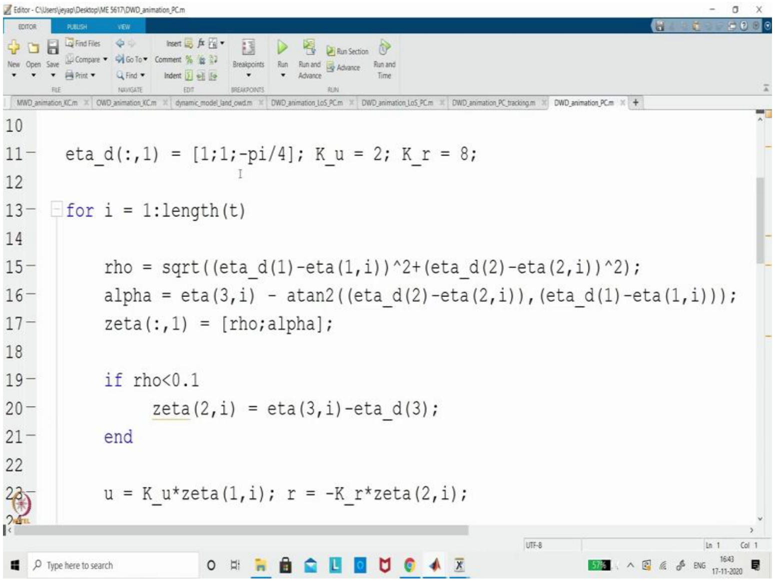
File name: APP_Example3_map_dis

All MATLAB files

```
; K_r = 8;

;1-1*cos(0.1*t(i));0.1*t(i)];
);0.1*sin(0.1*t(i));0.1];
i))^2+(eta_d(2,i)-eta(2,i))^2);
(2,i)),(eta_d(1,i)-eta(1,i)));
    if theta<0
        theta = 2*pi + theta;
    end
%     if t(i)>=62.8
%         theta = theta + 2*pi;
%     end
    alpha = eta(3,i) - theta;
```

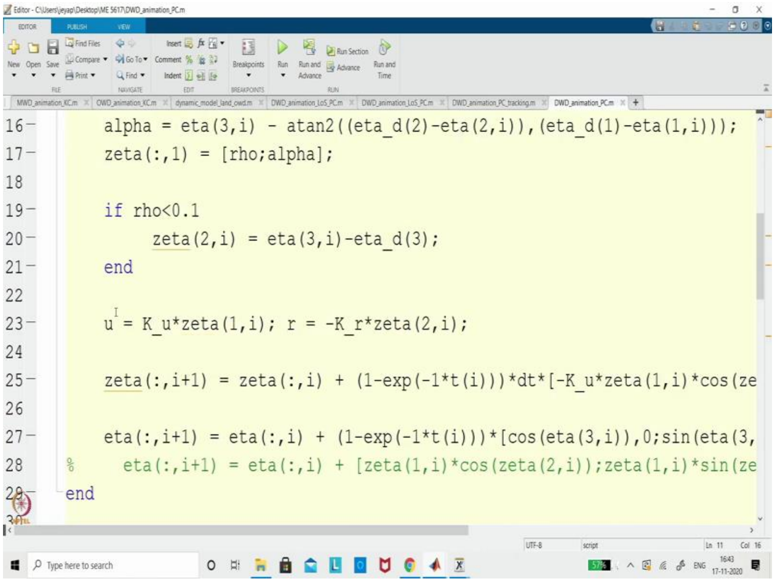
So, in order to get that so, I will actually like give that also here. So, then we will actually like take the other you can see configurations.

(Refer Slide Time: 08:14)



```
eta_d(:,1) = [1;1;-pi/4]; K_u = 2; K_r = 8;

for i = 1:length(t)

    rho = sqrt((eta_d(1)-eta(1,i))^2+(eta_d(2)-eta(2,i))^2);
    alpha = eta(3,i) - atan2((eta_d(2)-eta(2,i)),(eta_d(1)-eta(1,i)));
    zeta(:,1) = [rho;alpha];

    if rho<0.1
        zeta(2,i) = eta(3,i)-eta_d(3);
    end

    u = K_u*zeta(1,i); r = -K_r*zeta(2,i);
```

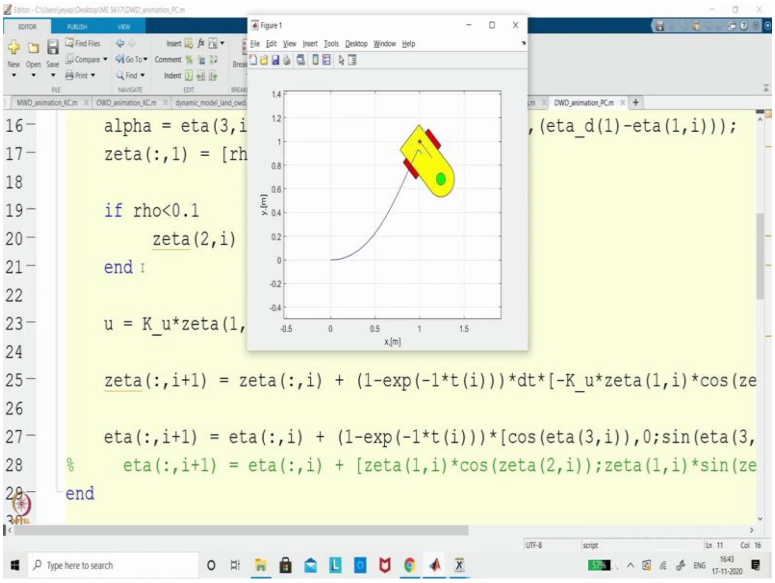
(Refer Slide Time: 08:18)



```
    alpha = eta(3,i) - atan2((eta_d(2)-eta(2,i)),(eta_d(1)-eta(1,i)));
    zeta(:,1) = [rho;alpha];

    if rho<0.1
        zeta(2,i) = eta(3,i)-eta_d(3);
    end

    u = K_u*zeta(1,i); r = -K_r*zeta(2,i);

    zeta(:,i+1) = zeta(:,i) + (1-exp(-1*t(i)))*dt*[-K_u*zeta(1,i)*cos(ze

    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,
%     eta(:,i+1) = eta(:,i) + [zeta(1,i)*cos(zeta(2,i));zeta(1,i)*sin(ze
end
```

So, you can see here. So, I am taking this 2 and I am trying to use this what you call polar coordinate form.

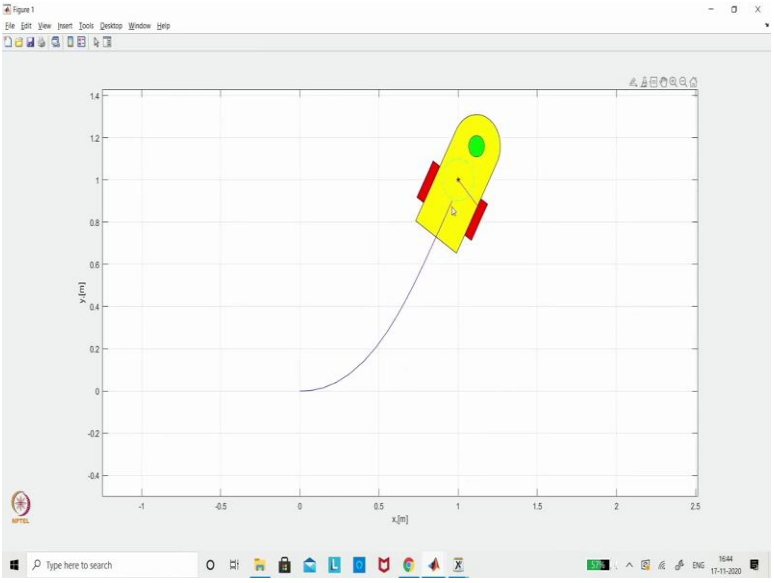
(Refer Slide Time: 08:23)



```
    alpha = eta(3,i                              ,(eta_d(1)-eta(1,i)));
    zeta(:,1) = [rh

    if rho<0.1
        zeta(2,i)
    end

    u = K_u*zeta(1,

    zeta(:,i+1) = zeta(:,i) + (1-exp(-1*t(i)))*dt*[-K_u*zeta(1,i)*cos(ze

    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,
%     eta(:,i+1) = eta(:,i) + [zeta(1,i)*cos(zeta(2,i));zeta(1,i)*sin(ze
end
```

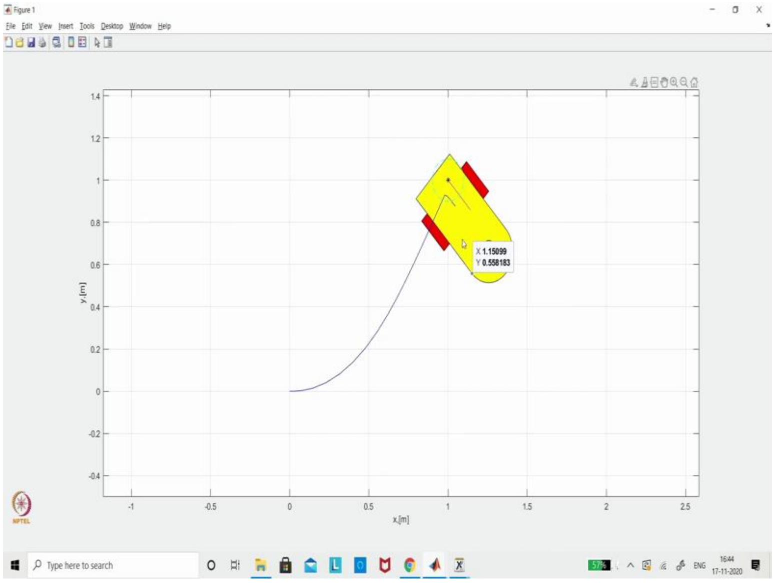
So, now if I actually I run this you can see it is actually like going the set point which is $\frac{1}{1}$ and $\frac{pi}{4}$. And I have actually like changed the condition here the ρ is actually like up to 10 centimeter you follow only line following after that you try to follow as a orientation. So, that is what it is actually like doing it.

(Refer Slide Time: 08:43)



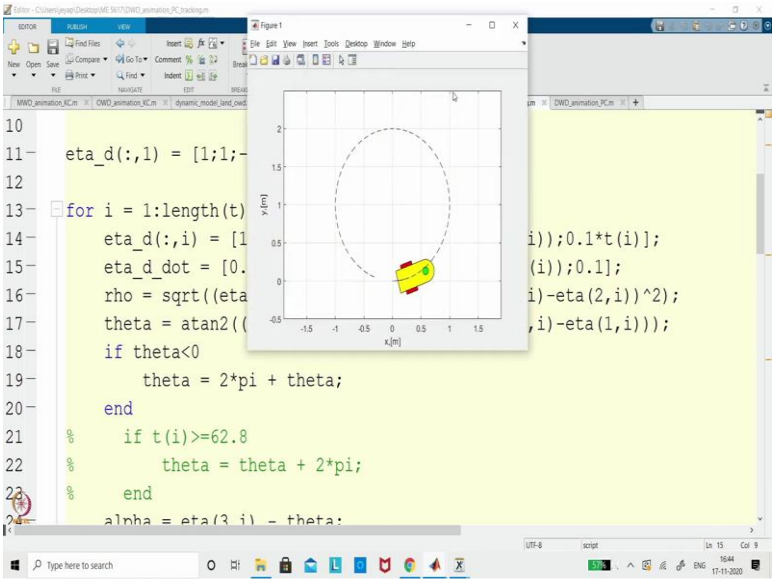
So, you can see like it start from 0 0 and try to go 1 and 1 and it is actually like trying to reach.

(Refer Slide Time: 08:49)



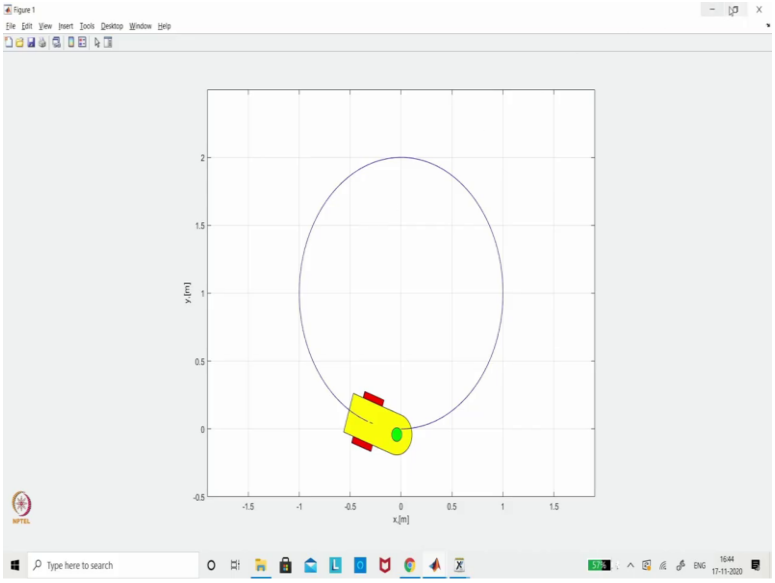
Once that circle of 10 centimeter reach it is trying to follow the orientation, but both are not directly controllable that is why there is a compromise between the position and orientation. Now you got it like what is nonholonomic vehicle, where you can apply right. If you are actually thinking about position tracking it is actually like working well, but if you are thinking about orientation also need to be controlled then it cannot be done in you can say nonholonomic conditions.

(Refer Slide Time: 09:17)



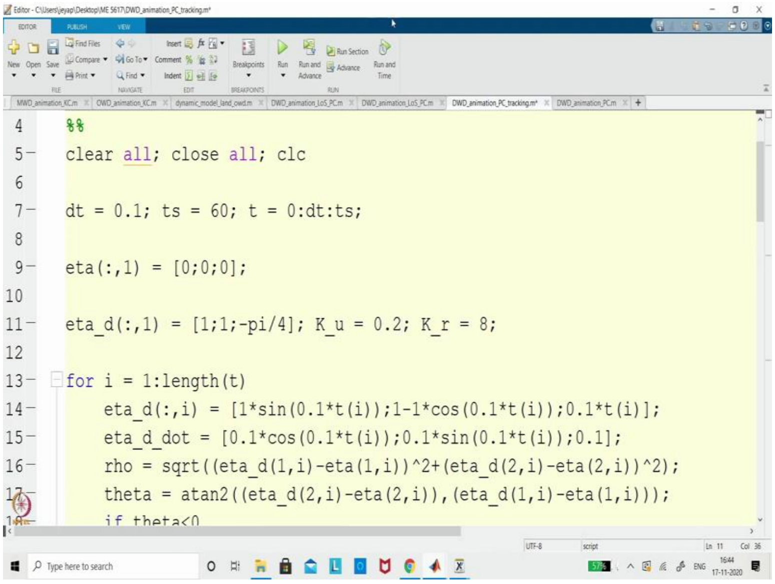
```

eta_d(:,1) = [1;1;-

for i = 1:length(t)
    eta_d(:,i) = [1                                           i));0.1*t(i)];
    eta_d_dot = [0.                                           (i));0.1];
    rho = sqrt((eta                                           i)-eta(2,i))^2);
    theta = atan2((                                           ,i)-eta(1,i)));
    if theta<0
        theta = 2*pi + theta;
    end
%     if t(i)>=62.8
%         theta = theta + 2*pi;
%     end
    alpha = eta(3,i) - theta;
```

(Refer Slide Time: 09:19)



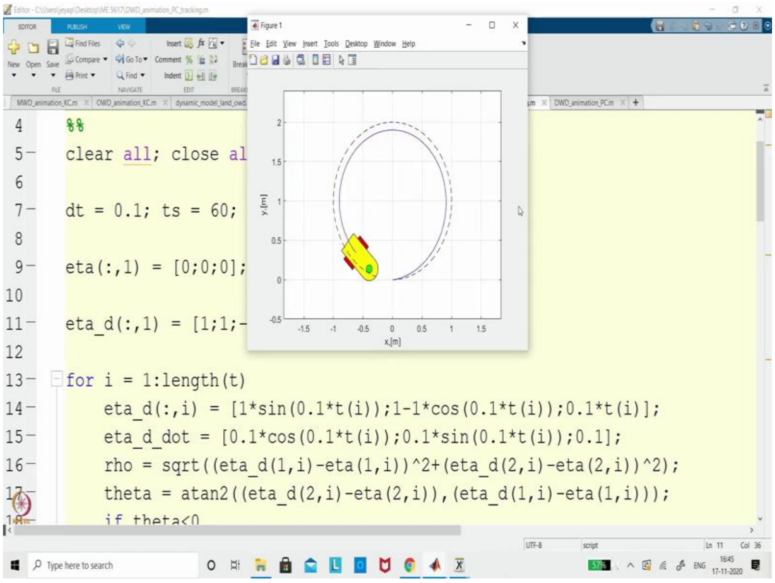
So, this is what the other case what I have shown as a circular profile following where this dotted circle is the given circular profile and this particular vehicle is actually like ask to follow it is doing it.

(Refer Slide Time: 09:30)


Editor - C:\Users\jeyap\Desktop\ME 5617\DWD_animation_PC_tracking.m*
EDITOR PUBLISH VIEW
New Open Save Find Files Compare Print Go To Find Insert Comment Indent Breakpoints Run Run and Advance Run Section Advance Run and Time
FILE NAVIGATE EDIT BREAKPOINTS RUN
MWD_animation_KC.m OWD_animation_KC.m dynamic_model_land_owd.m DWD_animation_LoS_PC.m DWD_animation_LoS_PC.m DWD_animation_PC_tracking.m* DWD_animation_PC.m
UTF-8 script Ln 11 Col 36
Type here to search

```
%%
clear all; close all; clc

dt = 0.1; ts = 60; t = 0:dt:ts;

eta(:,1) = [0;0;0];

eta_d(:,1) = [1;1;-pi/4]; K_u = 0.2; K_r = 8;

for i = 1:length(t)
    eta_d(:,i) = [1*sin(0.1*t(i));1-1*cos(0.1*t(i));0.1*t(i)];
    eta_d_dot = [0.1*cos(0.1*t(i));0.1*sin(0.1*t(i));0.1];
    rho = sqrt((eta_d(1,i)-eta(1,i))^2+(eta_d(2,i)-eta(2,i))^2);
    theta = atan2((eta_d(2,i)-eta(2,i)),(eta_d(1,i)-eta(1,i)));
    if theta<0
```

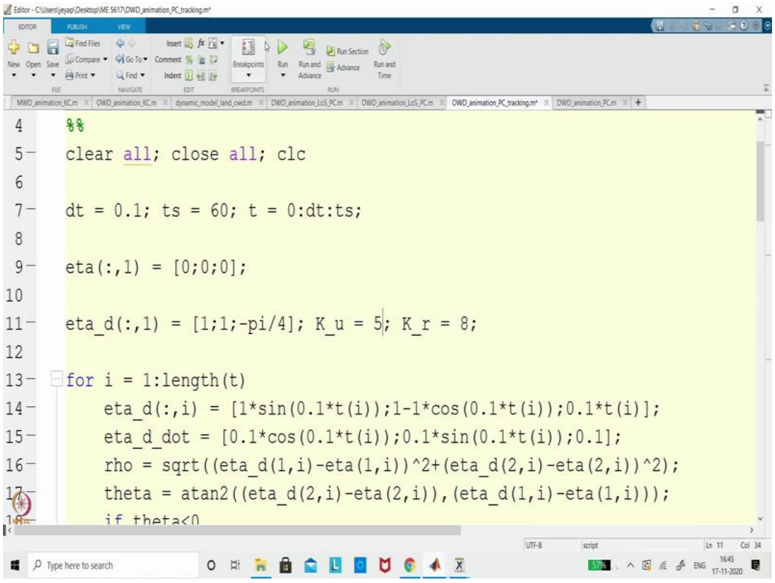
(Refer Slide Time: 09:35)



```
%%
clear all; close al

dt = 0.1; ts = 60;

eta(:,1) = [0;0;0];

eta_d(:,1) = [1;1;-

for i = 1:length(t)
    eta_d(:,i) = [1*sin(0.1*t(i));1-1*cos(0.1*t(i));0.1*t(i)];
    eta_d_dot = [0.1*cos(0.1*t(i));0.1*sin(0.1*t(i));0.1];
    rho = sqrt((eta_d(1,i)-eta(1,i))^2+(eta_d(2,i)-eta(2,i))^2);
    theta = atan2((eta_d(2,i)-eta(2,i)),(eta_d(1,i)-eta(1,i)));
    if theta<0
```

So, now you can actually like vary the control gain value for example, I am giving very slow control in the x you can say. So, you can see right. So, the error is actually like prolonged and it is actually like not you can say control why? Because the steady state error is still exist.

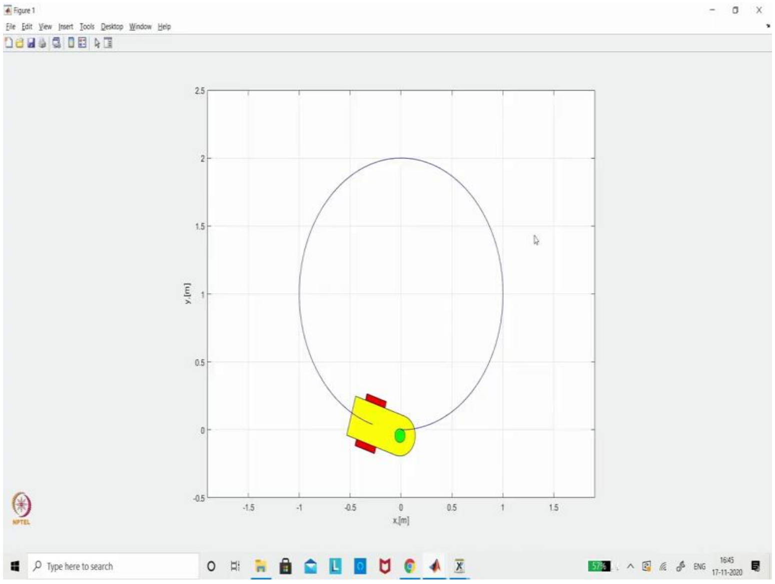
(Refer Slide Time: 09:47)



```
%%
clear all; close all; clc

dt = 0.1; ts = 60; t = 0:dt:ts;

eta(:,1) = [0;0;0];

eta_d(:,1) = [1;1;-pi/4]; K_u = 5; K_r = 8;

for i = 1:length(t)
    eta_d(:,i) = [1*sin(0.1*t(i));1-1*cos(0.1*t(i));0.1*t(i)];
    eta_d_dot = [0.1*cos(0.1*t(i));0.1*sin(0.1*t(i));0.1];
    rho = sqrt((eta_d(1,i)-eta(1,i))^2+(eta_d(2,i)-eta(2,i))^2);
    theta = atan2((eta_d(2,i)-eta(2,i)),(eta_d(1,i)-eta(1,i)));
    if theta<0
```

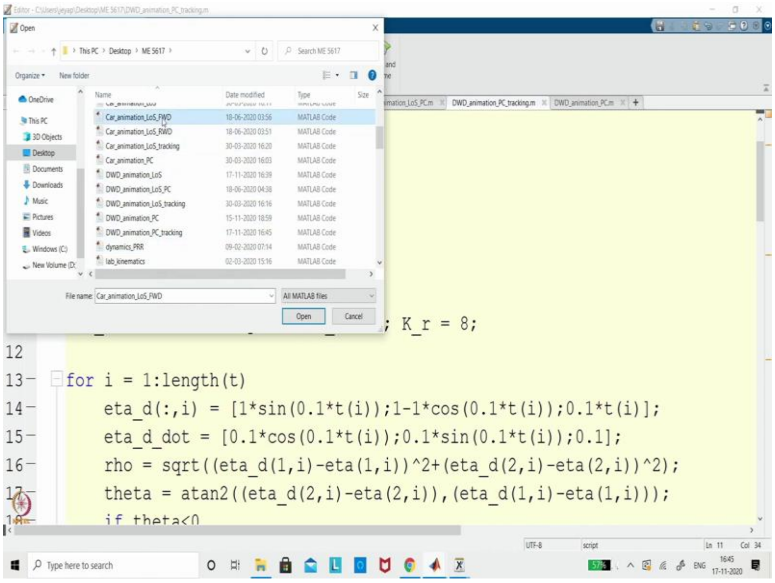
So, if I actually like increase this probably into 5, it would be faster right.

(Refer Slide Time: 09:50)



So, like that you can actually like see it and when you go to error profile, so, you can see that the error would be fastly converged to 0. So, these all you can actually like see it. So, this is one side where the differential wheel vehicle are coming into a picture. Now similar way you can see there are several nonholonomic vehicles are available right. So, one such vehicle is car like robot where there are 4 wheels, but all the 4 wheels are actually like not independently controlled. So, there are 2 wheels are actually like in single axle.

(Refer Slide Time: 10:32)



Name	Date modified	Type
Car_animation_LoS_FWD	18-06-2020 03:56	MATLAB Code
Car_animation_LoS_RWD	18-06-2020 03:51	MATLAB Code
Car_animation_LoS_tracking	30-03-2020 16:20	MATLAB Code
Car_animation_PC	30-03-2020 16:03	MATLAB Code
DWD_animation_LoS	17-11-2020 16:39	MATLAB Code
DWD_animation_LoS_PC	18-06-2020 04:38	MATLAB Code
DWD_animation_LoS_tracking	30-03-2020 16:16	MATLAB Code
DWD_animation_PC	15-11-2020 18:59	MATLAB Code
DWD_animation_PC_tracking	17-11-2020 16:45	MATLAB Code
dynamics_PRR	09-02-2020 07:14	MATLAB Code
lab_kinematics	02-03-2020 15:16	MATLAB Code

File name: Car_animation_LoS_FWD

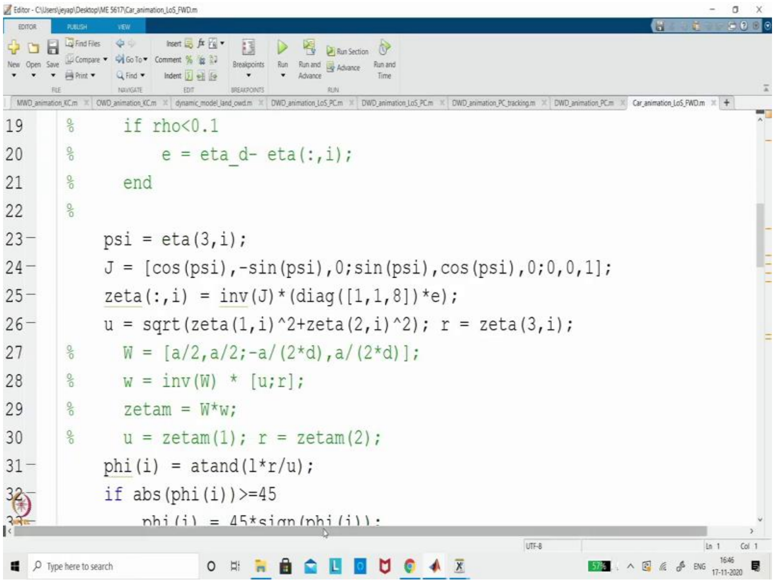
All MATLAB files

Open | Cancel

```
; K_r = 8;

for i = 1:length(t)
    eta_d(:,i) = [1*sin(0.1*t(i));1-1*cos(0.1*t(i));0.1*t(i)];
    eta_d_dot = [0.1*cos(0.1*t(i));0.1*sin(0.1*t(i));0.1];
    rho = sqrt((eta_d(1,i)-eta(1,i))^2+(eta_d(2,i)-eta(2,i))^2);
    theta = atan2((eta_d(2,i)-eta(2,i)),(eta_d(1,i)-eta(1,i)));
    if theta<0
```

So, in the sense in that one particular axle would be controlled with a traction and the same wheel can be steerable or vice versa ok. In that sense what I can do? I can actually take it that car scenario where the forward wheel drive line of sight method I am taking. So, now you can see that the W matrix would get changed.

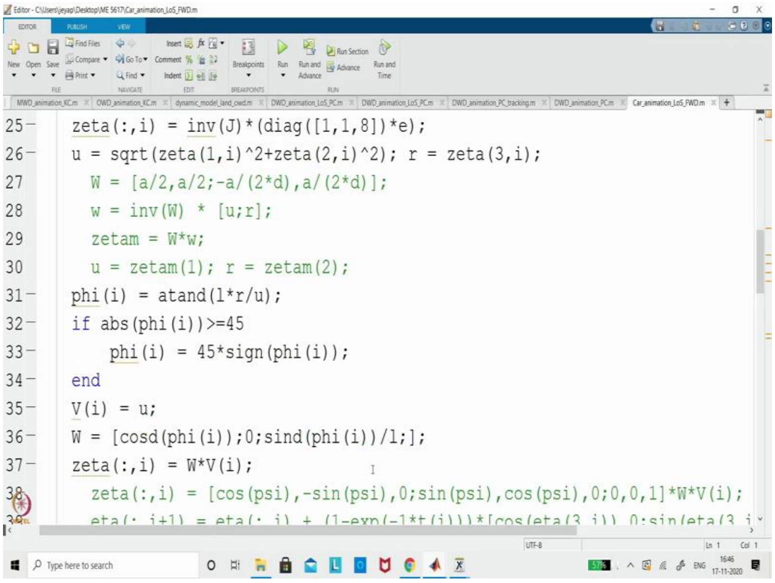
(Refer Slide Time: 10:41)



Editor - C:\Users\jeyap\Desktop\ME 5617\Car_animation_LoS_FWD.m

```
%     if rho<0.1
%         e = eta_d- eta(:,i);
%     end
%
    psi = eta(3,i);
    J = [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1];
    zeta(:,i) = inv(J)*(diag([1,1,8])*e);
    u = sqrt(zeta(1,i)^2+zeta(2,i)^2); r = zeta(3,i);
%     W = [a/2,a/2;-a/(2*d),a/(2*d)];
%     w = inv(W) * [u;r];
%     zetam = W*w;
%     u = zetam(1); r = zetam(2);
    phi(i) = atand(l*r/u);
    if abs(phi(i))>=45
        phi(i) = 45*sign(phi(i));
```

(Refer Slide Time: 10:45)



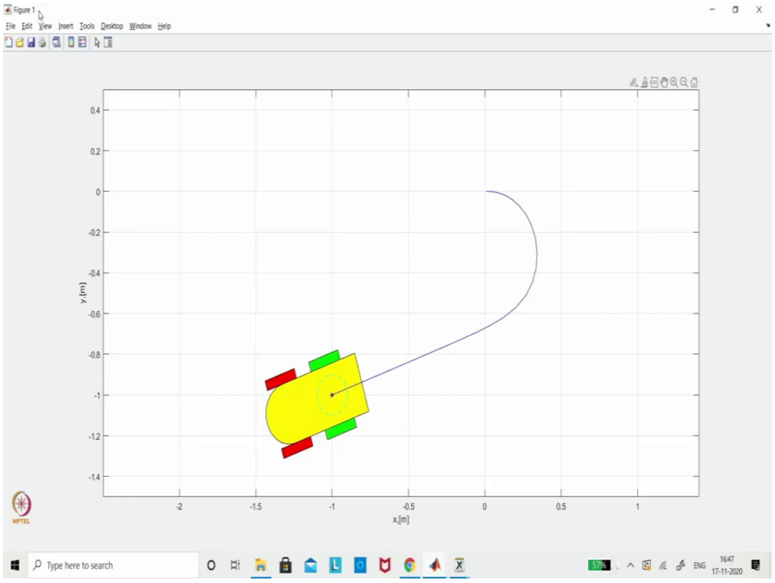
Editor - C:\Users\jeyap\Desktop\ME 5617\Car_animation_LoS_FWD.m

```
zeta(:,i) = inv(J)*(diag([1,1,8])*e);
u = sqrt(zeta(1,i)^2+zeta(2,i)^2); r = zeta(3,i);
  W = [a/2,a/2;-a/(2*d),a/(2*d)];
  w = inv(W) * [u;r];
  zetam = W*w;
  u = zetam(1); r = zetam(2);
phi(i) = atand(l*r/u);
if abs(phi(i))>=45
    phi(i) = 45*sign(phi(i));
end
V(i) = u;
W = [cosd(phi(i));0;sind(phi(i))/l;];
zeta(:,i) = W*V(i);
  zeta(:,i) = [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1]*W*V(i);
  eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(3,i
```

So, that you can actually like recall what we did in the what you call in our generalized wheel model. So, this is what we have done that you can apply and then get it. So, now what you need to have? So, you need to have actually like two things one is steering angle. So, the other one is actually like you call the forward velocity. So, both we are trying to calculate based on the line-of-sight method.

So, where u and r would be calculated. So, after that, that I will substitute in the W relation and then I will actually like calculate what is your you call the angular velocity of the individual wheels ok, individual wheel and as well as the case. First, I will run the simulation then I will explain.

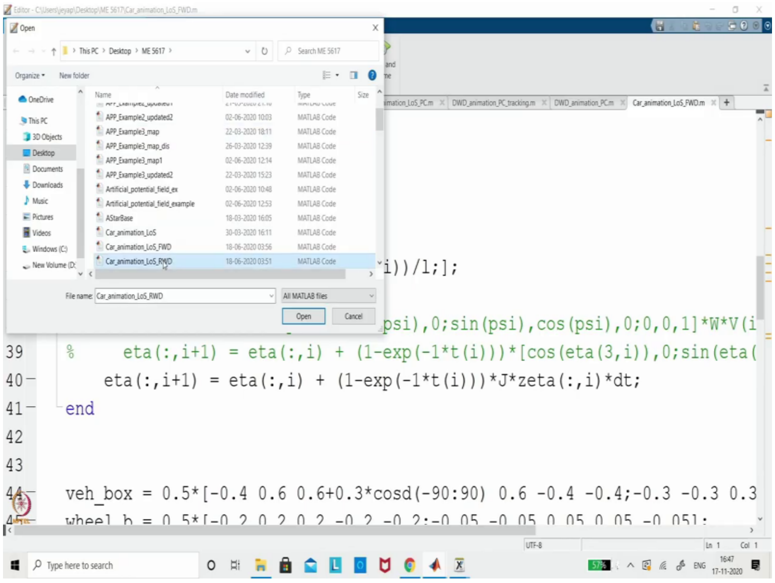
(Refer Slide Time: 11:25)



So, you can see here. So, this is the forward wheel. So, these two are powered in a single axle and this is the set point as given and it is actually trying to follow as a car what you have done right. So, now you can see that this is what we are actually like interested. This is also nonholonomic.

But here you can see that this is actually like forward wheel in the sense the both you can say forward wheel is actually like powered in a single axle and as well as steerable.

(Refer Slide Time: 11:54)



Editor - C:\Users\jeyap\Desktop\ME 5617\Car_animation_LoS_FWD.m

Open

This PC › Desktop › ME 5617 ›

Name	Date modified	Type	Size
APP_Example2_updated2	02-06-2020 10:03	MATLAB Code	
APP_Example3_map	22-03-2020 18:11	MATLAB Code	
APP_Example3_map_dis	26-03-2020 12:39	MATLAB Code	
APP_Example3_map1	02-06-2020 12:14	MATLAB Code	
APP_Example3_updated2	22-03-2020 15:23	MATLAB Code	
Artificial_potential_field_ex	02-06-2020 10:48	MATLAB Code	
Artificial_potential_field_example	02-06-2020 12:53	MATLAB Code	
AStarBase	18-03-2020 16:05	MATLAB Code	
Car_animation_LoS	30-03-2020 16:11	MATLAB Code	
Car_animation_LoS_FWD	18-06-2020 03:56	MATLAB Code	
Car_animation_LoS_RWD	18-06-2020 03:51	MATLAB Code	

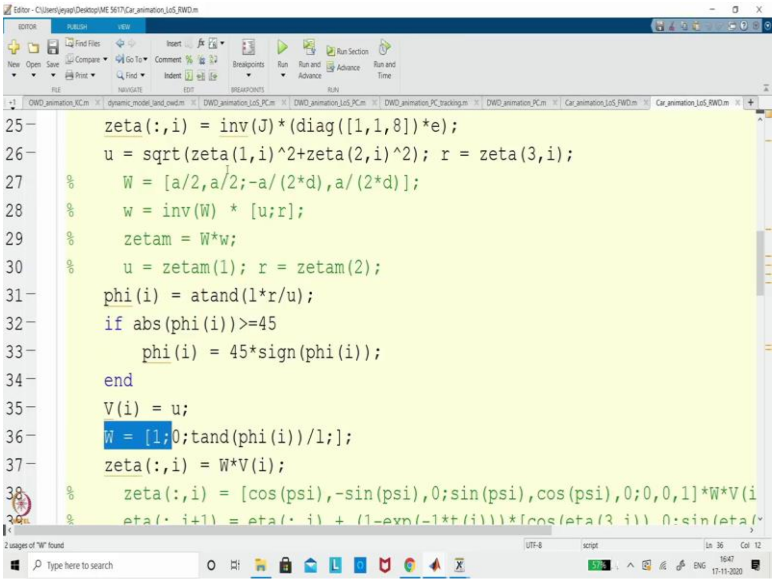
File name: Car_animation_LoS_RWD

All MATLAB files

```
i))/l;];

psi),0;sin(psi),cos(psi),0;0,0,1]*W*V(i
%     eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(
    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*J*zeta(:,i)*dt;
end


veh_box = 0.5*[-0.4 0.6 0.6+0.3*cosd(-90:90) 0.6 -0.4 -0.4;-0.3 -0.3 0.3
wheel_b = 0.5*[-0.2 0.2 0.2 -0.2 -0.2;-0.05 -0.05 0.05 0.05 -0.05];
```

The same thing we can actually like do it for what you call the rear wheel drive.

(Refer Slide Time: 12:00)



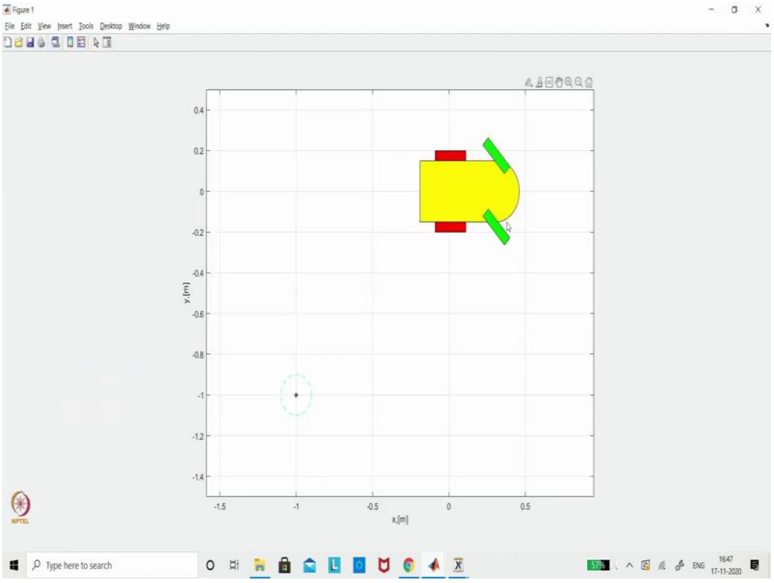
Editor - C:\Users\jeyap\Desktop\ME 5617\Car_animation_LoS_RWD.m

```
    zeta(:,i) = inv(J)*(diag([1,1,8])*e);
    u = sqrt(zeta(1,i)^2+zeta(2,i)^2); r = zeta(3,i);
%     W = [a/2,a/2;-a/(2*d),a/(2*d)];
%     w = inv(W) * [u;r];
%     zetam = W*w;
%     u = zetam(1); r = zetam(2);
    phi(i) = atand(l*r/u);
    if abs(phi(i))>=45
        phi(i) = 45*sign(phi(i));
    end
    V(i) = u;
    W = [1;0;tand(phi(i))/l;];
    zeta(:,i) = W*V(i);
%     zeta(:,i) = [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1]*W*V(i
%     eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(
```

2 usages of "W" found

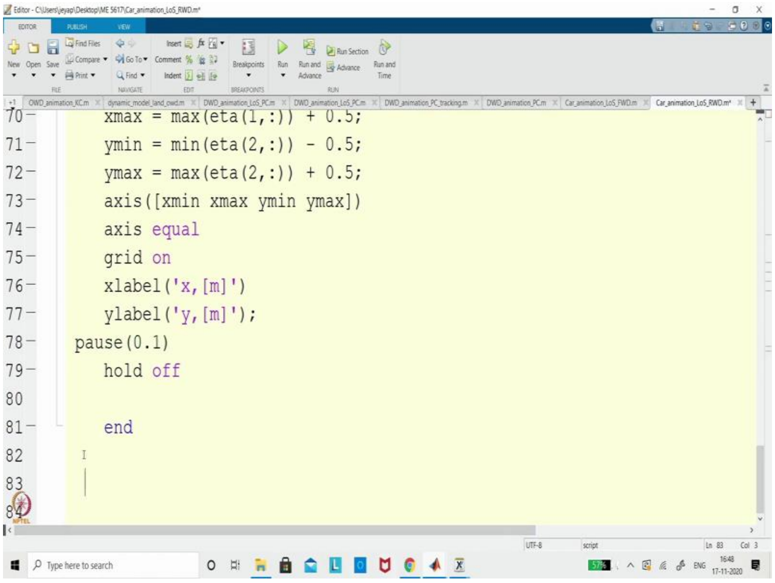
So, this is actually like rear wheel drive. Then you can see that the W matrix slightly modified where you compare to the forward wheel drive ok. There would be tanϕ would come. So, I know like you know by this time. So, how to derive this what you call generalized wheel model how to obtain this W for forward and rear. So, assuming that these two are available and now you can actually like get it.

(Refer Slide Time: 12:23)



So, the same situation I have given, but it is actually like the front wheel is actually steering connected and the rear wheel is actually like powered. So, now this is actually like trying to you call follow it. So, we will actually like see this is actually like taking time ok. So, I will actually like ok. So, these are there.

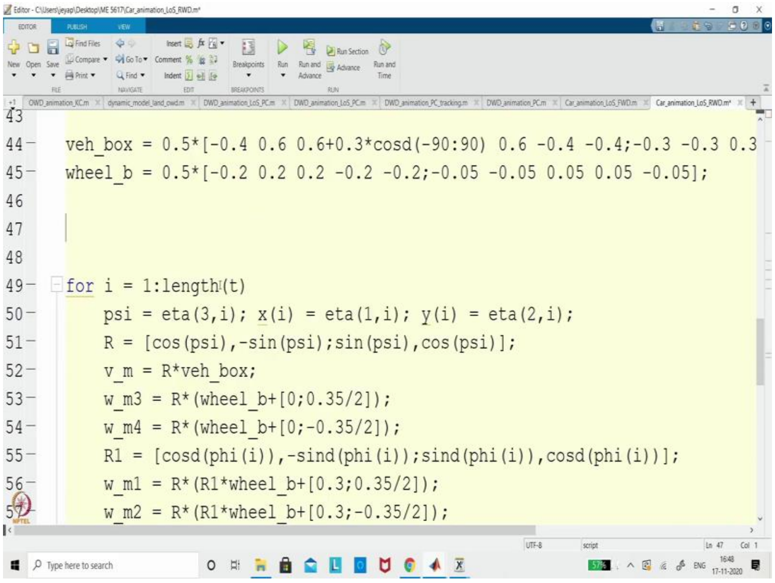
(Refer Slide Time: 12:40)



```
    xmax = max(eta(1,:)) + 0.5;
    ymin = min(eta(2,:)) - 0.5;
    ymax = max(eta(2,:)) + 0.5;
    axis([xmin xmax ymin ymax])
    axis equal
    grid on
    xlabel('x,[m]')
    ylabel('y,[m]');
pause(0.1)
    hold off

    end


```

So, I do not want to actually like record it. So, this is actually like what you call video recording.

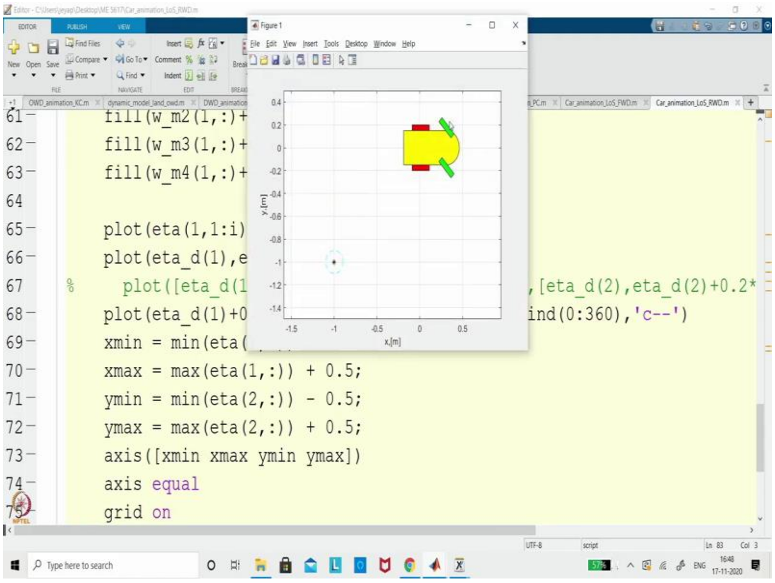
(Refer Slide Time: 12:47)



```
veh_box = 0.5*[-0.4 0.6 0.6+0.3*cosd(-90:90) 0.6 -0.4 -0.4;-0.3 -0.3 0.3
wheel_b = 0.5*[-0.2 0.2 0.2 -0.2 -0.2;-0.05 -0.05 0.05 0.05 -0.05];



for i = 1:length(t)
    psi = eta(3,i); x(i) = eta(1,i); y(i) = eta(2,i);
    R = [cos(psi),-sin(psi);sin(psi),cos(psi)];
    v_m = R*veh_box;
    w_m3 = R*(wheel_b+[0;0.35/2]);
    w_m4 = R*(wheel_b+[0;-0.35/2]);
    R1 = [cosd(phi(i)),-sind(phi(i));sind(phi(i)),cosd(phi(i))];
    w_m1 = R*(R1*wheel_b+[0.3;0.35/2]);
    w_m2 = R*(R1*wheel_b+[0.3;-0.35/2]);
```

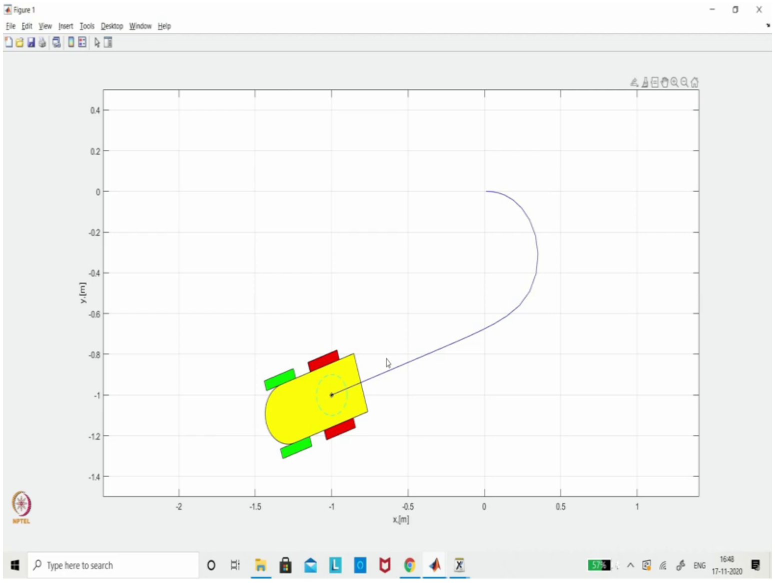
So, that is not supposed to be record here. So, I will just run this and we can actually like see it so how this will go?

(Refer Slide Time: 12:58)



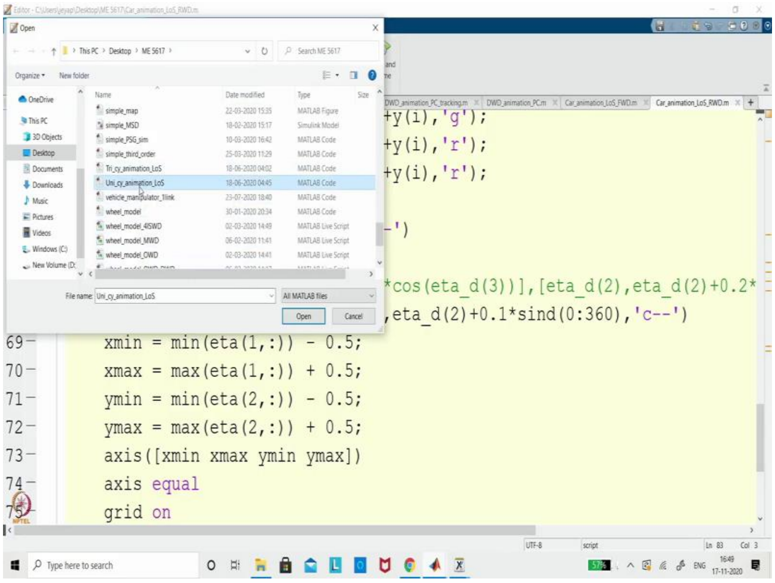
```
    fill(w_m2(1,:)+
    fill(w_m3(1,:)+
    fill(w_m4(1,:)+

    plot(eta(1,1:i)
    plot(eta_d(1),e
%     plot([eta_d(1                                  ,[eta_d(2),eta_d(2)+0.2*
    plot(eta_d(1)+0                                  ind(0:360),'c--')
    xmin = min(eta(
    xmax = max(eta(1,:)) + 0.5;
    ymin = min(eta(2,:)) - 0.5;
    ymax = max(eta(2,:)) + 0.5;
    axis([xmin xmax ymin ymax])
    axis equal
    grid on
```

(Refer Slide Time: 13:03)



So, you can see right. So, it is actually like taking a similar way, but only thing it is actually like slide variation would be in the control input because it is actually like rear wheel drive. The similar way you can even go with uni you can say cycle where only one wheel that would be connected with a few casters on the wheel base. So, that it can move it. In fact, in the last to last lecture I was discussed about this. So, we can see that.

(Refer Slide Time: 13:31)



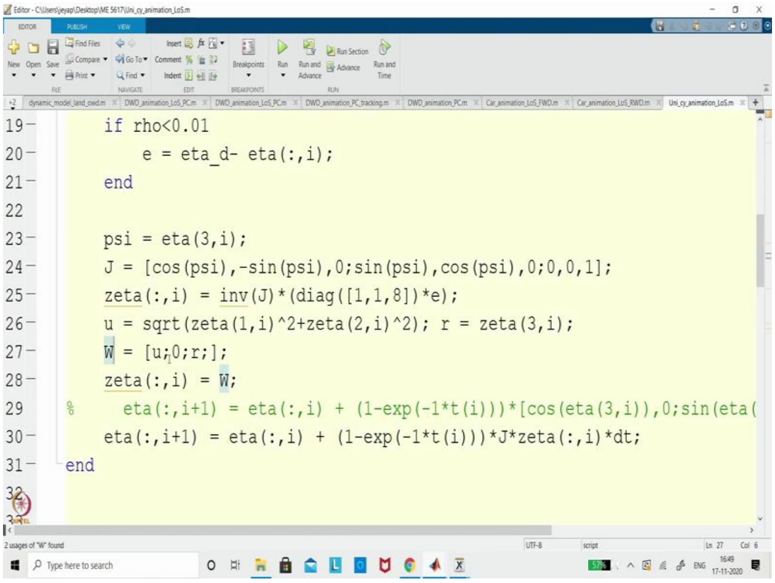
Name	Date modified	Type	Size
simple_map	22-03-2020 15:35	MATLAB Figure	
simple_MSD	18-02-2020 15:17	Simulink Model	
simple_PSG_sim	10-03-2020 16:42	MATLAB Code	
simple_third_order	25-03-2020 11:29	MATLAB Code	
Tri_cy_animation_LoS	18-06-2020 04:02	MATLAB Code	
Uni_cy_animation_LoS	18-06-2020 04:45	MATLAB Code	
vehicle_manipulator_1link	23-07-2020 18:40	MATLAB Code	
wheel_model	30-01-2020 20:34	MATLAB Code	
wheel_model_4ISWD	02-03-2020 14:49	MATLAB Live Script	
wheel_model_MWD	06-02-2020 11:41	MATLAB Live Script	
wheel_model_OWD	02-03-2020 14:41	MATLAB Live Script	

File name: Uni_cy_animation_LoS

```
+y(i),'g');
+y(i),'r');
+y(i),'r');

-')

*cos(eta_d(3))],[eta_d(2),eta_d(2)+0.2*
,eta_d(2)+0.1*sind(0:360),'c--')
    xmin = min(eta(1,:)) - 0.5;
    xmax = max(eta(1,:)) + 0.5;
    ymin = min(eta(2,:)) - 0.5;
    ymax = max(eta(2,:)) + 0.5;
    axis([xmin xmax ymin ymax])
    axis equal
    grid on
```

So, far that I am actually taking that another one which is what you call unicycle model. So, I will actually like take it that unicycle.

(Refer Slide Time: 13:40)

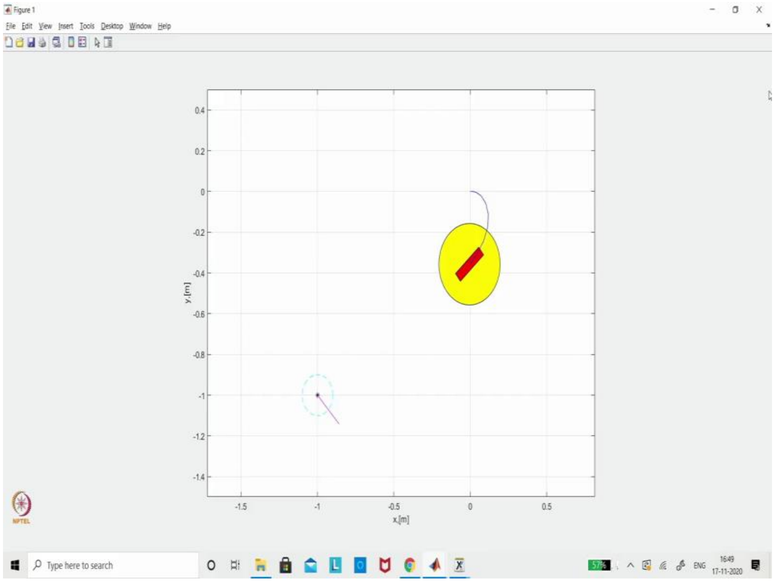


```
        if rho<0.01
            e = eta_d- eta(:,i);
        end

        psi = eta(3,i);
        J = [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1];
        zeta(:,i) = inv(J)*(diag([1,1,8])*e);
        u = sqrt(zeta(1,i)^2+zeta(2,i)^2); r = zeta(3,i);
        W = [u;0;r;];
        zeta(:,i) = W;
%         eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(
        eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*J*zeta(:,i)*dt;
    end
```

2 usages of "W" found

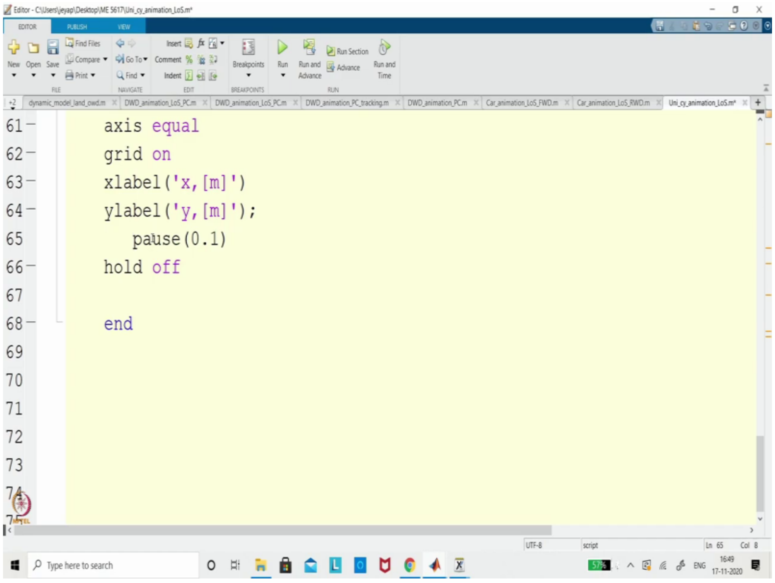
So, you can see in this case only the W would be having actually like 1 input. So, where u and r is directly controllable.

(Refer Slide Time: 13:49)



So, that is what we are actually trying to take. So, in the sense this is the unicycle I did not show the what you call the passive caster wheel. So, now you can see that this particular wheel configuration is trying to follow this. So, I have actually like change that configuration. So, here also we recorded.

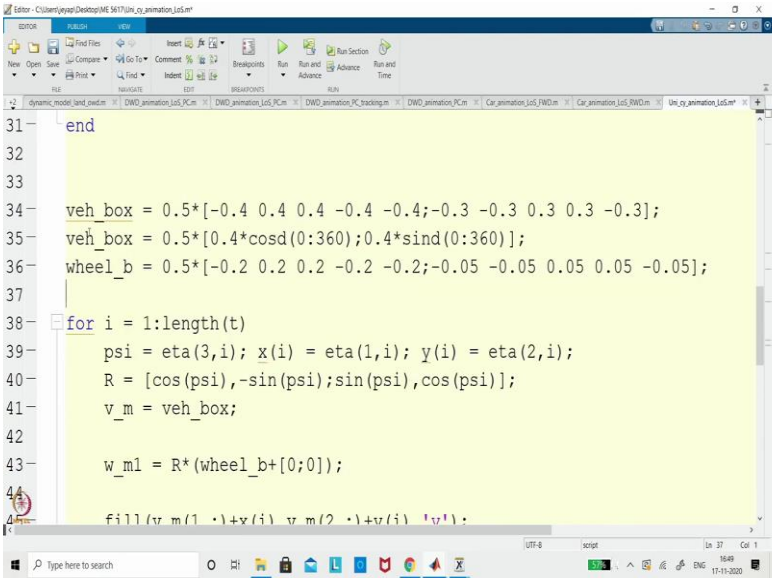
(Refer Slide Time: 14:03)



Editor - C:\Users\jeyap\Desktop\ME 5617\Uni_cy_animation_LoS.m*
EDITOR
PUBLISH
VIEW
Uni_cy_animation_LoS.m*

```
    axis equal
    grid on
    xlabel('x, [m]')
    ylabel('y, [m]');
       pause(0.1)
    hold off

    end
```

(Refer Slide Time: 14:13)

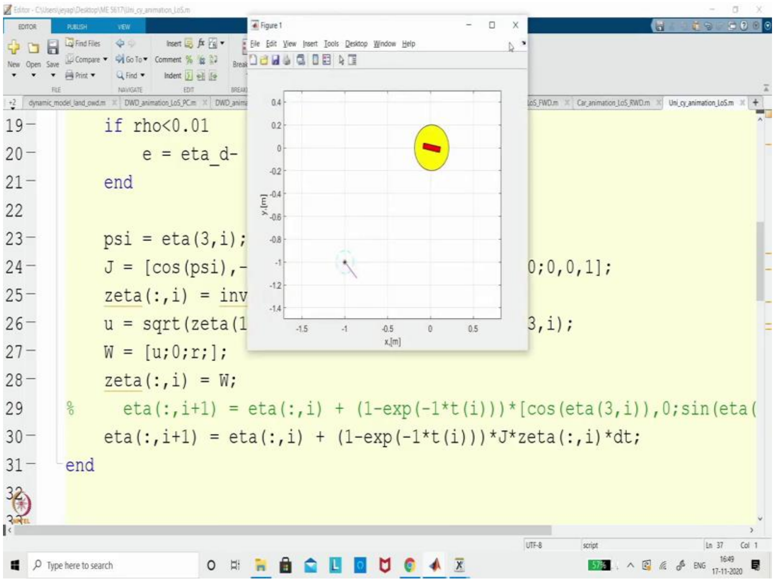


Editor - C:\Users\jeyap\Desktop\ME 5617\Uni_cy_animation_LoS.m*
EDITOR
PUBLISH
VIEW
Uni_cy_animation_LoS.m*

```
end


veh_box = 0.5*[-0.4 0.4 0.4 -0.4 -0.4;-0.3 -0.3 0.3 0.3 -0.3];
veh_box = 0.5*[0.4*cosd(0:360);0.4*sind(0:360)];
wheel_b = 0.5*[-0.2 0.2 0.2 -0.2 -0.2;-0.05 -0.05 0.05 0.05 -0.05];

for i = 1:length(t)
    psi = eta(3,i); x(i) = eta(1,i); y(i) = eta(2,i);
    R = [cos(psi),-sin(psi);sin(psi),cos(psi)];
    v_m = veh_box;

    w_m1 = R*(wheel_b+[0;0]);

    fill(v_m(1,:)+x(i),v_m(2,:)+y(i),'y');
```

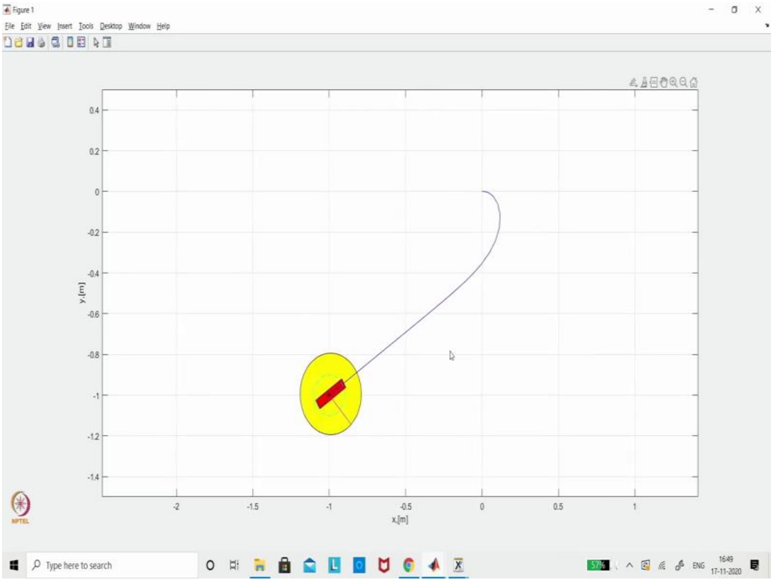
Because this particular case we are using it for some other benefit.

(Refer Slide Time: 14:15)



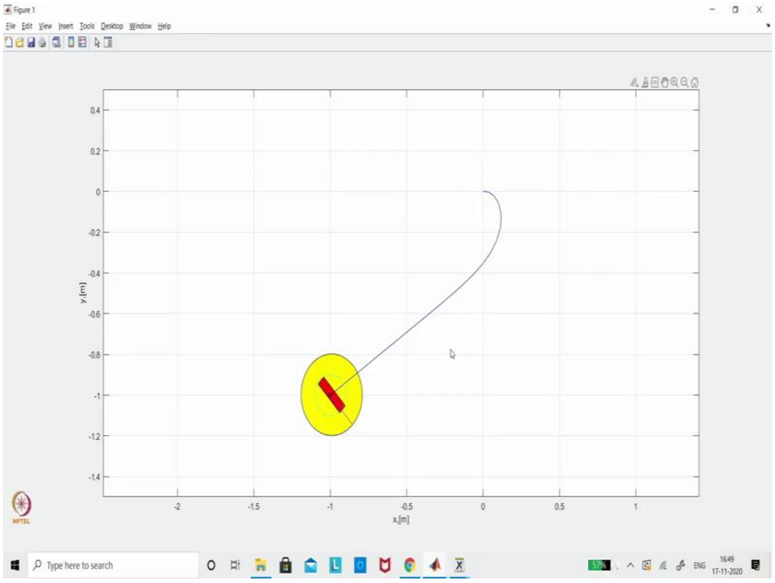
```
    if rho<0.01
        e = eta_d-
    end

    psi = eta(3,i);
    J = [cos(psi),-                                  0;0,0,1];
    zeta(:,i) = inv
    u = sqrt(zeta(1                                  3,i);
    W = [u;0;r;];
    zeta(:,i) = W;
%     eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(
    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*J*zeta(:,i)*dt;
end
```

(Refer Slide Time: 14:18)



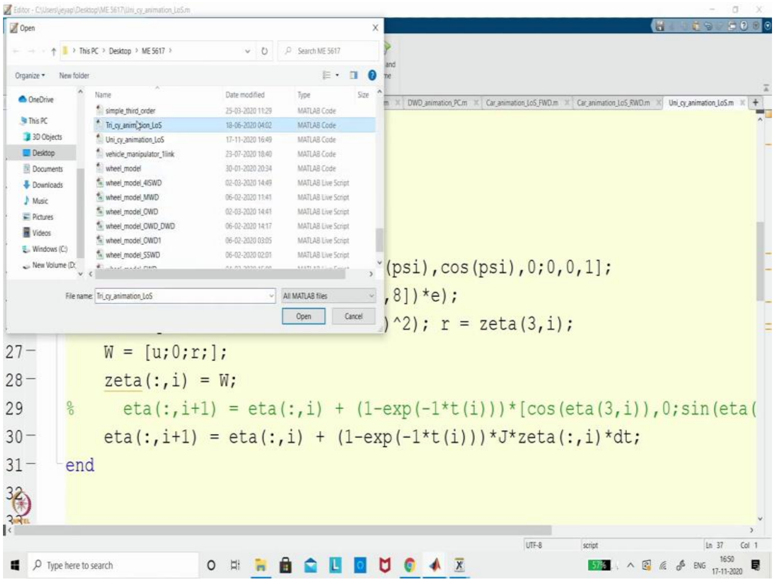
Because we can record this video. So, now you can see the same scenario only thing it is actually like there is no restriction on what you call sliding ok. So, it is actually like still nonholonomic, but the lateral resistant is not there. So, this is unicycle.

(Refer Slide Time: 14:31)



So, similarly only one left we not discussed that is a tricycle, I will take that tricycle also here and I will actually like run that.

(Refer Slide Time: 14:40)



Name	Date modified	Type	Size
simple_third_order	25-03-2020 11:29	MATLAB Code	
Tri_cy_animation_LoS	18-06-2020 04:02	MATLAB Code	
Uni_cy_animation_LoS	17-11-2020 16:49	MATLAB Code	
vehicle_manipulator_1link	23-07-2020 18:40	MATLAB Code	
wheel_model	30-01-2020 20:34	MATLAB Code	
wheel_model_4SWD	02-03-2020 14:49	MATLAB Live Script	
wheel_model_MWD	06-02-2020 11:41	MATLAB Live Script	
wheel_model_OWD	02-03-2020 14:41	MATLAB Live Script	
wheel_model_OWD_DWD	06-02-2020 14:17	MATLAB Live Script	
wheel_model_OWD1	06-02-2020 03:05	MATLAB Live Script	
wheel_model_SSWD	06-02-2020 02:01	MATLAB Live Script	

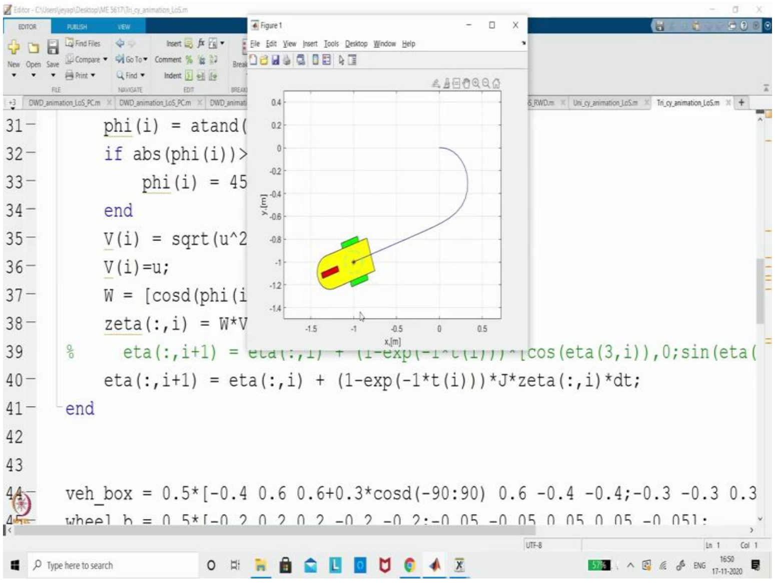
File name: Tri_cy_animation_LoS

All MATLAB files

```
(psi),cos(psi),0;0,0,1];
,8])*e);
)^2); r = zeta(3,i);
    W = [u;0;r;];
    zeta(:,i) = W;
%     eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(
    eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*J*zeta(:,i)*dt;
end
```

You can actually like get a feel how that will happen. So, you can see that tricycle model. The tricycle model in fact we derived in the what you call our own lectures. So, now you can actually like a put that into a case.

(Refer Slide Time: 14:50)



```
        phi(i) = atand(
        if abs(phi(i))>
            phi(i) = 45
        end
        V(i) = sqrt(u^2
        V(i)=u;
        W = [cosd(phi(i
        zeta(:,i) = W*V
%         eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*[cos(eta(3,i)),0;sin(eta(
        eta(:,i+1) = eta(:,i) + (1-exp(-1*t(i)))*J*zeta(:,i)*dt;
    end


    veh_box = 0.5*[-0.4 0.6 0.6+0.3*cosd(-90:90) 0.6 -0.4 -0.4;-0.3 -0.3 0.3
    wheel_b = 0.5*[-0.2 0.2 0.2 -0.2 -0.2;-0.05 -0.05 0.05 0.05 -0.05];
```

So, now you can see the vehicle trying to do it right. So, this is a forward wheel drive and, but it is a single wheel. So, this is what we have seen. So, now, what one can actually like see it? So, there is a W change the you can say the further performance will get changed right. So, that is what we are actually like trying to discuss.

So, what we can actually like see that all the combinations are actually like depend on only one thing which is what we have actually like derived as a W. So, now, you change the W. Your wheel you can say scenario or you can say behavior will get changed. So, that is why you can actually like say that they are nonholonomic and holonomic condition you should know if the W matrix rank is 3 then you are actually like safe all 3 states are controllable.

So, that is what we wanted. So, now you recall that particular lecture and see which are the vehicles are actually like you can say you can say holonomic. So, those are you can directly substitute what you have derived as a kinematic control. Only thing if you have more control input and your W^+ what you have taken right that would optimize the output or you can say the control inputs it will optimize.

But whereas, you have actually like a rank of the W is less than 3 then you have to actually like see which are the states you want to control, then you have to play a small tweak on the you can say the position input which you call desired trajectory that you have to see. So, now what we have done we have done the kinematic control that to like a simulation also we have done along with what you call the wheel configuration.

In the next lecture we will see the other class of you can say the motion control what you call dynamic control, then we will actually like move ahead further we will bring to the close loop control along with the dual loop scenario. So, with that we will see in the next lecture with the dynamic control, until then see you bye, take care.