

```
function [ state ] = FollowLine( state, lineToFollow, time, P )
%FollowLine inputs:
% state of the vehicle at the beginning
% line you wish the vehicle to follow
% the time to follow the line
% the parameters set P
%outputs are: the state after the time has elapsed.

    v = P.v_const; %constant velocity of 50 cm/sec

    %unpack pose from state
    pose = state(1:3);

    %follow the line lineToFollow for 10 seconds of simulation
    for t = 0:P.delta_t:time

        %TODO define the function to set the angular velocity (w) based on
        %distance from the line
        SE2Line = [cos(lineToFollow(3)), -sin(lineToFollow(3)), lineToFollow(1); ...
                    sin(lineToFollow(3)), cos(lineToFollow(3)), lineToFollow(2); ...
                    0 0 1];
        SE2Pose = [cos(pose(3)), -sin(pose(3)), pose(1); ...
                    sin(pose(3)), cos(pose(3)), pose(2); ...
                    0 0 1];
        e = inv(SE2Line)*SE2Pose;
        ky = .5; % Some scaling factor
        theta_c = lineToFollow(3) - atan(ky*e(2, 3));
        w = wrapAnglePi(theta_c - pose(3)); % Replace this line

        pose = propagatePose(pose, v, w, P.delta_t);
        state = [pose; v; w; state(6)+P.delta_t];
        drawCar(state, P);

    end
end
```