ME 435 Engineering Systems Lab

Level

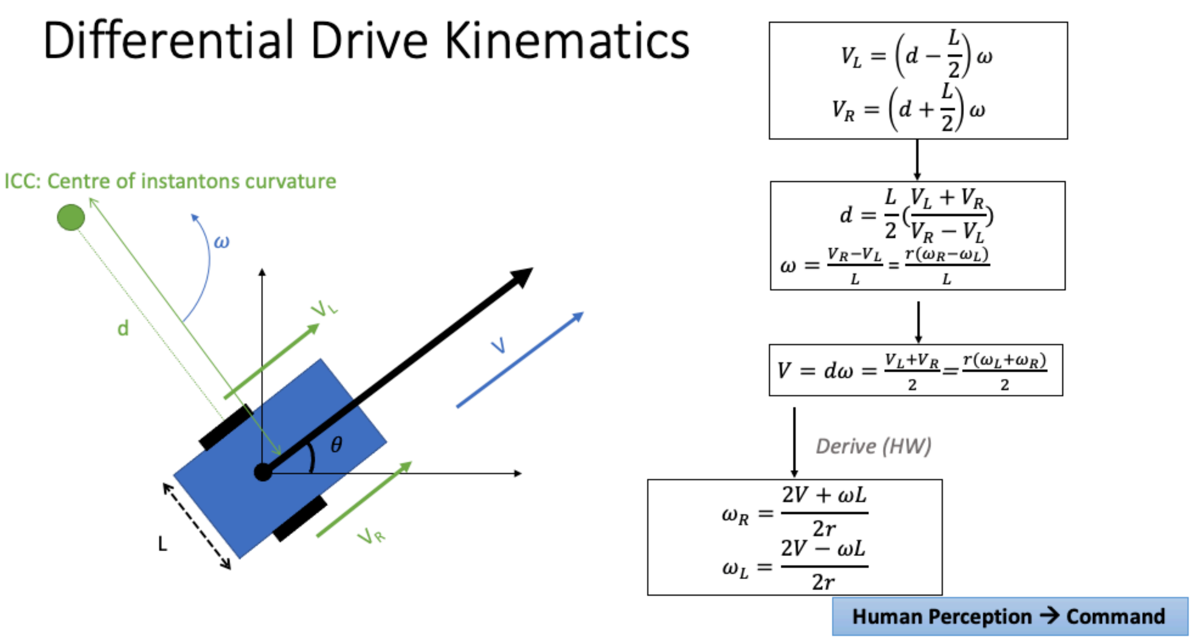
4

Autonomous Vehicles Lab Assignment

Name: ___Fan Han Hoon___

Part A – Derive Kinematic Equations

In the lecture, we derived equations for V_L, V_R, ω, d and V, as shown below. Use the question given in the first three boxes **to represent ω_L and ω_R in terms of ω, L, r, and V**. r is the wheel's radius. **Show your work.**



$$V_L = (d - \tfrac{L}{2})\,w \Rightarrow w = \frac{V_L}{d-\frac{L}{2}} \quad -(1)$$

$$V_R = (d + \tfrac{L}{2})\,w \Rightarrow w = \frac{V_R}{d+\frac{L}{2}} \quad -(2)$$

$(1) = (2)$

$$\frac{V_L}{d-\frac{L}{2}} = \frac{V_R}{d+\frac{L}{2}}$$

$$V_L\left(d+\frac{L}{2}\right) = V_R\left(d-\frac{L}{2}\right)$$

$$d(V_L - V_R) = -\left(\frac{V_L}{2} + \frac{V_R}{2}\right)L$$

$$d(V_R - V_L) = \left(\frac{V_L + V_R}{2}\right)L$$

$$d = \frac{L}{2}\frac{(V_L + V_R)}{(V_R - V_L)}$$

$$W = \frac{V_R - V_L}{L} = \frac{r(w_R - w_L)}{L} \quad -(1)$$

$$V = dw = \frac{V_L + V_R}{2} = \frac{r(w_L + w_R)}{2} \quad -(2)$$

From (1)

$$\frac{r(\omega_R - \omega_L)}{L} = \omega$$

From (2)

$$r\omega_L = 2V - r\omega_R$$

$$r\omega_R - r\omega_L = \omega L$$

$$r\omega_R = \omega L + r\omega_L$$

$$r\omega_R = \omega L + 2V - r\omega_R$$

$$2r\omega_R = \omega L + 2V$$

$$\boxed{\omega_R = \frac{2V + \omega L}{2r}}$$

From (2)

$$\frac{r(w_L + w_R)}{2} = V$$

From (1)

$r w_R = wL + r w_L$

$$r w_L = 2V - r w_R$$

$$r w_L = 2V - wL - r w_L$$

$$2 r w_L = 2V - wL$$

$$\boxed{w_L = \frac{2V - wL}{2r}}$$

Part B – Understand Lane Detection using Image Processing Technique.

Download the MATLAB LiveScript file, UAV_Individual_Assignment.mlx, and the road.png files from Canvas. Complete the LiveScript and answer questions 1-5.

You will also need the following material to complete this part of the assignment:

1. **Image Processing Toolbox** by MathWorks is installed.
2. **Take 2-3 pictures of a road in different environmental conditions**. For example, in the morning, in the evening, at night, with varying signs of road or lane designs compared to the provided image. You can choose any condition. The only requirement is that the road is clear in the picture and similar to a vehicle's point of view.

SAFETY SAFETY SAFETY

- **NEVER record the video while you are driving alone! Only do it when you are in the passenger seat!**
- **NEVER record the picture when a car is driving towards you! Take pictures on the sidewalk. That should be sufficient.**

Copy and paste your answer to the fives questions to the following space:

Your Answer for Question 1:

```
% Code is provided.


% Create a polygon mask
[height, width]= size(grayimage) ;
xi=[width*0.4, width*1.65, width*0.35, width*0.1];
yi=[height, height*2/3, height*2/3, height];


BW2 = poly2mask(xi,yi,height,width);


% Combine boundary detected image with polygon mask
BW1 = BW2 & BW1;
imshow(BW1)
```

<u>Figure 1 shows the trapozaoid mask result.</u>

<u>Your Answer for Question 2:</u>

The fill gap parameter restricts the maximum distance between the two points in the line while the min length parameter specifies the minimum length of the line to be detected.

If I increase the fill gap value, it will result in a fewer but longer lines. On the other hand, a decrease in fill gap value will in more but shorter lines. Similiarly, an increase in Min length value will result in shorter lines getting deleted and longer lines getting detected. Decrease in Min length value will result in only detecting shorter lines.

<u>Your Answer for Question 3:</u>

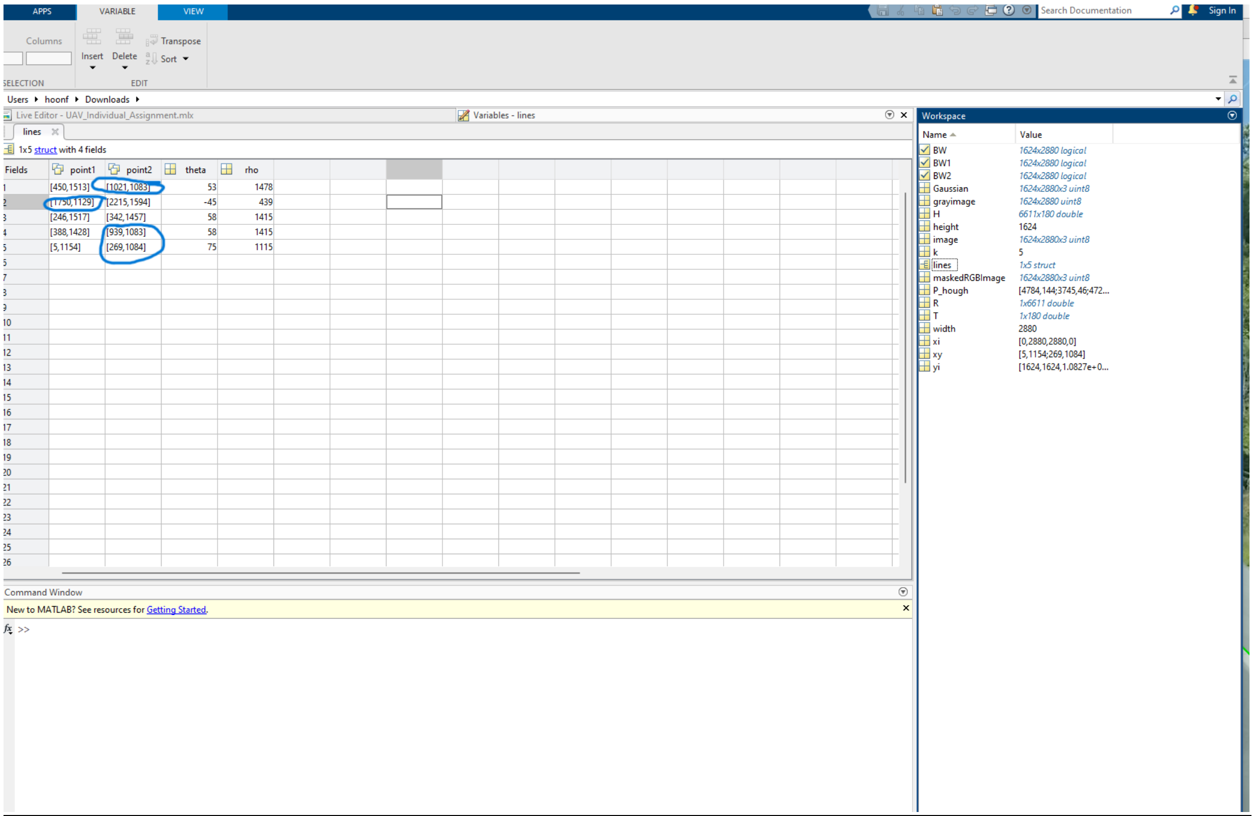


Figure 2 shows the values stored in line workspace with end-points circled.

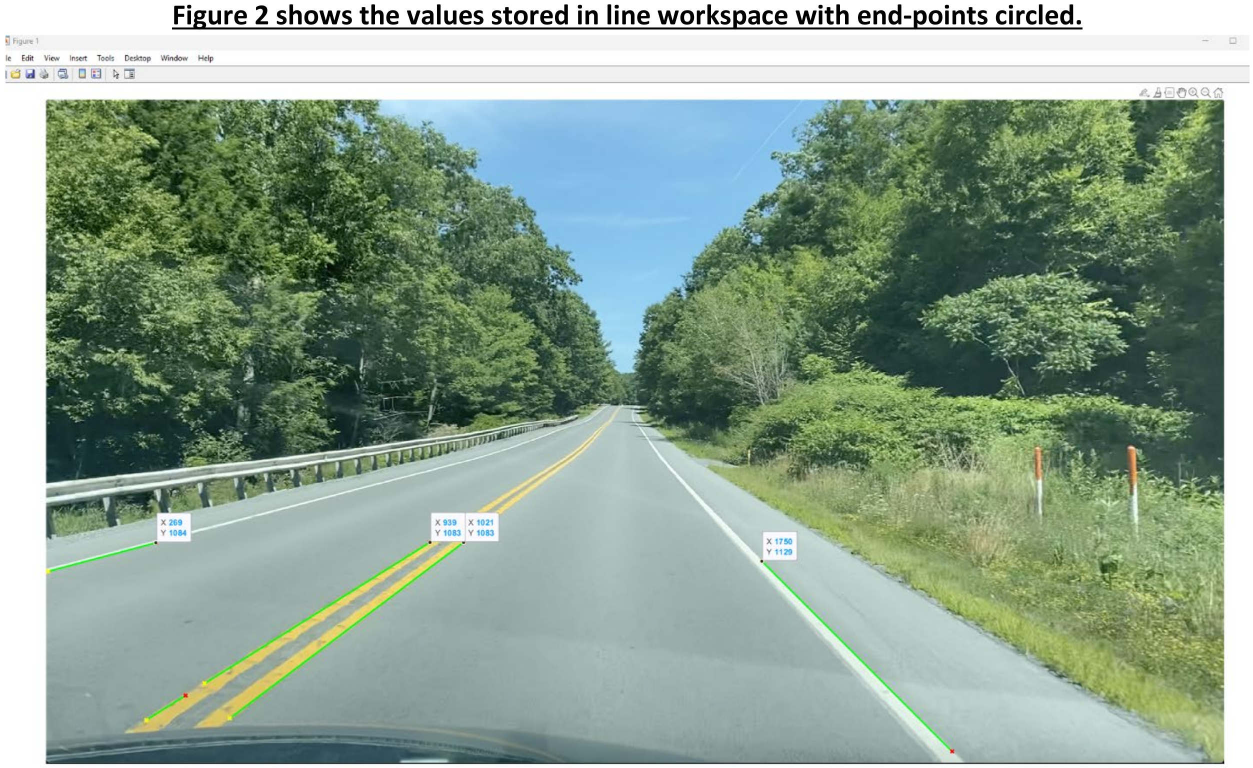


Figure 3 shows the xy coordinates of the end-point in the image.

Your Answer for Question 4:

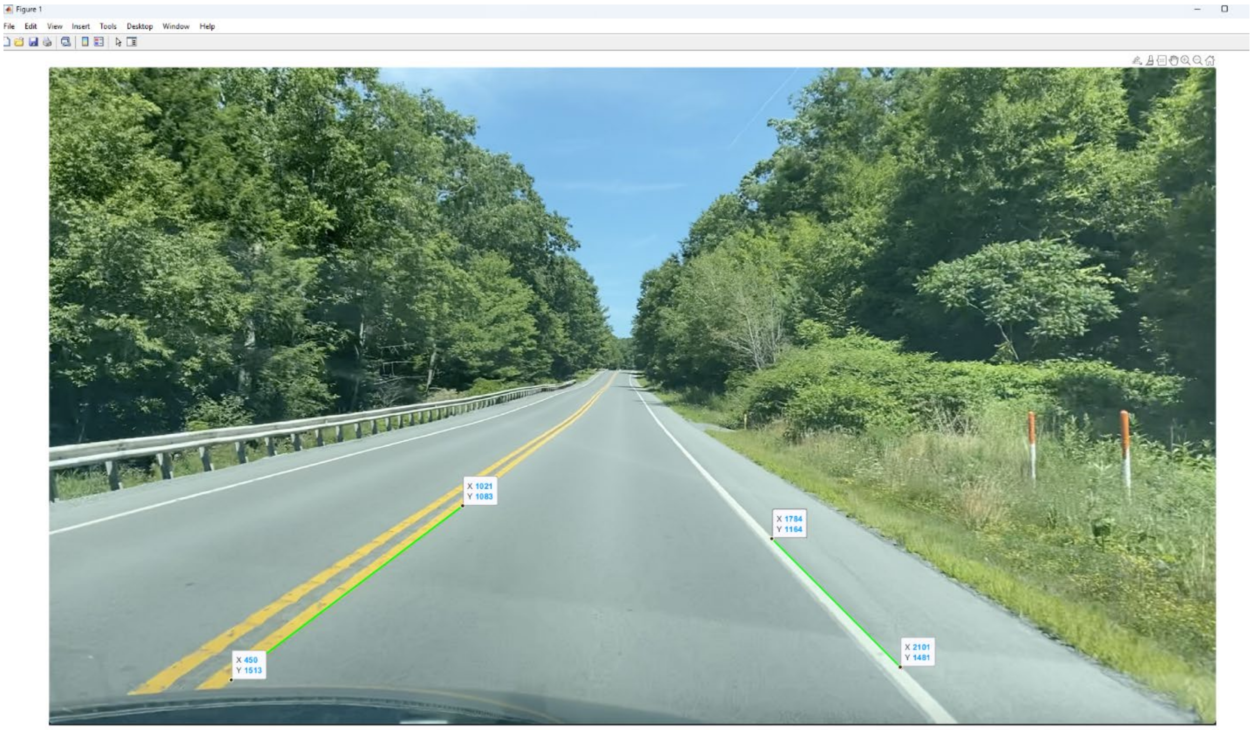


Figure 4 shows the xy coordinates of the end-point in the image.

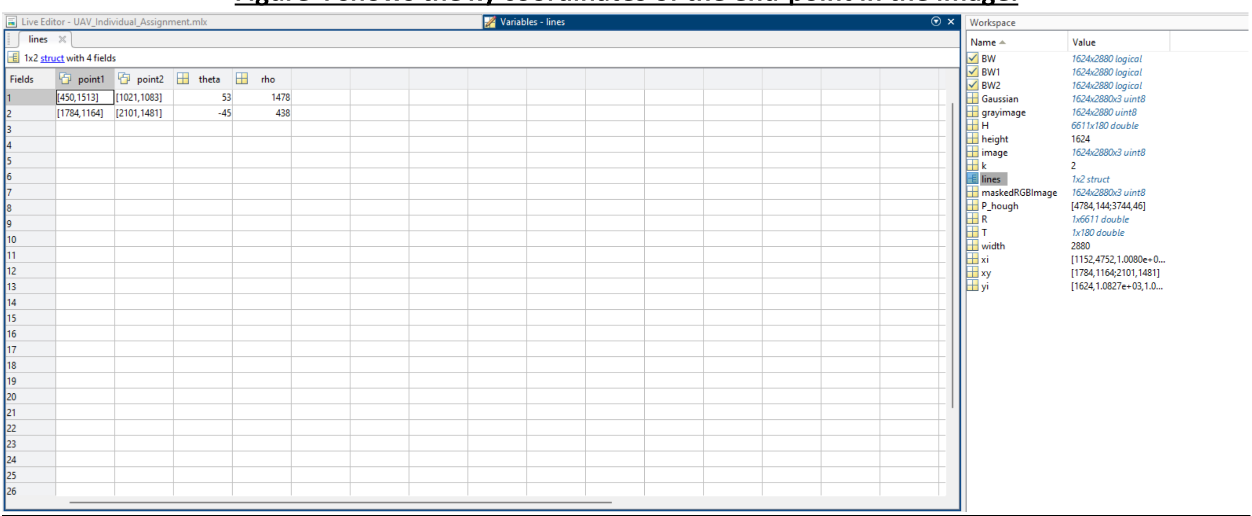
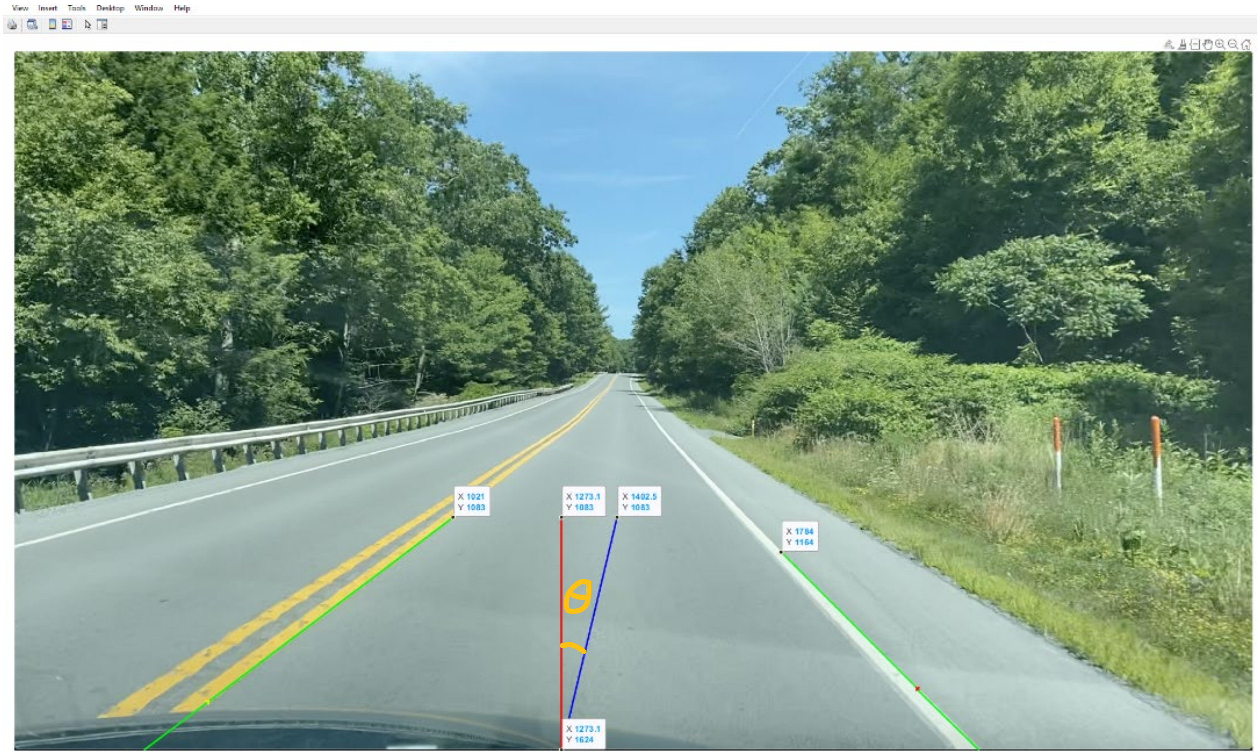


Figure 5 shows the values stored in line workspace.



<u>Figure 6 shows the image used to calculate the heading angle</u>

```
% Copy and paste the example code here to visualize the outcome.


   firstrow= [lines(1).point1; lines(1).point2];
   secondrow=[lines(2).point1; lines(2).point2];
plot([302.19,450],[1624,1513],'LineWidth',2,'Color','green');
plot([2244,2101],[1624,1484],'LineWidth',2,'Color','green');


% Define the coordinates of the initial and heading points
Initial_coordinate = [(302.19 + 2244)/2, 1624];
Heading_coordinate = [(firstrow(2,1)+secondrow(1,1))/2, firstrow(2,2)];


plot([Initial_coordinate(1), Heading_coordinate(1)], [Initial_coordinate(2),
Heading_coordinate(2)], 'LineWidth', 2, 'Color', 'Blue');


% Original intented final heading
Final_coordinate=[Initial_coordinate(1), Heading_coordinate(2)];
plot([Initial_coordinate(1), Initial_coordinate(1)], [Initial_coordinate(2),
Final_coordinate(2)], 'LineWidth', 2, 'Color', 'red');
\
Opp=sqrt((Heading_coordinate(1)-Final_coordinate(1))^2);
A=sqrt((1624-Heading_coordinate(2))^2);
```

```
Heading_angle=atan(Opp/A)
Heading_angle = 0.2348
```

$(x, 0) \quad (450, 1513) \quad (1021, 1083)$

$y = mx + c$

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

$$= \frac{(1083 - 1513)}{(1021 - 450)}$$

$$= -0.7539$$

$$c = y - mx$$

$$= 1513 + (0.7539)(450)$$

$$= 1822.53$$

$$y = -0.7539x + 1822.55$$

$$1624 = -0.7539x + 1822.55$$

$$x = 302.19$$

$(x, 0) \quad (2101, 1481) \quad (1784, 1164)$

$$m = \frac{(1481 - 1164)}{(2101 - 1784)}$$

$$= 1$$

$$y - 1164 = 1(x - 1784)$$

$$y = x - 620$$

$$1624 = x - 620$$

$$x = 2244$$

Your Answer for Question 5:

For the images that I took with my Samsung S22, the images performed poorly under the same gaussian filtering and thus the following greyscale and line segmeners extraction performed poorly. Since I don't have a car at state college, I took some random images of the roads that were available around West Campus. The original filter was focused on filtering the background noise in the image where only yellow and white lines will be the target of focus for line segmeners extraction. However, in some of the image that I took, there were no obvious markings on the road where the boundary of the road was indicated with concrete walk path instead of the white lines seen in the example image.

In order to improve the lane detection process, I would implement another line of code which allow users to select different gaussian filter as I optimize the filters based on the image captured. If I were to design the program for a long run, I would design a code where it converts the image to grayscale initially and use deep learning techniques to let the program to select the desired filters. However, the deep learning portion would still require tradional manual training dataset that was mentioned previously.