

FINAL PROJECT REPORT
MAE 5810

Pinhole Camera Tracking and Path Planning Based on Two-wheeled Vehicle Differential Steering Model

Yang Jiao
NetID: yj286
10 Dec 2021

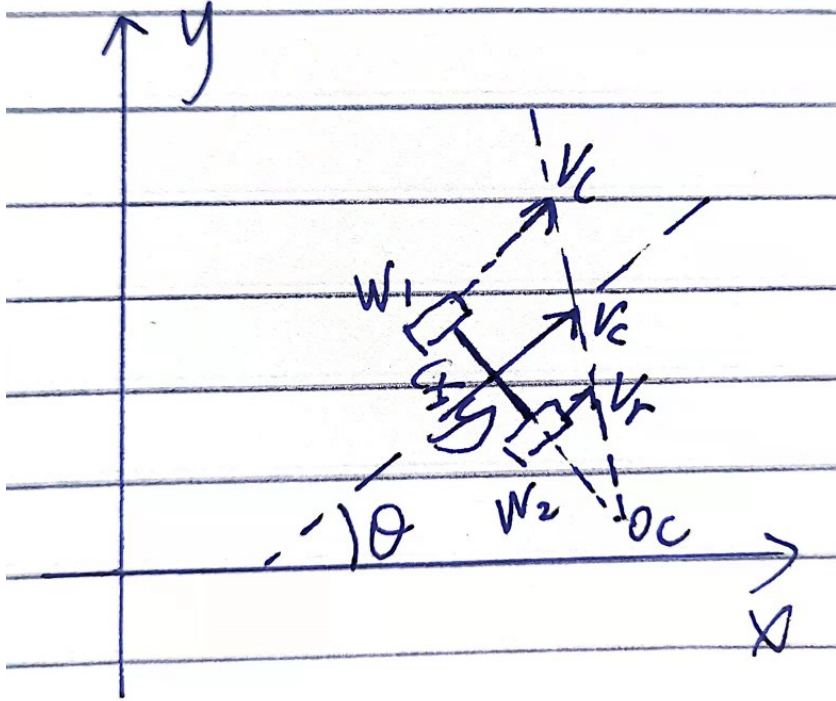
- The results are only briefly shown here in the demo powerpoint
- For detailed derivation process and descriptions can be found in the pdf version of the report as:

[Final_Project_Report-yj286.pdf](#)

ABSTRACT

- This article introduced the establishment of differential steering model for two-wheeled vehicles, and realized the trajectory tracking and path planning for two-wheeled vehicles by combining the pinhole camera model constructed in the mid-term project. At the same time, the A*-based heuristic search algorithm was explained considering the possible environmental obstacles in the actual application scenarios, and the simulation results were illustrated.

1. Differential steering model for two-wheeled vehicles



$$v_l = \frac{\phi_l \cos(\theta)}{2}$$

$$v_r = \frac{\phi_r \cos(\theta)}{2}$$

$$l = \frac{v_r}{w_r} - \frac{v_l}{w_l}$$

$$w_c = \frac{v_r - v_l}{l}$$

$$R = \frac{v_c}{w_c} = \frac{l(v_r + v_l)}{2(v_r - v_l)}$$

$$x' = v_c \cos \theta = \frac{v_r + v_l}{2} \cos \theta$$

$$y' = v_c \sin \theta = \frac{v_r + v_l}{2} \sin \theta$$

$$\theta' = w_c = \frac{v_r - v_l}{l}$$

$$x(t+1) = x(t) + v_c \cos \theta \times T$$

$$y(t+1) = y(t) + v_c \sin \theta \times T$$

$$\theta(t+1) = \theta(t) + w_c \times T$$

$$v_r = v_c + \frac{w_c l}{2}$$

$$v_l = v_c - \frac{w_c l}{2}$$

2. Pinhole Camera Model

- $(\mathbf{q}_1)_V = \begin{bmatrix} \frac{a}{2} \\ \frac{b}{2} \\ \frac{b}{2} \end{bmatrix}, (\mathbf{q}_2)_V = \begin{bmatrix} \frac{a}{2} \\ \frac{b}{2} \\ -\frac{b}{2} \end{bmatrix}, (\mathbf{q}_3)_V = \begin{bmatrix} -\frac{a}{2} \\ \frac{b}{2} \\ -\frac{b}{2} \end{bmatrix}, (\mathbf{q}_4)_V = \begin{bmatrix} -\frac{a}{2} \\ \frac{b}{2} \\ \frac{b}{2} \end{bmatrix}, (1)$

- $(\mathbf{q}_\ell)_B = [(\mathbf{q}_\ell)_V^T \quad \lambda]^T, \ell = 1, \dots, 4 (2)$

- $\mathbf{H}_\psi \triangleq \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{H}_\phi \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} (3)$

$$(\mathbf{q}_\ell)_I = (\mathbf{H}_\phi^T \mathbf{H}_\psi^T)^{-1} (\mathbf{q}_\ell)_B$$

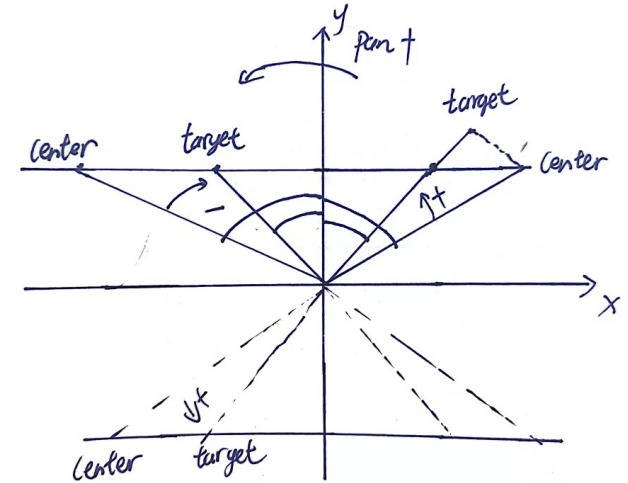
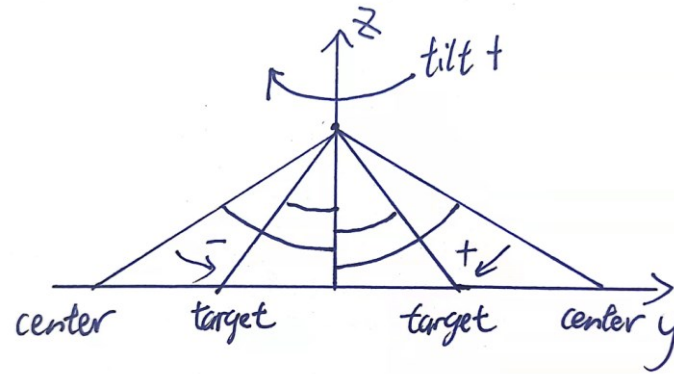
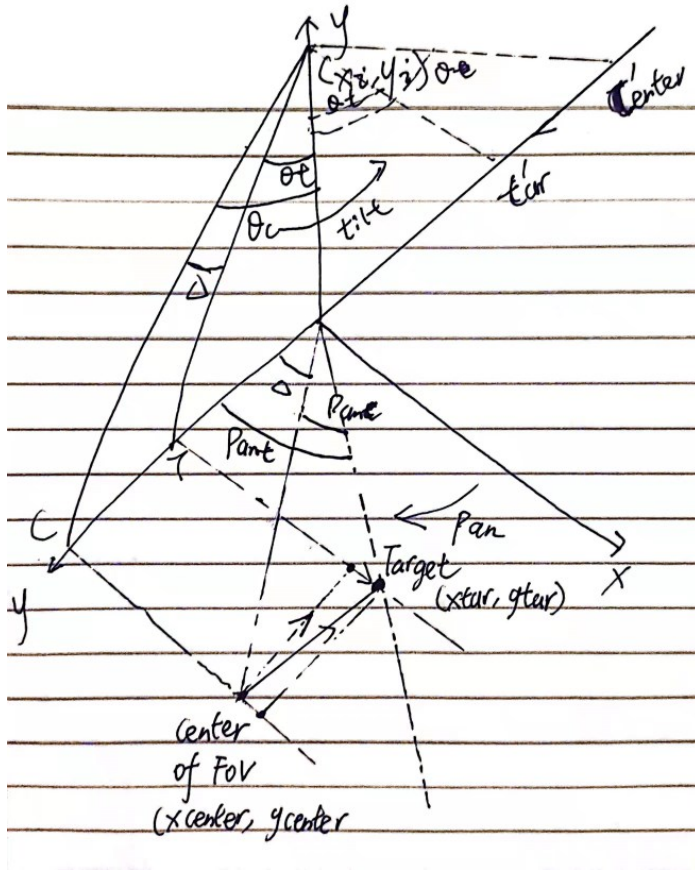
- $$\begin{aligned} &= [(\mathbf{H}_\psi \mathbf{H}_\phi)^{-1}]^T (\mathbf{q}_\ell)_B \quad (4) \\ &= \mathbf{H}_\psi \mathbf{H}_\phi (\mathbf{q}_\ell)_B, \ell = 1, \dots, 4 \end{aligned}$$

- $(\mathbf{q}'_\ell)_I = \rho_\ell (\mathbf{q}_\ell)_I, \ell = 1, \dots, 4 (5)$

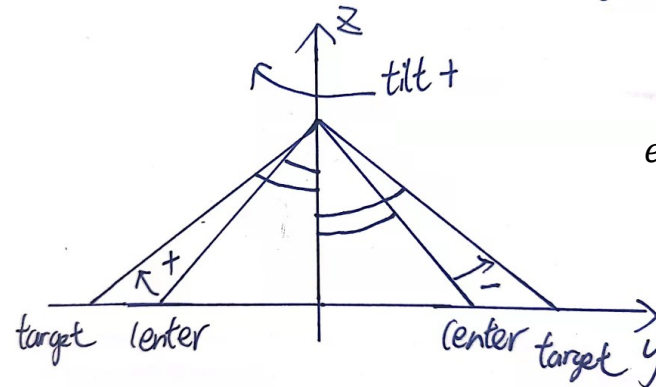
- $\rho_{1,4} = \frac{-\mathbf{x}_{(3)}}{\left(\frac{b}{2} \sin \phi + \lambda \cos \phi\right)} \rho_{2,3} = \frac{-\mathbf{x}_{(3)}}{\left(-\frac{b}{2} \sin \phi + \lambda \cos \phi\right)} (6)$

$$\begin{aligned} \xi_1 &= \mathbf{X} + \mathbf{q}'_1 \\ \xi_2 &= \mathbf{X} + \mathbf{q}'_2 \\ \xi_3 &= \mathbf{X} + \mathbf{q}'_3 \\ \xi_4 &= \mathbf{X} + \mathbf{q}'_4 \end{aligned}$$

3. Distance /Heading angle Control and Tracking



$$\text{errorpan} = \text{atan}\left(\text{abs}\left(\frac{x_{center}}{y_{center}}\right)\right) - \text{atan}\left(\text{abs}\left(\frac{x_{tar}}{y_{tar}}\right)\right);$$



$$\text{errortilt} = \text{atan}\left(\frac{\text{lenc tilt}}{z_i}\right) - \text{atan}\left(\frac{\text{lent tilt}}{z_i}\right);$$

- $\theta_{goal} = \text{atan}\left(\frac{y_{goal}-y_{trac}}{x_{goal}-x_{trac}}\right);$
- $\theta_{error} = \theta - \theta_{goal};$
- $error_{sum} = error_{sum} + \theta_{error}$

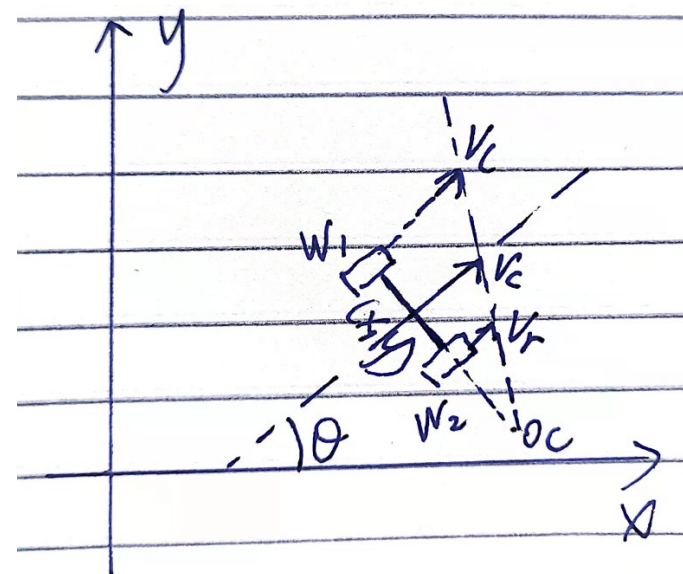
Control:

- $w = -k_p * (\theta_{error}) + k_i * error_{sum} + k_d * (\theta_{error} - pre_{error});$
- $dist = \left((x_{trac} - x_{goal})^2 + (y_{trac} - y_{goal})^2 \right)^{\frac{1}{2}};$
- $v = k2 * dist;$

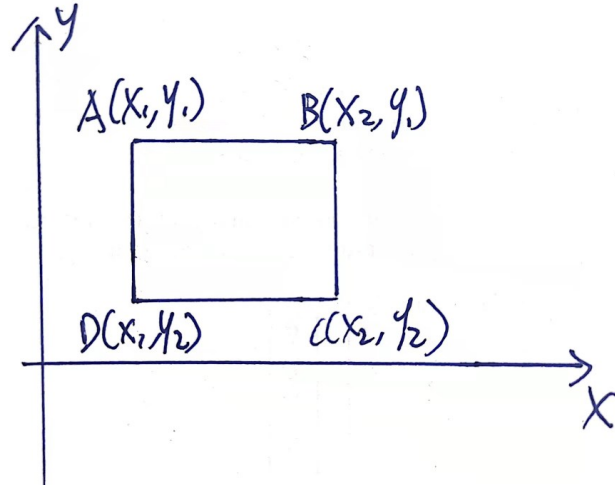
$$v_r = v + w * \frac{L}{2} \quad v_l = v - w * \frac{L}{2}$$

Dynamic model:

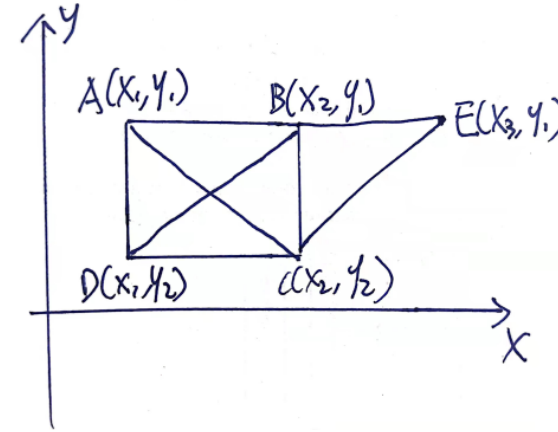
- $v = (v_l + v_r)/2$
- $w = (v_r - v_l)/L$
- $x_{trac} = x_{trac} + v * \cos(\theta) * T$
- $y_{trac} = y_{trac} + v * \sin(\theta) * T$
- $\theta = \theta + w * T$



4. *A*-Based Heuristic Search Algorithm for path planning*



$$F(n) = G(n) + H(n)$$



$$H(n) = \overline{AB} + \overline{BC} = |x_1 - x_2| + |y_1 - y_2|$$

$$G(n) = \overline{AB} = |x_1 - x_2|;$$

$$H(n) = \overline{BE} = \sqrt{(x_2 - x_3)^2 + (y_1 - y_2)^2}$$

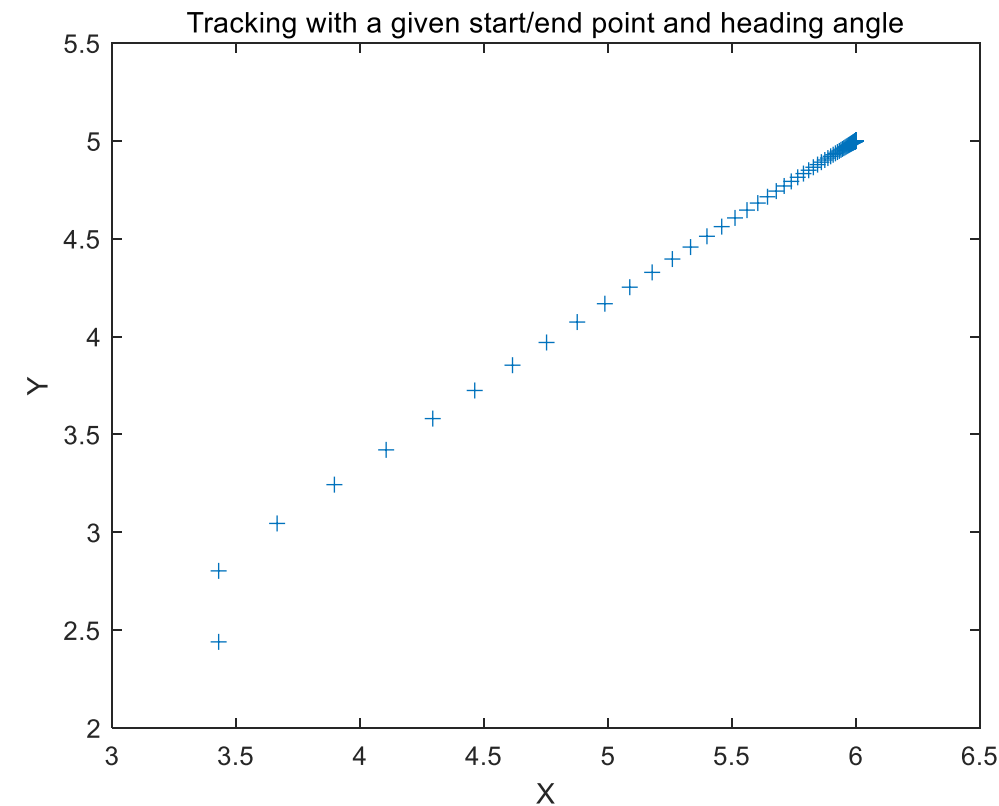
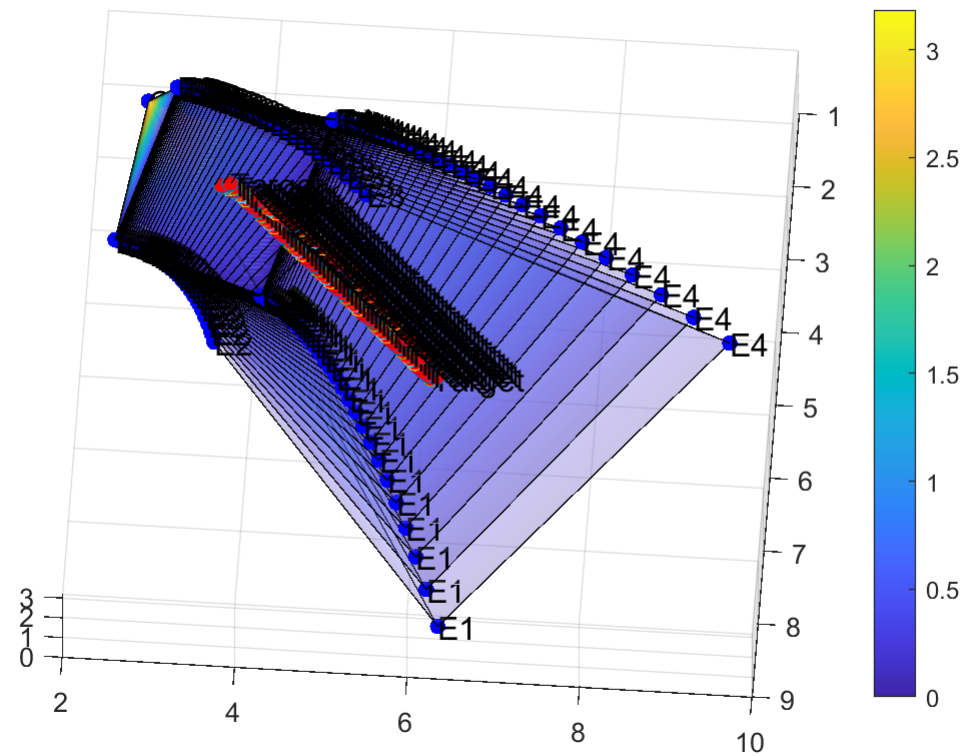
$$F(n) = G(n) + H(n) = \overline{AB} + \overline{BE}$$

$$= |x_1 - x_2| + \sqrt{(x_2 - x_3)^2 + (y_1 - y_2)^2}$$

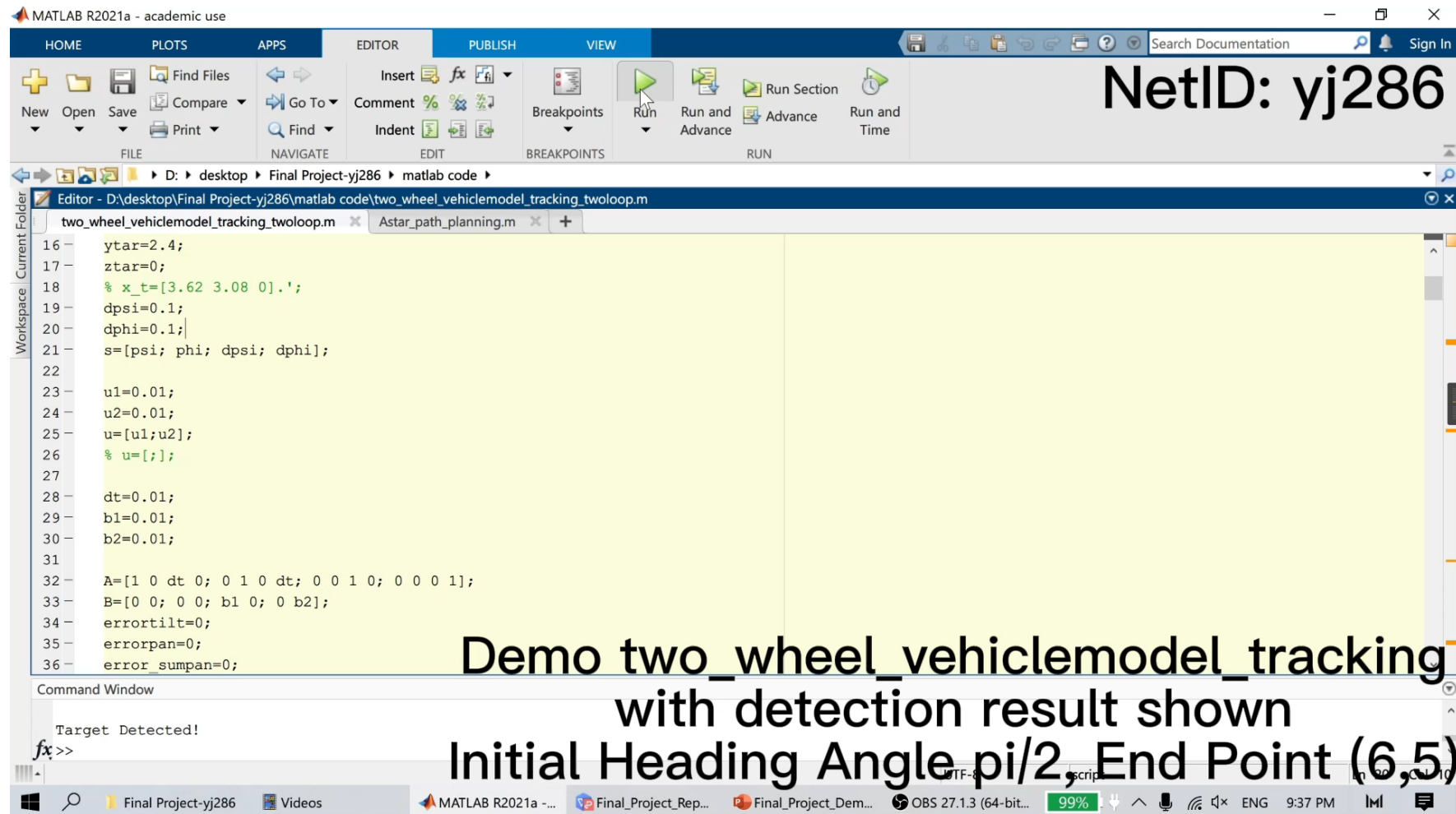
The search process of the A-star algorithm is implemented in the code as follows.

- 1' Calculate the costs of all the blocks that are immediately next to the present one.
- 2' Add them to valid list
- 3' Find block with minimum total cost from valid list, make it parent for next iteration and make it unavailable for next minimum total cost comparison
- 4' Put that in in_valid list so it won't be visited again
- 5' Keep doing this until the block with minimum cost is not target block.

Simulation Results of two_wheel_vehiclemodel_tracking



Video also included in the folder if is not playable here



The image shows the MATLAB R2021a interface with the following components:

- Top Bar:** MATLAB R2021a - academic use, tabs for HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. A search bar and "Sign In" button are on the right.
- Toolbars:** FILE (New, Open, Save, Compare, Print), NAVIGATE (Go To, Find), EDIT (Insert, Comment, Indent), BREAKPOINTS, and RUN (Run, Run and Advance, Run Section, Advance, Run and Time).
- Workspace:** Shows the current folder path: D:\desktop\Final Project-yj286\matlab code.
- Editor:** Displays the script `two_wheel_vehiclemodel_tracking_twoloop.m` with the following code:

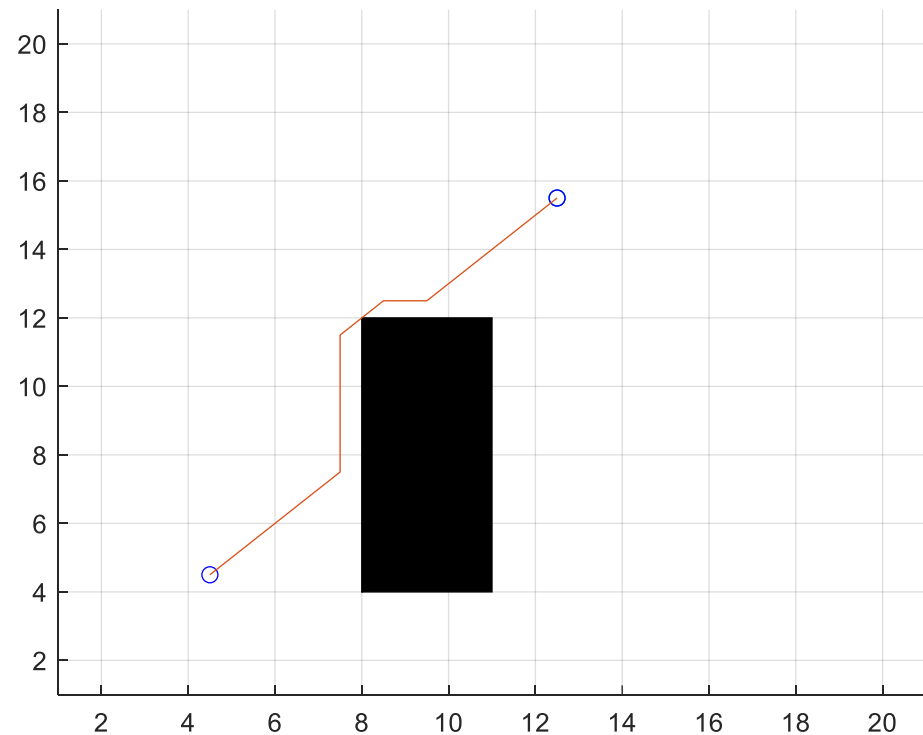
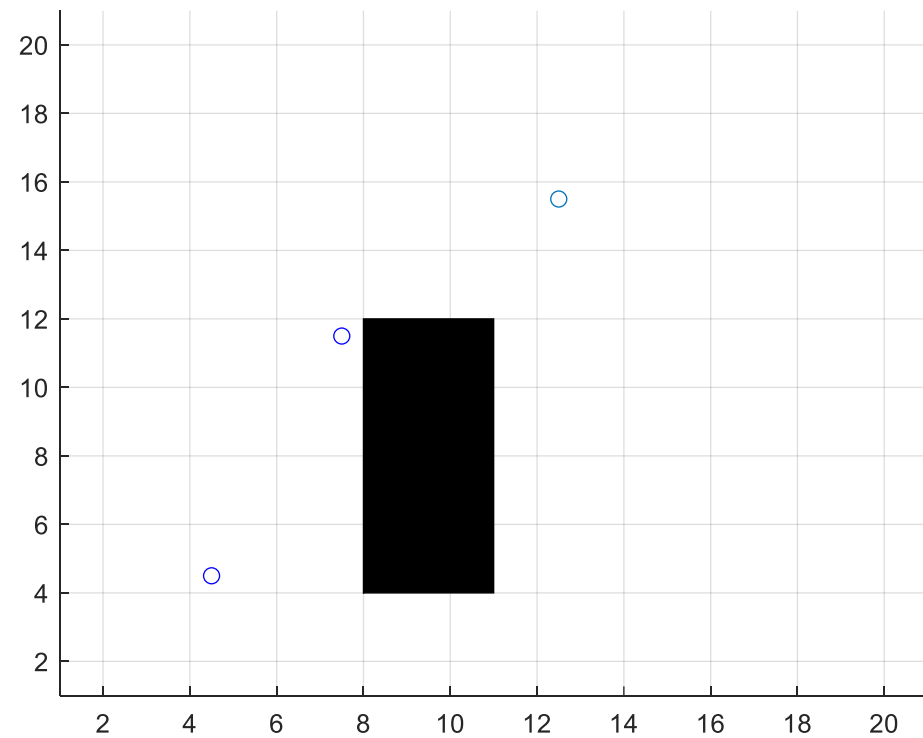
```
16- ytar=2.4;
17- ztar=0;
18- % x_t=[3.62 3.08 0].';
19- dpsl=0.1;
20- dphi=0.1;
21- s=[psi; phi; dpsl; dphi];
22-
23- u1=0.01;
24- u2=0.01;
25- u=[u1;u2];
26- % u=[;];
27-
28- dt=0.01;
29- b1=0.01;
30- b2=0.01;
31-
32- A=[1 0 dt 0; 0 1 0 dt; 0 0 1 0; 0 0 0 1];
33- B=[0 0; 0 0; b1 0; 0 b2];
34- errorilt=0;
35- errorpan=0;
36- error_sumpn=0;
```
- Command Window:** Displays the message "Target Detected!" and the prompt `fx>>`.
- Taskbar:** Shows the Windows taskbar with the following open applications: Final Project-yj286, Videos, MATLAB R2021a, Final_Project_Rep..., Final_Project_Dem..., OBS 27.1.3 (64-bit...), and a system tray with 99% battery, network, and volume icons. The system clock shows 9:37 PM on 10/10.

NetID: yj286

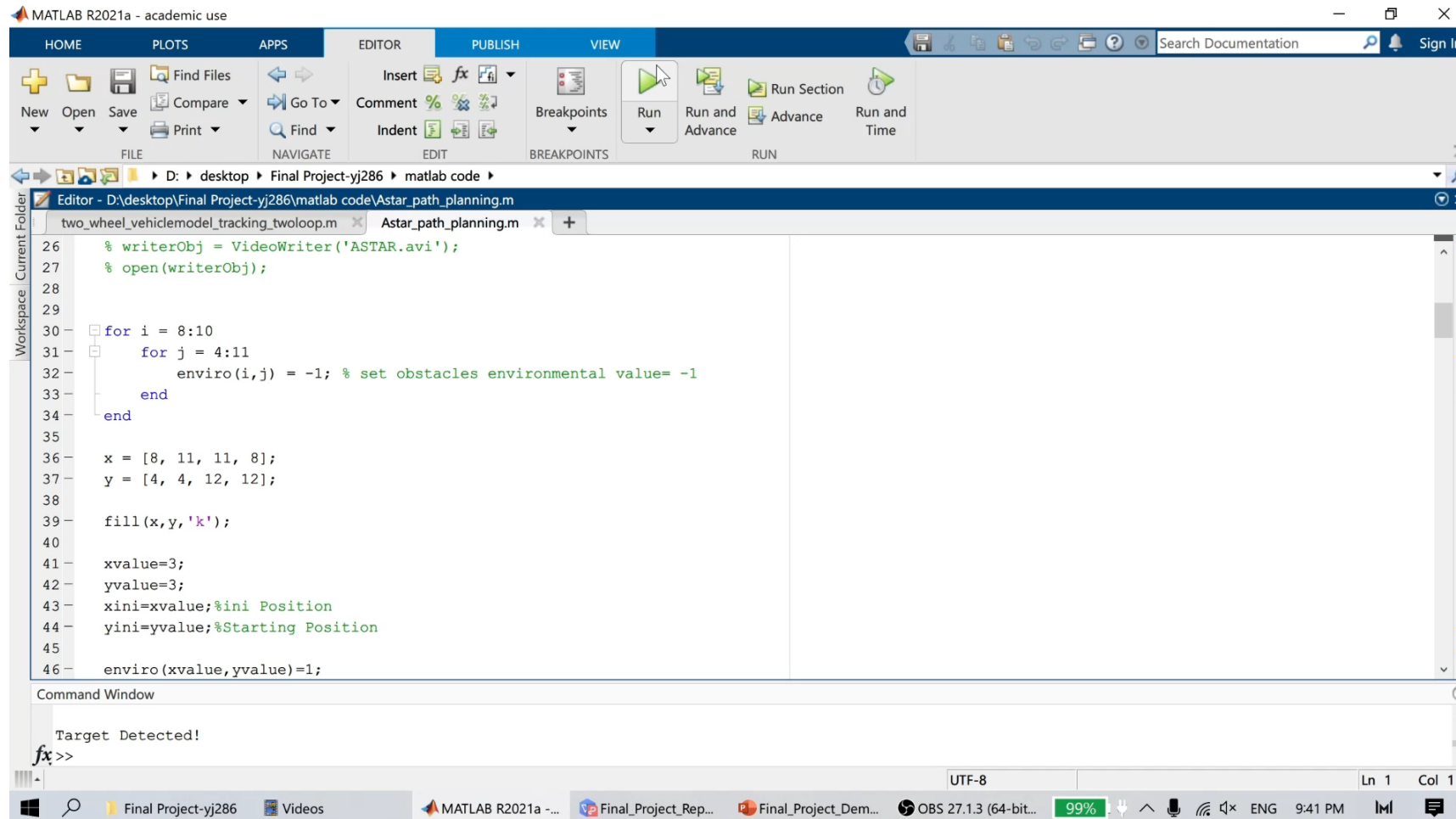
Demo two_wheel_vehiclemodel_tracking with detection result shown

Initial Heading Angle $\pi/2$, End Point (6,5)

Simulation Results of Astar_path_planning



Video also included in the folder if is not playable here



The image shows the MATLAB R2021a - academic use interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The toolbar contains icons for New, Open, Save, Find Files, Compare, Print, Go To, Find, Insert, Comment, Indent, Breakpoints, Run, Run and Advance, Run Section, Advance, and Run Time. The current file is 'Astar_path_planning.m' located at 'D:\desktop\Final Project-yj286\matlab code'. The script content is as follows:

```
26 % writerObj = VideoWriter('ASTAR.avi');
27 % open(writerObj);
28
29
30 for i = 8:10
31     for j = 4:11
32         enviro(i,j) = -1; % set obstacles environmental value= -1
33     end
34 end
35
36 x = [8, 11, 11, 8];
37 y = [4, 4, 12, 12];
38
39 fill(x,y,'k');
40
41 xvalue=3;
42 yvalue=3;
43 xini=xvalue;%ini Position
44 yini=yvalue;%Starting Position
45
46 enviro(xvalue,yvalue)=1;
```

The Command Window shows the message 'Target Detected!' and the prompt 'fx>'. The status bar at the bottom indicates 'UTF-8', 'Ln 1 Col 1', and the system clock shows '9:41 PM'.