UBC ENGR509

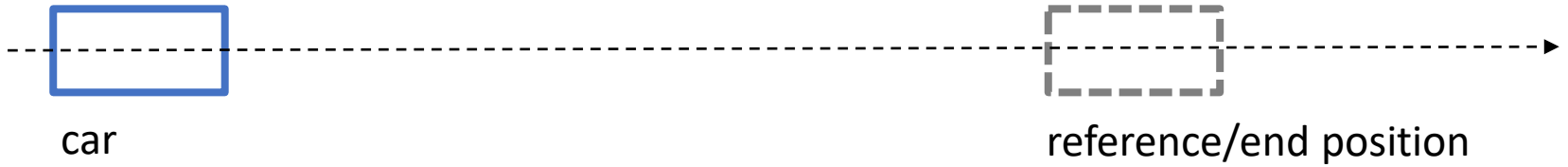
Intelligent Wireless Robotics

Lab on Model Predictive Control

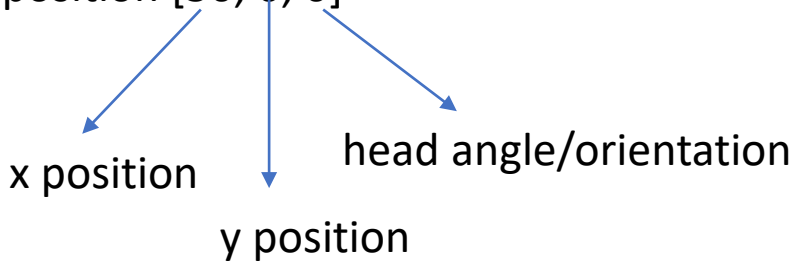
Pengxia Wu

Implement 1 – Parking Control 1D (Assignment)

- Task: Drive the car from start position to end position



Control the car from starting position [0, 0, 0]
to the end position [50, 0, 0]



Starting code: mpc_1d_st.py

Complete the methods: *motion_model* and *cost_func*

How to respect the speed limit 30km/h?

Implement 1 – Parking Control 1D (Assignment)

- **Model**: Will be used to predict the next state based on current state and control inputs.

$$x_{t+1} = x_t + \dot{x}\, dt \qquad \dot{x} = v_t$$

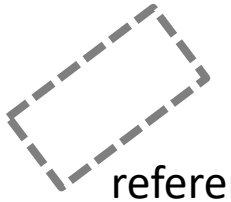
$$v_{t+1} = v_t + \dot{v}\, dt - v_t / 25 \qquad \dot{v} = pedal_t * 5$$

Assume a natural resistive force from the air friction

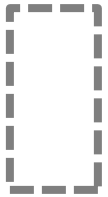
- **Cost function** :
  - Suppose a sequence of control inputs *u=[pedal_0, steering_1, pedal_2, steering_3, … pedal_(2N-2), steering_(2N-1)]*, along the prediction horizon *N*, for any time instance *t*, predict x_{t+1}, compute the cost and add to the total cost.
- **Optimization and Control (Done by the simulator):** Compute and return the optimal control inputs under constraints. Perform the first action from the computed optimal control inputs.
  - Actuator constraint: *pedal* is in the range [-1, 1] = [full brake, full throttle]; *steering* is bounded to [-0.0, 0.0] since 1D case does not need steering control.
  - Use the solver scipy.optimize.minimize.

Implement 2 – Parking Control 2D (Assignment)

- Task 2: Park the car into the reference positions



reference 1 = [10, 10, 0]



reference 2 = [10, 2, 3*3.14/2]

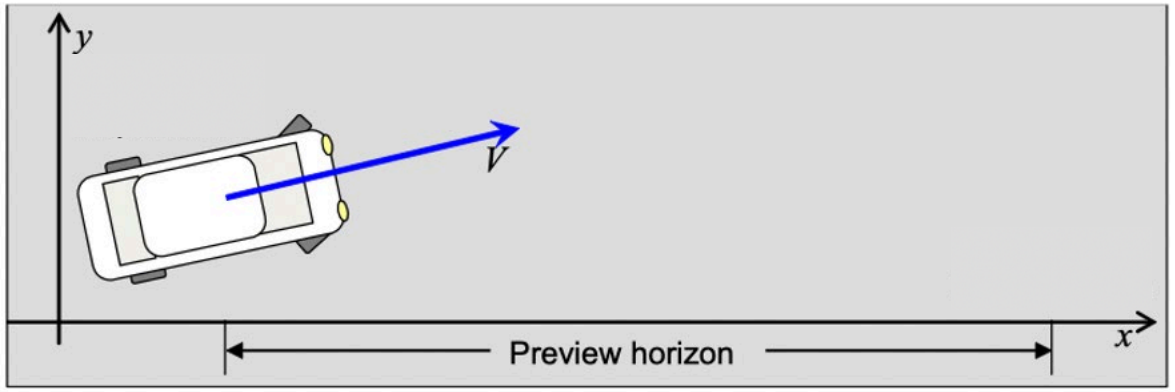
How to drive smoothly?

The purpose is not only parking perfectly into the reference positions but also as smooth as possible to avoid frequent wheel jerking or frequent throttling/braking.

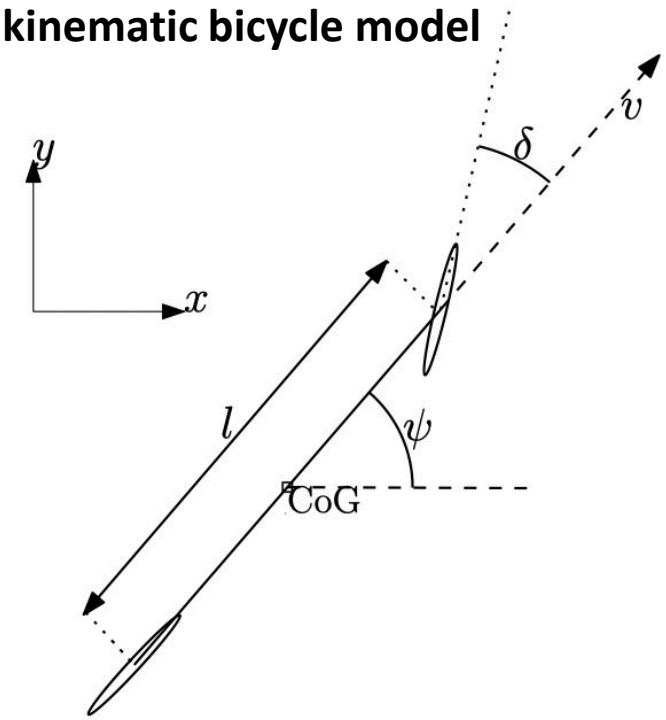
Starting code: mpc_2d_st.py

Complete the methods: *motion_model* and *cost_func*

Hint of 2D model



kinematic bicycle model



$\mathbf{x}_t = [x_t\ y_t\ \psi_t\ v_t]$

$x_{t+1} = x_t + \dot{x}\ dt$

$y_{t+1} = y_t + \dot{y}\ dt$

$\psi_{t+1} = \psi_t + \dot{\psi}\ dt$

$v_{t+1} = v_t + \dot{v}\ dt - v_t/25$

$\dot{x} = v_t \cos(\psi_t)$

$\dot{y} = v_t \sin(\psi_t)$

$\dot{\psi} = \dfrac{v_t \tan \delta}{l}$ ➡ where car length $l = 2.5$ m

$\dot{v} = \alpha * 5$ ➡ Acceleration from the pedal input is (pedal position * 5)

ψ: angle/orientation of the car

α: pedal position in the range $\alpha \in [-1, 1]$ = [full brake, full throttle]

δ : steering angle with the in the range $\delta \in [-0.8, 0.8]$

Implement 3 – Parking with Obstacle Avoidance (Assignment)

- Task 3: Drive the car from start position to end position while avoiding the obstacle

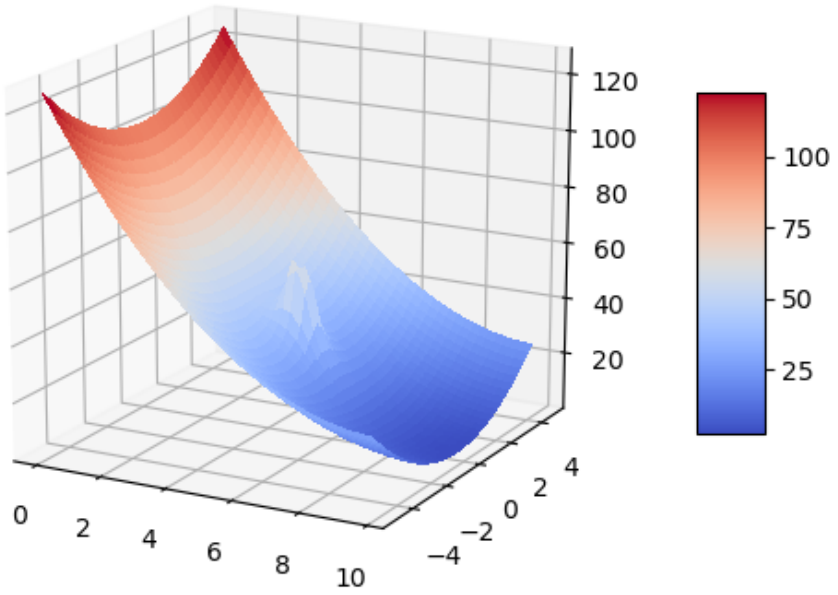
obstacle

Starting code: mpc_2d_obs_st.py

Complete the methods: *motion_model* and *cost_func*

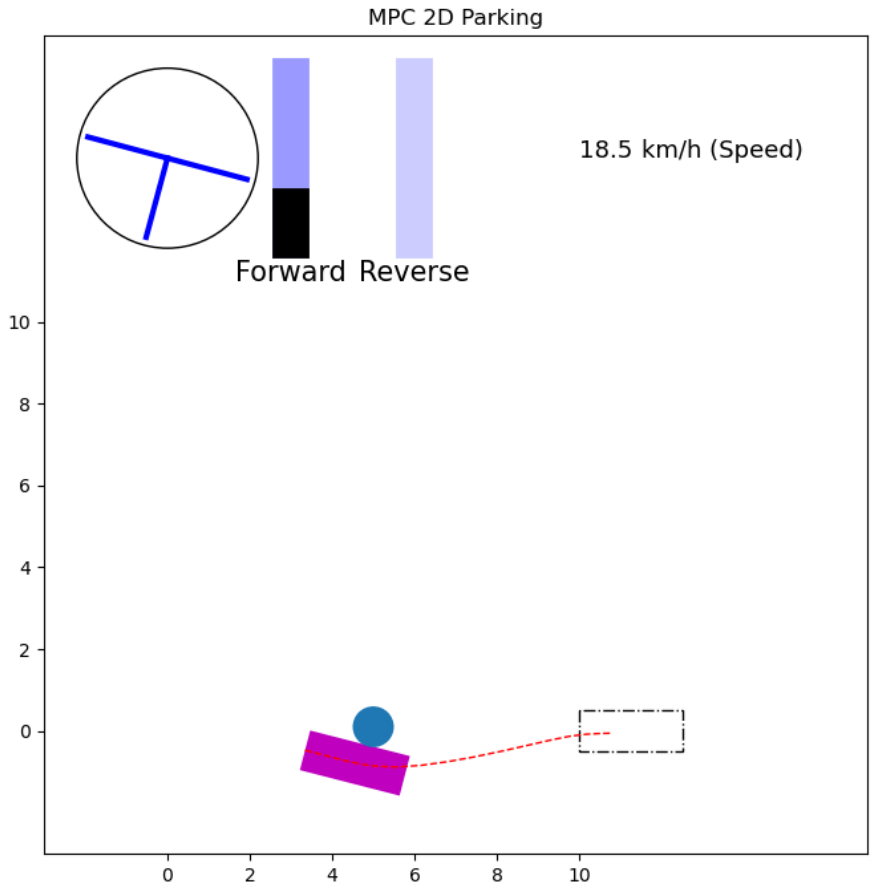
Implement 3 – Parking with Obstacle Avoidance (Assignment)

Hint: Visualize your cost function



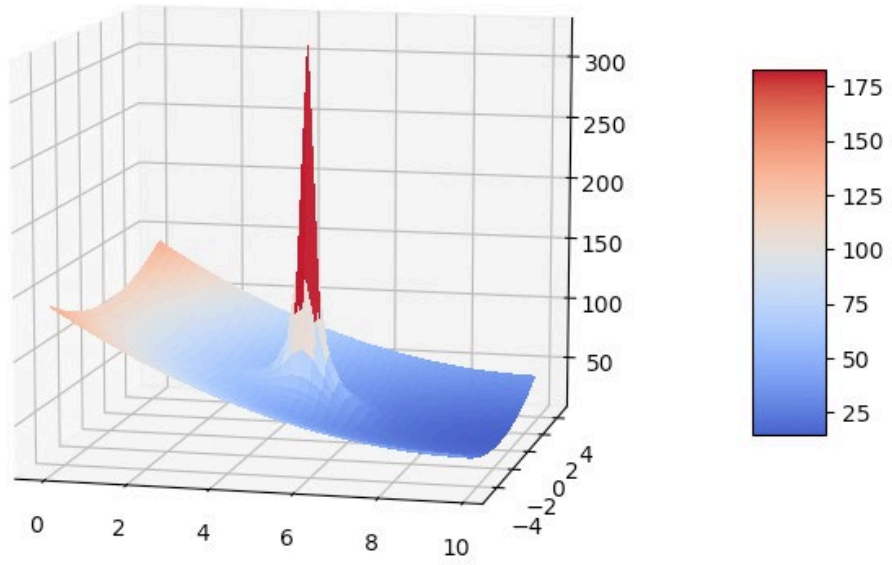
Cost function 1

Result 1



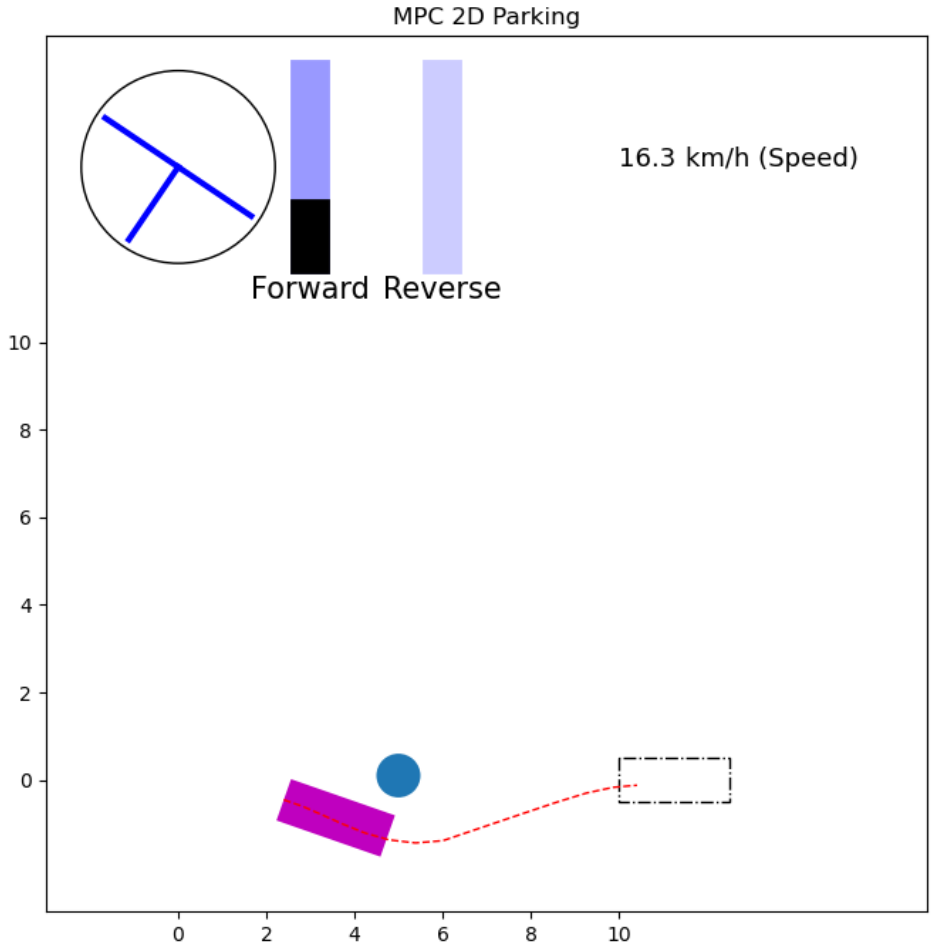
Implement 3 – Parking with Obstacle Avoidance (Assignment)

Hint: Visualize your cost function

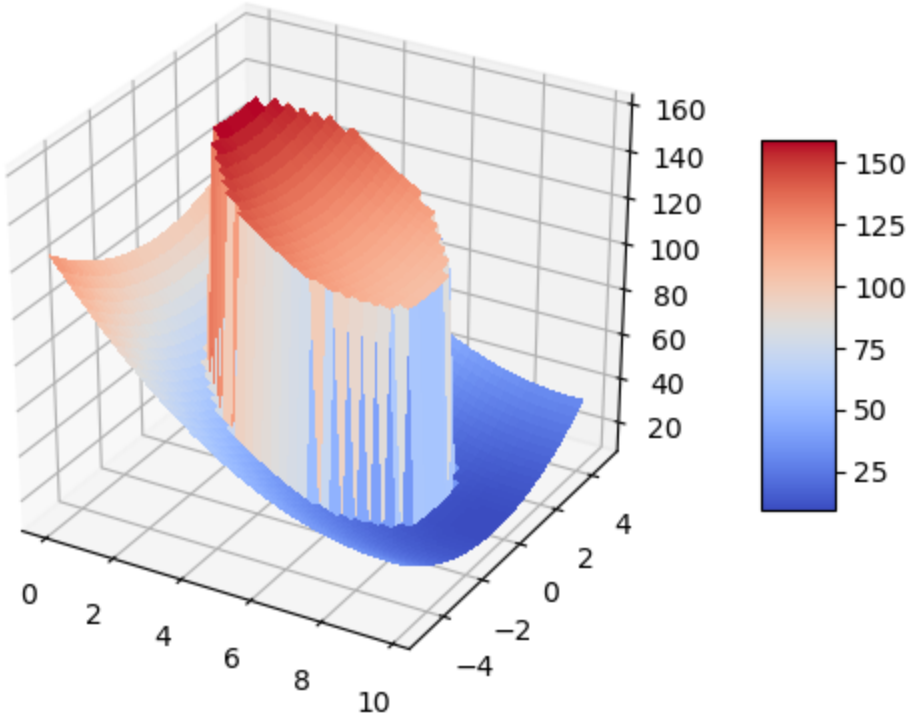


Cost function 2

Result 2



Implement 3 – Parking with Obstacle Avoidance (Assignment)



Cost function 3

Result 3