Coursework 3: Control Barrier Functions and Control Lyapunov Functions for Obstacle Avoidance

EEEN60122 & 40122 Applied Control and Autonomous Systems
(2024-25, 2nd Semester)

Lanlan Su

1 Modeling the Robot and Problem Setup

The robot used in this coursework follows a **unicycle model**:

$$\dot{x} = v\cos(\theta), \quad \dot{y} = v\sin(\theta), \quad \dot{\theta} = \omega, \quad \dot{v} = a, \tag{1}$$

where:

- x, y represent the robot's position in the plane.
- θ is the heading angle.
- v is the linear velocity.
- ω is the angular velocity (control input).
- a is the forward acceleration (control input).

The state and control vectors are given by

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ v \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \omega \\ a \end{bmatrix}.$$

The objective is to navigate from an initial position $(x_0, y_0) = (0, 0)$ to a goal location $(x_g, y_g) = (6, 6)$ by designing the control input (ω, a) while avoiding a circular obstacle using a **Control Barrier Functions (CBFs)** and **Control Lyapunov Functions (CLFs)**-based **Quadratic Programme (QP)**.

2 Pre-Lab Section

2.1 Generating Obstacles Based on Student ID

Generate an obstacle based on **the last digit of your student ID number**. The obstacle must be positioned such that it **blocks the straight-line path** from the initial position $(0, 0)$ to the goal $(6, 6)$, while ensuring that it does **not** block the initial and goal positions. Compute the centre and radius of the circular obstacle (x_o, y_o, R) as follows:

- **X-coordinate:** The last digit of Student 1 ID determines x_o. Compute it as
$$x_o = \big[(\text{last digit of Student 1 ID}) \mod 4\big] + 1.$$
- **Y-coordinate:** The last digit of Student 2 ID determines y_o. Compute it as
$$y_o = \big[(\text{last digit of Student 2 ID}) \mod 4\big] + 1.$$
- **Obstacle Radius:** The radius is determined based on the absolute difference between the coordinates and then adding 0.5:
$$R = 0.5 + |y_o - x_o|.$$
- **Answer:** $(x_o, y_o, R) =$ *[Provide your values]. You will need to implement this in your MATLAB code.*

2.2 Computing the Second Derivative of the Barrier Function

Consider the safety constraint:

$$B(x, y) = (x - x_o)^2 + (y - y_o)^2 - R^2, \tag{2}$$

where (x_o, y_o) is the obstacle centre and R is its radius[1]. The safety region is defined as
$$\{(x, y) \in \mathbb{R}^2 \mid B(x, y) \geq 0\}.$$

- Compute the **first derivative** $\dot{B}$ and the **second derivative** $\ddot{B}$ based on the robot model.
- **[10 marks] Answer:**
- Determine the relative degree of B with respect to the control inputs ω and a, and explain why a higher-order control barrier function (HOCBF) is needed.
- **[10 marks] Answer:**

[1] In the MATLAB code, an additional safety threshold $d_{\min} \geq 0$ is introduced in the barrier function to ensure a minimum safe distance from the obstacle. You may set $d_{\min}$ to 0 or a small positive value.

2.3 The Choice of Control Lyapunov Function (CLF)

Consider the CLFs consisting of two components:

$$V_1(\mathbf{x}) = \left(\theta - \arctan\left(\frac{y_g - y}{x_g - x}\right)\right)^2 + \lambda\left[(x - x_g)^2 + (y - y_g)^2\right], \quad V_2(\mathbf{x}) = (v - v_{\text{nom}})^2,$$

where $\lambda > 0$ is the weight on the distance term in V_1, and $v_{\text{nom}} = 1$ m/s is the nominal speed.

- Describe the status of the robot when both V_1 and V_2 approach 0.
- **[10 marks] Answer:**
- Compute the derivative of V_1 and V_2, respectively, and explain why a **distance-only** CLF such as $V = (x - x_g)^2 + (y - y_g)^2$ cannot be used.
- **[10 marks] Answer:**

3 Tasks for the Lab

3.1 CBF Constraints

We set the class of $\mathcal{K}$ functions to be linear functions with positive coefficients, i.e., $k_i > 0$ for $i \in \{1, 2, \ldots\}$. Define:

$$\psi_0(x, y) = B(x, y),$$
$$\psi_1(x, y) = \dot{\psi}_0(x, y) + k_1\psi_0(x, y) = \dot{B}(x, y) + k_1 B(x, y).$$

The CBF constraint adopted in the MATLAB code is given by

$$\ddot{B}(x, y) + k_1\dot{\psi}_0(x, y) + k_2\psi_1(x, y) \geq 0.$$

The forward-invariant set is

$$\{(x, y) \in \mathbb{R}^2 \mid \psi_0(x, y) \geq 0\} \cap \{(x, y) \in \mathbb{R}^2 \mid \psi_1(x, y) \geq 0\}.$$

- Suppose the system is initially within the safety zone (i.e., $B(\mathbf{x}(0)) \geq 0$). Discuss how different values of $k_1 > 0$ affect the forward-invariant set.
- **[20 marks] Answer:**

3.2 CLF Constraints

Consider the CLFs:

$$V_1(\mathbf{x}) = \left(\theta - \arctan\left(\frac{y_g - y}{x_g - x}\right)\right)^2 + \lambda\left[(x - x_g)^2 + (y - y_g)^2\right], \quad V_2(\mathbf{x}) = (v - v_{\text{nom}})^2,$$

with $v_{\text{nom}} = 1$ m/s. To enforce stability, the following CLF constraints are imposed:

$$\dot{V}_1 + \gamma_1 V_1 \leq \delta,$$
$$\dot{V}_2 + \gamma_2 V_2 \leq \delta,$$

where $\delta \in \mathbb{R}$ is a relaxation variable introduced to guarantee feasibility of the QP. Here, $\gamma_1 > 0$ and $\gamma_2 > 0$ dictate the exponential decay rates of V_1 and V_2, respectively.

3.3 Decision Variables in the QP Formulation

The **decision variables** in the Quadratic Programme (QP) are:

$$\begin{bmatrix} \omega \\ a \\ \delta \end{bmatrix},$$

where δ is the relaxation variable ensuring the QP remains feasible. The cost function is defined as

$$\begin{bmatrix} \omega \\ a \\ \delta \end{bmatrix}^\top \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & p \end{bmatrix} \begin{bmatrix} \omega \\ a \\ \delta \end{bmatrix},$$

with $p > 0$ penalizing the relaxation variable δ. The QP is formulated as:

$$\begin{aligned} \min_{\omega, a, \delta} \quad & \begin{bmatrix} \omega \\ a \\ \delta \end{bmatrix}^\top \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & p \end{bmatrix} \begin{bmatrix} \omega \\ a \\ \delta \end{bmatrix}, \\ \text{s.t.} \quad & \begin{bmatrix} A_{cbf} \\ A_{clf1} \\ A_{clf2} \end{bmatrix} \begin{bmatrix} \omega \\ a \\ \delta \end{bmatrix} \leq \begin{bmatrix} b_{cbf} \\ b_{clf1} \\ b_{clf2} \end{bmatrix}, \end{aligned}$$

where $A_{cbf} \in \mathbb{R}^{1\times 3}$ and $b_{cbf} \in \mathbb{R}$ define the CBF constraint, $A_{clf1} \in \mathbb{R}^{1\times 3}$ and $b_{clf1} \in \mathbb{R}$ define the CLF constraint for V_1, and $A_{clf2} \in \mathbb{R}^{1\times 3}$ and $b_{clf2} \in \mathbb{R}$ define the CLF constraint for V_2.

- Let
$$A = \begin{bmatrix} A_{cbf} \\ A_{clf1} \\ A_{clf2} \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} b_{cbf} \\ b_{clf1} \\ b_{clf2} \end{bmatrix}.$$
Derive A and b in your report. **Then implement these matrices in your MATLAB code.**

- **[20 marks] Answer:**

3.4 MATLAB Code

Understanding the Given MATLAB Code:

- The provided MATLAB script simulates a **unicycle robot** using a **QP-based control strategy** with CBFs and CLFs.
- Carefully study the script and identify the roles of different components (CBFs for safety, CLFs for goal convergence).
- After setting the obstacle and completing A and b as described above in the code, run the script.

Parameter Tuning:

- **Tunable Parameters:** In your MATLAB implementation, you are required to tune the relaxation variable penalty p, γ_1, and k_1. Based on your observations, explain how adjusting these parameters affects the robot's behaviour. Please include screenshots of the trajectory plots in your report.
- **[20 marks] Answer:**

4 Submission

- A **PDF report** containing written answers and trajectory plots. Your report will be marked out of 100 marks. The report must be a maximum of 6 pages. page with the name and IDs is allowed.
- The **modified MATLAB script** submitted as a `.m` file.
- All responses should be well-structured and clear.