Safety in Dynamical System Using Control Barrier Function

Qadeer Ahmed
Director Mobility System Labs

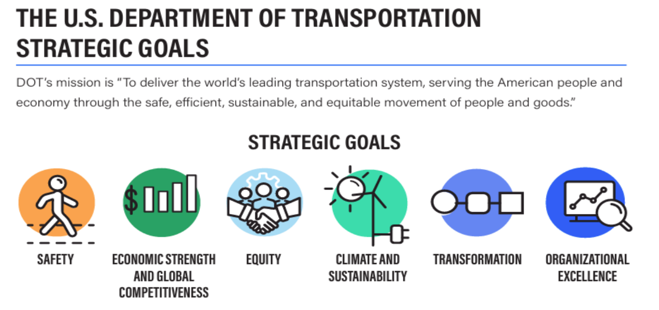
THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

Motivation

<u>SAFETY</u> is the highest priority for autonomous driving


Self-driving Waymo car kills dog amid increasing concern over robotaxis

Collision occurred as canine ran out from behind another car, but autonomous vehicle could not stop in time to avoid contact


THE U.S. DEPARTMENT OF TRANSPORTATION STRATEGIC GOALS

DOT's mission is "To deliver the world's leading transportation system, serving the American people and economy through the safe, efficient, sustainable, and equitable movement of people and goods."

STRATEGIC GOALS

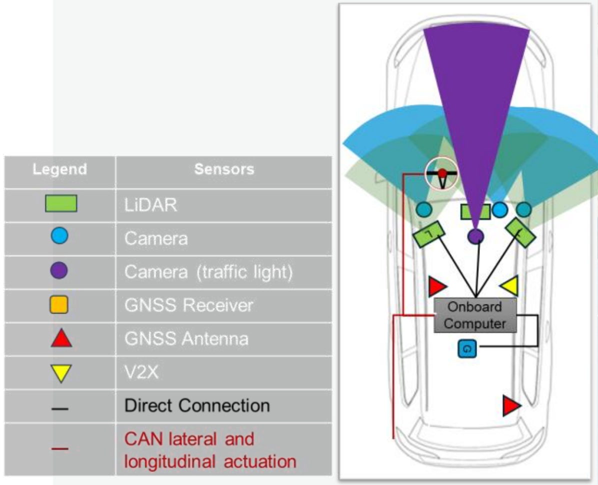
SAFETY
ECONOMIC STRENGTH AND GLOBAL COMPETITIVENESS
EQUITY
CLIMATE AND SUSTAINABILITY
TRANSFORMATION
ORGANIZATIONAL EXCELLENCE

The traffic and the environment in which Autonomous Vehicles (AVs) operate are dynamic and highly unpredictable.

[1]. Ahmed, Qadeer, and Vishnu Renganathan. *Cybersecurity and Digital Trust Issues in Connected and Automated Vehicles*. No. EPR2024009. 2024.

Introduction

Our journey from here



- A vehicle with fully autonomous lateral and longitudinal actuation, equipped with a fully functional sensor stack

to here and beyond…

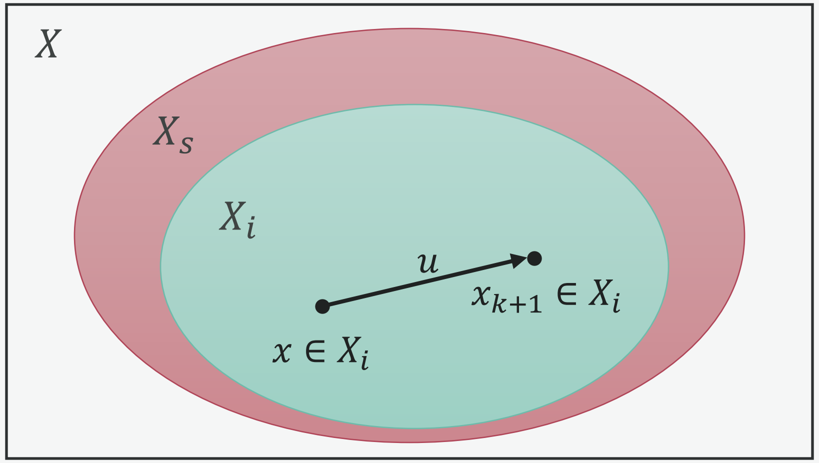
- Dynamic obstacle avoidance with safety control for fast execution on nonlinear systems

Introduction

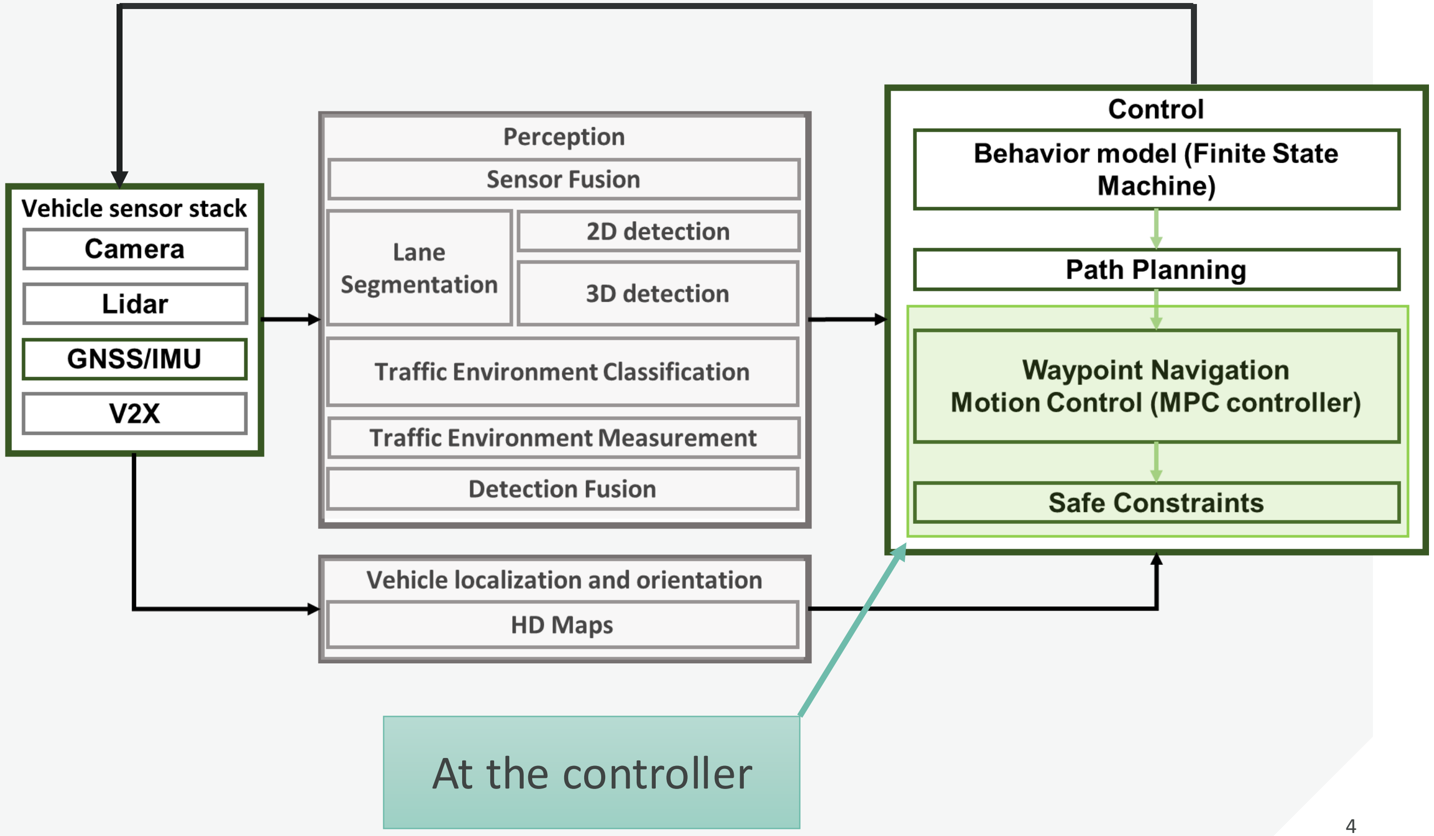
What is safe control?

Def: Controlled invariant sets

- If X_s is the set of admissible states
- X_i is a controlled invariant set (safe set) **IFF**
  - $X_i \subset X_s$
  - $\forall x \in X_i, \exists u$ such that $x_{k+1} \in X_i$



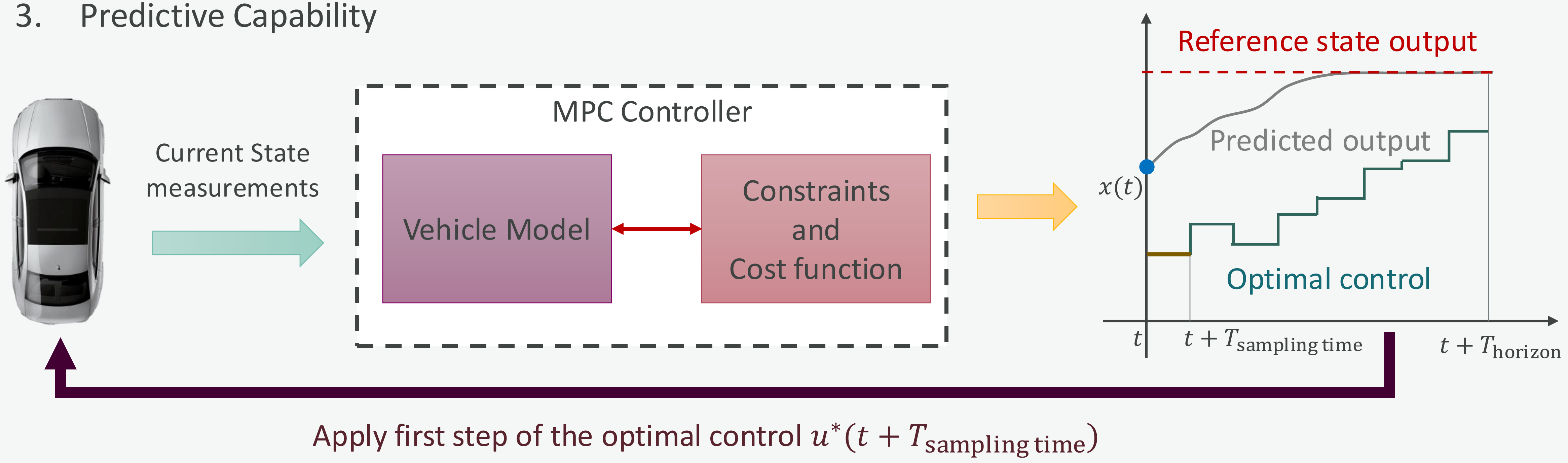
On the AV stack, where do we start?



Model Predictive Controller (MPC)

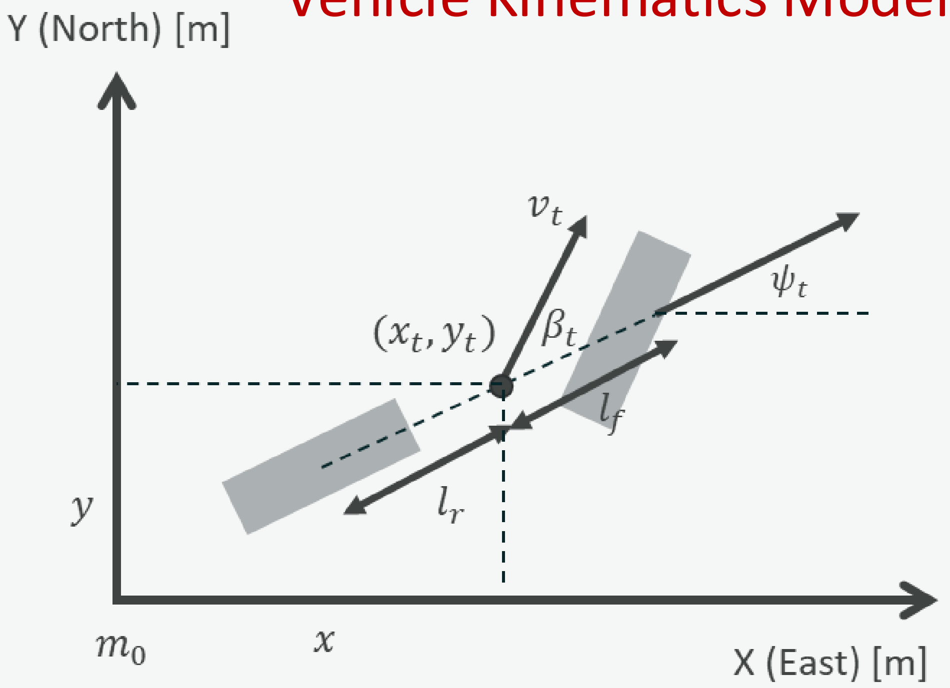
Why MPC and not any other controller?

1. Optimal controller
2. Explicit Constraint Handling
3. Predictive Capability



MPC – Vehicle Dynamics

Vehicle Kinematics Model



a: *acceleration, and* ω: *steering angle*

l_r: *center to rear wheel,* l_f: *center to front wheel*

t: *time,* dt: *sampling time*

Discrete-time vehicle dynamics

$$\beta_t = \tan^{-1}\left(\frac{l_r}{l_f + l_r} * \tan(\psi_t)\right) \quad \text{(slip angle)}$$

$$x_{t+1} = x_t + dt * v_t * (\cos(\psi_t + \beta_t)) \quad \text{(x position)}$$

$$y_{t+1} = y_t + dt * v_t * (\sin(\psi_t + \beta_t)) \quad \text{(y position)}$$

$$\psi_{t+1} = \psi_t + dt * \frac{v_t}{l_r} * sin\beta_t \quad \text{(yaw angle)}$$

$$v_{t+1} = v_t + dt * a_t \quad \text{(velocity)}$$

States: $\begin{bmatrix} x \\ y \\ \psi \\ v \end{bmatrix}$ Inputs: $\begin{bmatrix} a \\ \omega \end{bmatrix}$

MPC - Constraints

- <u>Initial State Constraint</u>: Constraint the initial state as the current position to predict the next state

$$x_0 = x_t, y_0 = y_t, \psi_0 = \psi_t, v_0 = v_t$$

Initialize the states for the optimization problem

- <u>State Bound Constraint</u>: Constraint the minimum and maximum velocity

$$v_{min} < v_k < v_{Max}$$

Velocity bounds for safe navigation and collision avoidance

- <u>Input Bound Constraint</u>: Constrain the minimum and maximum steering and acceleration

$$\omega_{min} < \omega_k < \omega_{Max}$$

$$a_{min} < a_k < a_{Max}$$

Input bound constraints to impose feasible actuation on the vehicle

- <u>Input Rate Bound Constraint:</u>

$$\dot{a}_{min} - a_s < \Delta a_t < \dot{a}_{Max} + a_s$$

$$\dot{\omega}_{min} - \omega_s < \Delta a_t < \dot{\omega}_{Max} + \omega_s$$

Input rate bound constraints to avoid high actuation rates and to have smooth inputs

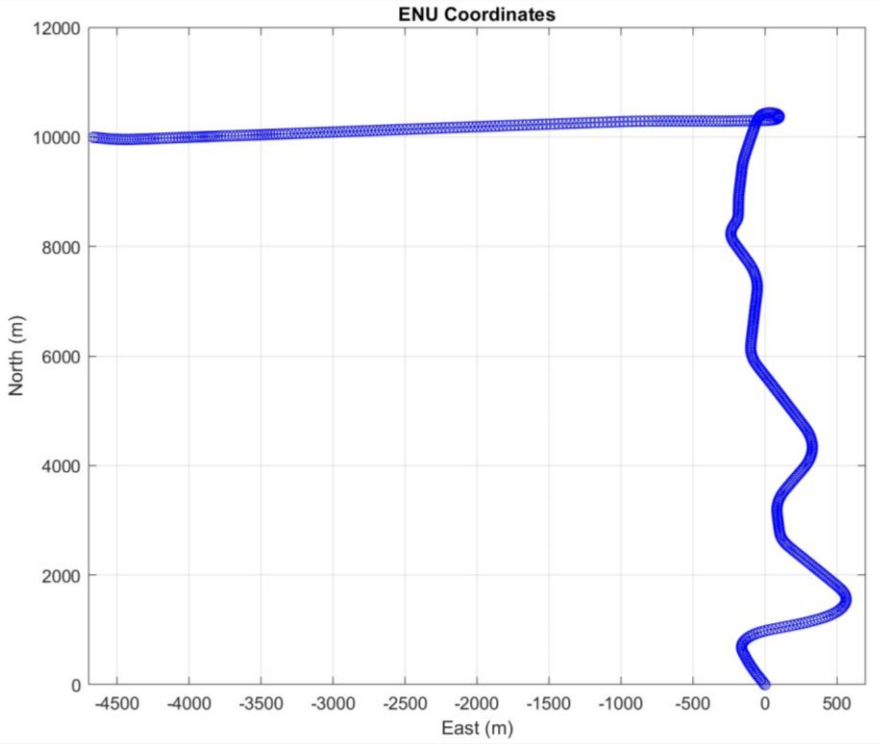
MPC – Cost Functions

What is reference set point?

It is the reference trajectory

- The set of GPS waypoints the AV is intended to follow
- We get this information from a Global path planner

For simplicity in error tracking and easier integration with perception, we convert the Lat and Lon coordinates to cartesian coordinates - East-North-Up (ENU)



MPC – Cost Functions

Error-based cost function:

$$J_x = Q_x(x_t - x_{ref})Q_x^T$$

$$J_y = Q_y(y_t - y_{ref})Q_y^T$$

$$J_v = Q_v(v_t - v_{ref})Q_v^T$$

$$J_\psi = Q_\psi(\psi_t - \psi_{ref})Q_\psi^T$$

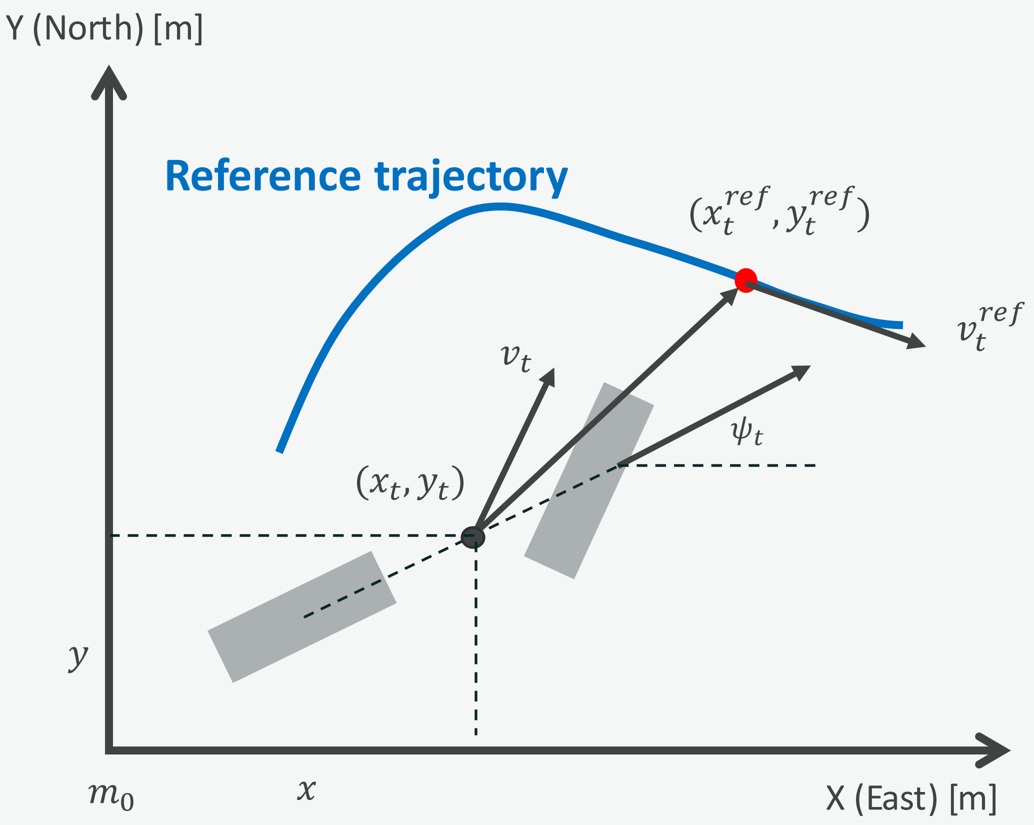
$$J_u = R(u_{t+1} - u_t)R^T$$

$$J_{total} = J_x + J_y + J_v + J_\psi + J_u$$

Optimization Goal: Minimize the J_{total}

⇒ Minimize the deviation from the reference trajectory

Vehicle Kinematics Model



MPC formulation and its performance analysis

$$J(x_t, u_t) = \sum_{k=0}^{N-1} \left(\left\| x_{t+k|t} - x_{t+k|t}^{\text{ref}} \right\|_Q^2 + \left\| u_{t+k|t} - u_{t+k-1|t} \right\|_R^2 \right)$$

$$J_t^*(x_t) = \underset{u_{t:t+N-1|t}}{\text{argmin}} \; p(x_{t+N|t}) + J(x_t, u_t)$$ → Cost function

Such that $x_{t+k+1|t} = f\left(x_{t+k|t}, u_{t+k|t}\right)$ → Vehicle dynamics

$$u_{min} < u_{t+k|t} < u_{max}$$ → Input Constraints

$$\dot{u}_{min} < u_{t+k} - u_{t+k-1} < \dot{u}_{max}$$ → Input rate constraints

$$p\left(x_{t+N|t}\right) \in X$$ → Terminal cost

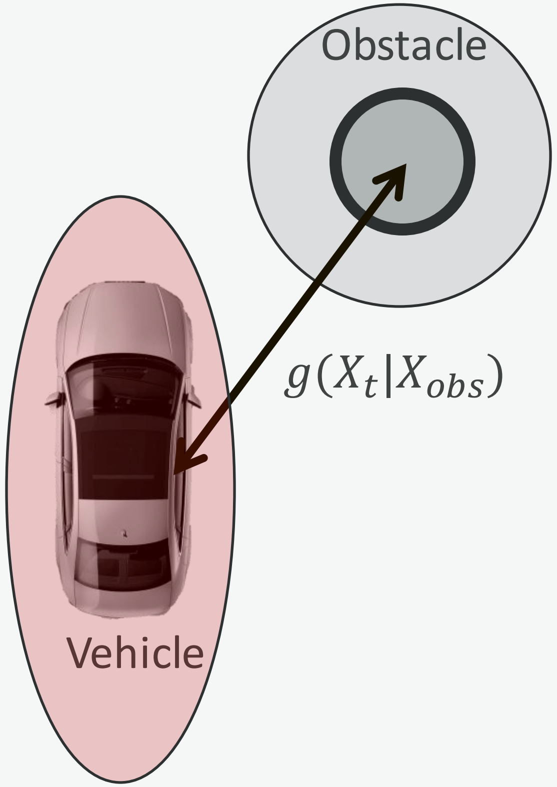
Feasibility

Stability

Optimality

Safety-Distance Constraints (MPC-DC)

Nonlinear MPC with discrete-time safe-distance constraints $\xrightarrow{u(x)}$



Define a distance constraints function $g(X_t|X_{obs})$

where $X_t = \big(x(t), y(t)\big)$, $X_{obs} = (x_{obs},\ y_{obs})$, and

D_{safe} is the safety distance.

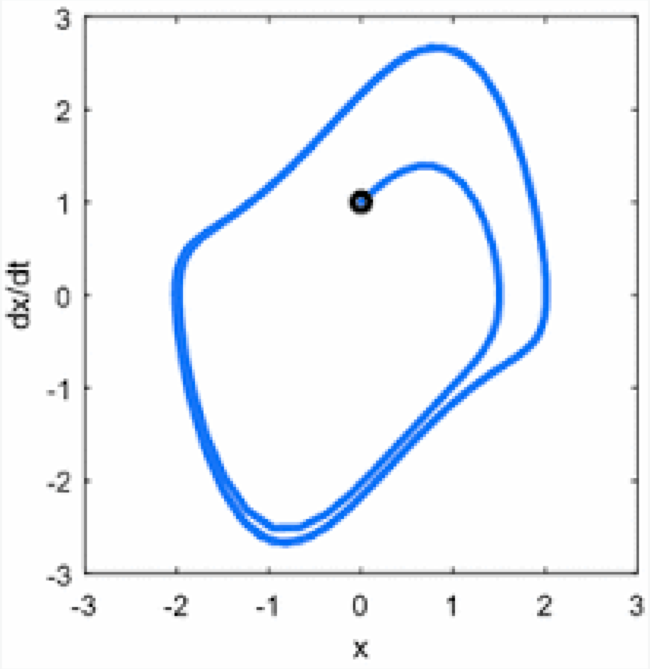
$$g(X_t|X_{obs}) = \sqrt{(X_t - X_{obs})^2} - D_{safe} \geq 0$$

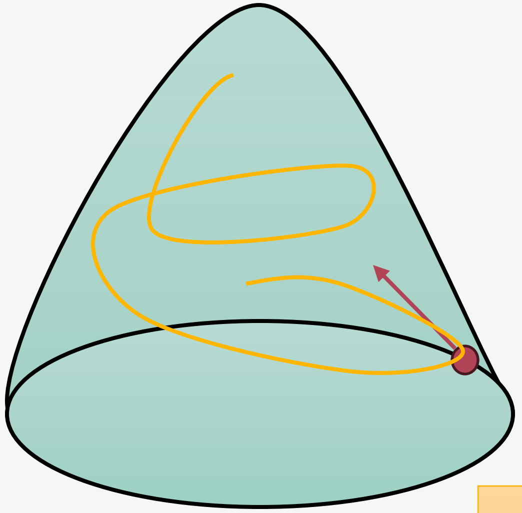
The vehicle has a safety radius, the obstacle has a safety radius, and the safety is imposed through the constraints

Safety Constraints – Control Barrier Function (CBF)

Define a function $h: \mathbb{R}^n \rightarrow \mathbb{R}$, S is the closed safe set

For a nonlinear function $\dot{x} = f(x)$





$$S = \{x \in \mathbb{R}^n : h(x) \geq 0\}$$

$$\text{Int}(S) = \{x \in \mathbb{R}^n : h(x) > 0\}$$

$$\partial S = \{x \in \mathbb{R}^n : h(x) = 0\}$$

$$\boxed{S = \{x \notin S, h(x(t)) < 0\}}$$

Unsafe condition

$\dot{h}(x) \geq 0$
$\forall x \in \partial S$

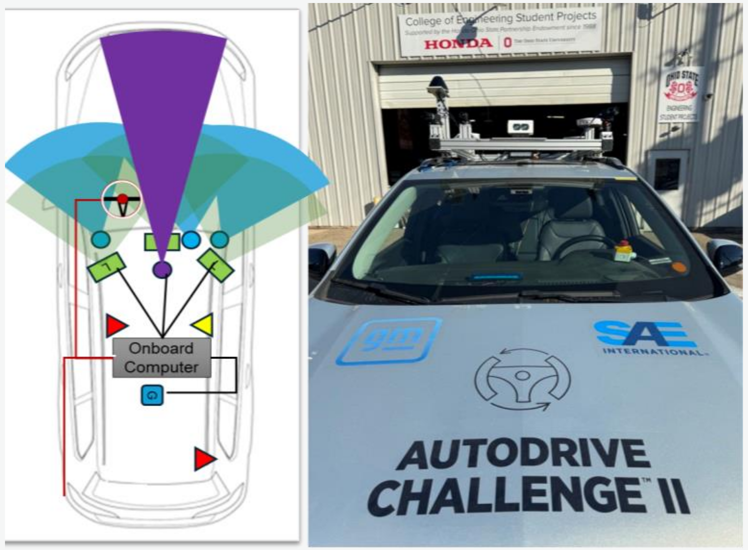
- There exists a **function h** that is always **greater than zero**
- As we approach the boundary of the safe-set, the **function h increases**

Now, for the function $h(x)$, $\dot{h}(x)$ can be implemented in discrete-time and continuous-time

Zeng, Jun, Bike Zhang, and Koushil Sreenath. "Safety-critical model predictive control with discrete-time control barrier function." In *2021 American Control Conference (ACC)*, pp. 3882-3889. IEEE, 2021.

Safe Controller
Safety constraints for obstacle avoidance

AV test vehicle



Control Loop

Vehicle dynamics

Vehicle sensor stack
- Camera
- Lidar
- GNSS/IMU

Lateral and longitudinal control

$u(x)$

Controller Formulation

(I) Nonlinear MPC with discrete-time safe-distance constraints $\rightarrow u(x)$

(II) Nonlinear MPC with discrete-time CBF constraints $\rightarrow u(x)$

(III) MPC $u^{\mathrm{MPC}} \notin \mathrm{K}_{\mathrm{CBF}}$ $\rightarrow$ QP $u^{\mathrm{CBF}} \in K_{\mathrm{CBF}}$ $\rightarrow u^{\mathrm{CBF}}(x)$

Key Takeaway

Any control action $\boldsymbol{u} \in \boldsymbol{U}$, that keeps CBF $h(x) \geq 0$, renders the safe set S forward invariant.
Then $\boldsymbol{u}$ is a safe control input

Platform and Testing setup

Tool: Python(ROS2, CasADi)

Predefined information:

- Reference Trajectory
- Obstacle Information (assume that it's from perception)

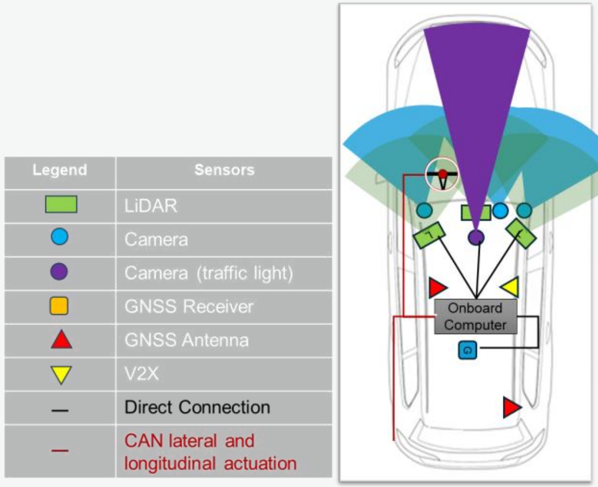
Real-time input:

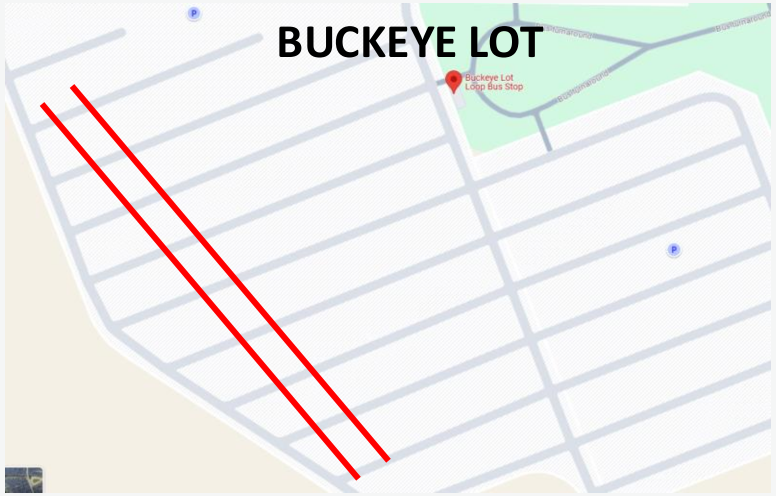
- GPS (20Hz)
- IMU(100Hz)
- Velocity (CAN BUS)

Safe Analysis:

$$d_{min} = \min(\sqrt{(X_t - X_{obs})^2}, 0)$$
$$d_{sc} = R_v + R_o + safe\ dis$$
$$d_{cc} = R_v + R_o$$

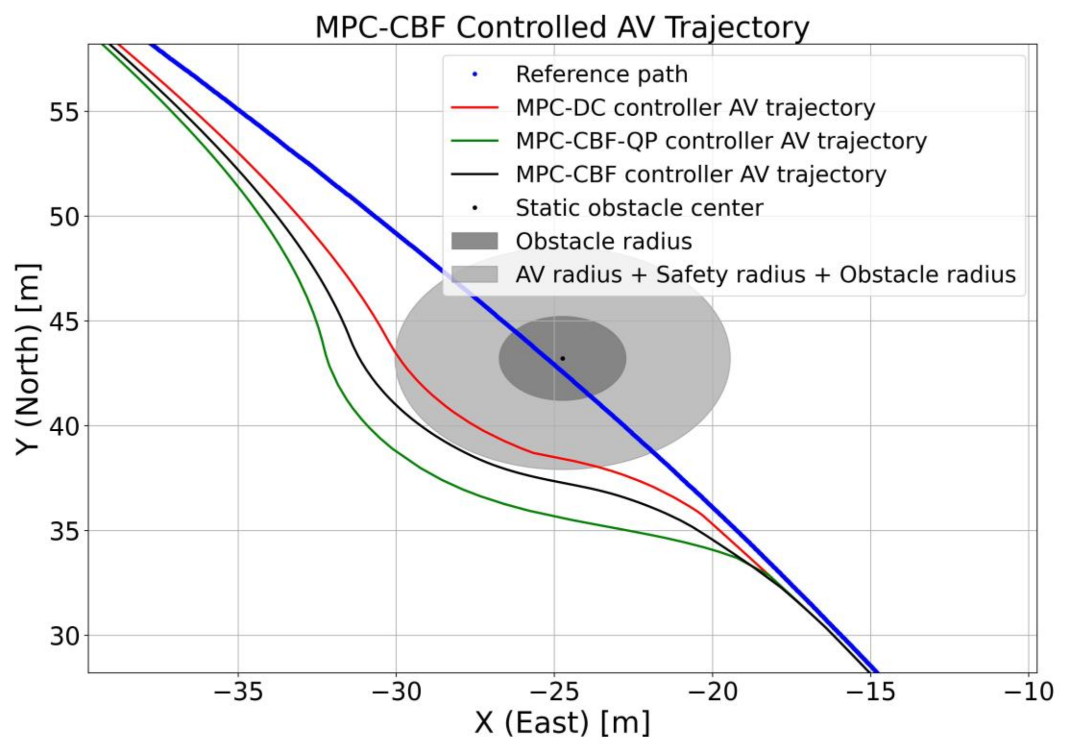
- Safe condition: $d_{min} \geq d_{sc}$
- Unsafe condition: $d_{cc} < d_{min} < d_{sc}$
- Collision condition: $d_{min} \leq d_{cc}$





Scenario I – Static Obstacle Avoidance

Safe distance = 1.5m, obstacle = 2m, vehicle = 1.8

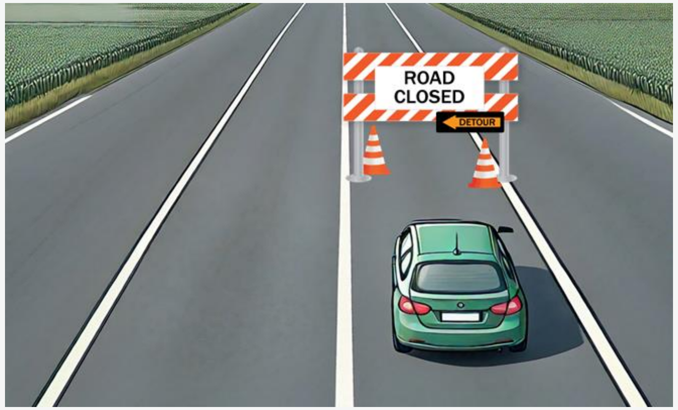


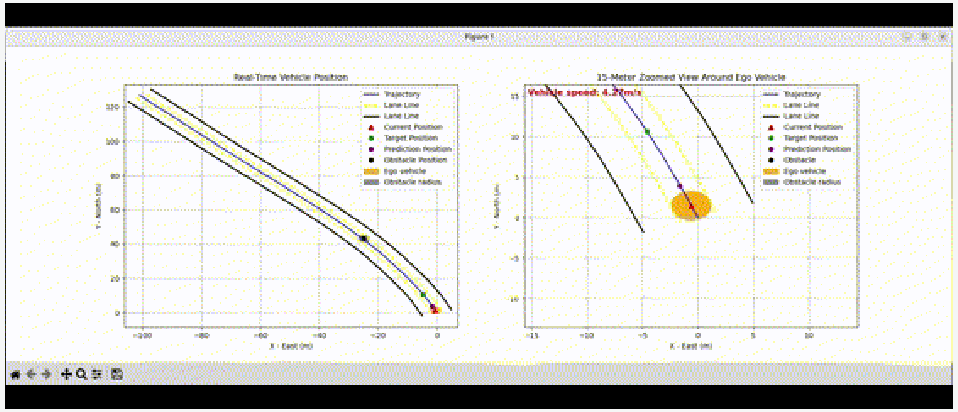
Controller	Avg. computation time [s]	Total run time [s]	Min dis to obs [m]
MPC-DC	0.0268 ± 0.0135	8.428	4.25
MPC-CBF	0.0328 ± 0.0173	10.623	5.4387
MPC-CBF-QP	0.0188 ± 0.0111	5.342	6.895

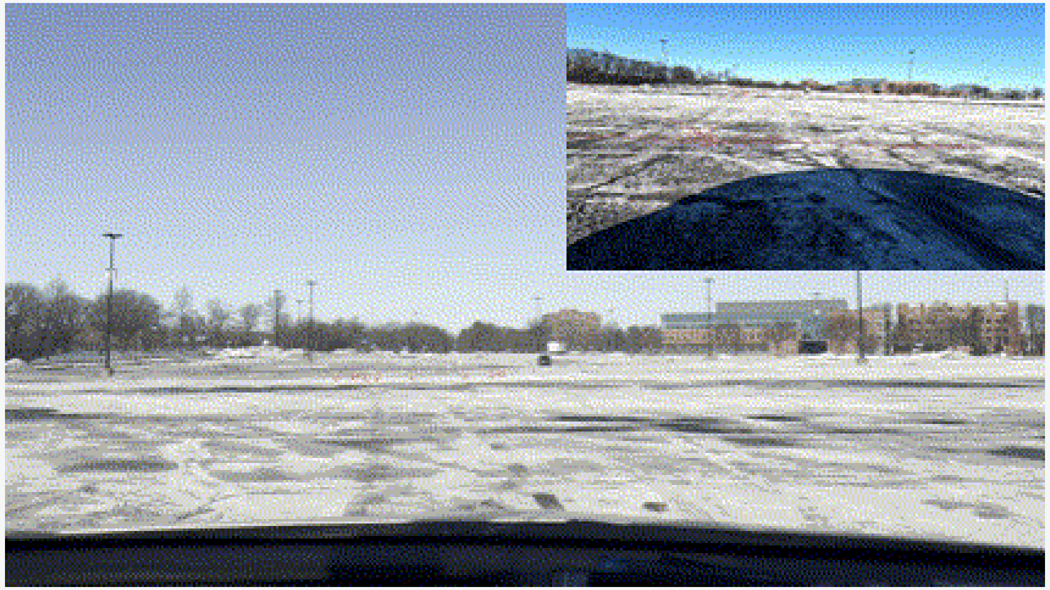
Controller	$d_{min}[m]$	d_{sc} / d_{cc} [m]	Safe Analysis
MPC-DC	4.748	5.3 / 3.8	Unsafe
MPC-CBF	5.32	5.3 / 3.8	Safe
MPC-CBF-QP	7.872	5.3 / 3.8	Safe

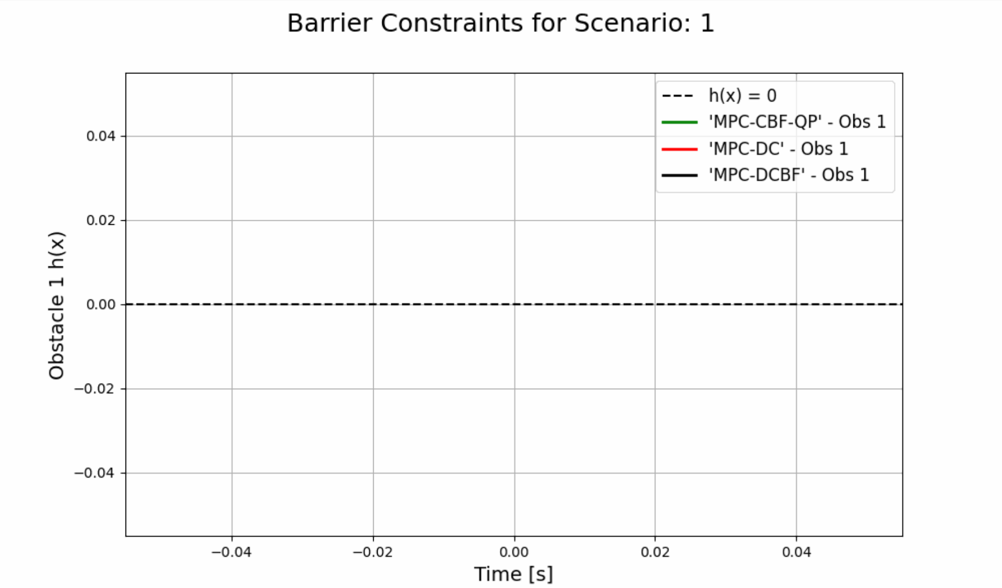
Details: https://github.com/OSU-CAR-MSL/NMPC-CBF-AV

Scenario I – Static Obstacle

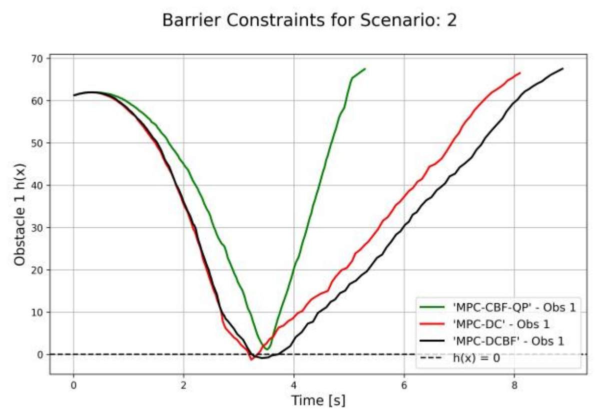


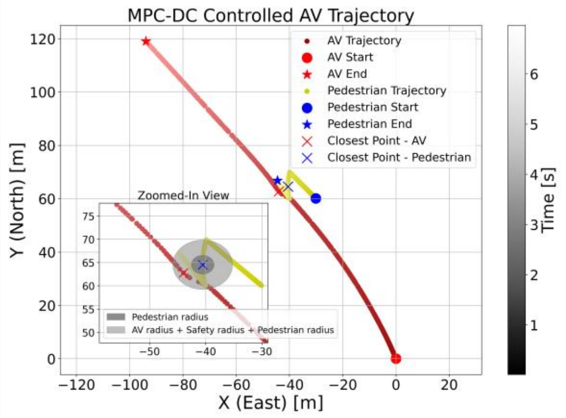


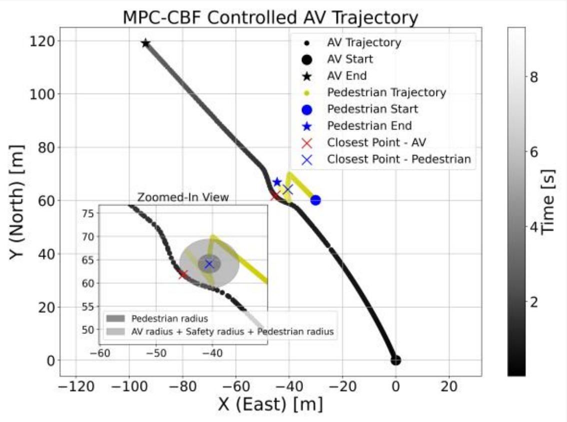


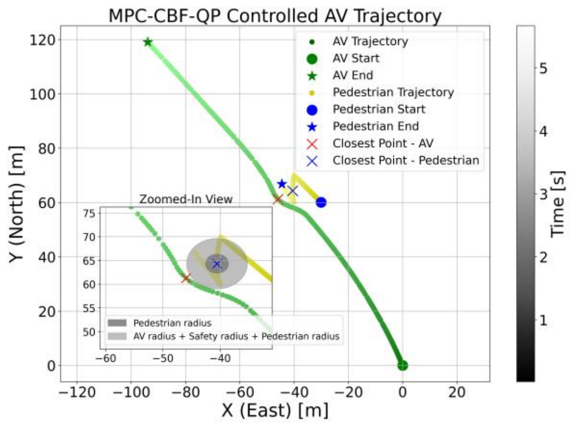


Scenario II – Sudden Pedestrian Interaction







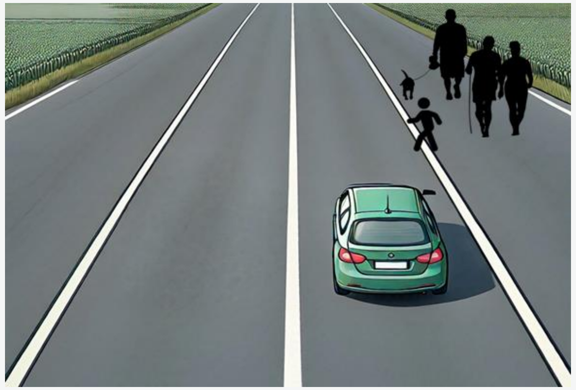


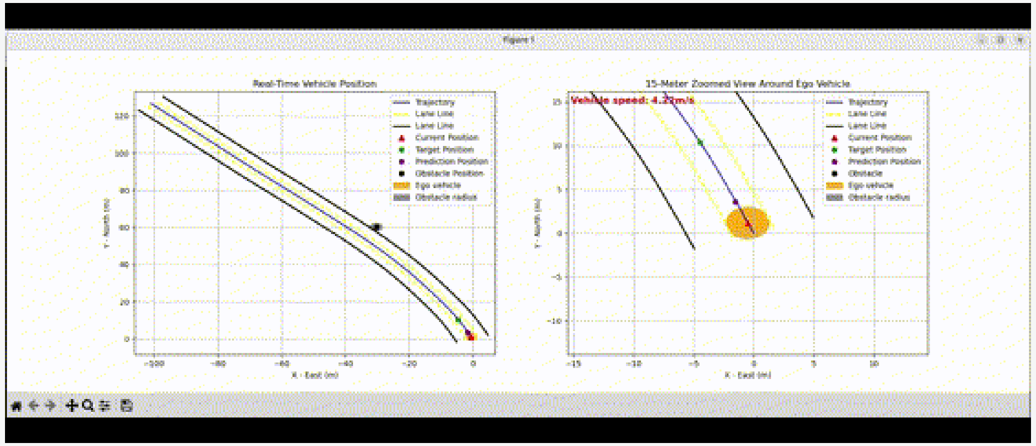
Controller	Avg. computation time [s]	Total run time [s]	Min dis to obs [m]
MPC-DC	0.0256 ± 0.0151	6.966	3.847
MPC-CBF	0.0322 ± 0.0217	9.357	5.208
MPC-CBF-QP	0.0216 ± 0.0117	5.666	6.895

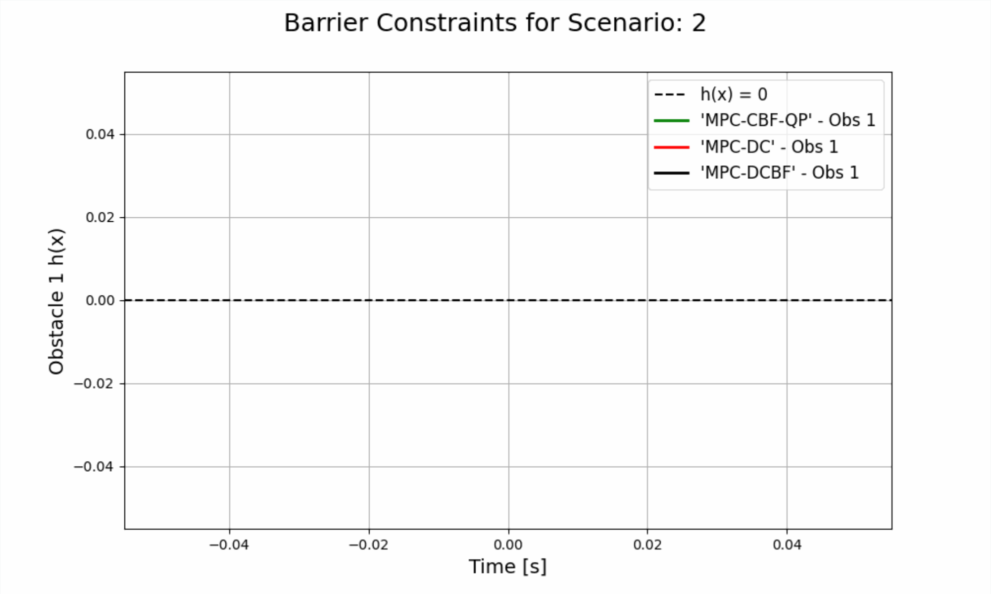
Safe distance = 3m, pedestrian = 1m, vehicle = 1.8

Controller	$d_{min}[m]$	d_{sc} / d_{cc} [m]	Safe Analysis
MPC-DC	4.557	5.8 / 2.8	Unsafe
MPC-CBF	4.929	5.8 / 2.8	Unsafe
MPC-CBF-QP	6.9565	5.8 / 2.8	Safe

Scenario II – Sudden Pedestrian







Summary

So, with all this effort can we guarantee 100% safety? **NO**

- All these efforts provide a mathematical framework to enforce safety.
- There are practical considerations:
  1. Perfect working of sensors and perception pipeline
  2. Finding the optimal or at least feasible solution – based on computation limits
  3. Uncertainties in measurement, model and the environment

Thank You!

Questions!

Special Thanks to:

Yuvraj Singh, Qizhe Xu, Javed Nur Uddin, Shengzhe Tan, Derin Durak