# Obstacle Avoidance in Dense Environments using MPC

Pei-An Hsieh, Zhenzhen Shao , Zi-Yan Liu

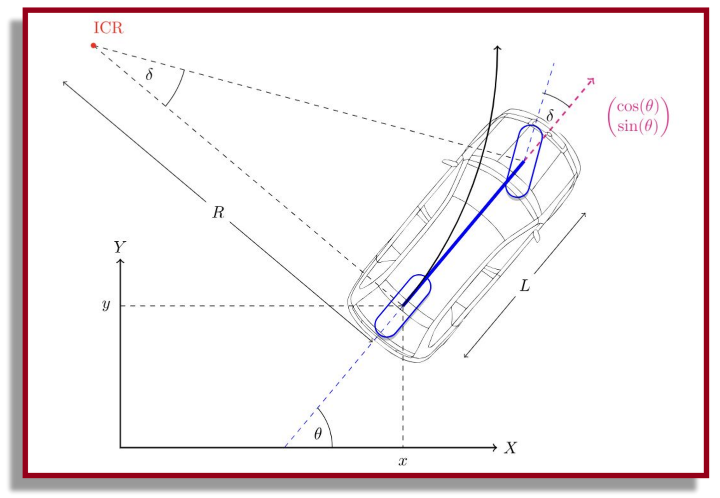
## INTRODUCTION/MOTIVATION

- Introduction
  - Implemented local planning approaches, such as Model Predictive Contouring Control (MPCC) and dynamic windows (DWA), to experiment and improve their abilities to safely navigate a mobile robot in dynamic, unstructured environments.
- Motivation
  - Robots may fail in a crowded environment (freezing robot problem) or act too aggressively and cause collision
  - Unstructured and non-convex environment make optimization hard and slow


## BACKGROUND

- Dynamic Model: Bicycle

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ \theta \\ v \end{pmatrix} = \begin{pmatrix} v\cos(\theta) \\ v\sin(\theta) \\ v\tan(\delta)/L \\ a \end{pmatrix}$$




Picture from https://thomasfermi.github.io/Algorithms-for-Automated-Driving/Control/BicycleModel.html

- Model Predictive Contouring Control:

$$J^* = \min_{u} \sum_{k=0}^{N-1} J(z_k, u_k, \theta_k) + J(z_N, \theta_N)$$

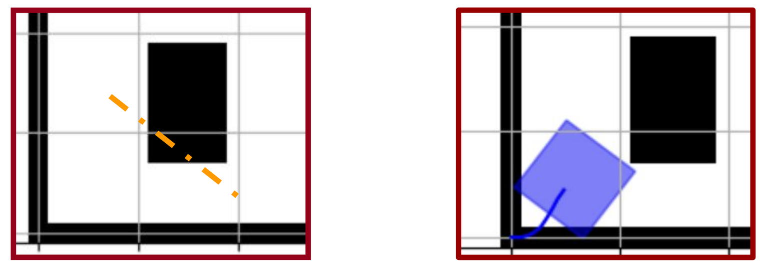
$$\text{s.t.:} \quad z_{k+1} = f(z_k, u_k), \quad \theta_{k+1} = \theta_k + u_k T$$

$$c^j_{k,\text{stat},l}(z_k) > 0, \quad \forall j \in \{1,\dots,n_c\}, l \in \{1,\dots,4\}$$

$$c^j_{k,\text{obst},j}(z_k) > 1, \quad \forall j \in \{1,\dots,n_c\}, \forall \text{obst}$$

## METHODS

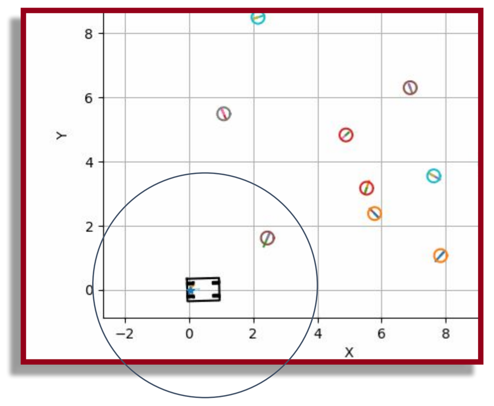
- Model Predictive Contouring Control
- Unstructured and Non-Convex environment:
  - Designing constraints are difficult and tricky
  - Static Obstacles: Approximate free space around the robot with convex rectangular regions [1]



  - Dynamic Obstacles: Use euclidean distance [1]
  - Cost function: Quadratic Cost on position error, reference speed error, and a cost function that penalizes being near to dynamic obstacles [1]

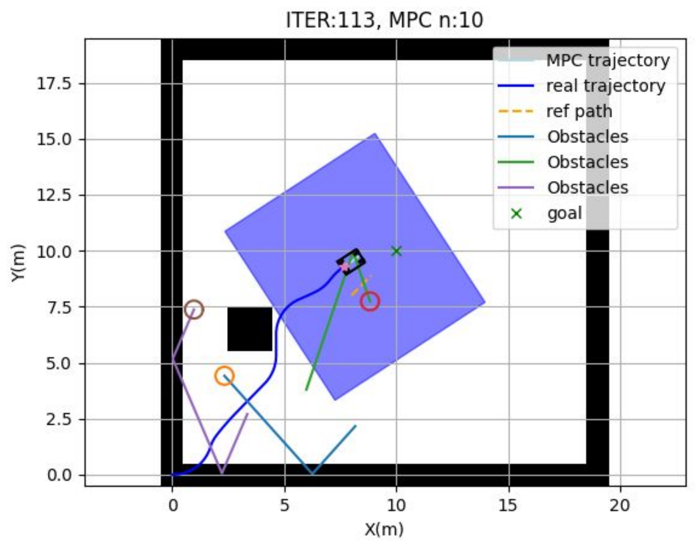
$$J_{\text{repulsive}}(z_k) = Q_R \sum_{i=1}^{n} \left( \frac{1}{(\Delta x_k)^2 + (\Delta y_k)^2 + \gamma} \right)$$

  - Obstacle Window: Consider only the obstacles that will reach the robot in a given time interval



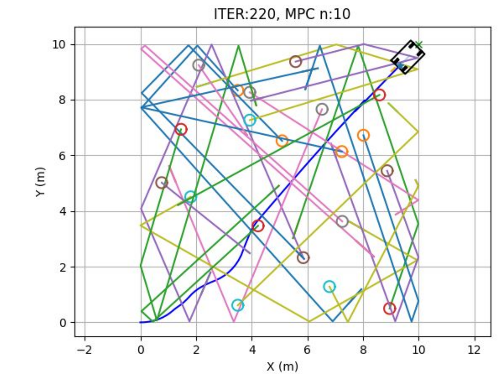
## RESULTS/FUTURE WORKS

- Results
  - By implementing the method above, we achieved essential static and dynamic obstacle avoidance in simple environments.

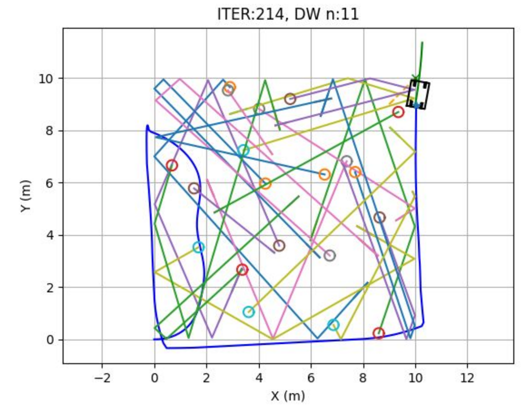



## RESULTS Continued

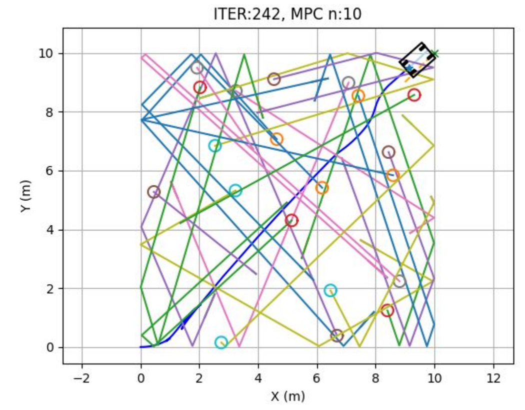
- Particularly interested in how good its dynamic obstacle avoidance capability is when we scale up the number of obstacles, we tested it under a pure dynamic environment with 10 dynamic obstacles.
- Compared with DWA [2], the results show that the MPC outperforms the DWA in safety aspects.
- An interesting phenomenon we discovered is that when we filter the obstacles by obstacle window, we achieve better results in safety instead of time.




Original MPCC
Time to reach the Goal: 22.1 s
Collisions: 1.8 s
Collision Percentage: 8.14 %
Collision Speed: 0.35 m/s
Calculation Speed: 0.503 s




DWA
Time to reach the Goal: 21.5 s
Collisions: 6.2 s
Collision Percentage: 28.8 %
Collision Speed: 2.59 m/s
Calculation Speed: 0.0482 s




MPCC with Obstacle window
Time to reach the Goal: 24.3 s
Collisions: 1.3 s
Collision Percentage: 5.35 %
Collision Speed: 0.13 m/s
Calculation Speed: 0.279 s

- Future Works:
  - Details on how obstacle windows affect the result
  - Include human reactions to the robot's presence in MPC since multiple research show that cooperative planners work better than noncooperative planners.

## REFERENCES


- [1] Brito, B., Floor, B., Ferranti, L., & Alonso-Mora, J. (2019). Model predictive contouring control for collision avoidance in unstructured dynamic environments. *IEEE Robotics and Automation Letters*, *4*(4), 4459-4466.
- [2] Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, *4*(1), 23-33..