MADR: MPC-guided Adversarial DeepReach

Ryan Teoh[1,*], Sander Tonkens[2,*], William Sharpless[2], **Aijia (Annie) Yang[2]**, Zeyuan Feng[3], Somil Bansal[3], and Sylvia Herbert[2]

[1] UCLA, [2] UC San Diego (Mechanical and Aerospace Engineering) Safe Autonomous Systems Lab, [3] Stanford University. [*]Authors contributed equally

Scan for paper & videos

annieyaj.github.io

Introduction and Theory

Motivation: Safety in Robotics

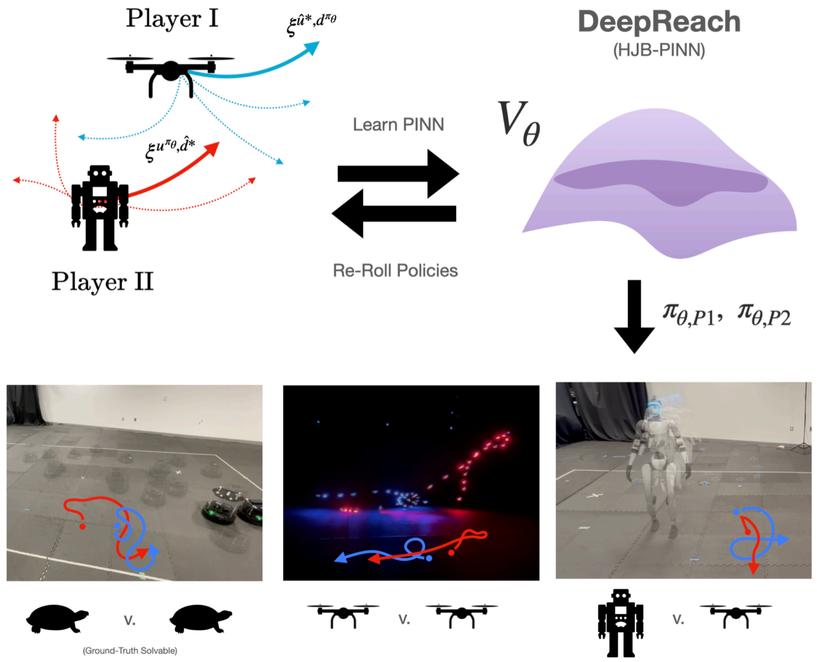
- Robots operate in uncertain environments
- Disturbances and adversarial interactions are common
- Safety guarantees are crucial in real-life deployment

Problem

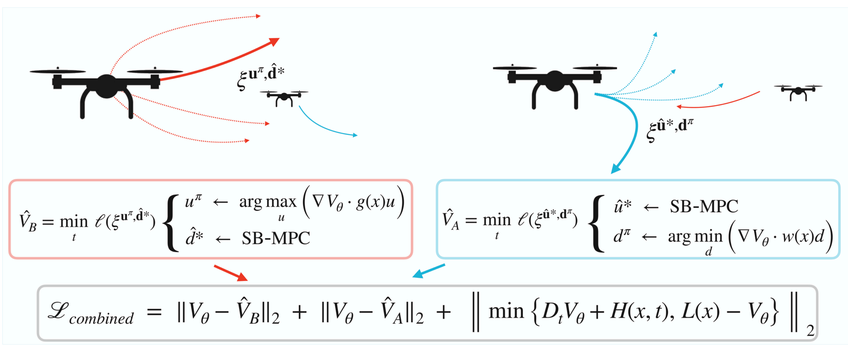
Traditional HJ Reachability suffers from the "Curse of Dimensionality"

Solution

MADR combines DeepReach neural network approximation and Robust Model Predictive Control trajectories => find optimal "cost"



Graphical depiction of the MPC formulation combining both players' rollouts in the loss function for the proposed MPC-guided adversarial PINN training:

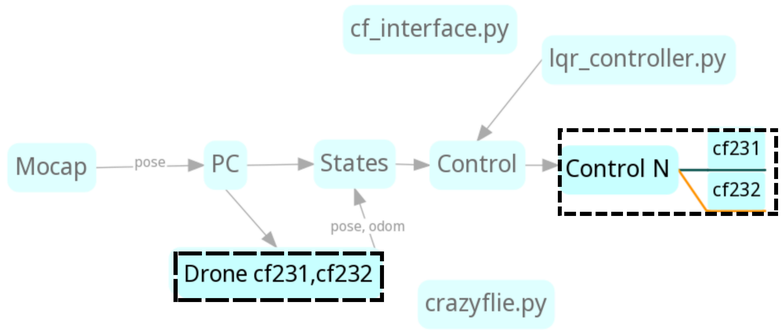


Infrastructure

System Deployment Pipeline

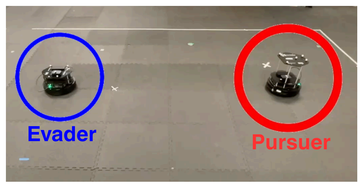
MADR algorithm => Safety controller => ROS control nodes => Mocap + Control commands => Robots

Extending Single-agent Infrastructure to Multi-agent

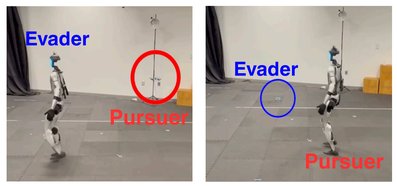


Experiments

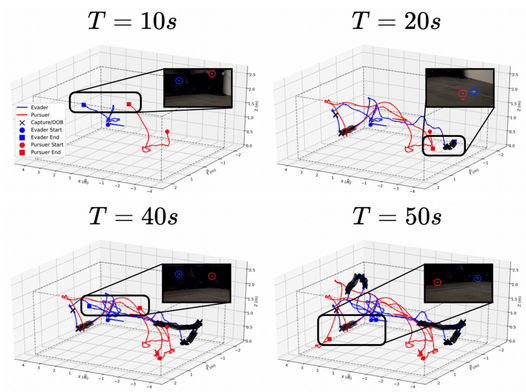
Dubin's Cars (Turtlebots)



Humanoid vs. Crazyflie (Drone)



Crazyflies (Drones)



Baselines (other methods we compared against)

- **ISAACS** - Adversarial game-theoretic reinforcement learning scheme used to solve robustness against disturbances, learning safety policy and value function separately
- **Ground Truth** - Dynamic Programming Grid-Based Methods
- **Sampling-Based MPC** - Traditional SB-MPC-based approach, maximizing/minimizing distance
- **Vanilla DeepReach** - DeepReach without MPC supervision

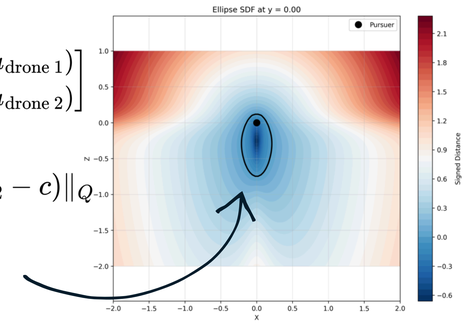
System:

$$\dot{x} = \begin{bmatrix} \dot{x}_{\text{drone 1}} \\ \dot{x}_{\text{drone 2}} \end{bmatrix} = \begin{bmatrix} f_{\text{drone 1}}(x_{\text{drone 1}}, u_{\text{drone 1}}) \\ f_{\text{drone 2}}(x_{\text{drone 2}}, u_{\text{drone 2}}) \end{bmatrix}$$

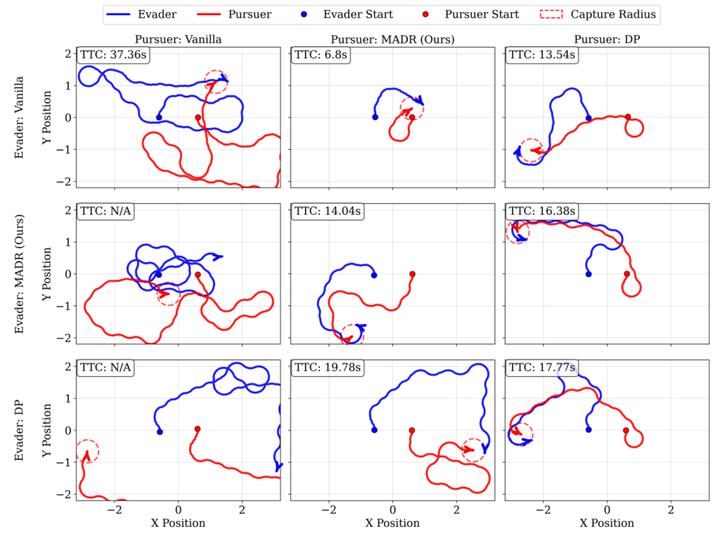
Cost Function:

$$\ell_{PE}(x) = \|x_{p,\text{drone 1}} - (x_{p,\text{drone 2}} - c)\|_Q$$

Computed Capture Set:
States where the pursuer guarantees optimal positioning



Results



Average time-to-capture (seconds) for each evader-pursuer policy pairing, evaluated over six selected initial states with large relative separation:

Evader (↑) \ Pursuer (↓)	DP	MADR	MADR-FOLLOW	Vanilla
DP	31	84	37	355
MADR	15	54	25	243
Vanilla	11	11	8	72

Contributions and Future Work

- #1: multi-agent control infrastructure
- #2: robot communication and data collection
- #3: MADR controller integration into control stack
- #4: hardware experiments, testing, and debugging
- Future: compare against other baselines and integrate humanoid dynamics

To appear in IEEE International Conference on Robotics and Automation (ICRA) 2026
Project page: https://land-dev.github.io/madr/

UC San Diego JACOBS SCHOOL OF ENGINEERING

idea engineering student center