Fast UAV Trajectory Generation using Bilevel Optimization

Weidong Sun*, Gao Tang*, and Kris Hauser

Presented by Yifan Zhu

Duke University

*Equal Contribution

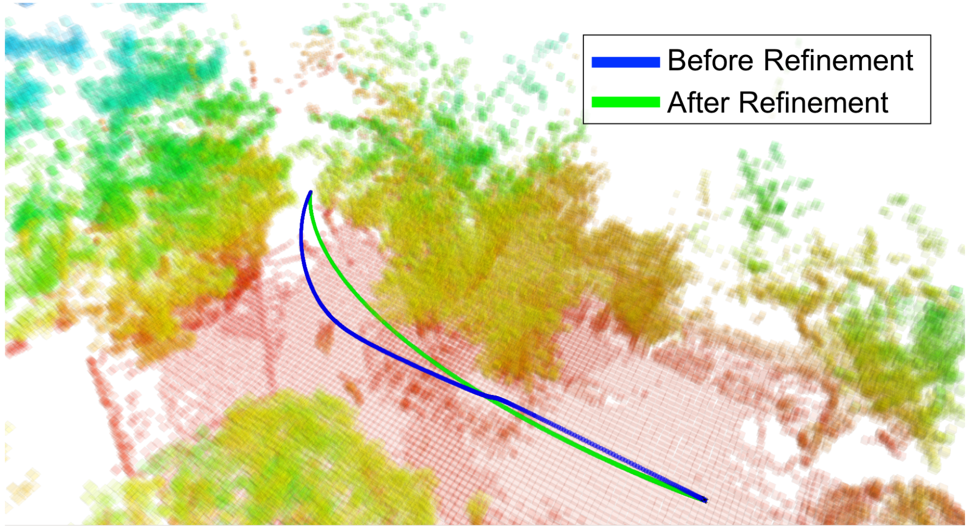
DUKE ROBOTICS

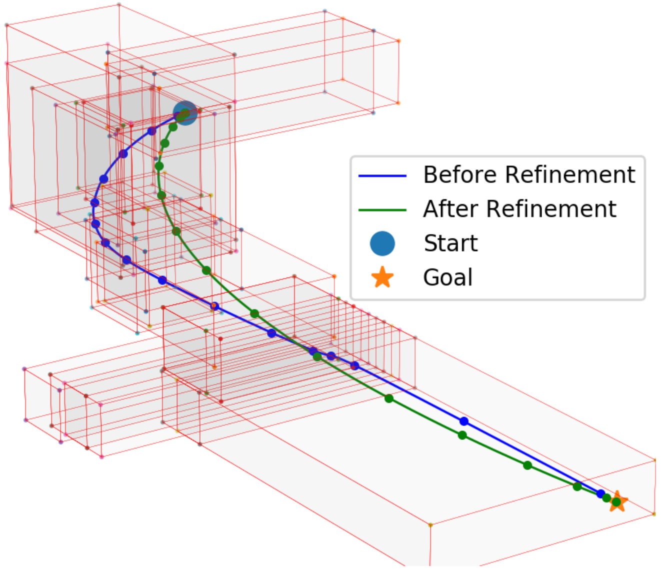
IML Intelligent Motion Lab

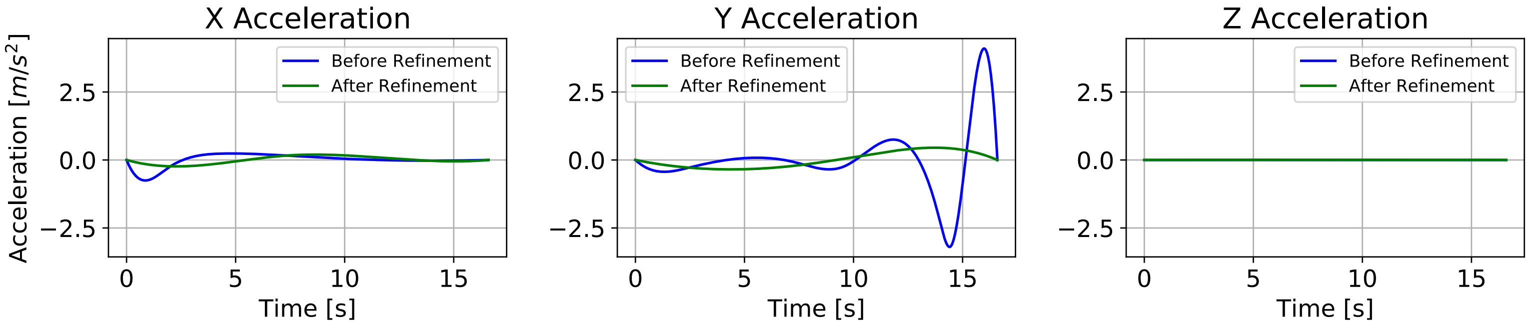
Motivation

- Time allocation for spline trajectories is important, but hard

	Jerk	Length
Before[1]	45.06	20.7m
After	0.15	19.9m







[1] F. Gao et al., "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," *ICRA*, 2018

Why hard?

- Time enters optimization nonlinearly
- Time is refined by gradient descent, but gradient is hard to compute.

Formulation

For a flight corridor with n segments, use a piecewise Bézier curve of order d:

$c \in R^{3n(d+1)}$: control points of the curve

$y \in R^{n}_{++}$: time allocation

minimize $J = \underbrace{c^T P(y) c}_{\text{Jerk}} + w \underbrace{\mathbf{1}^T y}_{\text{Traversal time}}$

Quadratic in c
Nonlinear in y

subject to $G(y)c \leq h$

Trajectory stays in flight corridor
Velocity/acceleration stay in the bound

$L(y)c = m$

C^2 continuity at knot points
Trajectory starts/ends at initial/final state

$Ay \leq b, Cy = d$

Fixed traversal time (optional),
Time is positive

Bilevel Formulation

minimize $J = c^T P(y) c + w\, 1^T y$

subject to $G(y) c \leq h$
$L(y) c = m$
$Ay \leq b, Cy = d$

→

minimize $J = c^T P(y) c + w\, 1^T y$

subject to $\mathrm{c} \in \operatorname{argmin} \{J:\ G(y)c \leq h, L(y)c = m\}$
$Ay \leq b, Cy = d$

We use constrained gradient descent:

$$y = y - \alpha \mathrm{proj}_{\mathrm{A,C}}(\nabla_y J^\star(y))$$

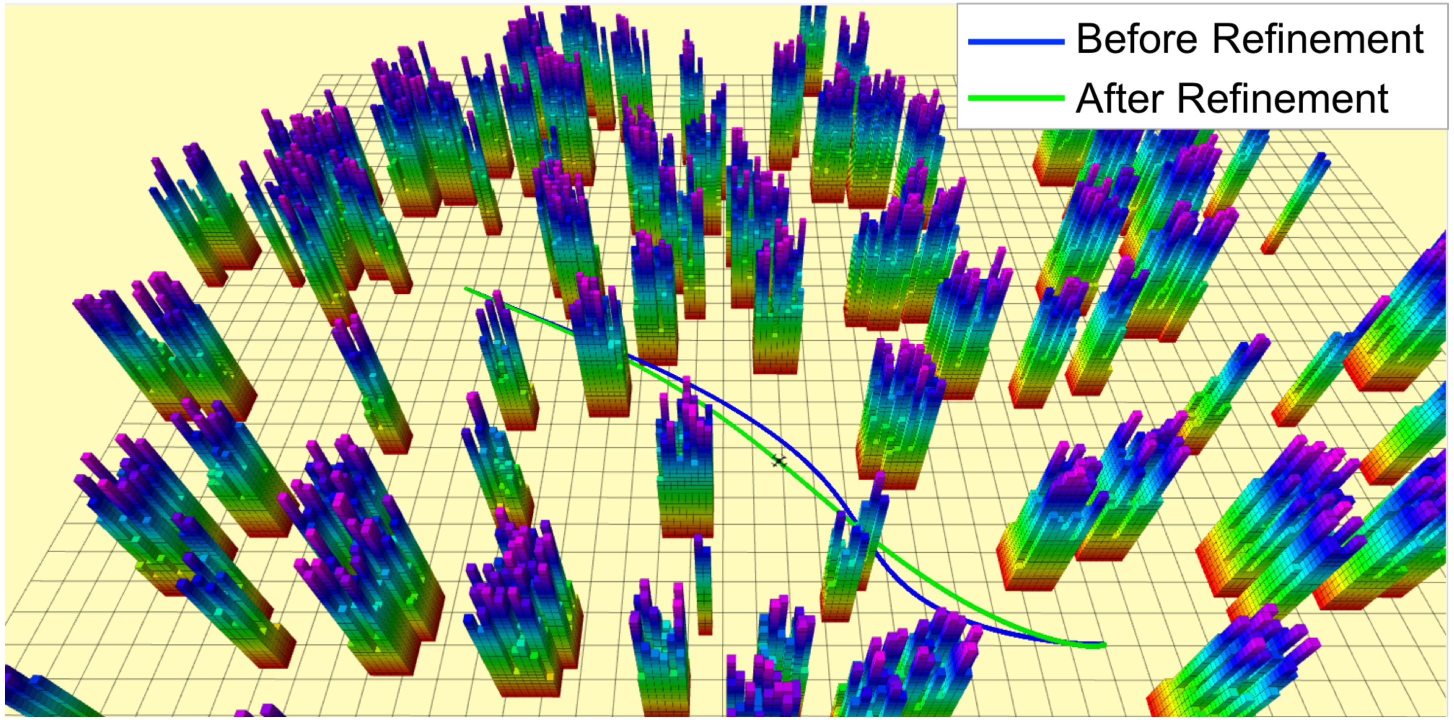
Gradient computation (from sensitivity analysis of parametric NLPs):

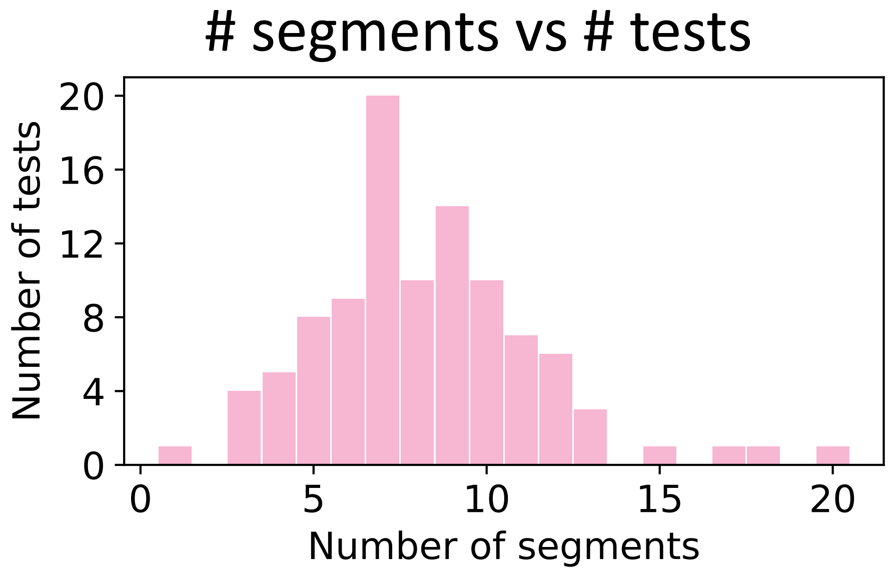
$$\nabla_y J^\star(y) = \nabla_y J + \lambda^T \nabla_y (\, G(y)c - h) + \nu^T \nabla_y (L(y)c\ - m)$$

λ, ν: Lagrange multipliers, which can be obtained “for free” by solving the QP

Numerical experiments: real-time performance

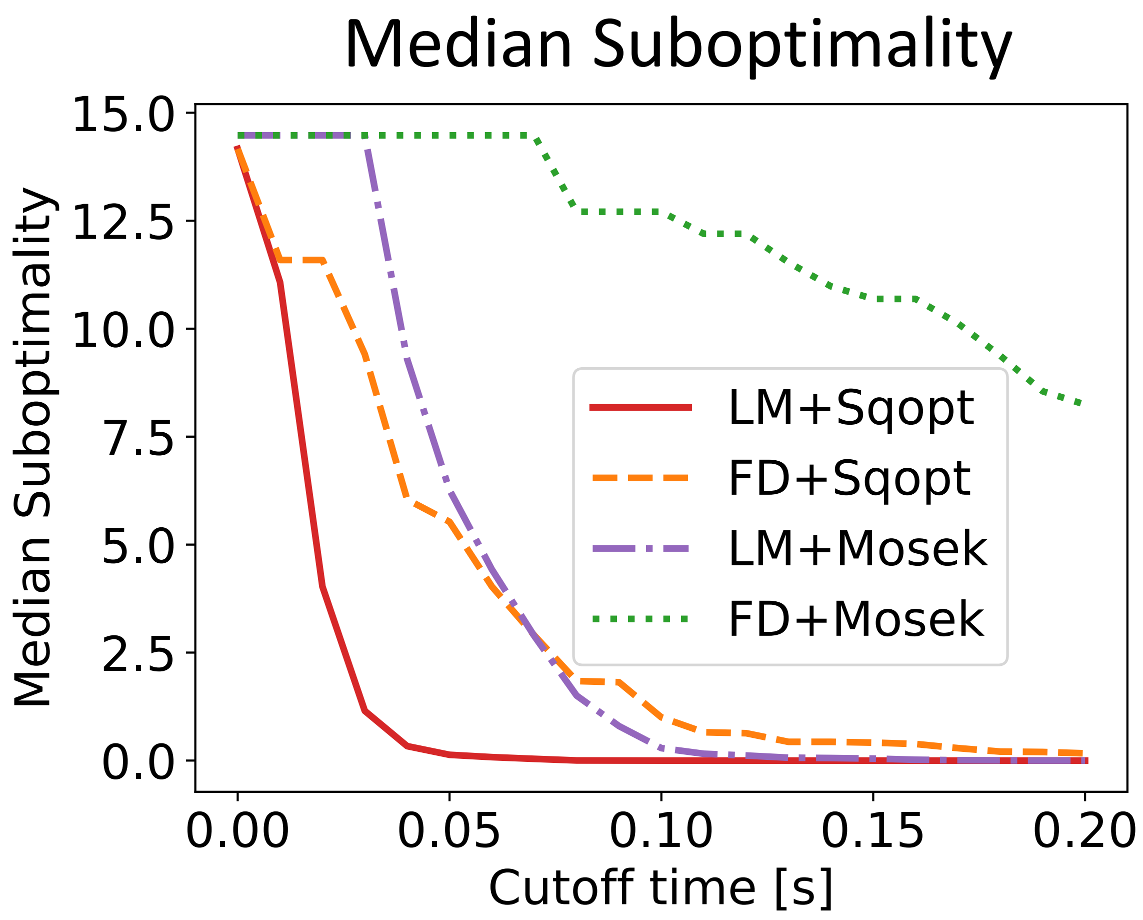
- 100 tests: Random environment + random start/goal, fixed T, $w = 0$.
- We solve c, y to optimal.

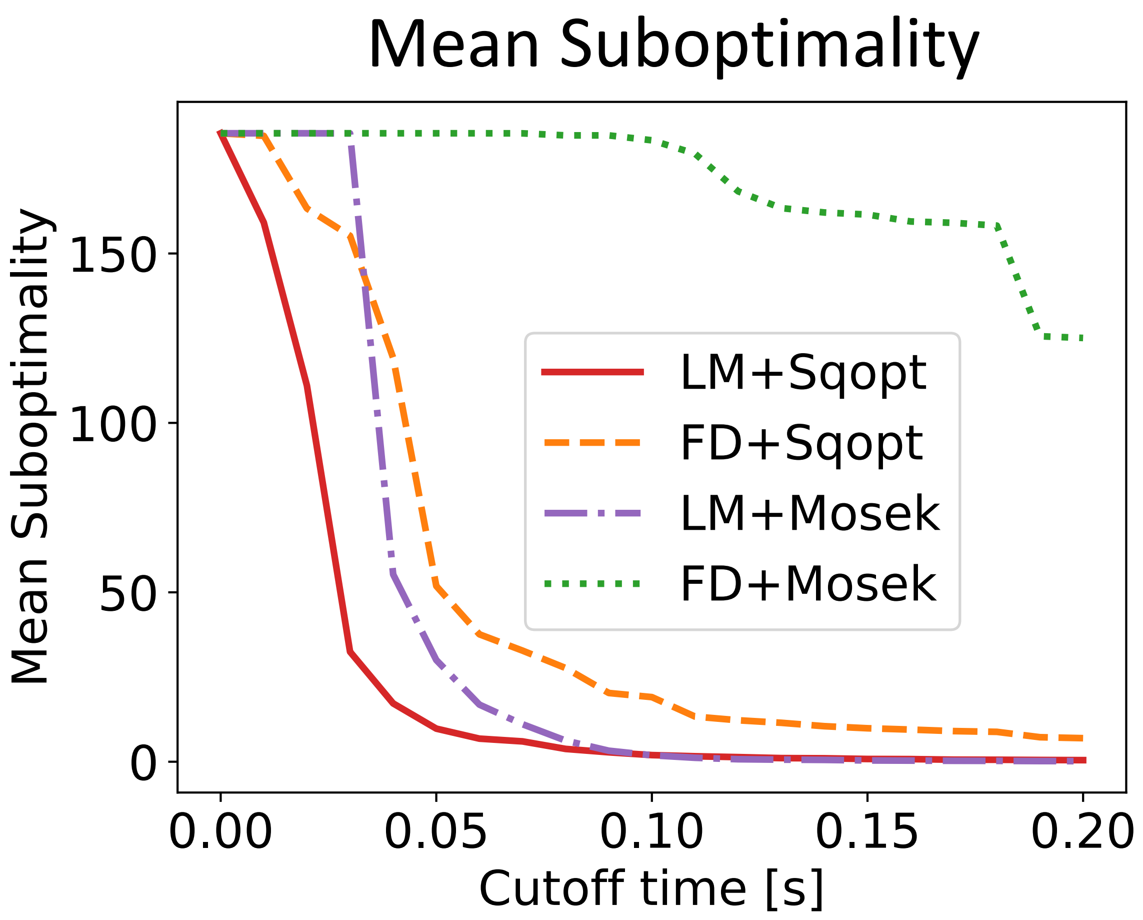




Numerical experiments: real-time performance

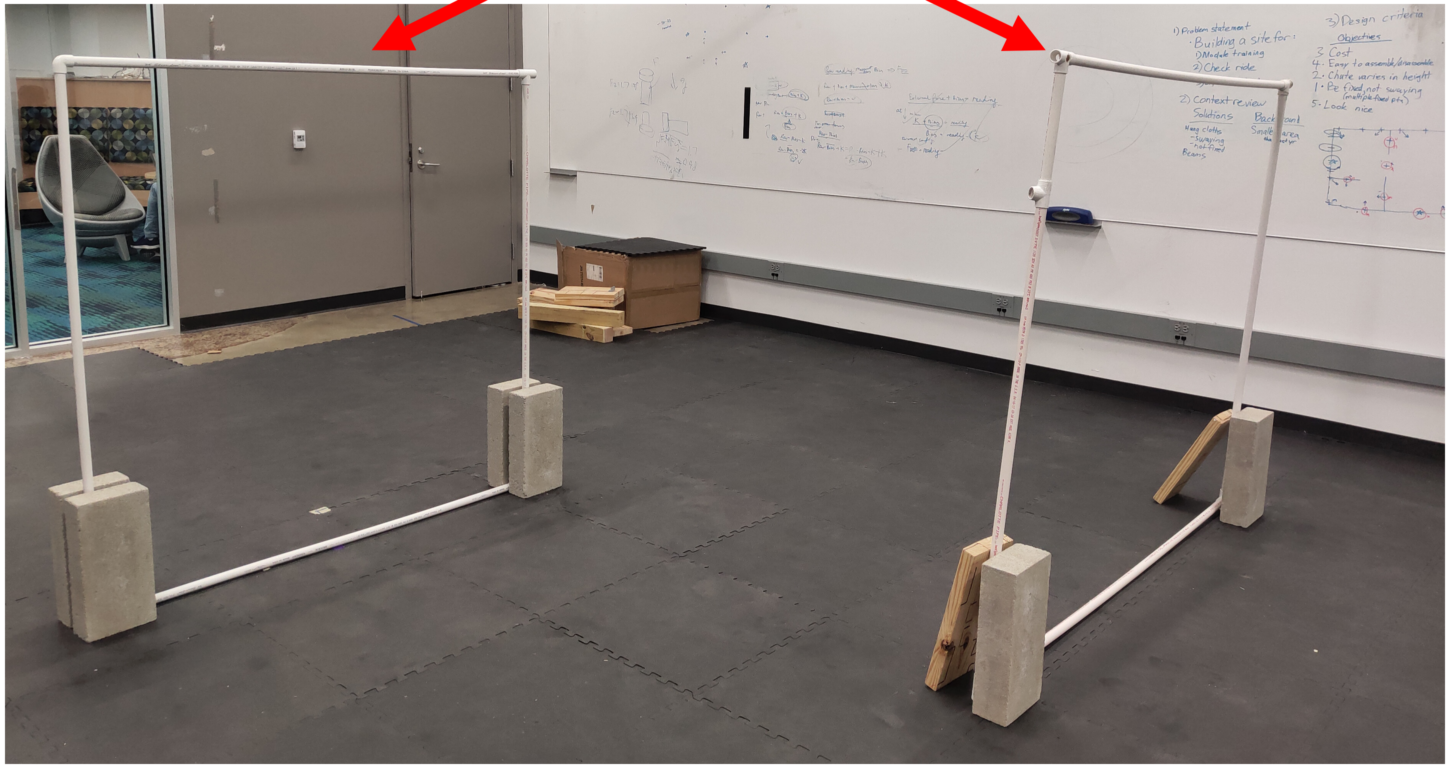
- Our method (LM) vs finite difference (FD)
- 2 QP solvers are used: Sqopt (active-set), Mosek (interior-point)



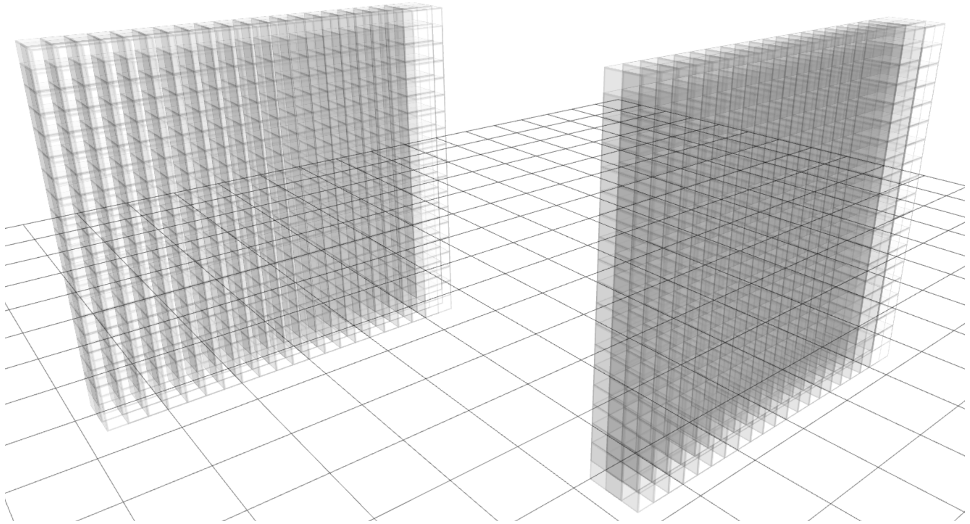


Physical Experiment Setup

Frames treated as walls



Physical layout

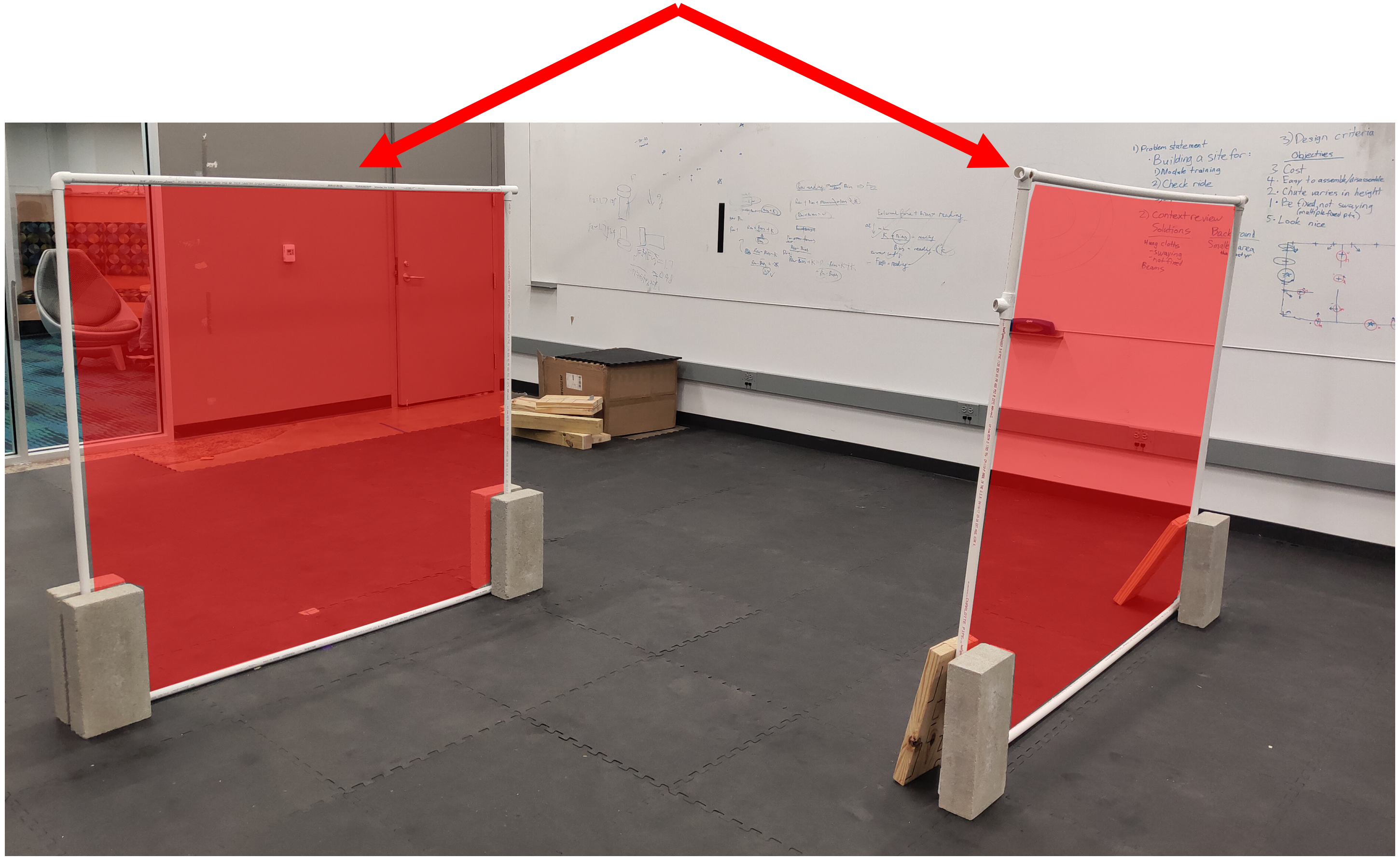


Rviz visualization

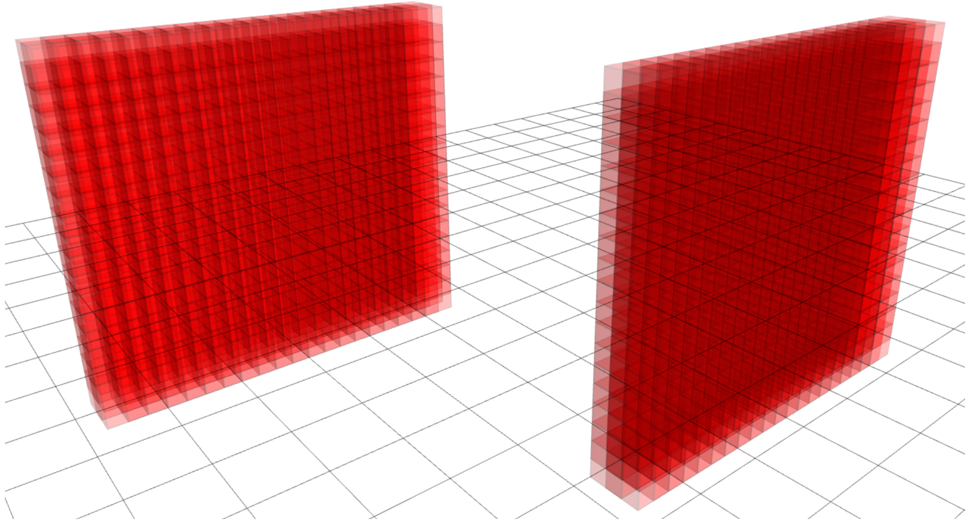
We use the Crazyflie 2.1

Physical Experiment Setup

Frames treated as walls



Physical layout



Rviz visualization

We use the Crazyflie 2.1

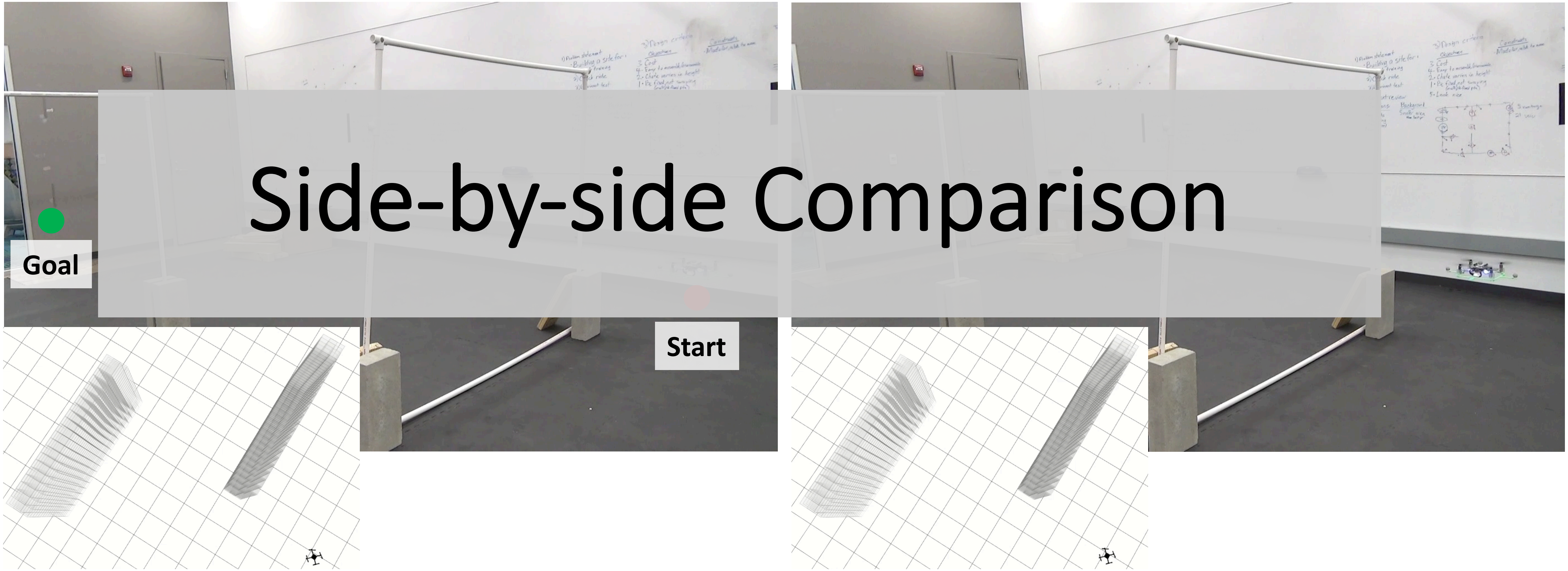
[1] F. Gao et al., "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," *ICRA*, 2018

Experiment 1

Our method plans a faster trajectory than state-of-the-art [1] with the same jerk

Gao et al. [1], $T = 5.32$s, Jerk=39

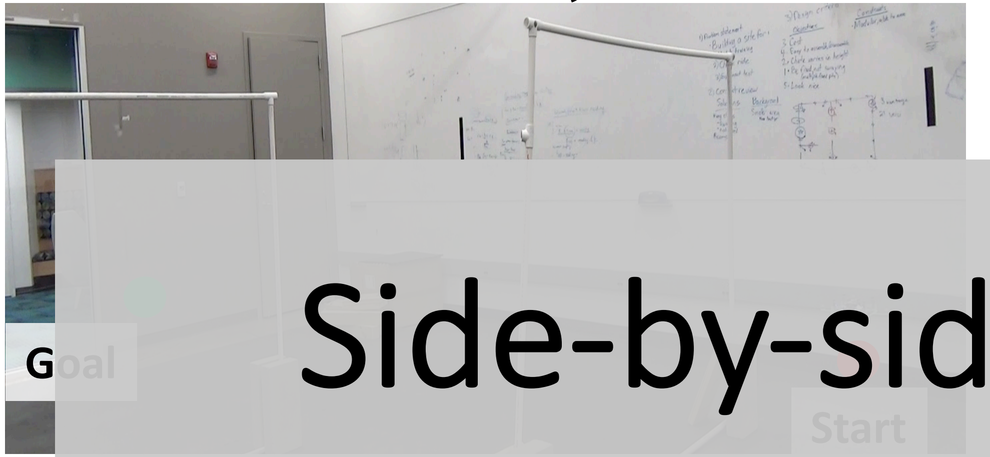
Ours, $T = 4.36$s, Jerk=39



Experiment 2

Our method can control aggressiveness using time penalty w (Plan time ~10 ms)

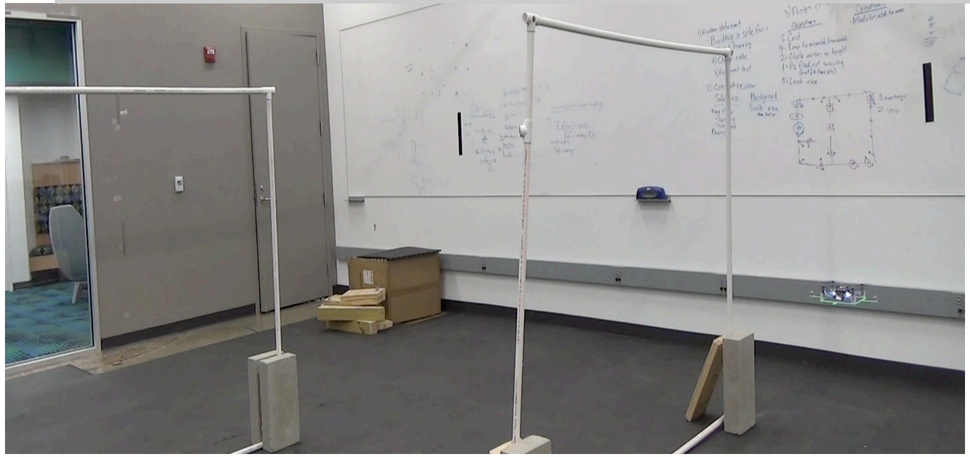
$w = 10, T = 5.60\text{s}, \text{Jerk} = 11.2$



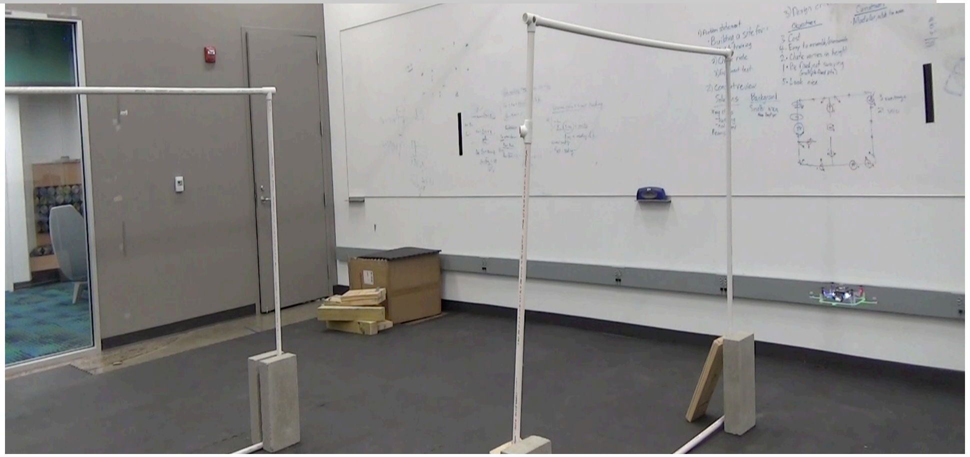
$w = 20, T = 4.96\text{s}, \text{Jerk} = 19.9$

Side-by-side Comparison

$w = 40, T = 4.42\text{s}, \text{Jerk} = 36.1$



$w = 80, T = 4.01\text{s}, \text{Jerk} = 64.7$



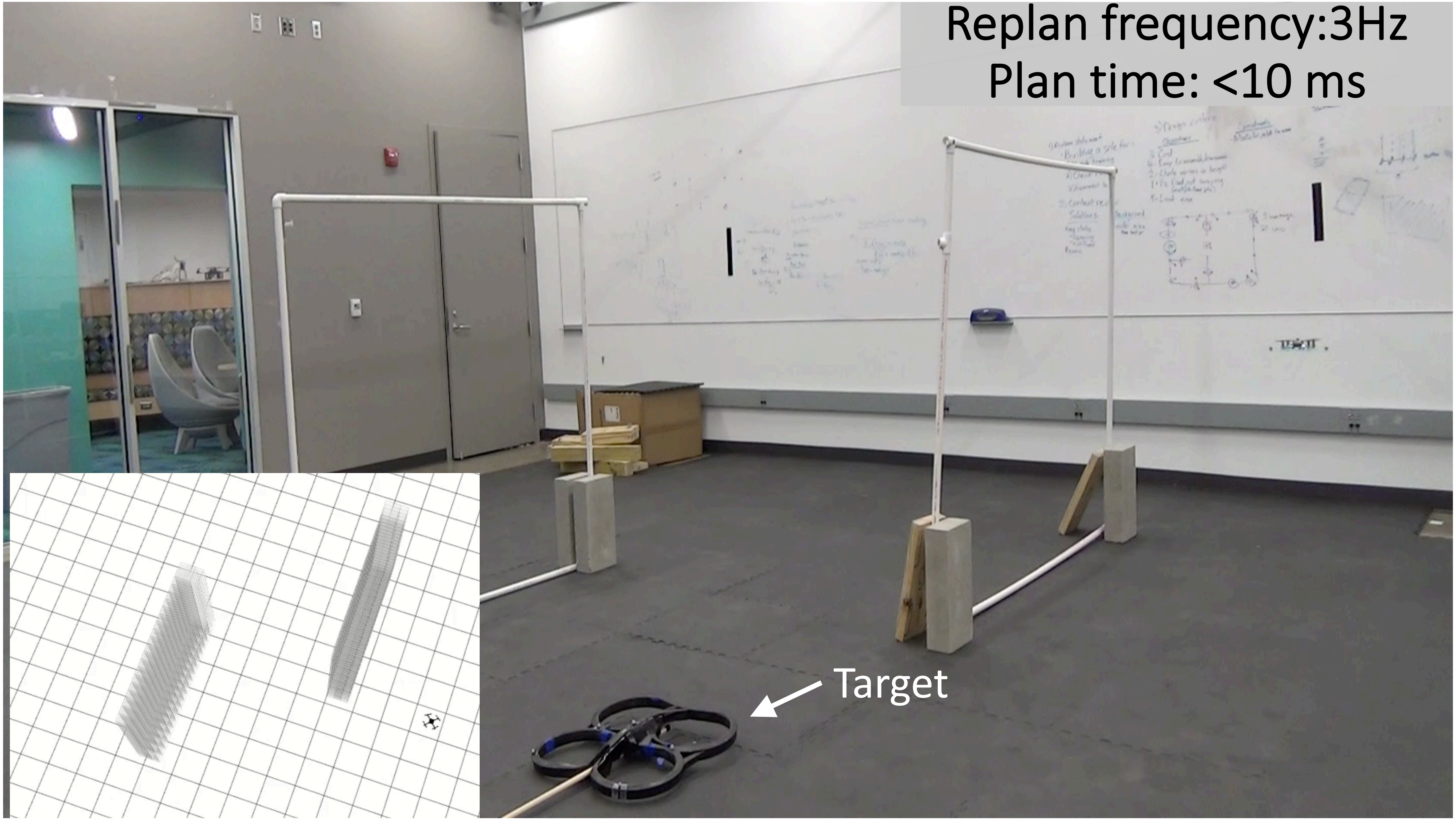
Experiment 3
Tracking a dynamic goal

Target

Moved by human

Quadrotor

Goal is 0.5m above the target


Replan frequency:3Hz
Plan time: <10 ms
Target

Thank you