Simultaneous System Identification and Model Predictive Control with No Dynamic Regret

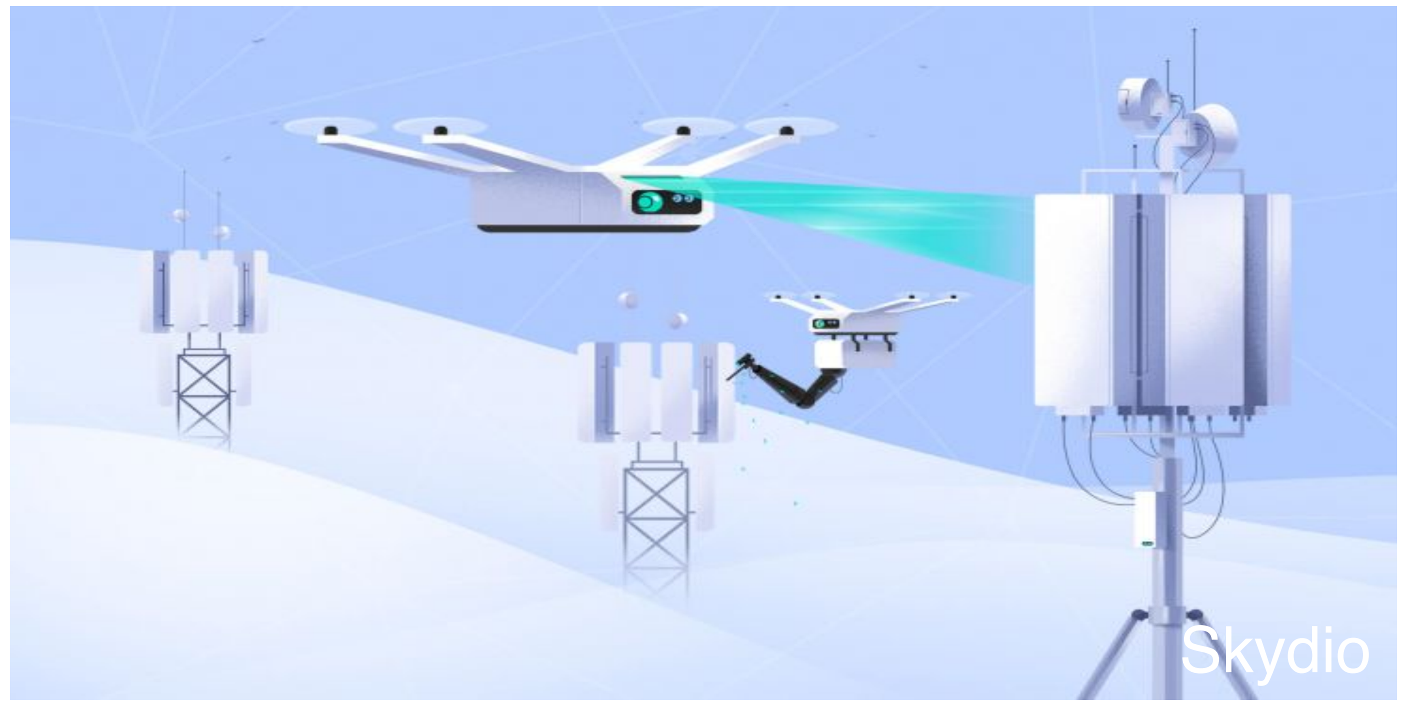
Hongyu Zhou, Vasileios Tzoumas

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Motion Control Tasks that Require Accuracy and Agility

Drone Delivery

Inspection & Maintenance



Target Tracking

Goal: Generate control inputs to achieve agile and accurate motion control.[1,2,3]

Challenges: Dynamics and/or disturbances that are unknown, difficult-to-model, adaptive:

I. *Drone delivery*: Packages with unknown weights.

II. *Inspection and maintenance*: Wind, drag, ground effects.

III. *Target tracking*: Targets with unknown dynamics.

---

[1] Ackerman, IEEE Spectrum '13

[2] Chen, Liu, Shen, IROS '16

[3] Seneviratne, Dammika, et al., Acta Imeko '18

Model Predictive Control Under Uncertainty

All above scenarios are control problems under uncertainty

Goal: Find control input to minimize a look-ahead cumulative loss:

$$\begin{aligned}
\min_{u_t,\ \ldots,\ u_{t+N-1}} \quad & \sum_{k=t}^{t+N-1} c_k\left(x_k, u_k\right) \\
\text{subject to} \quad & x_{k+1} = f\left(x_k\right) + g\left(x_k\right) u_k + h\left(z_k\right), \\
& u_k \in \mathcal{U}, \\
& k \in \{t, \ldots, t+N-1\}.
\end{aligned}$$

convex

known system dynamics

unknown uncertainty

control constraints

Model Predictive Control Under Uncertainty

All above scenarios are control problems under uncertainty

Goal: Find control input to minimize a look-ahead cumulative loss:

$$\begin{aligned} \min_{u_t,\ \ldots,\ u_{t+N-1}} \quad & \sum_{k=t}^{t+N-1} c_k\left(x_k, u_k\right) \\ \text{subject to} \quad & x_{k+1} = f\left(x_k\right) + g\left(x_k\right) u_k + h\left(z_k\right), \\ & u_k \in \mathcal{U}, \\ & k \in \{t, \ldots, t+N-1\}. \end{aligned}$$

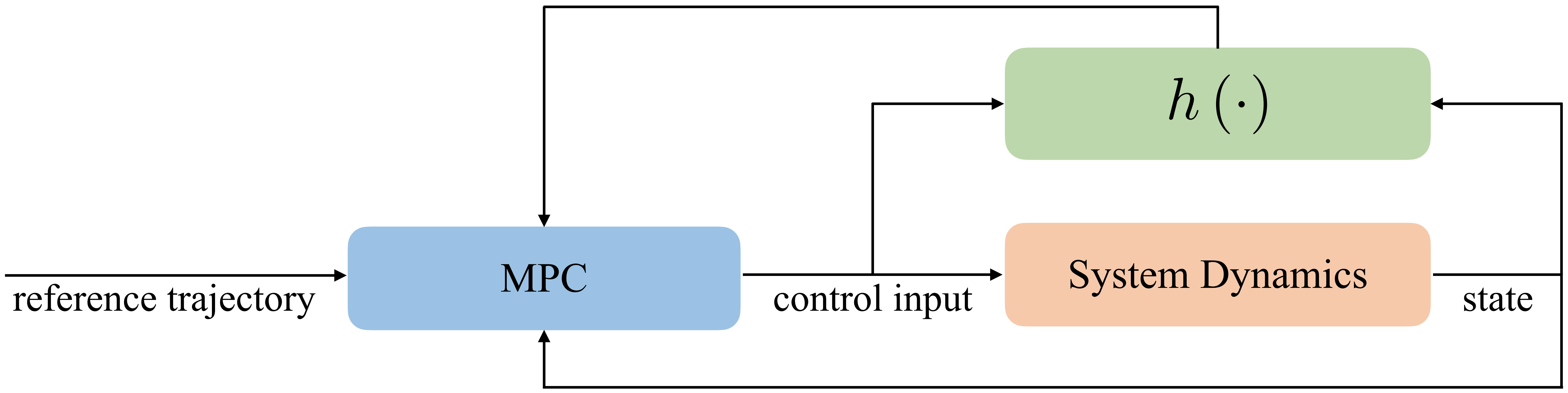
convex

known system dynamics

unknown uncertainty

control constraints

Ideally: if $h(\cdot)$ is known:



Model Predictive Control Under Uncertainty

All above scenarios are control problems under uncertainty

Goal: Find control input to minimize a look-ahead cumulative loss:

$$
\begin{aligned}
\min_{u_t,\ \ldots,\ u_{t+N-1}} \quad & \sum_{k=t}^{t+N-1} c_k\left(x_k, u_k\right) \\
\textbf{subject to} \quad & x_{k+1} = f\left(x_k\right) + g\left(x_k\right) u_k + \hat{h}\left(z_k\right), \\
& u_k \in \mathcal{U}, \\
& k \in \{t, \ldots, t+N-1\}.
\end{aligned}
$$

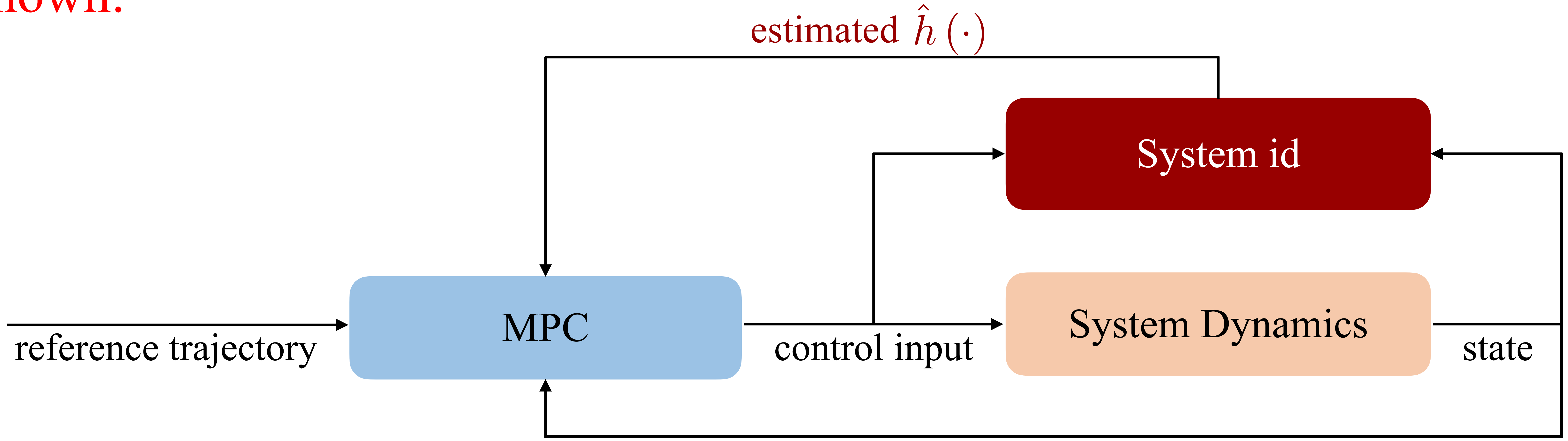
convex

known system dynamics

estimated uncertainty

control constraints

But $h(\cdot)$ is unknown:



System Identification given a Function Basis

Assume: A function basis $\{\Phi(z_t, \theta_1), \ldots, \Phi(z_t, \theta_M)\}$ such that:

$$\hat{h}(z_t; \alpha) \triangleq \frac{1}{M} \sum_{i=1}^{M} \Phi(z_t, \theta_i)\, \alpha_i$$

Goal: Find $\alpha_i, \ldots, \alpha_M$ online

Examples:

- **Reproducing Kernels in Hilbert Spaces**: Universal approximation theorem:[4]

  For appropriately chosen $\alpha^\star$ and Φ, and for $\theta_1, \ldots, \theta_M$ sampled from appropriate distribution ν, then with high probability:

  $$\|h(\cdot) - \hat{h}(\cdot; \alpha^\star)\|_\infty = \mathcal{O}\left(1/\sqrt{M}\right).$$

  [TRO '25]

- **Neural Networks:** Similarly to above but where:

  $\theta_1, \ldots, \theta_M$ the trained parameters
  Φ the trained neural network model as basis functions

- **Koopman Observables:** $\Phi(h(z_t)) \triangleq A\Phi(h(z_{t-1})) + B\Psi(h(z_{t-1}), z_t)$
  $\Phi(\cdot)$ and $\Psi(\cdot, \cdot)$ given Koopman observable functions
  A and B to be learned online

  [ACC '25]

[4] Boffi et al., JMLR '22

System Identification given a Function Basis

Assume: A function basis $\{\Phi(z_t, \theta_1), \ldots, \Phi(z_t, \theta_M)\}$ such that:

$$\hat{h}(z_t; \alpha) \triangleq \frac{1}{M} \sum_{i=1}^{M} \Phi(z_t, \theta_i)\, \alpha_i$$

Goal: Find $\alpha_i, \ldots, \alpha_M$ online

Examples:

- **Reproducing Kernels in Hilbert Spaces**: Universal approximation theorem:[4]

  For appropriately chosen $\alpha^\star$ and Φ, and for $\theta_1, \ldots, \theta_M$ sampled from appropriate distribution ν, then with high probability:

  $$\|h(\cdot) - \hat{h}(\cdot; \alpha^\star)\|_\infty = \mathcal{O}\left(1/\sqrt{M}\right).$$

  [TRO '25]

- **Neural Networks**: Similarly to above but where:

  $\theta_1, \ldots, \theta_M$ the trained parameters
  Φ the trained neural network model as basis functions

- **Koopman Observables:** $\Phi(h(z_t)) \triangleq A\Phi(h(z_{t-1})) + B\Psi(h(z_{t-1}), z_t)$
  $\Phi(\cdot)$ and $\Psi(\cdot, \cdot)$ given Koopman observable functions
  A and B to be learned online

  [ACC '25]

[4] Boffi et al., JMLR '22

Simultaneous System Identification and Model Predictive Control

Problem

At each $t = 1, \ldots, T$,

- estimate the unknown disturbance $\hat{h}(\cdot)$; → i.e., update estimate of α
- identify a control input u_t using MPC.

The goal is to minimize dynamic regret:

$$\sum_{t=1}^{T} c_t\left(x_t, u_t, h(z_t)\right) - \sum_{t=1}^{T} c_t\left(x_t^\star, u_t^\star, h(z_t^\star)\right).$$

Suboptimality Metric against Optimal Control Policies in Hindsight

Definition (Dynamic Regret)

Assume a total time horizon of operation T, and loss functions c_t, $t = 1, \ldots, T$. Then, *dynamic regret* is

$$\text{Regret}_T^D = \sum_{t=1}^{T} c_t\left(x_t, u_t, h(z_t)\right) - \sum_{t=1}^{T} c_t\left(x_t^\star, u_t^\star, h(z_t^\star)\right),$$

where $x_t^\star$ and $u_t^\star$ are the optimal trajectory and control input in hindsight, and the cost c_t depends on the unknown disturbance h explicit.

Remark:

- The regret is sublinear if $\lim_{T\to\infty} \frac{\text{Regret}_T^D}{T} \to 0$, which implies $c_t\left(x_t, u_t, h(z_t)\right) - c_t\left(x_t^\star, u_t^\star, h(z_t^\star)\right) \to 0$.
- h adapts (possibly differently) to the state and control sequences (x_1, u_1), $\ldots$, (x_T, u_T) and $(x_1^\star, u_1^\star)$, $\ldots$, $(x_T^\star, u_T^\star)$ since h is a function of the state and the control input.

Offline Learning for Control[5]

- collects data offline and trains neural-networks or Gaussian-process models BUT
  - data-collection can be expensive and time-consuming
  - may not generalize to unseen environments

Robust Control[6]

- select control input over a look-ahead horizon BUT
  - conservative since assuming worst-case disturbances

[5] Sánchez-Sánchez et al., '18; Carron et al., RAL '19; Torrente et al., RAL '21; Shi et al., ICRA '19; O'Connell, et al., SR '22; …
[6] Goel et al., '20; Sabag et al., ACC '21; Goel et al., L4DC '21; Martin et al., L4DC '22; Didier et al., L-CSS '22; Zhou et al., CDC '23; …

State of the Art

Adaptive Control[7]

- estimates uncertainty and compensates control input with the estimated uncertainty BUT
  - do not learn a model of uncertainty for predictive control

Non-Stochastic Control[8]

- updates control input online to adapt to observed uncertainty BUT
  - sensitive to tuning parameters
  - do not learn a model of uncertainty for predictive control

[7] Slotine, '91; Krstic, et al., '95; Ioannou et al., '96; Tal et al., TCST '20; Wu, et al., '23; Das et al., '24; Jia, et al., TRO '23; …

[8] Agarwal et al., ICML '19; Hazan et al., ALT '20; Gradu et al., L4DC '23; Zhou et al., CDC '23; Zhou et al., RAL '23; ...

Algorithm: Simultaneous Sys-ID and MPC

Initialization:

- Gradient descent learning rate η, number of random Fourier features M, domain set $\mathcal{D}$, estimated parameter $\hat{\alpha}_{i,1} \in \mathcal{D}$;
- Randomly sample $\theta_i \sim \nu$ and formulate $\Phi(\cdot, \theta_i)$, where $i \in \{1, \ldots, M\}$;

At each iteration $t = 1, \ldots, T$:

1. Apply control input u_t using MPC with $\hat{h}(\cdot) \triangleq \frac{1}{M} \sum_{i=1}^{M} \Phi(\cdot, \theta_i)\, \hat{\alpha}_{i,t}$;
2. Observe state x_{t+1}, and calculate disturbance via $h(z_t) = x_{t+1} - f(x_t) - g(x_t)u_t$;
3. Formulate estimation loss $l_t(\hat{\alpha}_t) \triangleq \| h(z_t) - \frac{1}{M} \sum_{i=1}^{M} \Phi(z_t, \theta_i)\, \hat{\alpha}_{i,t} \|^2$;
4. Calculate gradient $\nabla_t \triangleq \nabla_{\hat{\alpha}_t} l_t(\hat{\alpha}_t)$;
5. Update $\hat{\alpha}'_{t+1} = \hat{\alpha}_t - \eta \nabla_t$;
6. Project $\hat{\alpha}'_{i,t+1}$ onto $\mathcal{D}$, *i.e.*, $\hat{\alpha}_{i,t+1} = \Pi_{\mathcal{D}}(\hat{\alpha}'_{i,t+1})$, for $i \in \{1, \ldots, M\}$.

No Dynamic Regret

Theorem [TRO '25]

Our algorithm with $\eta = \mathcal{O}\left(1/\sqrt{T}\right)$ achieves $\text{Regret}_T^D \leq \mathcal{O}\left(T^{\frac{3}{4}}\right)$.

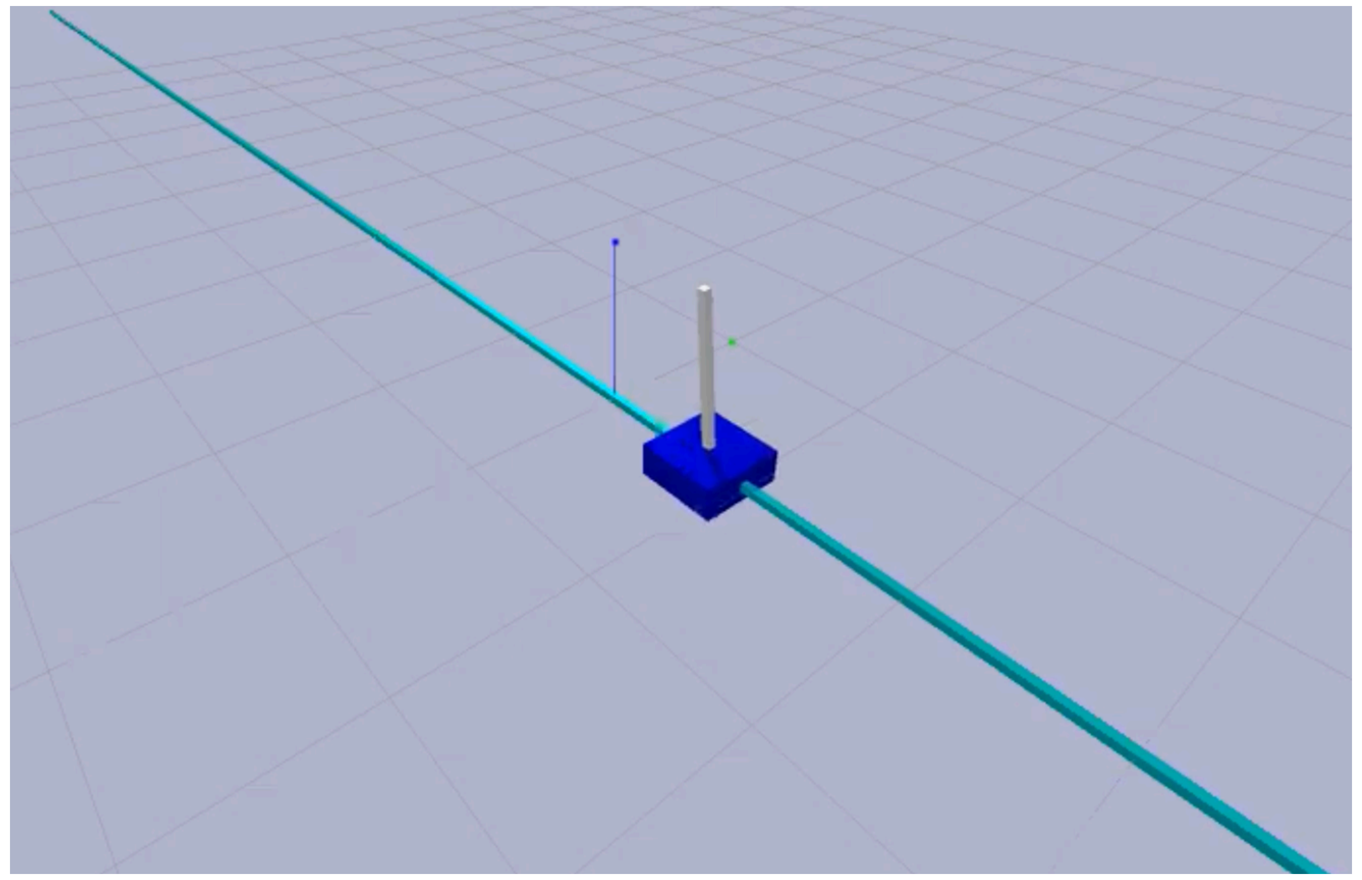
Remark:

- Our algorithm converges asymptotically to the optimial controller since $\lim_{T\to\infty} \frac{\text{Regret}_T^D}{T} \to 0$.

Technical Assumptions:

- Lipschitzness of $c_t(\cdot,\cdot)$ and $\hat{h}(\cdot)$.
- Stability of MPC for the estimated system.

Simulations on Cart-Pole Stabilization with Inaccurate Model



Goal:

- Stabilize the cart-pole around the upright position of the pole

Setup:

- The model parameters (pole length & mass, cart mass) are inaccurate

Simulations on Cart-Pole Stabilization with Inaccurate Model

Goal:

- Stabilize the cart-pole around the upright position of the pole

Setup:

- The model parameters (pole length & mass, cart mass) are inaccurate

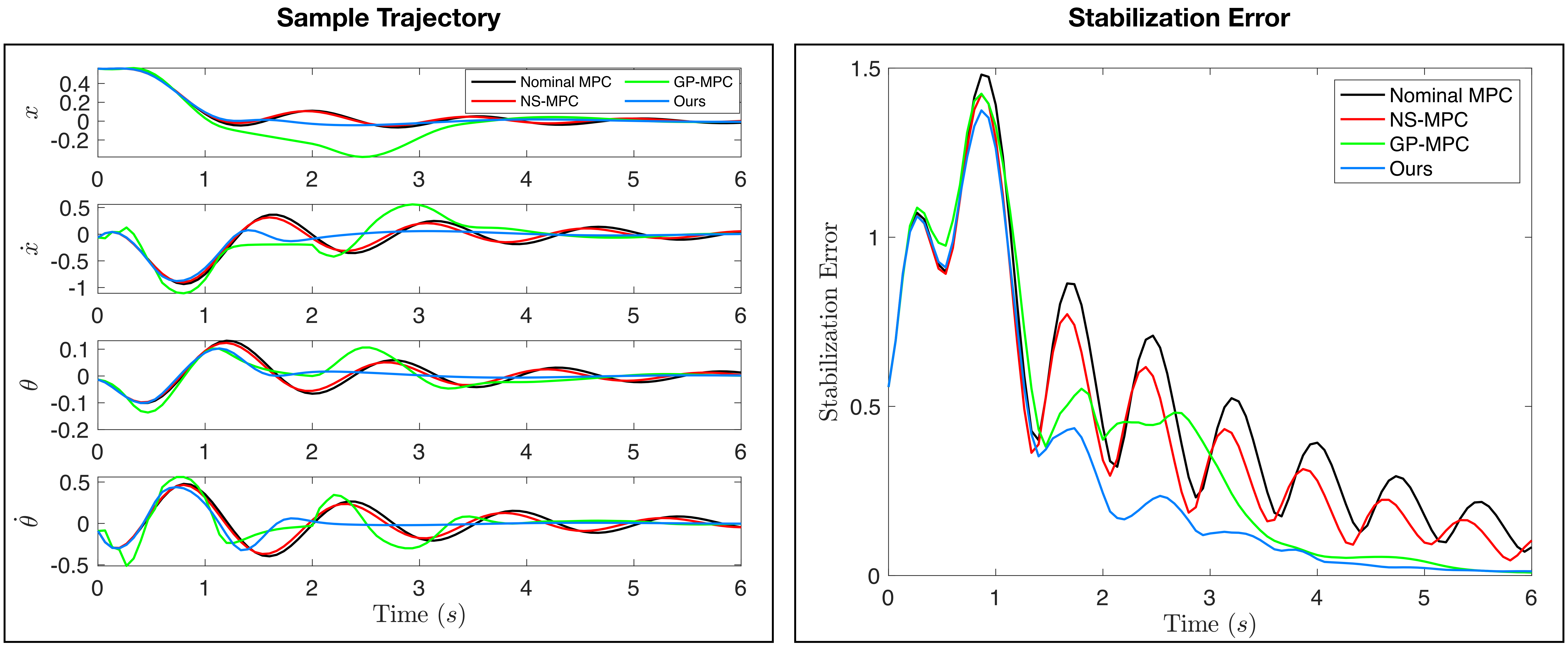
Compared algorithms: (i) Non-stochastic MPC[10] and (ii) Gaussian Process MPC[11]

Ours

NS-MPC

GP-MPC

[10] Zhou et al., RAL '23 [11] Hewing et al., TCST '19



Results:

- Our method achieves fastest stabilization while other algorithms:
  - NS-MPC has marginal improvement over Nominal MPC
  - GP-MPC has larger deviation than our method

Simulations on Trajectory Tracking with Unknown Aerodynamic Effect

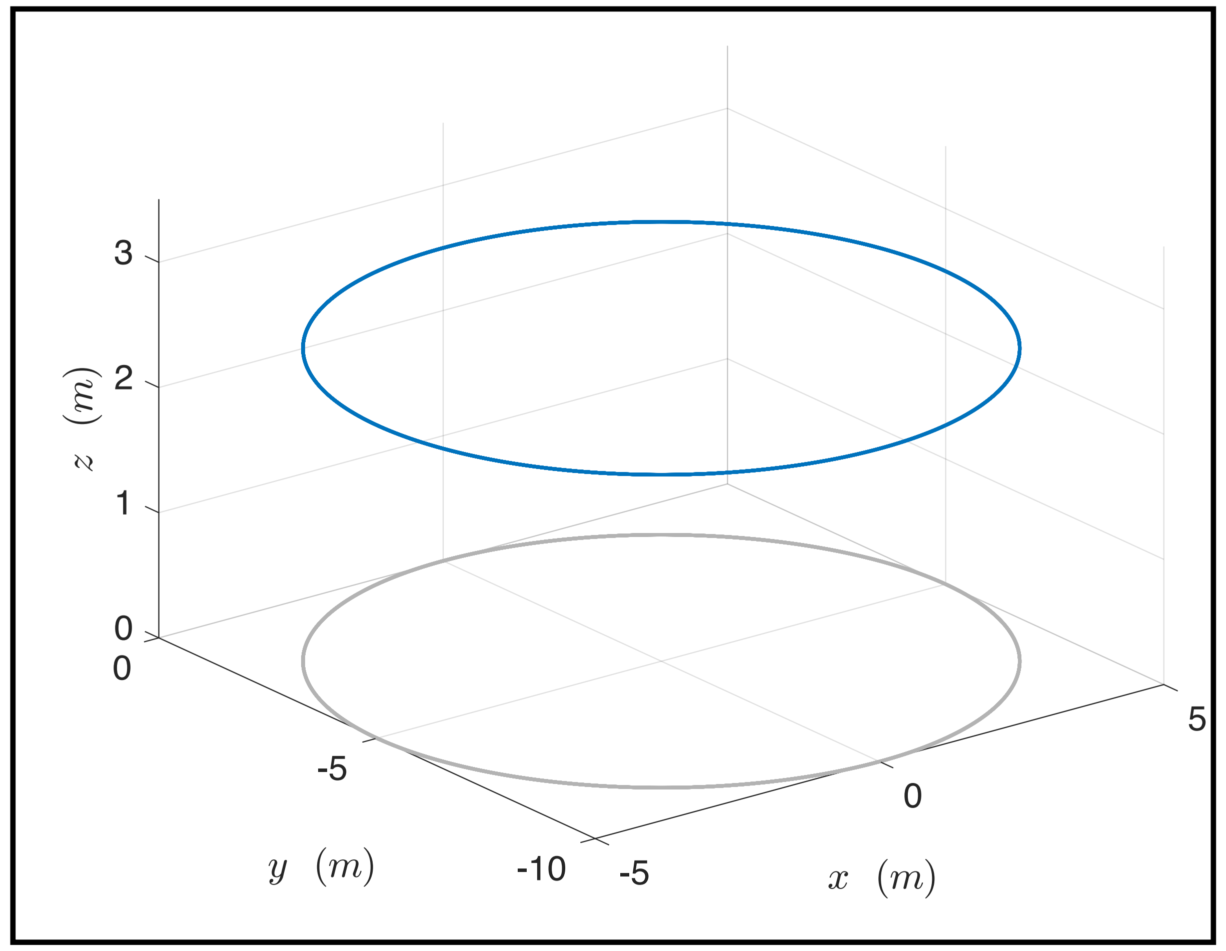
Goal:

- Track reference trajectories with a drone

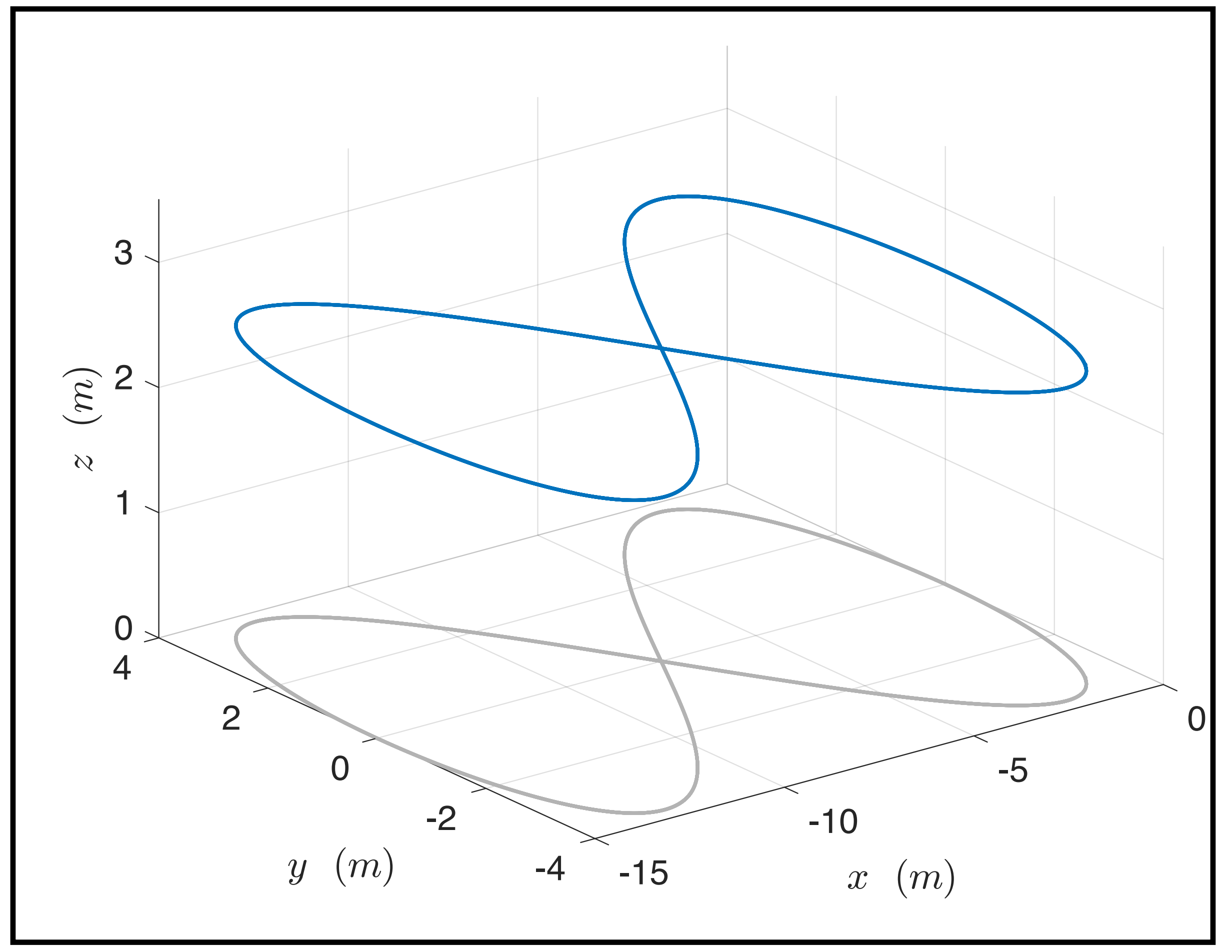
Setup:

- The system dynamics of the drone are corrupted with unknown drag effects

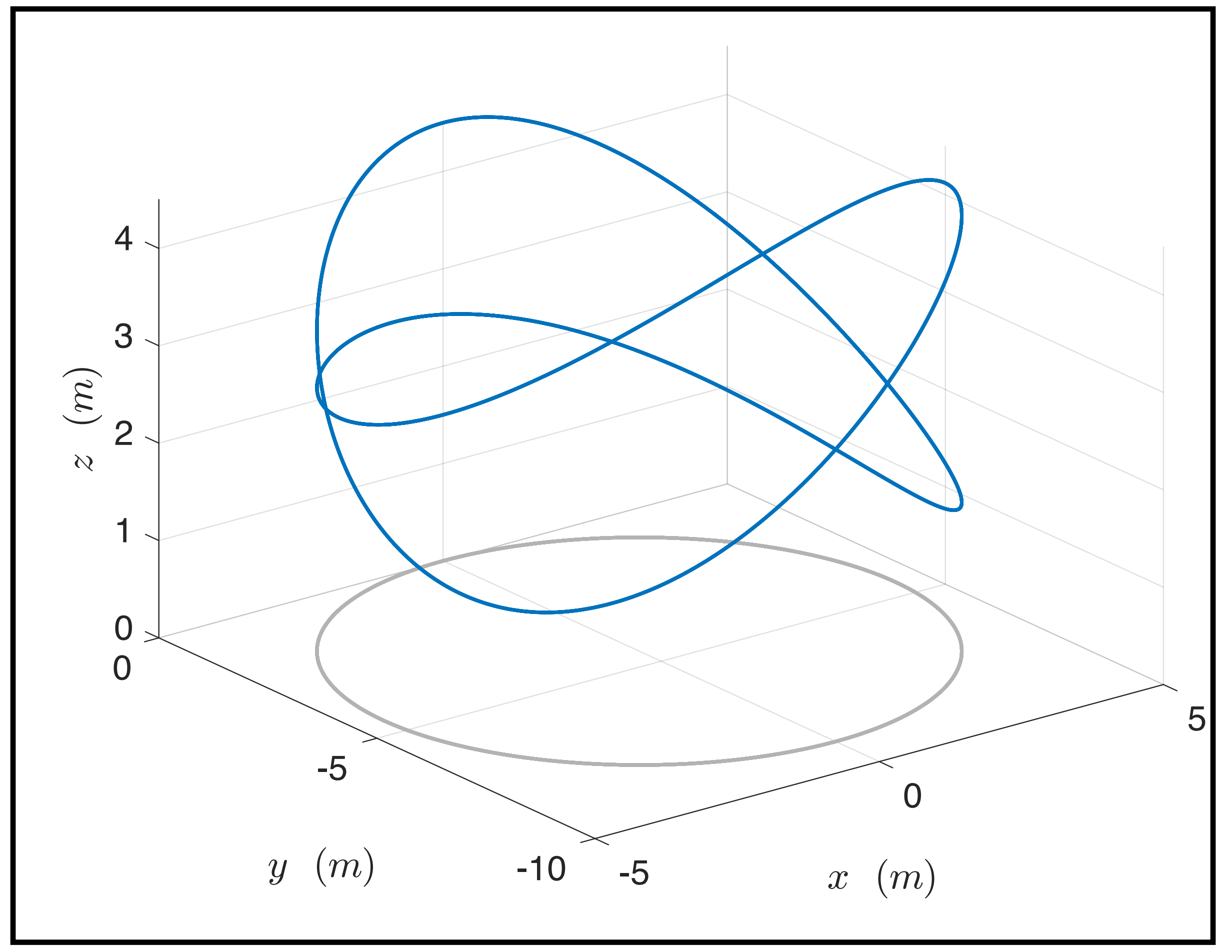
Circle



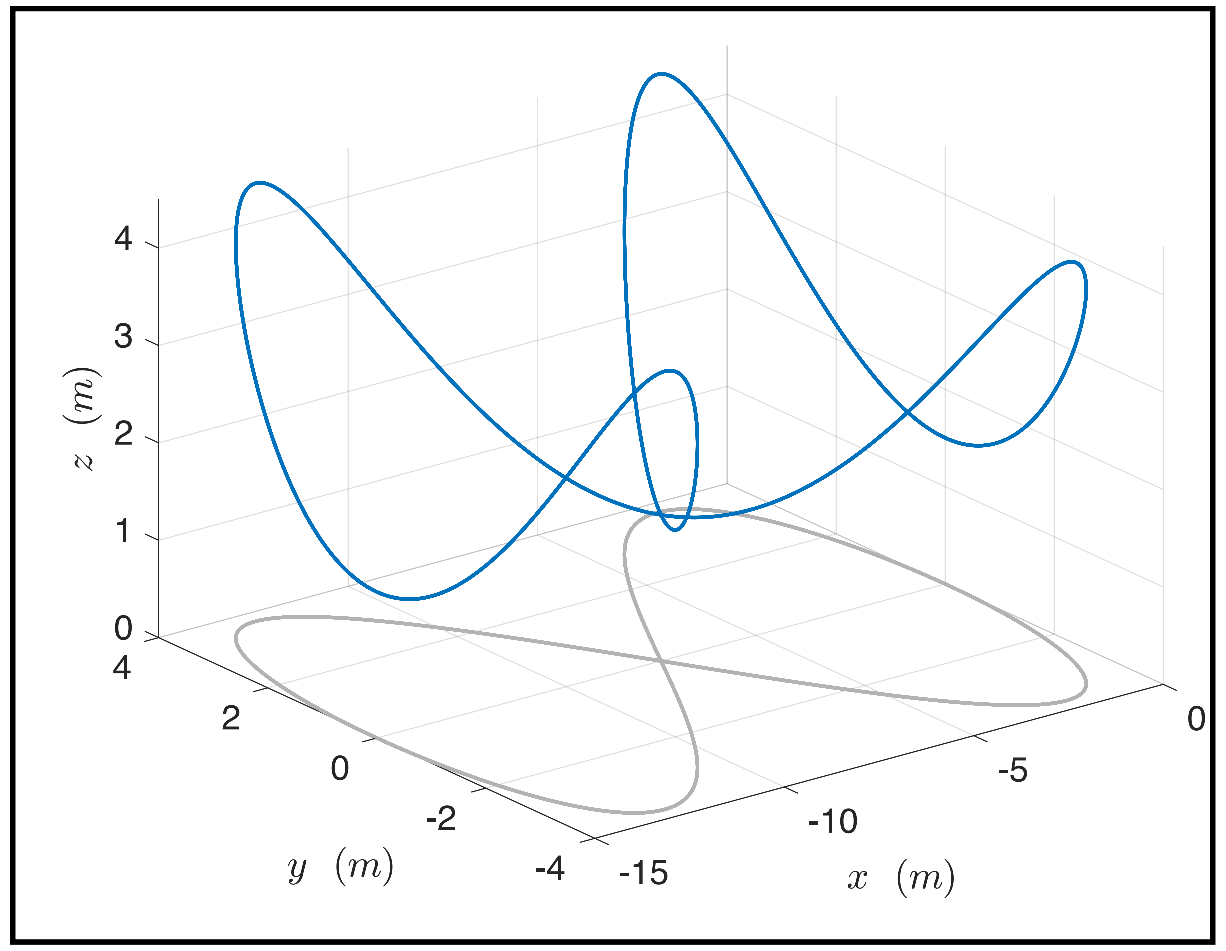
Lemniscate



Wrapped Circle



Wrapped Lemniscate



Simulations on Trajectory Tracking with Unknown Aerodynamic Effect

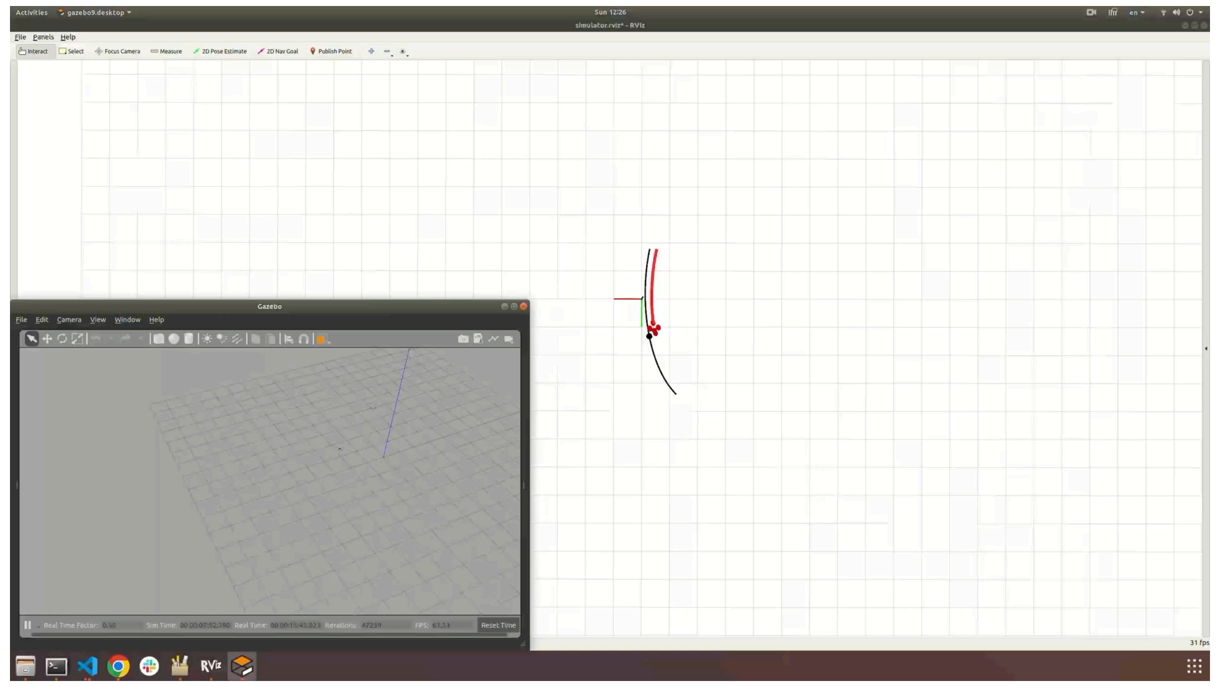
Goal:

- Track reference trajectories with a drone

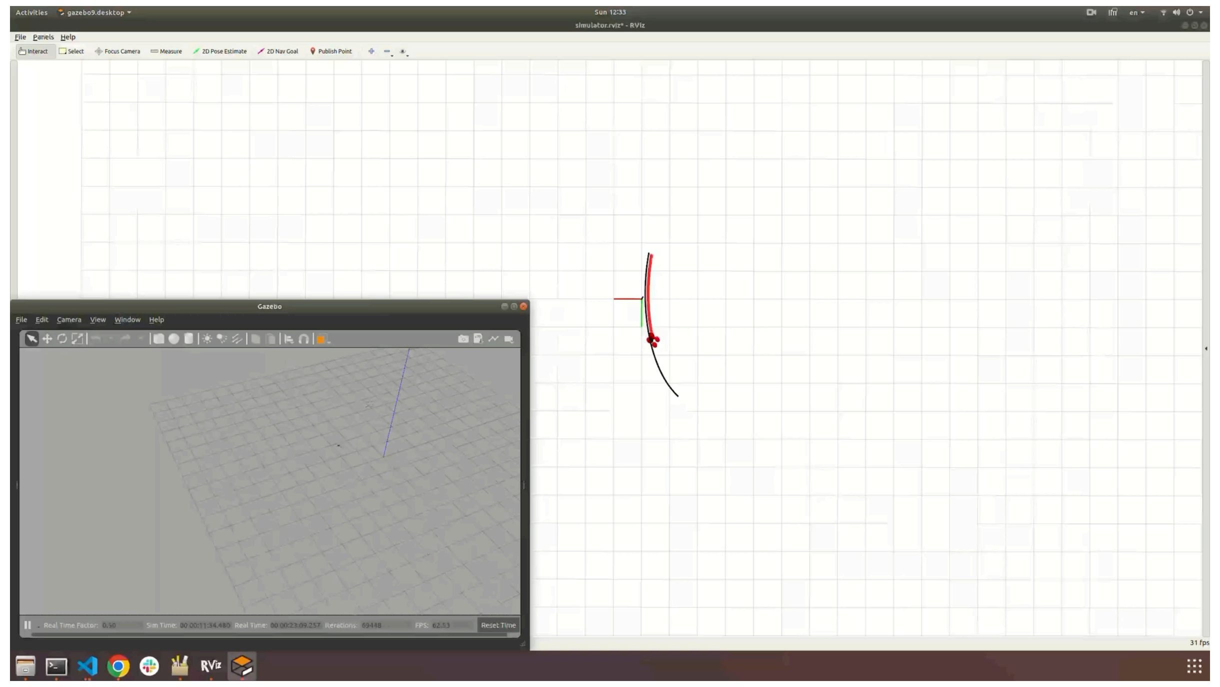
Setup:

- The system dynamics of the drone are corrupted with unknown drag effects

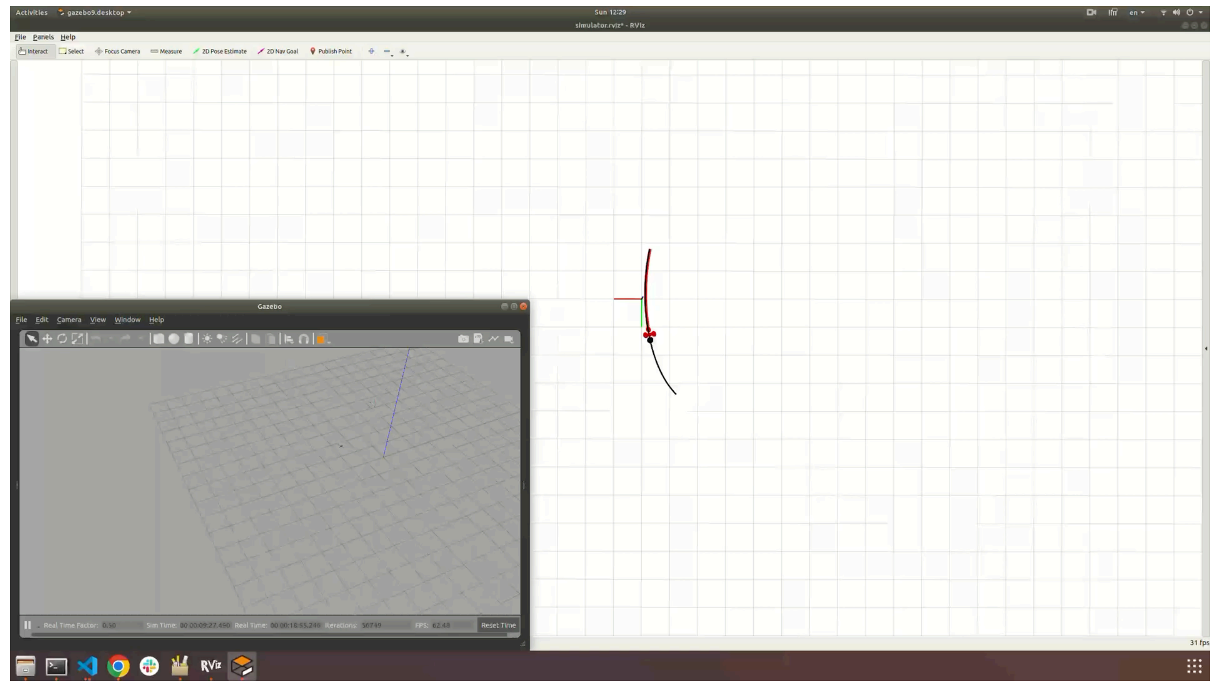
Compared algorithms: (i) Nominal MPC, (ii) Gaussian Process MPC[12], and (iii) Ours w/ INDI inner loop[13]



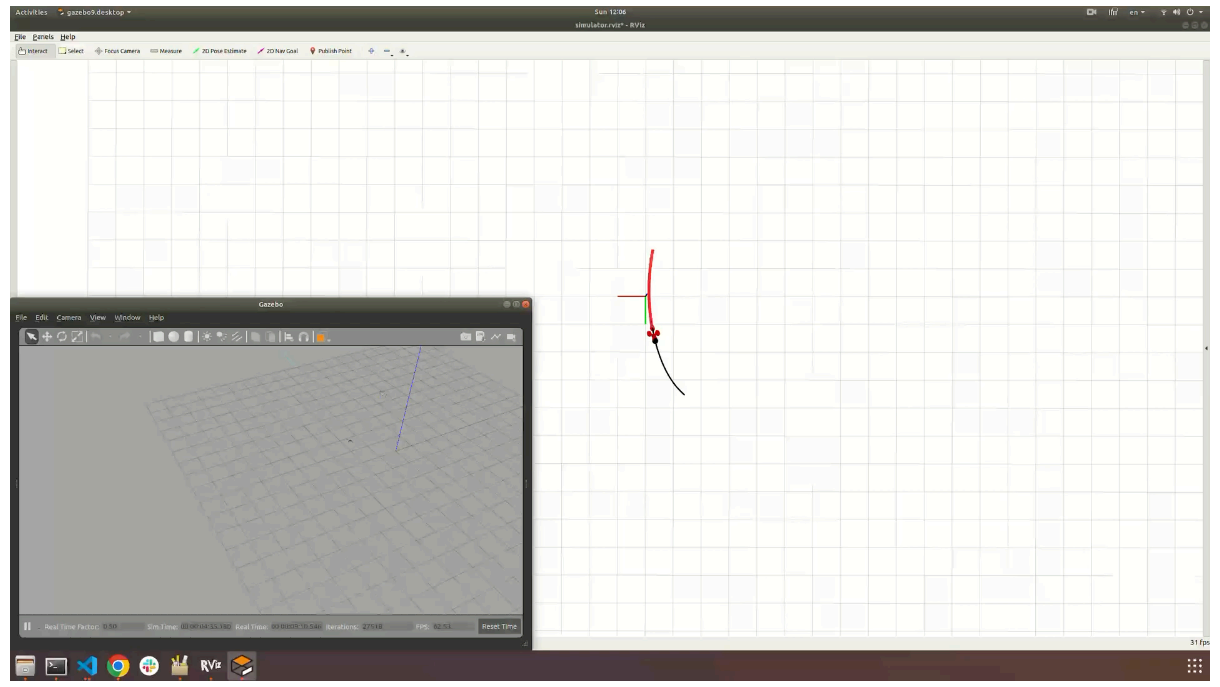
Nominal MPC



GP-MPC



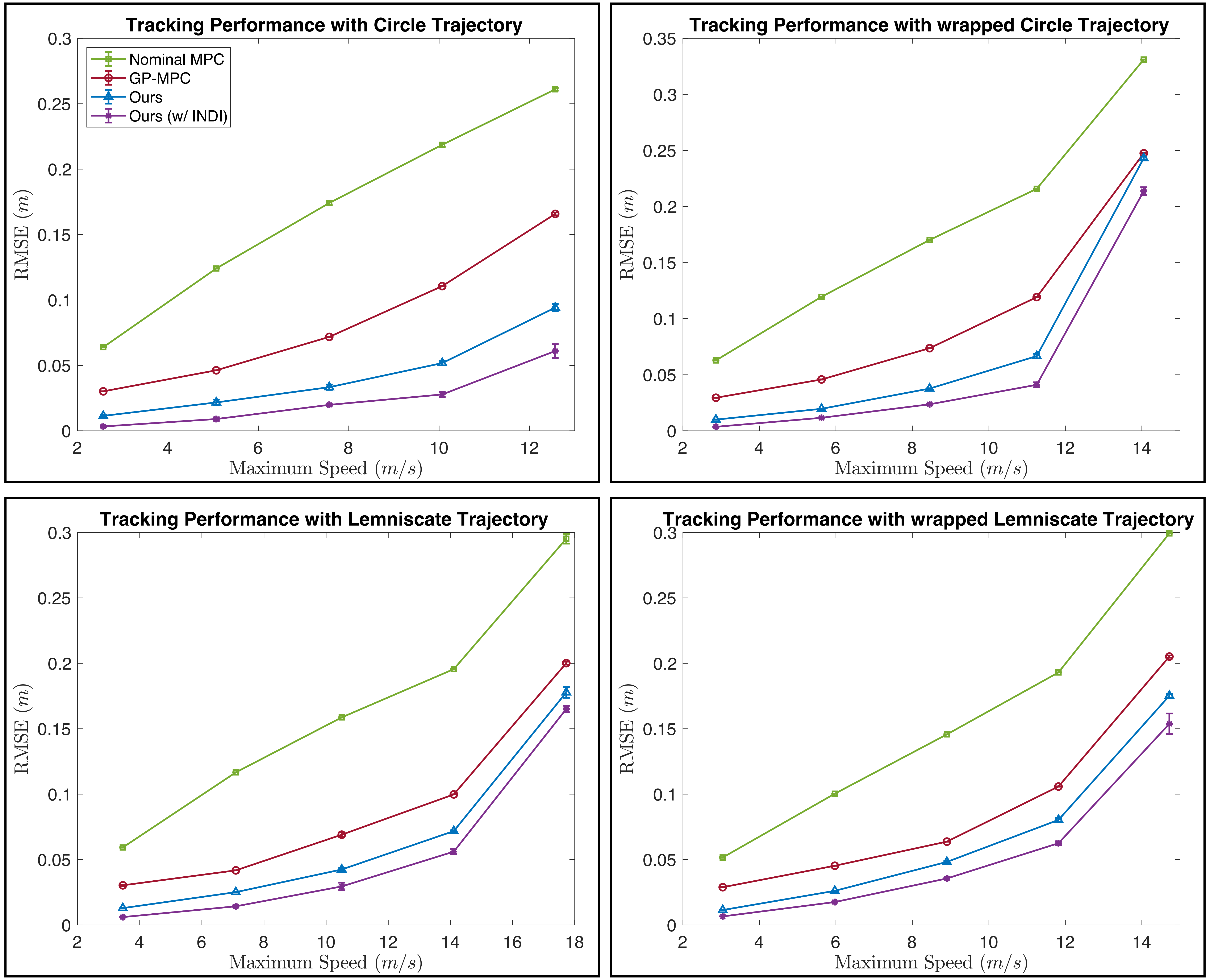
Ours



Ours w/ INDI

[12] Z Torrente et al., RAL '21 [13] Tal et al., TCST '20

Our Algorithm Achieves Lowest Tracking Errors



Hardware on Trajectory Tracking with Unknown Aerodynamic Effect

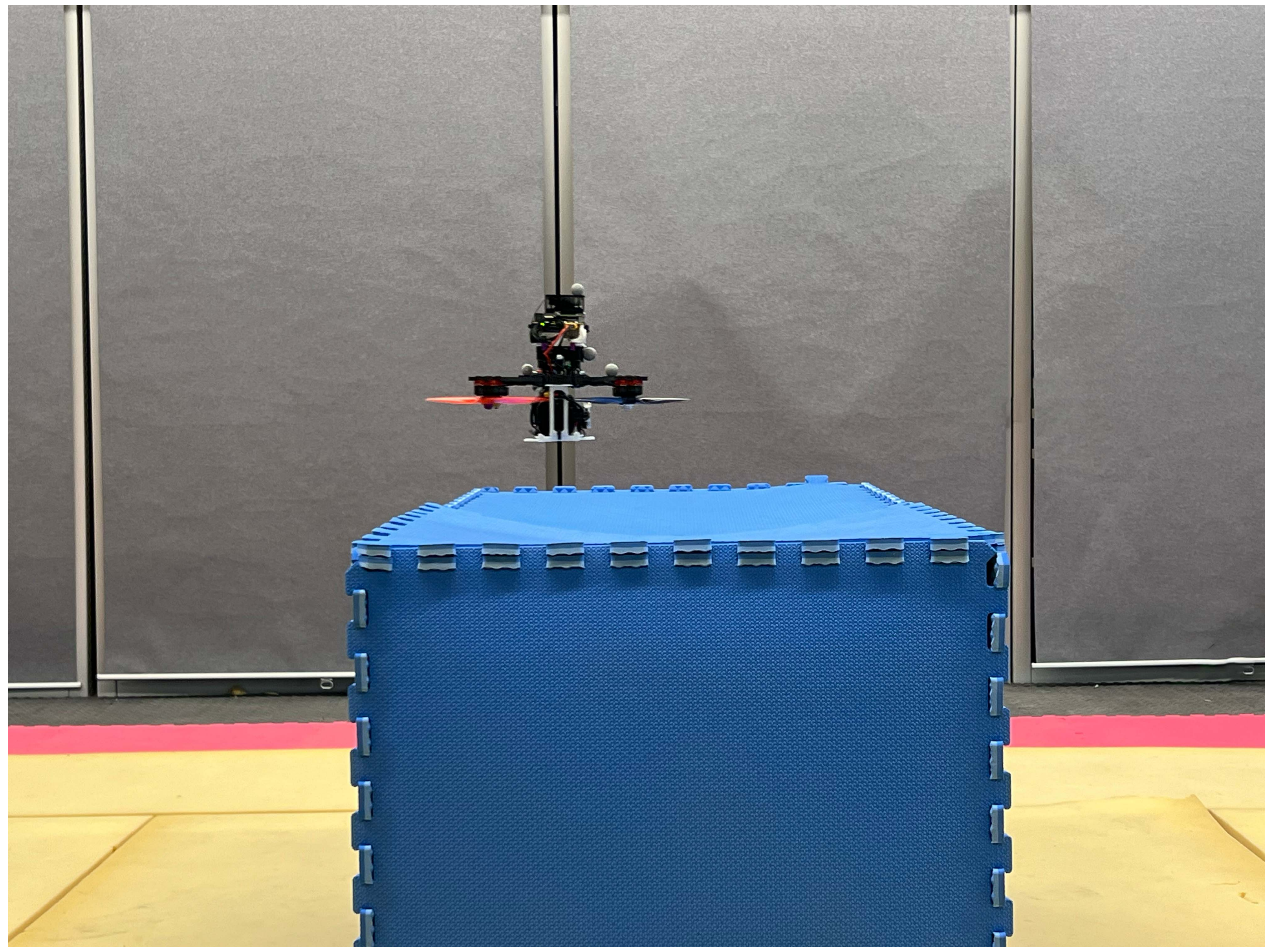
Goal:

- Track a circular trajectory with a drone

Setup:

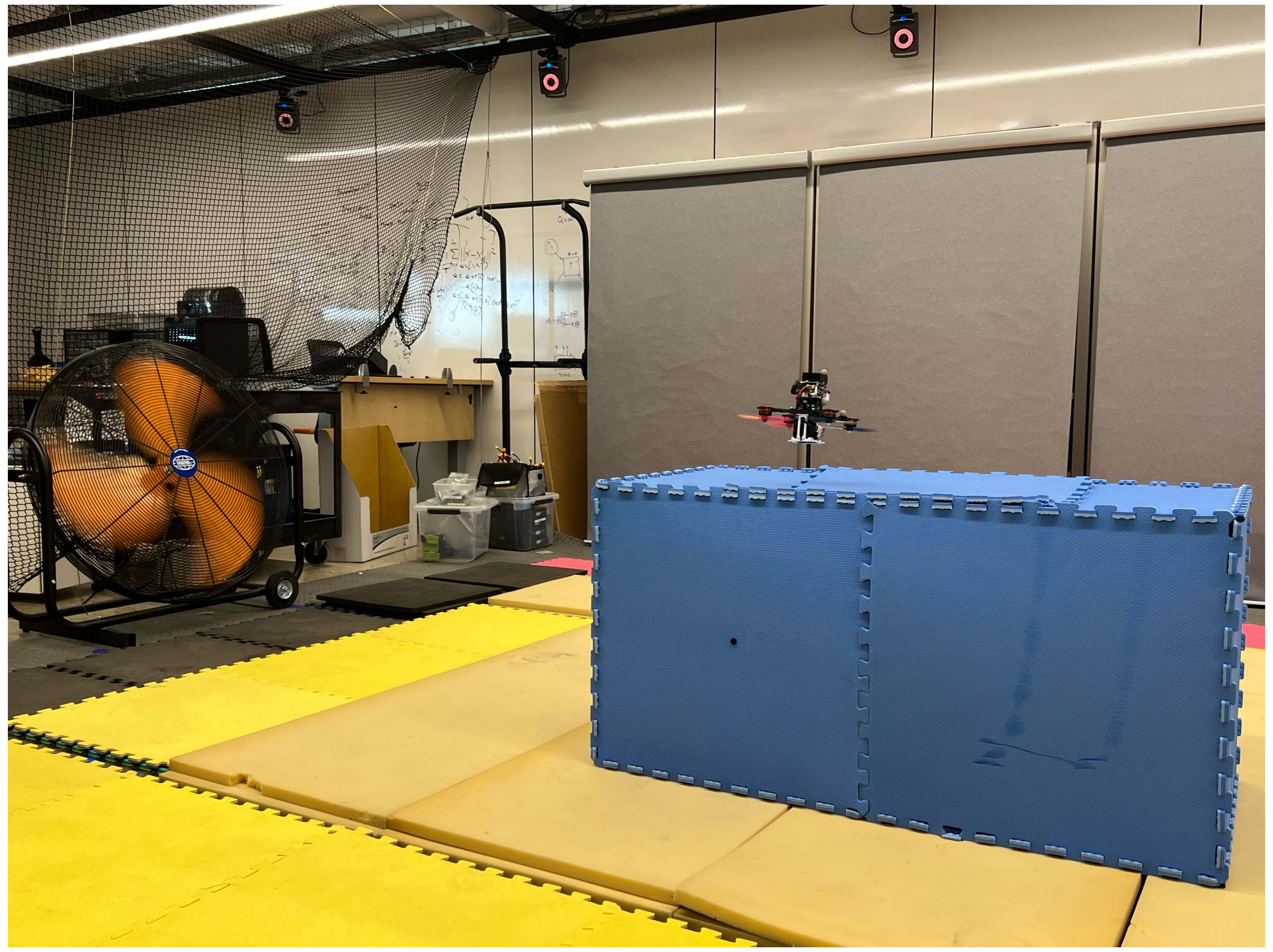
- The circular trajectory is 1*m* in diameter
- The speed is 0.8 *m/s*
- The drone suffers from drag, voltage drop, communication delay, and:

(i) ground effect



(ii) wind disturbances

(iii) ground effect + wind disturbances



Trajectory Tracking with Ground Effect & Wind Disturbances

Hardware on Trajectory Tracking with Unknown Aerodynamic Effect

Goal:

- Track a circular trajectory with a drone

Setup:

- The circular trajectory is 1*m* in diameter
- The speed is 0.8 *m/s*
- The drone suffers from: drag, voltage drop, communication delay, and (i) ground effect, (ii) wind disturbances, and (iii) ground effect + wind disturbances

Compared algorithms: (i) Nominal MPC and (ii) L1 adaptive MPC[14]

---

[14] Wu et al., TCST '25

Our Algorithm Achieves Lowest Tracking Errors

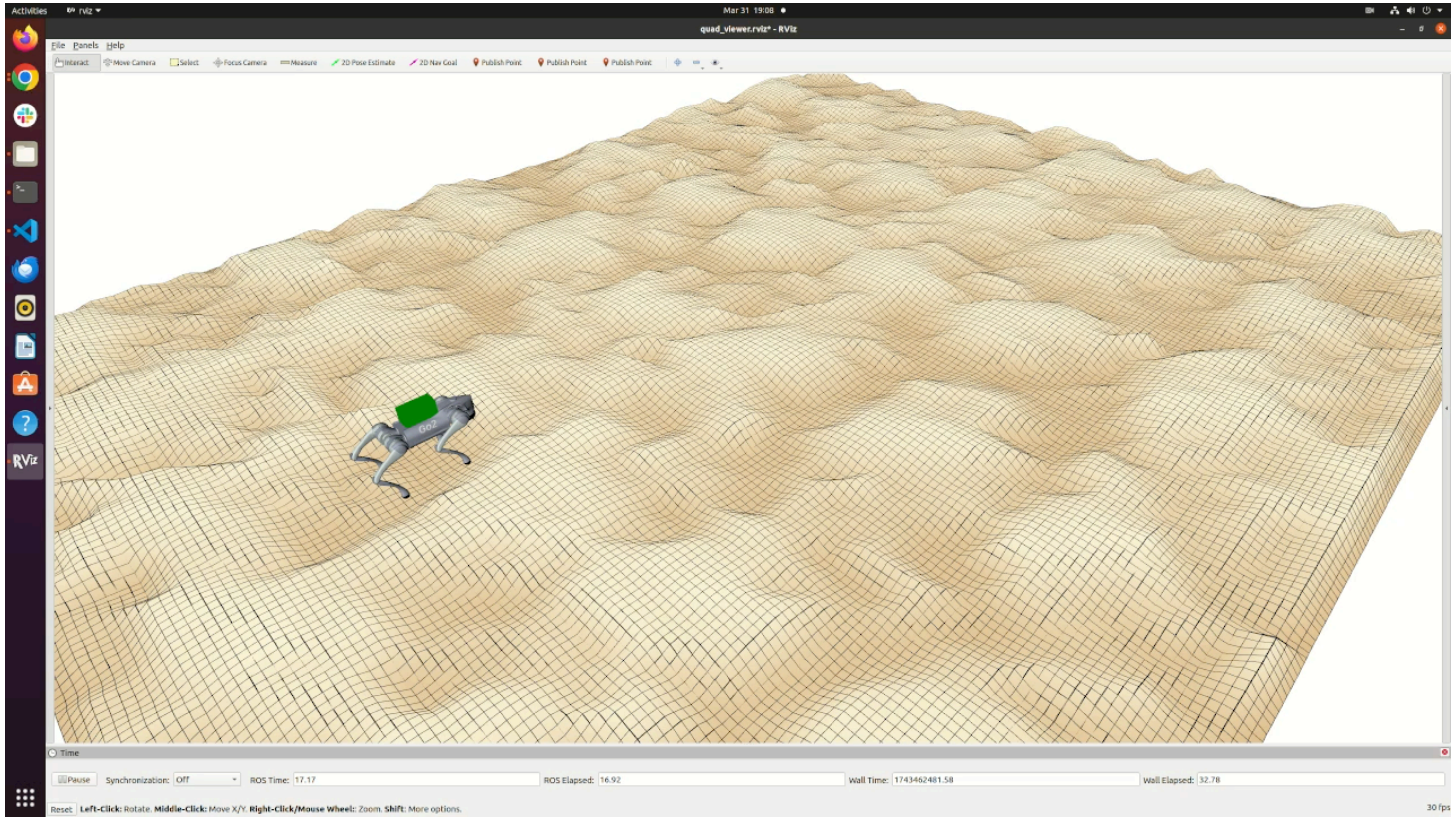
Results:

- Ours achieves lower tracking errors than L1-MPC, due to the benefit of predictive model of unknown disturbances
- Nominal MPC crashes under ground effect & wind disturbances

Summary and Extensions

Online control algorithm for partially unknown control-affine systems with:

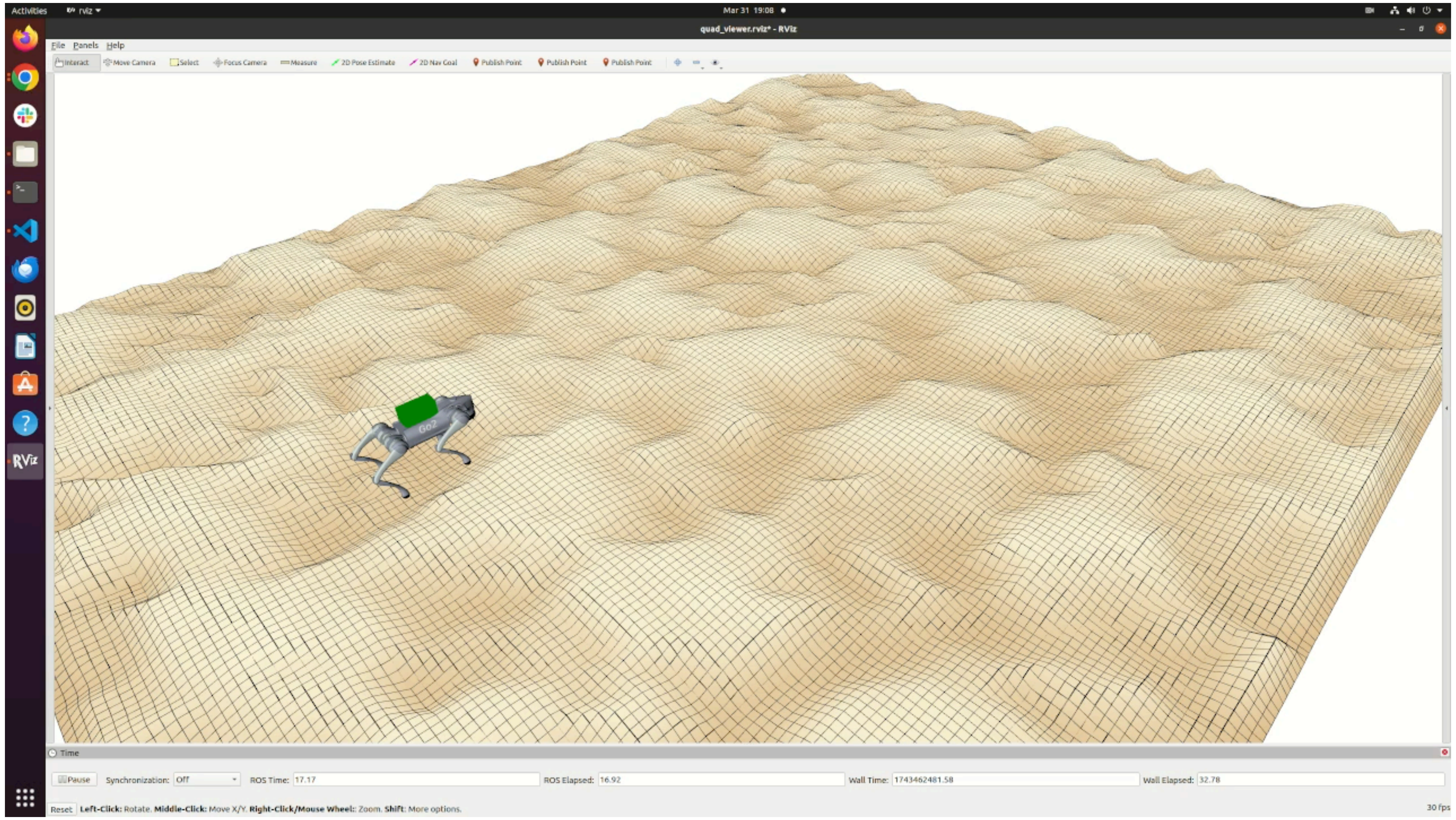
- simultaneous system identification and model predictive control
- no-dynamic-regret performance guarantees



Summary and Extensions

Online control algorithm for partially unknown control-affine systems with:

- simultaneous system identification and model predictive control
- no-dynamic-regret performance guarantees



Extensions:

- Hybrid systems
- Active feature selection