Zero-Shot Sim2Real-Transfer of Neural Controllers

University of Zurich[UZH]

ROBOTICS & PERCEPTION GROUP

Leonard Bauersfeld, Davide Scaramuzza

Motivation

Vision-based drone racing is a challenging task that raises fundamental questions in robotics research:

- fly a given number of laps faster than the opponent
- sequence of gates with known positions
- speeds > 80 km/h

Goal

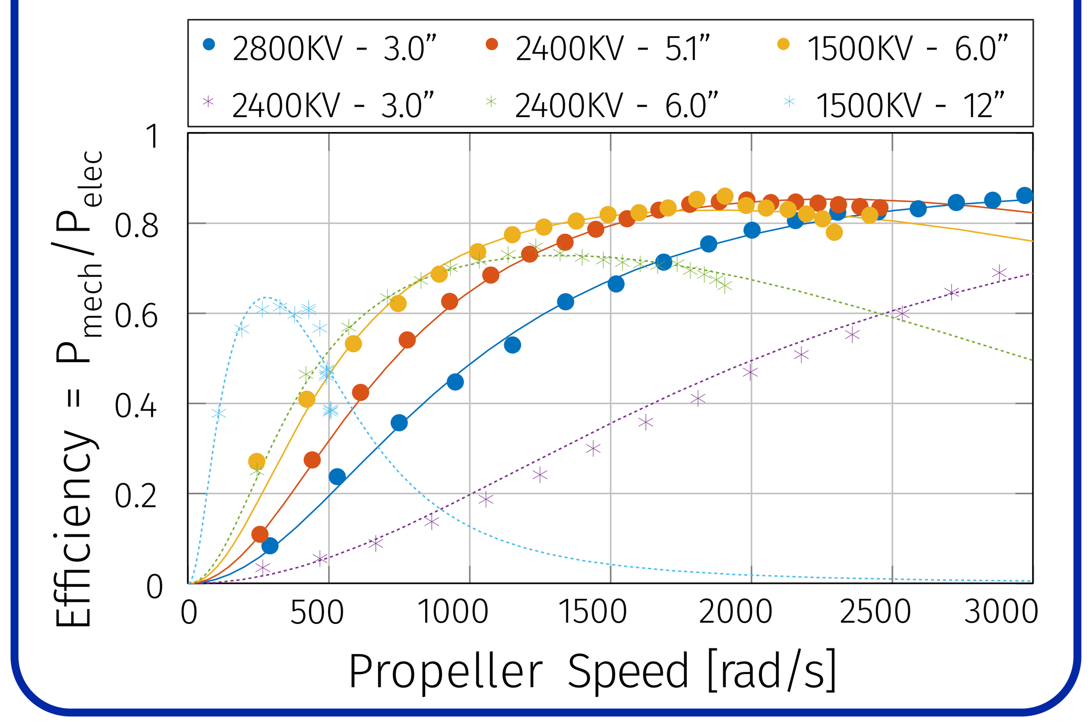
Develop a simulation and training pipeline that enables zero-shot transfer from the simulation to the real world.

Aerodynamics Model:

- hybrid model
  - quadratic fit model for propeller forces/torques
  - body drag using cuboid model
  - data-driven augmentation
- first-principles model very fast but not high fidelity
- data-driven component: model fitted real-world force/torque measurements
- polynomial terms of velocities, bodyrates, motorspeeds
- requires drone-specific data

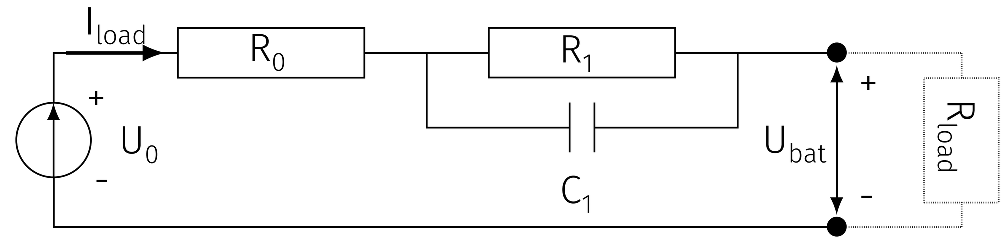
Graybox Motor Model:

- nonideal motor efficiency
- identified from real-word measurement data
- all recommended motor-propeller pairings: ≈ 0.7



Battery Model:

- one-time-constant model
- R_0, R_1, C_1 identified from data (over 2h of measurement data)

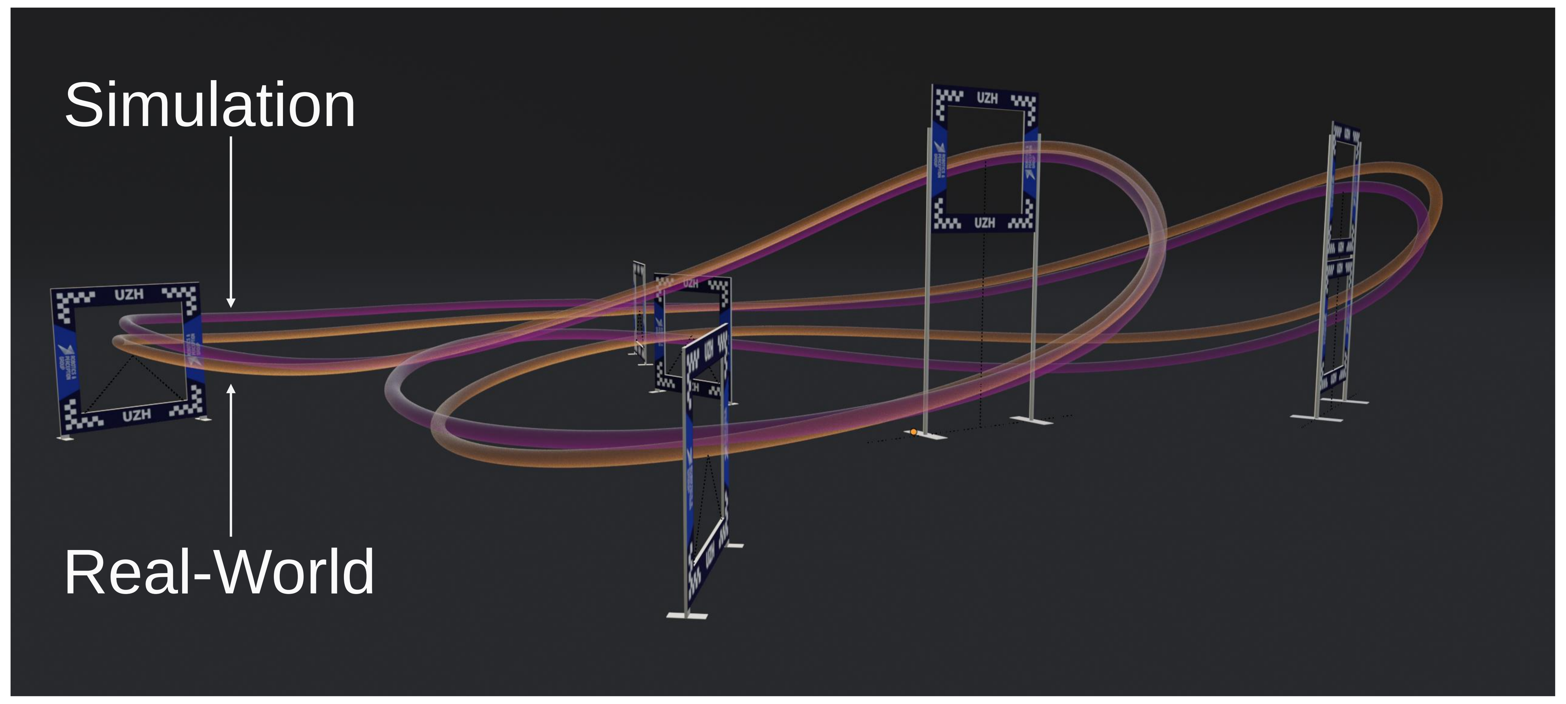


- able to simulate
  - dynamic loads
  - effective capacity
  - recovery effects
- improved accuracy over Peukert model

State-Based Autonomous Drone Racing

- requires external motion-capture system
- near-perfect state estimate @ 400Hz
- greatly facilitates sim2real transfer

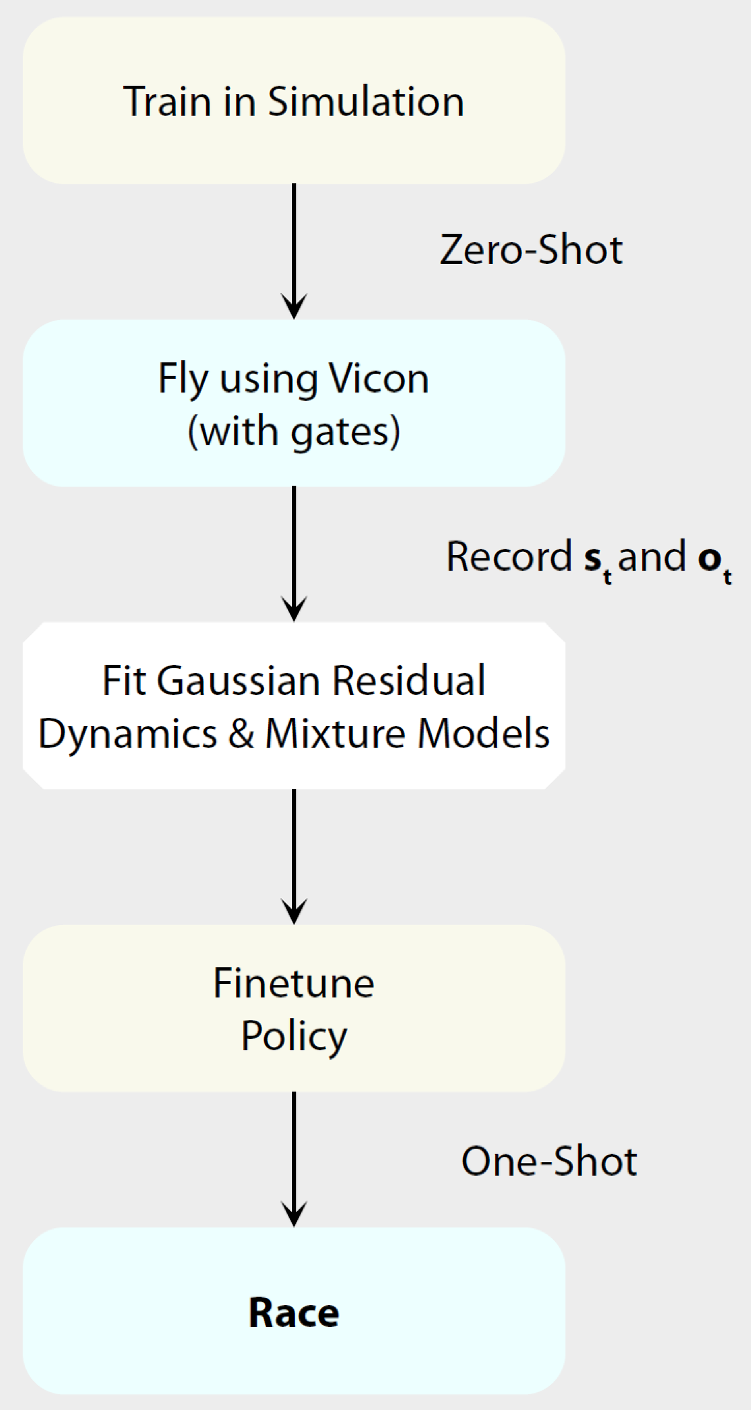
- simplistic models are sufficient
- use domain randomization
  - mass
  - inertia
  - thrust
- high-fidelity reduce sim2real gap further



Vision-Based Autonomous Drone Racing

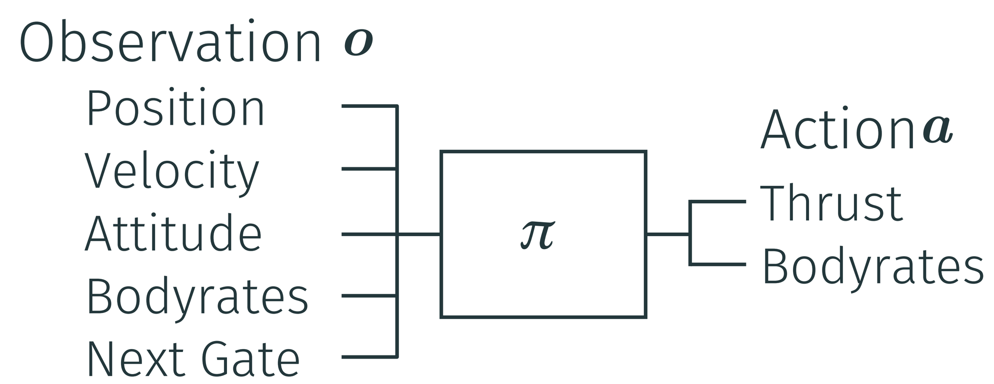
- all sensing and computation onboard of the vehicle
- relies only IMU & camera
- uncertain state-estimate
  - noisy state-estimation (e.g. gate detetions)
  - systematic errors (e.g. camera miscalibration)
  - state-dependent measurement accuracy
- even with high-fidelity dynamics models, perception uncertainty makes finetuning necessary.

Training pipeline



Reinforcement Learning

- controller trained purely in simulation using PPO
- 2 layer MLP with 512 neurons per layer



- reward:
  - progress $r_{prog} = \lambda_1 (d_{Gate}(t-1) - d_{Gate}(t))$
  - perception $r_{perc} = \lambda_2 \exp(\lambda_3 * \delta_{cam}^4)$
  - smoothness $r_{cmd} = \lambda_4 \,|\, a(t-1) - a(t) |$
  - crash $r_{crash} = \begin{cases} -5.0 & \text{if crash} \\ 0 & \text{otherwise} \end{cases}$
  - total reward $r_{prog} + r_{perc} + r_{cmd} + r_{crash}$

Sponsors:

robotics Swiss National Centre of Competence in Research

erc

FNSNF Swiss National Science Foundation

Horizon2020 European Union Funding for Research & Innovation