- Topic and Motivation
  - Minimum-time flight for VTOL-Drone
  - Push platform to its limits → to explor full potential
  - Use Cases:
    - Rescue
    - Delivery
    - Transportation

Image

- Our Approach
  - 2D Electra VTOL3,
  - 3D Quadcopter → more dim in observation space
    → other action space + double the hidden units

difference to primary source paper

- We produce as output PWM (pulse duration modulation) for each propeller individually. → Low level controller
- Paper produce collective Thrust + Body rate commands
  → low-level PID-controller then produces speed command for each rotor.

- What has been done so far
  - Problem reduced to 2D
  - Stabalize Slow / fast
  - Back to 3D case
  - → Image: Waypoints connected with lines

- Training Strategy

↓ Reward scaling

pretraining slow model

↓ Reward scaling

fast model

Lot of our time spent on Modeling of Reward

- Reward
  - Fly along trajectory → reward progress in each time step.
  - reward for reaching waypoint within certain distance

- Penalties
  - high body rates
  - falling

→ differences in phases

- slow phase: encourage velocity range and flying close to trajectory

fast phase:
- encourage faster velocities
- + scale down progress reward such that reaching points is rewarded more → in comparison result: more stable fast flight
- + penalty for body rate scaled down → allowing more rapid maneuvers

- **Plots 2D**

Success rate / velocity per step for slow + fast mode

- Success rate: how much drones reach every waypoint → almost 100%. After 400 000 steps
- Successfull drones: completion time goes down and converges
  → drone goes faster

- **Videos 2D**

Old: Overshoting was a problem

Now: Drone knows how to corrigate it

Plots 3D

- Reward: goes up in both cases
  ↑ slow learning: learned to takeoff at ~10 Mio steps.

  ⇒ Success rate goes suddenly up.

  ⇒ Fast learning: velocity at 4 meter per seconds.

- Isn't time optimal yet
  ⇒ pushing this further requires slowly increasing encouraged velocity and more training. → This takes some time!

More recent:

Office Test

Retraining with horizontal trajectories.

PID : 12.45

Ours : 3.55 → $7 \frac{m}{s}$ (Success rate 100%)

⇒ Further training actually will increase velocity.

Possible further Research

- Curriculum Learning: to slowly increase velocities in fast training
- Training with D Rand: Drone gets different weights during training s.t. its robust to model parameters ⇒ for sim to real
- Using phy. Model with latency: latency between sensor reading and action ~2ms

SAC / DDPG

Off policy → more Sample efficient

But we couldn't achieve convergence yet.