Time-optimal Flying a VTOL Drone

Leonardo Igler and Larissa Rickler

Topic and Motivation

- Minimum-time flight for VTOL-Drone
- Push platform to its limits
- Use Cases:
    - Rescue
    - Delivery
    - Transportation

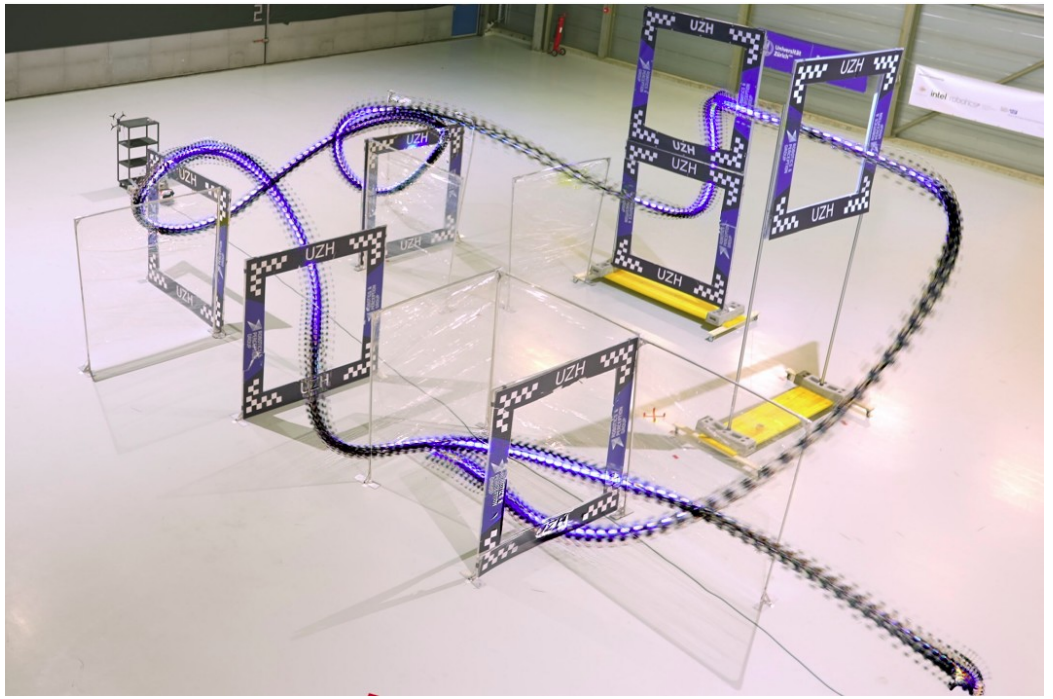
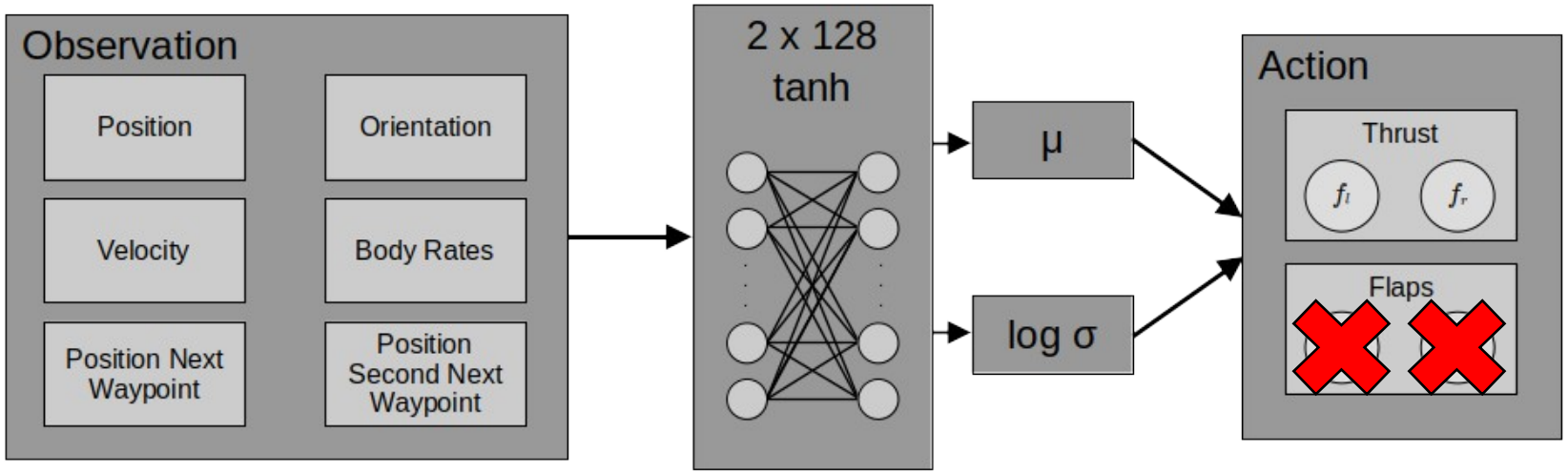
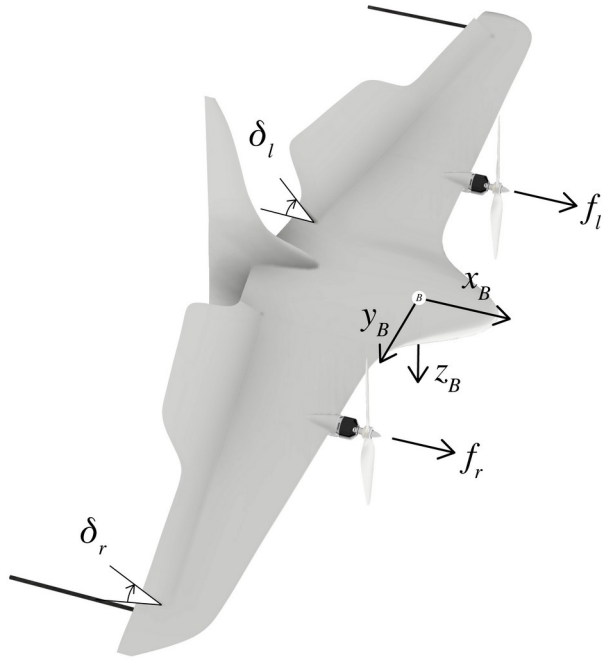


Fig. 1: Quadrotor races through a race track in the real world.

Our Approach

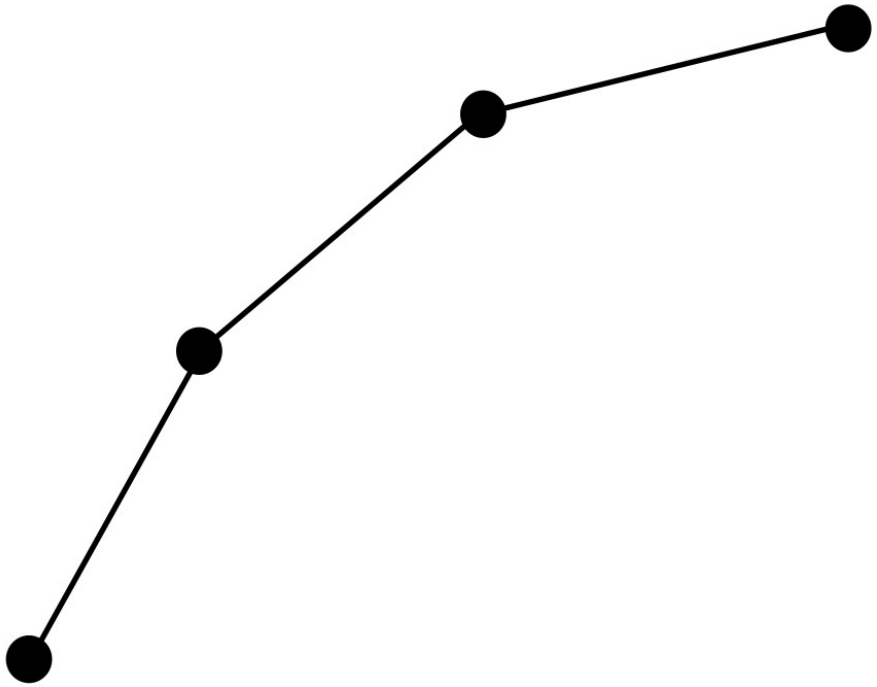
- Actor Critic
- Adam Optimizer





What has been done so far

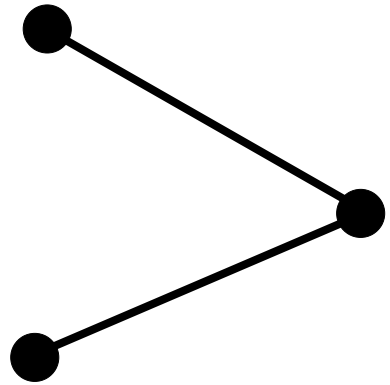
- Problem reduced to 2D
- Steer towards single points
- Fly along small random trajectories



Reward Function

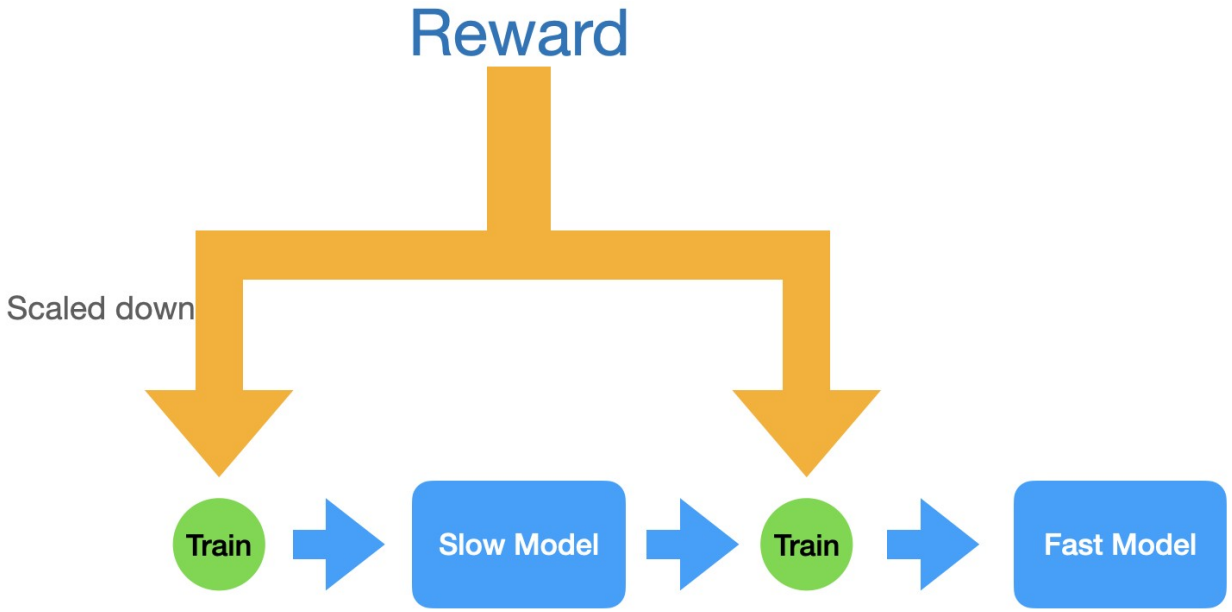
$$r(t) = k_p \cdot \Delta progress(t) + k_s \cdot progress(t) + k_{wp} \cdot r_{wp} - k_\omega \cdot \|\omega\| - fall$$

- Encourage line progression
- Encourage reaching points
- Penalize spinning
- Penalize inactivity

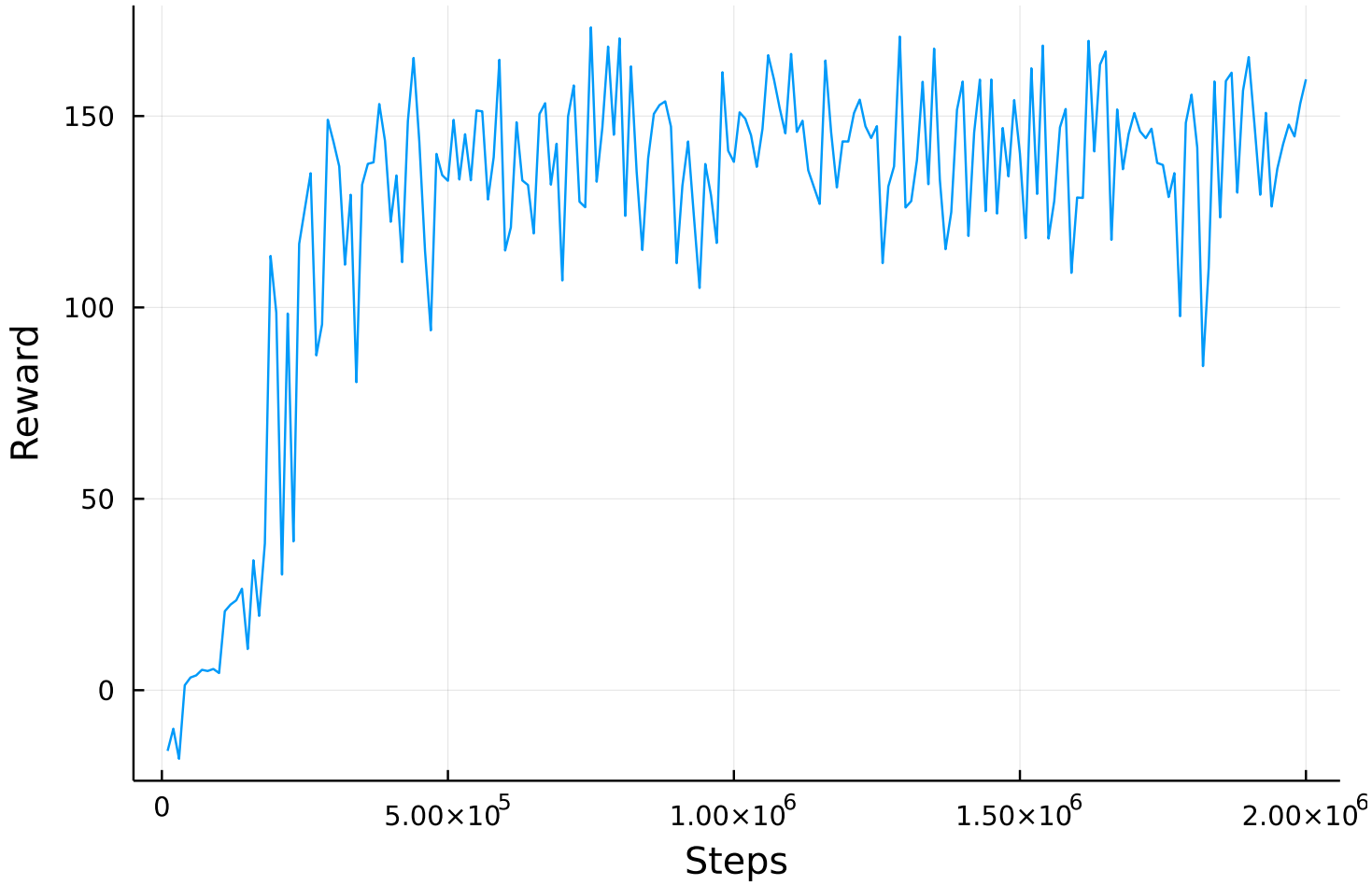


Training Strategy

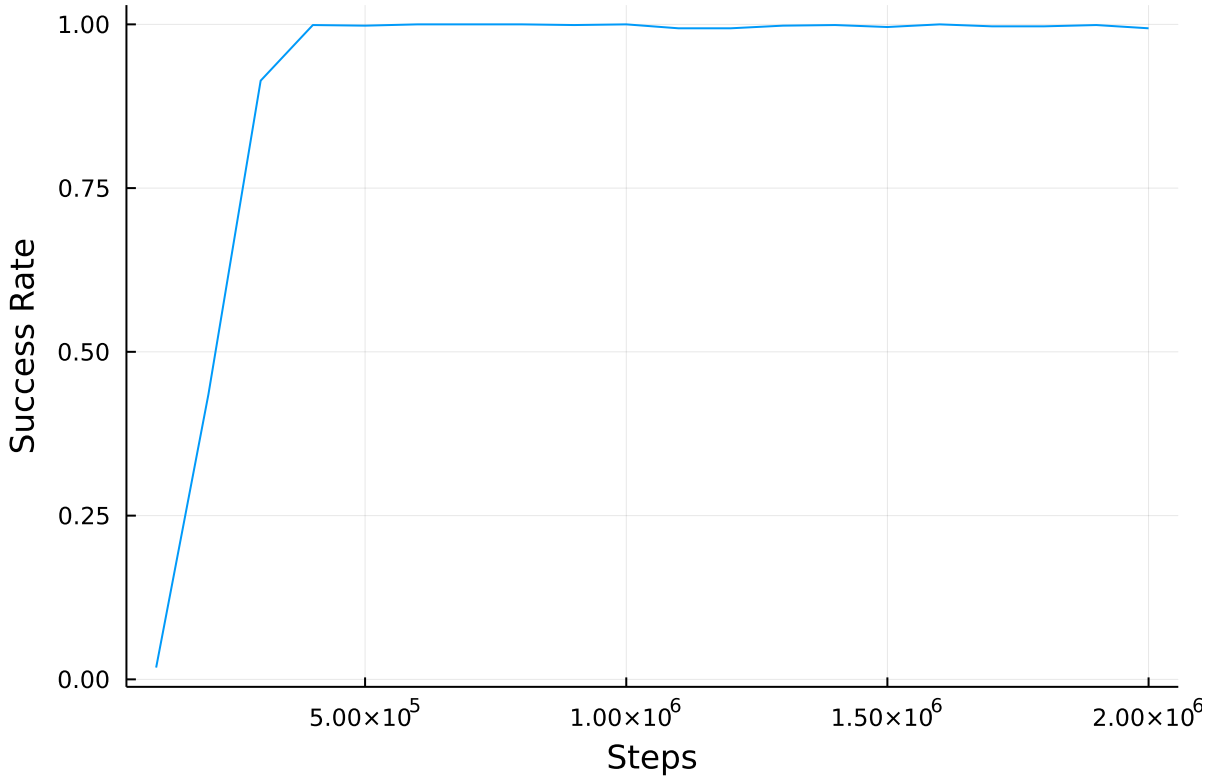
- Two Phases: Slow and Fast
- Slow: Learn to follow path
- Fast: Time optimality

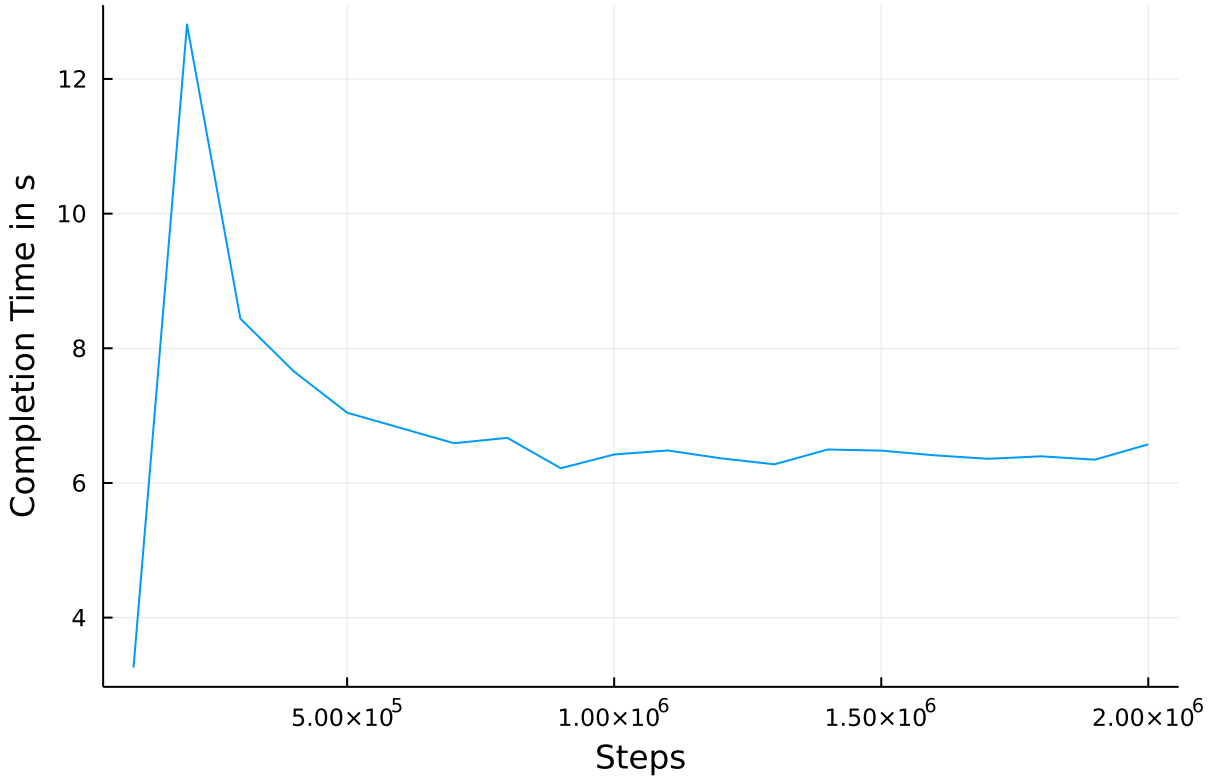


SLOW PLOTS

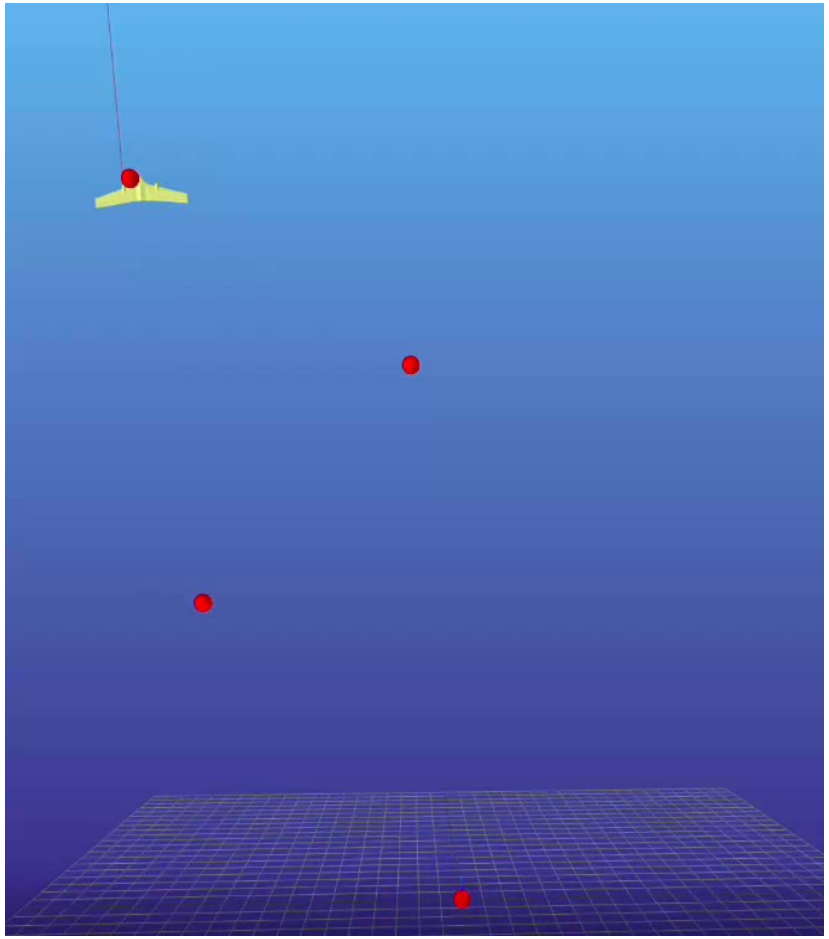


SLOW PLOTS


1.00
0.75
0.50
0.25
0.00
Success Rate
5.00×10^5
1.00×10^6
1.50×10^6
2.00×10^6
Steps

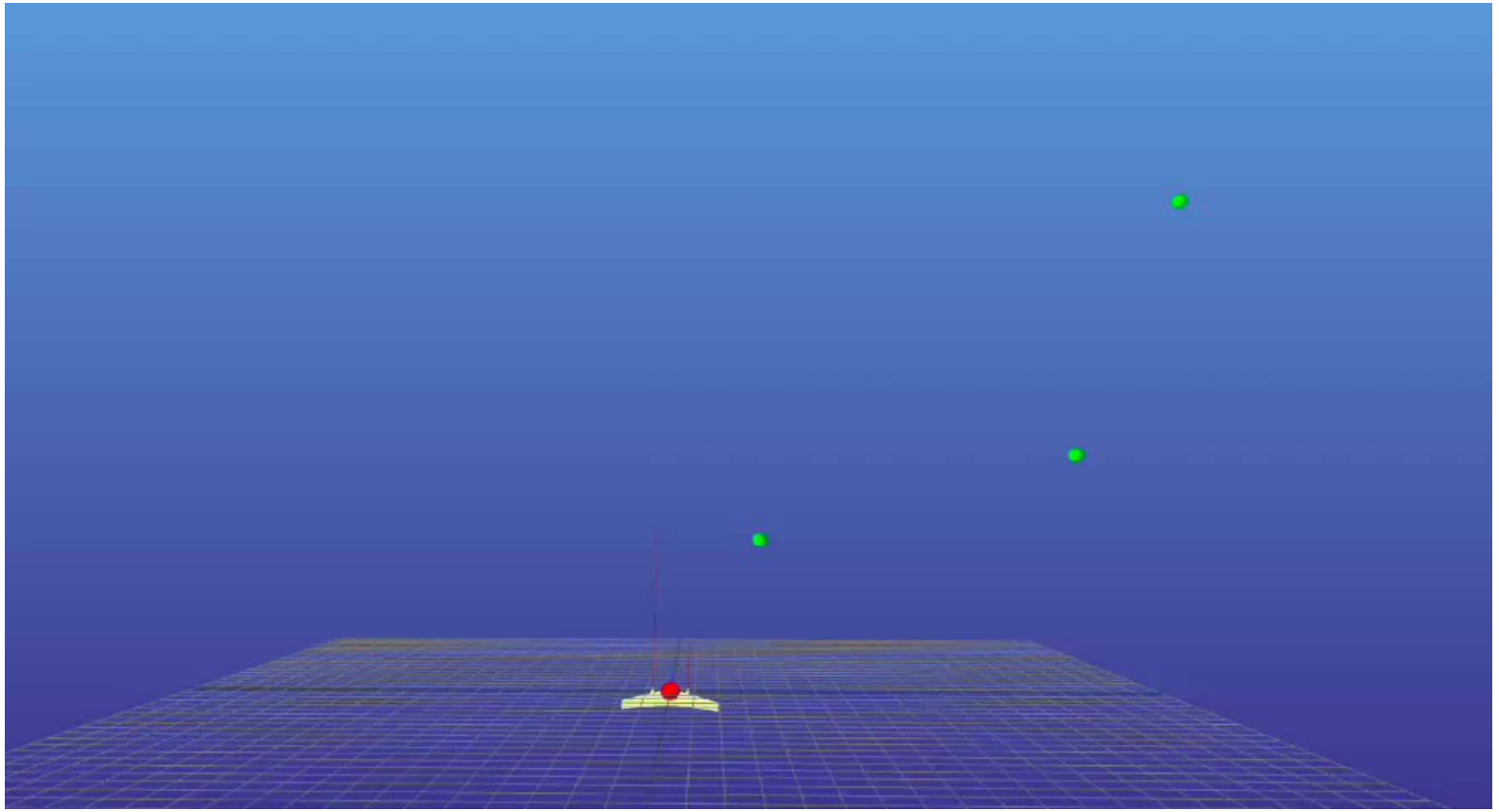

12
10
8
6
4
Completion Time in s
5.00×10^5
1.00×10^6
1.50×10^6
2.00×10^6
Steps

SLOW VIDEO



https://youtu.be/jzqBFTEnowo

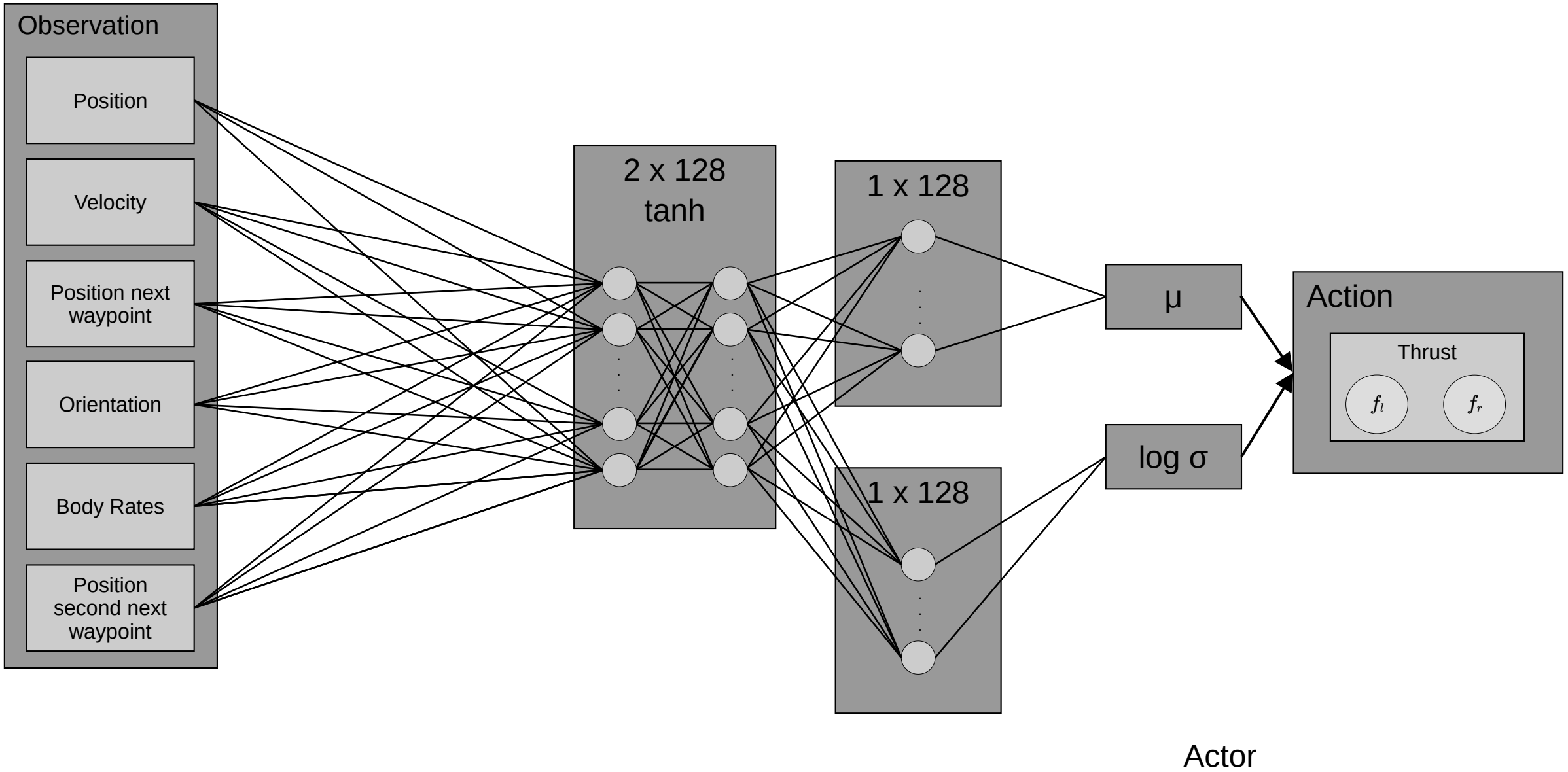
FAST VIDEO



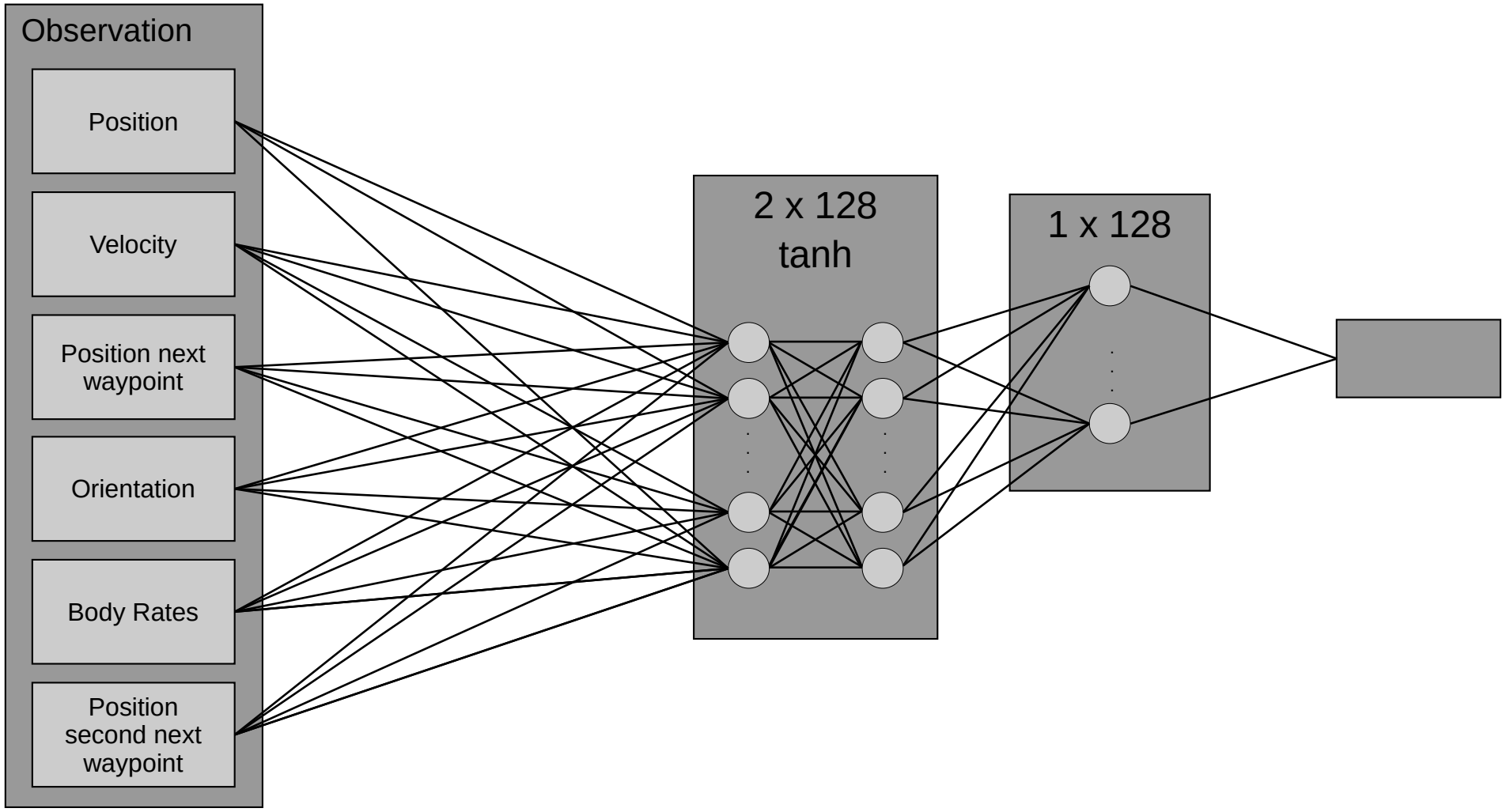
https://youtu.be/IRfzMXrSIUQ

Next Steps

Appendix


Observation
Position
Velocity
Position next waypoint
Orientation
Body Rates
Position second next waypoint
2 x 128 tanh
1 x 128
1 x 128
μ
log σ
Action
Thrust
f_l
f_r

Actor


Observation
Position
Velocity
Position next waypoint
Orientation
Body Rates
Position second next waypoint
2 x 128 tanh
1 x 128

Critic

Reward Function

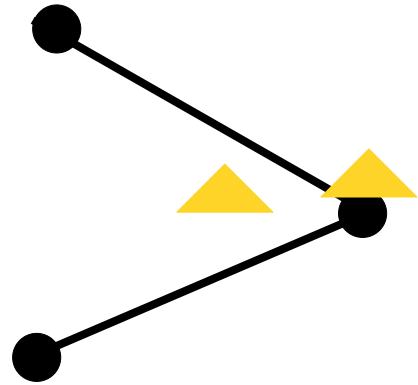
$$r(t) = k_p \cdot \Delta progress(t) + k_s \cdot progress(t) + k_{wp} \cdot r_{wp} - k_\omega \cdot \|\omega\| - fall$$

- Encourage line progression in each step
- $k_p = 5.0 \cdot s_m$

Reward Function

$$r(t) = k_p \cdot \Delta progress(t) + \boxed{k_s \cdot progress(t)} + k_{wp} \cdot r_{wp} - k_\omega \cdot \|\omega\| - fall$$

- Encourage line progression
  - Counteracts to possible singularities in $\Delta progress$
- $k_s = s_m \cdot (2 v_{max} \Delta t) / length_{traj}$



Reward Function

Progress calculation

$$
\begin{aligned}
l(\boldsymbol{p}), \boldsymbol{\psi}(\boldsymbol{p}) &= \underset{l(\boldsymbol{p}), \psi(\boldsymbol{p})}{\arg\min} \|\boldsymbol{p} - \boldsymbol{\psi}(\boldsymbol{p})\| \\
\text{s.t. } & \boldsymbol{\psi}(\boldsymbol{p}) = \boldsymbol{g}_{l(\boldsymbol{p})} + t(\boldsymbol{g}_{l(\boldsymbol{p})+1} - \boldsymbol{g}_{l(\boldsymbol{p})}), \\
& t = \frac{(\boldsymbol{p} - \boldsymbol{g}_{l(\boldsymbol{p})}) \cdot (\boldsymbol{g}_{l(\boldsymbol{p})+1} - \boldsymbol{g}_{l(\boldsymbol{p})})}{\left\|\boldsymbol{g}_{l(\boldsymbol{p})+1} - \boldsymbol{g}_{l(\boldsymbol{p})}\right\|^2}, \\
& l(\boldsymbol{p}) \in \{1, \ldots, n-1\} \ , \ t \in [0, 1].
\end{aligned}
$$

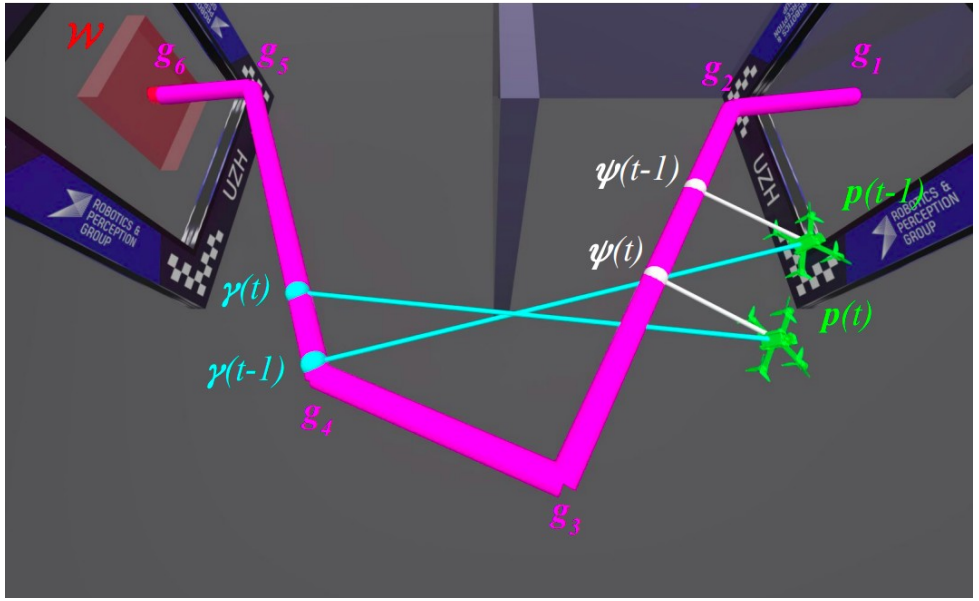


Fig. 2: Illustration of guiding path between waypoints g. Nearest point on the guiding path ψ from drone position p is used to calculate progress reward $r_p(t)$ and $r_s(t)$ while the next and second next waypoints are used as a part of observation.

Reward Function

Scale down reward of progress.

Encourages velocity in certain range.

$$s_m = s_{v_{max}} s_{v_{min}} s_{gd},$$

$$s_{v_{max}} = \begin{cases} 10^{v_{max} - \|\boldsymbol{v}\|} & \text{if } \|\boldsymbol{v}\| > v_{max}, \\ 1 & \text{otherwise,} \end{cases}$$

$$s_{v_{min}} = \begin{cases} 10^{\|\boldsymbol{v}\| - v_{min}} & \text{if } \|\boldsymbol{v}\| < v_{min}, \\ 1 & \text{otherwise,} \end{cases}$$

$$s_{gd} = \begin{cases} e^{-\|\boldsymbol{p} - \boldsymbol{\psi}(\boldsymbol{p})\| + d_{max}} & \text{if } \|\boldsymbol{p} - \boldsymbol{\psi}(\boldsymbol{p})\| > d_{max}, \\ 1 & \text{otherwise.} \end{cases}$$

Reward Function

$$r(t) = k_p \cdot \Delta progress(t) + k_s \cdot progress(t) + \boxed{k_{wp} \cdot r_{wp}} - k_\omega \cdot \|\omega\| - fall$$

- Encourage reaching points
  - Only added if waypoint is passed within distance $< dist_{tol}$
- $k_{wp} = 10.0 \cdot \#waypoints$
- $r_{wp} = exp(-dist(drone, waypoint)/dist_{tol})$

Reward Function

$$r(t) = k_p \cdot \Delta progress(t) + k_s \cdot progress(t) + k_{wp} \cdot r_{wp} - \boxed{k_\omega \cdot \|\omega\|} - fall$$

- Penalize spinning
- $k_\omega = 0.01$

Reward Function

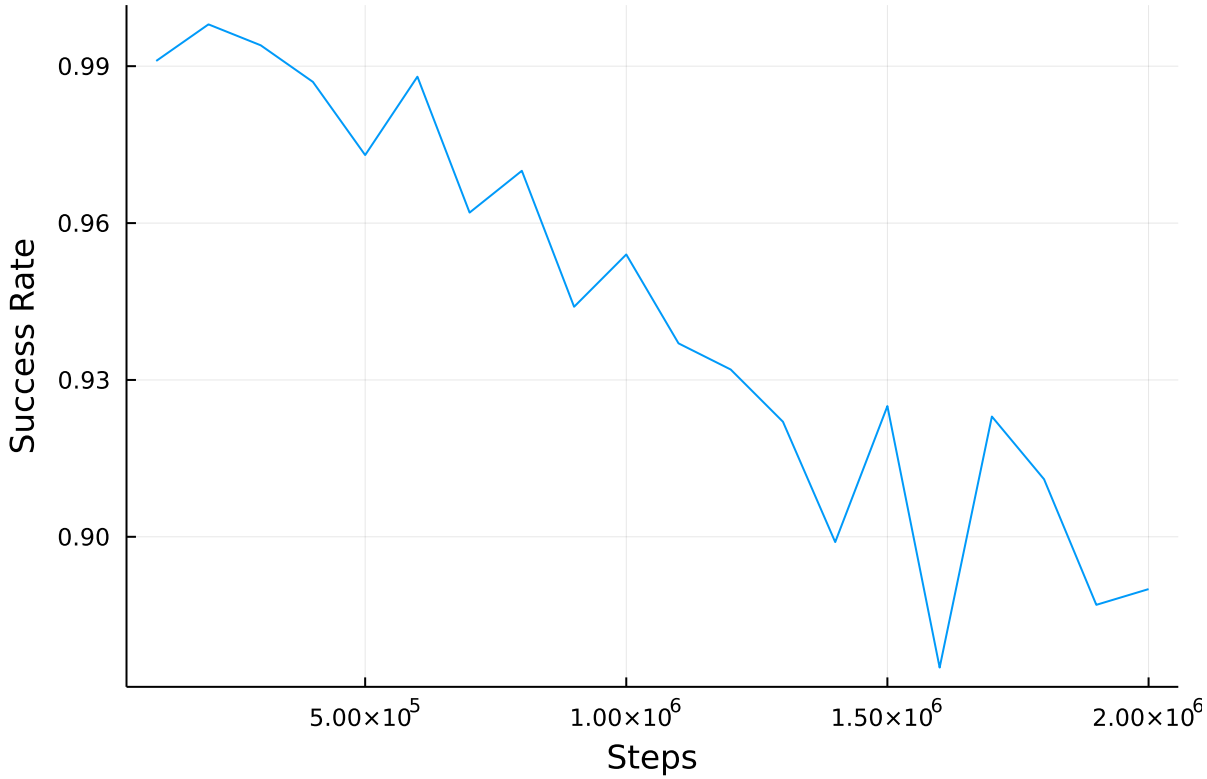
$$r(t) = k_p \cdot \Delta progress(t) + k_s \cdot progress(t) + k_{wp} \cdot r_{wp} - k_\omega \cdot \|\omega\| - \boxed{fall}$$

- Penalize inactivity
  - Only subtracted if hight falls under -2
- $fall = 1$

Termination Criteria

- Body rates to high
- Velocity to high
- Drone falls below -5
- Too much time passed
- Too far off path
- All waypoints reached

FAST PLOTS


Success Rate
0.99
0.96
0.93
0.90
5.00×10^5
1.00×10^6
1.50×10^6
2.00×10^6
Steps

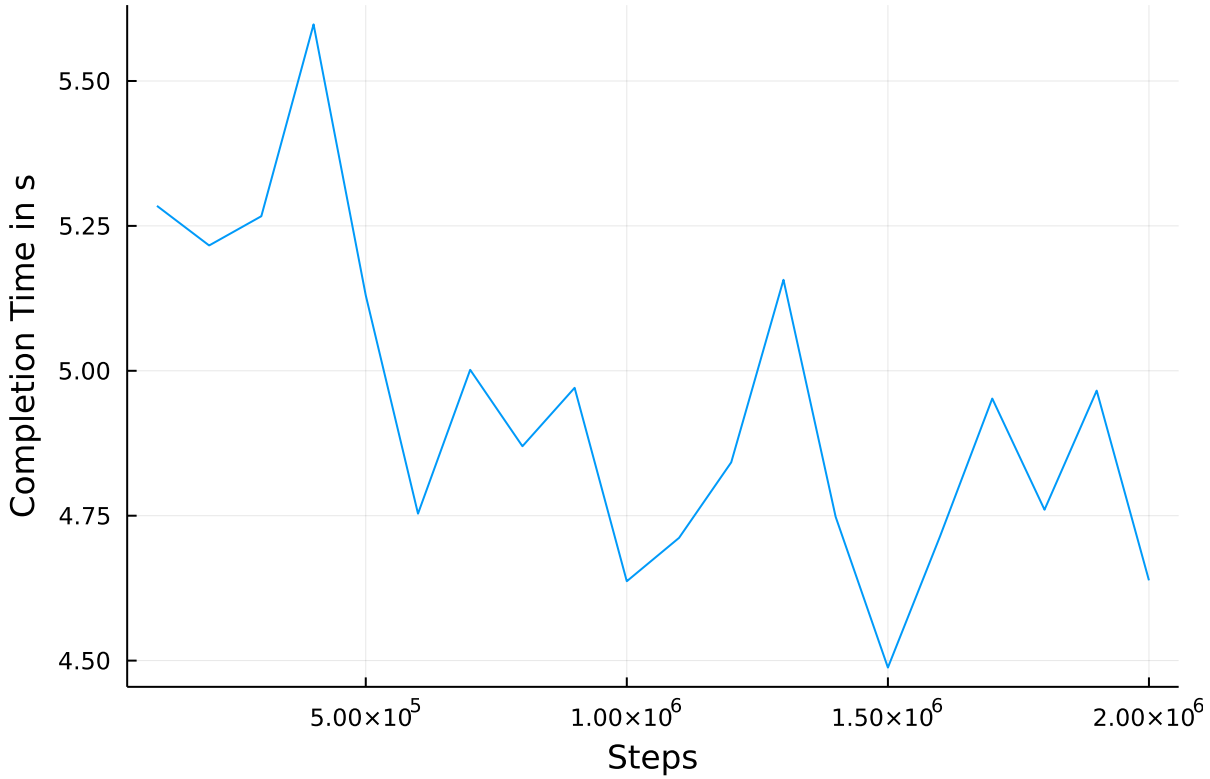

Completion Time in s
5.50
5.25
5.00
4.75
4.50
5.00×10^5
1.00×10^6
1.50×10^6
2.00×10^6
Steps

Image Sources

From https://arxiv.org/pdf/2203.15052v1.pdf

- Fig. 1: Our quadrotor races through a complex race track inthe real world.
- Fig. 2: Illustration of guiding path between waypoints g. Nearest point on the guiding path ψ from drone position p is used to calculate progress reward $r_p(t)$ and $r_s(t)$ while the next and second next waypoints are used as a part of observation.
  https://arxiv.org/pdf/2203.15052v1.pdf