Time-optimal Flying a VTOL Drone

Larissa Rickler
(03697651)
larissa.rickler@tum.de

Leonardo Igler
(03702035)
leonardo.igler@tum.de

***Abstract*—This milestone report describes our group's first experiments in training a drone to follow simple 2D trajectories using deep reinforcement learning methods.**

I. Methods

We used an actor-critic algorithm as approximator. The applied optimization algorithm is Adam. The actor network produces actions learned via the Proximal Policy Optimization (PPO). Both actor and critic are multilayer perceptrons (MLP), see Table I and Figure 1 for details on their architectures.

The chosen observation space and reward function are based on the suggestions of Penicka et al. [1]. The observations consist of two parts: the VTOL drone's state (world position, body orientation, as well as the translational and rotational velocities) and the relative path to the next two waypoints. The VTOL drone model provided by Flyonic [2] allows four possible actions: regulating thrusts of two propellers and adjusting angles of two flaps. In Figure 1 both observation and action space are illustrated.

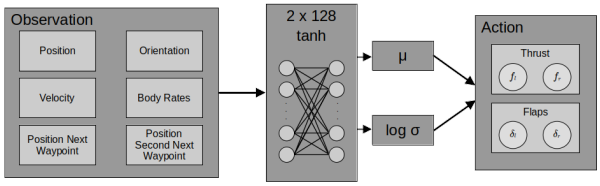


Fig. 1: Illustration of the actor network with the used observation and action space.

The reward function consist of five parts: encouraging progression along the trajectory (connected lines between n waypoints $g_1, ..., g_n$) in every time step, rewarding the overall reached distance along the trajectory, rewarding reached waypoints, penalizing high body rates ω and penalizing inactivity. To calculate the progress at position p we define the closest point on the trajectory as $\psi(p)$ and its corresponding line segment index as $l(p)$. See an example illustration in Figure 2. The total reward function $r(t)$ at time t is defined as

$$r(t) = k_p r_p(t) + k_s s(p(t)) + k_{wp} r_{wp} + k_\omega \|\omega\| - fall,$$

with the reached distance along the trajectory $s(p(t))$ and the progress reward at each time step $r_p(t)$ being defined as

$$s(p) = \sum_{i=1}^{l(p)-1} \|g_{i+1} - g_i\| + \|\psi(p) - g_{l(p)}\|,$$
$$r_p(t) = s(p(t)) - s(p(t - \Delta t)).$$

The reached distance $s(p(t))$ is part of the reward function to counteract possible singularities due to sharp corners of the trajectory and the minimum distance projection. The reward r_{wp} was payed, if the drone reached a waypoint within a distance d_{wp} smaller than a tolerance r_{tol} for the first time. The penalty term $fall$ was only subtracted if the drone fell under -1m due to inactivity. The contribution of each reward part is scaled by the hyperparameters k_p, k_s, k_{wp} and k_ω.

We initially focused on learning a slow and stable flying policy, henceforth referred to as *slow learning phase*. Therefore, the hyperparameters k_s and k_p were scaled by a factor s_m that encouraged translation velocities between v_{min} and v_{max}. This factor also motivates the drone to stay within a distance of d_{max} to the trajectory. The scaling factor s_m is calculated as

$$s_m = s_v \cdot s_{gd},$$
$$s_v = \min(1, 10^{v_{max} - ||v||}) \cdot \min(1, 10^{||v|| - v_{min}}),$$
$$s_{gd} = \min(1, e^{-\|p - \psi(p)\| + d_{max}}).$$

For our training process the values in Table II were used.

II. Experiments

All the experiments were executed inside the Flyonic's physic simulation environment [2]. For the scope of our group's milestone report, we reduced our experiments to a 2D plane (xz-plane). We have disabled all forces that would conflict with this 2D-world, as well as the possible actions regarding the flaps of the drone. It is noteworthy that the gravity is still enabled, having an orientation antiparallel to the z-axis.

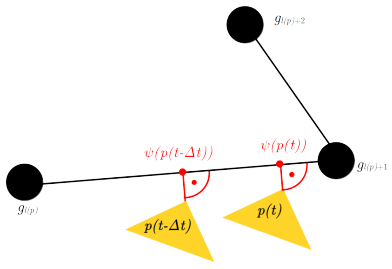
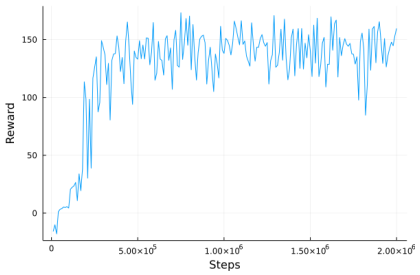


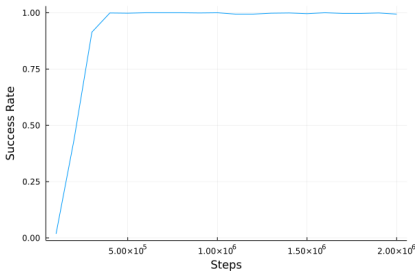
Fig. 2: Illustration of trajectory with three waypoints. The nearest point $\psi(p)$ on trajectory from drone at position p and its corresponding line segment index $l(p)$ is used to calculate the progress in each time step Δt and the reached distance.

TABLE I: Network architecture.

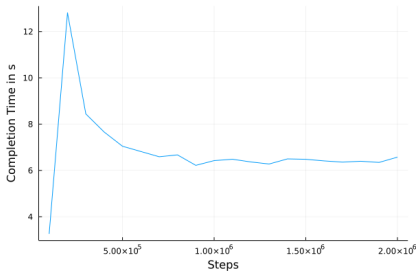
Network	Type	Input	Hidden Layer	Activation	Last Layer	Output	Output Size
actor	MLP	observation	2x128	tanh	1x128 1x128	action mean action variance	2 2
critic	MLP	observation	2x128	tanh	1x128	value	1



(a) Reward per step for the slow learning phase in 2D.



(b) Success rate per step for the slow learning phase in 2D.



(c) Completion time of successful attempts per step for the slow learning phase in 2D.

Fig. 3: Experimental setup described in Section II. Note that in 3c the completion time in step 1 is set to zero, since there were no successful flights.

TABLE II: Parameters of the algorithm.

Variable	Value	Variable	Value
v_{min} [m/s]	1.0	v_{max} [m/s]	3.0
Δt [s]	0.025	$fall$ [-]	1
n	4	k_{wp} [-]	$10.0n$
r_{tol} [m]	0.5	r_{wp}	$\exp(d_{wp}/r_{tol})$
k_p [-]	5.0	k_ω [-]	0.01
k_s [-]	$\frac{2v_{max}\Delta t}{\sum_{i=1}^{n-1}\|g_{i+1}-g_i\|}$	d_{max} [m]	0.2

The drone is trained for a total of two million time steps with a step size of Δt. In each episode a trajectory with 4 waypoints (including the starting point in the origin) is generated as follows: The derivation from the previous point is randomly sampled from the uniform distributions $\mathcal{U}(-7,+7)$ for the x-axis and $\mathcal{U}(+1.5,+7)$ for the z-axis. Then, the guiding trajectory is given by the three straight lines connecting the waypoints.

A flight is considered a success, if each of the four points is passed within a tolerance of r_{tol}. This is also reflected through the chosen termination criteria. An episode terminates once one of the following conditions is met:

- Drone falls 5 meters below the initial level
- Episode takes longer than $10.0 \cdot n$ seconds
- Drone deviates more than 5 meter from trajectory
- All waypoints were reached

During training, every 100,000 steps the current state of the model is saved for further evaluation, which is topic of Section III.

III. Results

Based on 10,000 different random trajectories, we evaluated the proposed method (slow learning phase) with respect to the following performance criteria: achieved reward, success rate on reaching every point within the tolerance and completion time of successful flights.

In Figure 3a the reward per step is plotted. We observed an increasing reward in the first 500,000 steps. The value seems to have converged by then. Next, the success rate is evaluated. The success rate increases in a fast pace. After 500,000 steps the rate already consistently reaches values up to 100%, as seen in Figure 3b. We have also investigated the average completion time of a successful flight, visualized in Figure 3c. The average time decreases and converges to approximately 6.5s. Given that the average trajectory length is about 13.5m in our experimental setup, the resulting average velocity of a successful episode is approximately $2.07\frac{\text{m}}{\text{s}}$. A video of two successful example episodes can be found in [3].

IV. Future Work

As stated in the previous section, the drone performs well for the models trained with the scaled reward functions. However, in order to achieve the desired time optimality, an unscaled reward should be applied. Regarding the performance of time optimal training strategy, further investigations need to be conducted.

Once the 2D case is optimized, the achieved progress will be generalized to the case of 3D trajectories. By then, the drone's whole set of action (propellers and flaps) will be unlocked, aiming to achieve time optimality using the full potential of the drone.

References

[1] R. Penicka, Y. Song, E. Kaufmann, and D. Scaramuzza, "Learning minimum-time flight in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7209–7216, Jul. 2022. [Online]. Available: https://doi.org/10.1109%2Flra.2022.3181755
[2] F. Süberkrüb, "Flyonic," Available at http://docu.flyonic.de.
[3] L. Igler and L. Rickler, "Slow flight." [Online]. Available: https://youtu.be/jzqBFTEnowo