Autonomous Navigation of UAV in Large-scale Unknown Complex Environment with Deep Reinforcement Learning

Chao Wang, Jian Wang, Xudong Zhang, and Xiao Zhang

Department of Electronic Engineering
Tsinghua University

UAV navigation: creating a smarter city

- Autonomous navigation in large-scale unknown complex environment
  - Drone delivery: delivering goods in cities, emergency treatment
  - Anti-terrorism: remote investigation, military strike



Challenges of large-scale unknown complex environment

Large-scale	Unknown	Complex
Environment covers several square kilometers	Environment is totally stochastic	Obstacles are dense

Hard-coded path planning is intractable
SLAM-based navigation is intractable
Sensing-and-avoidance-based navigation is inefficient

More intelligent algorithms need to be developed to cope with more complex environment

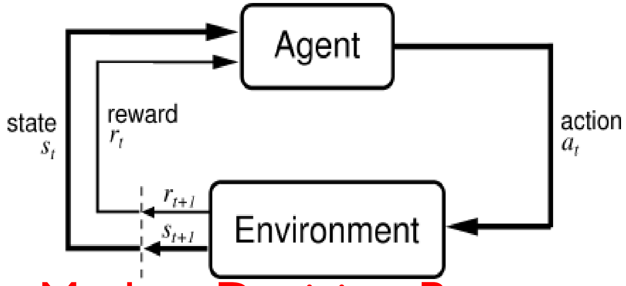
SLAM, simultaneously Localization and Mapping, is generally used to navigate and localize in indoor environment
Sensing-and-avoidance is already used by Amazon to deliver goods in countryside

Modeling UAV navigation as a reinforcement learning problem

UAV navigation: a sequential decision making problem

Observing environment → Taking proper action → Observing environment → Taking proper action → ……

Reinforcement Learning: learning to solve sequential decision making



Markov Decision Process

- State: s_t — sensory output
- Action: a_t — control profile
- Dynamic: $p(s_{t+1}|s_t,a_t)$ — unknown but stationary
- Reward: $p(r_t|s_t,a_t)$ — need to be designed

State profile and action profile

- Deep reinforcement directly takes high-dimensional sensory outputs as states[1]

State profile
- GPS to obtain the distance between target and present position of UAV
- gyroscope to obtain the first-perspective direction of UAV
- Other sensors to sense local environment, such as range finders to sense obstacles

Action profile
- gyroscope to obtain the first-perspective direction of UAV

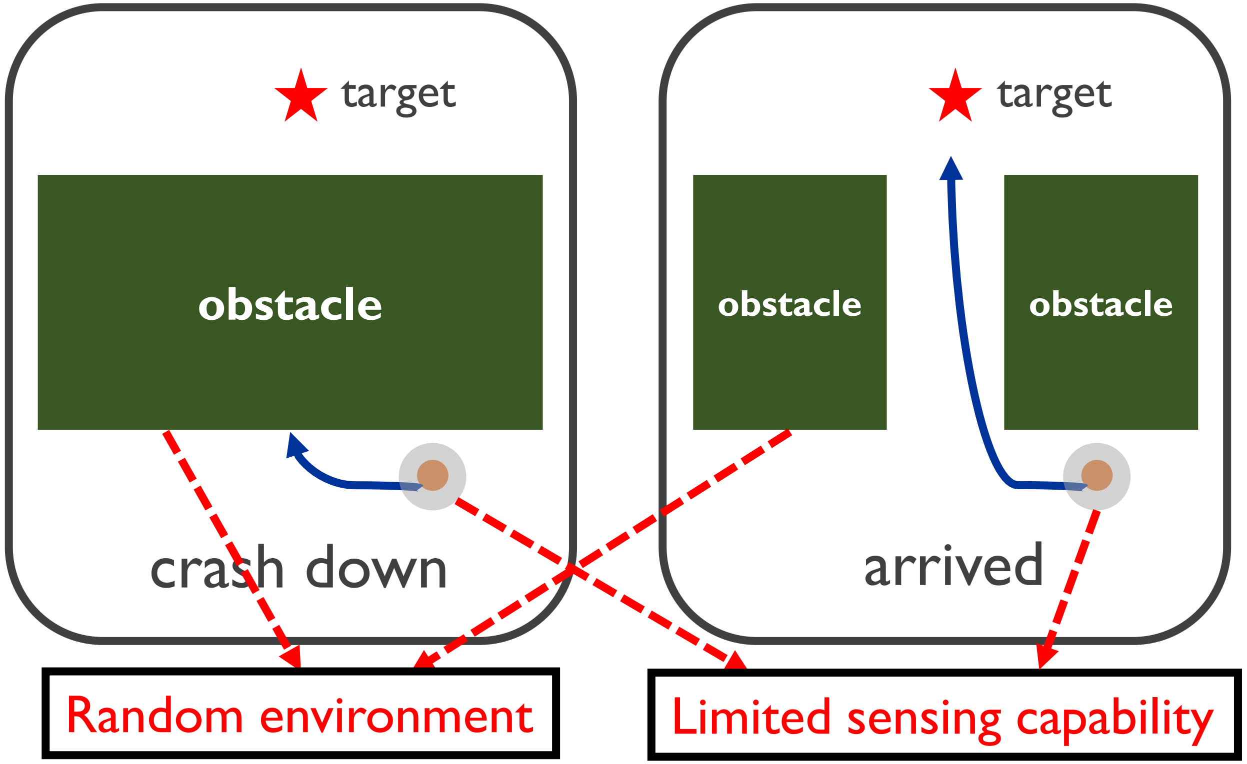
Reward

- Sparse reward
  - Agent would be rewarded only if it arrives at the target position
- Non-sparse reward
  - Agent would be rewarded whenever and wherever

Target reward	Rewarded if UAV approaches the target
Obstacle penalty	Penalized if UAV approaches any obstacles
Free space reward	Rewarded if UAV's first perspective points to free space
Transition penalty	Penalized as long as UAV moves forward

Partial observability of states

- Random environment
- Limited sensing capability
- Memoryless learning agent



Agent's action should be determined by its history observation and action trajectories

Attacking partial observability

- Policy function: projecting history trajectories to actions

$$a_t \sim \pi_\theta\left(a_t \mid h_t\right) \qquad \text{where} \quad h_t = [o_0, a_0, \cdots, a_{t-1}, o_t]$$

observations

actions

history trajectory

- Define value function and action-value function as

$$V^{\pi_\theta}\left(h_t\right) = E_{\tau_1}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,|h_t\right] \qquad Q^{\pi_\theta}\left(h_t, a_t\right) = E_{\tau_2}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,|h_t, a_t\right]$$

$$\tau_2 \sim p\left(h_{t+1}|h_t, a_t\right)\pi\left(a_{t+1}|h_{t+1}\right)p\left(h_{t+2}|h_{t+1}, a_{t+1}\right)\pi\left(a_{t+2}|h_{t+2}\right)\cdots$$

$$\tau_1 \sim \pi\left(a_t|h_t\right)p\left(h_{t+1}|h_t, a_t\right)\pi\left(a_{t+1}|h_{t+1}\right)p\left(h_{t+2}|h_{t+1}, a_{t+1}\right)\pi\left(a_{t+2}|h_{t+2}\right)\cdots$$

Attacking partial observability

- Define target function as

$$J(\theta)=\sum_{h_0} V^{\pi_\theta}(h_0)$$

Policy gradient

- Gradient of the target function

$$\frac{\partial J(\theta)}{\partial \theta}=\sum_{h_0}\sum_{h}\rho_{h_0}^{\pi_\theta}(h)\sum_{a}\frac{\partial \pi_\theta(h,a)}{\partial \theta}Q^{\pi_\theta}(h,a)$$

- Deterministic policy

$$a=\mu_\theta(h)$$

$$\pi_\theta(a|h)=\delta(a-\mu_\theta(h))$$

$$\frac{\partial J(\theta)}{\partial \theta}=\sum_{h_0}\sum_{h}\rho_{h_0}^{\mu_\theta}(h)\boxed{\frac{\partial Q^\theta(h_t,\mu^\theta(h_t))}{\partial a}\frac{\partial \mu^\theta(h_t)}{\partial \theta}}$$

Partially observable VS fully observable

- Gradient of the target function of fully observable MDP

$$\frac{\partial J(\theta)}{\partial \theta} = \sum_{s_0}\sum_{s}\rho_{s_0}^{\pi_\theta}(s)\sum_{a}\frac{\partial \pi_\theta(s,a)}{\partial \theta}Q^{\pi_\theta}(s,a)$$

$$\frac{\partial J(\theta)}{\partial \theta} = \sum_{s_0}\sum_{s}\rho_{s_0}^{\mu_\theta}(s)\frac{\partial Q^{\theta}(s_t,\mu^{\theta}(s_t))}{\partial a}\frac{\partial \mu^{\theta}(s_t)}{\partial \theta}$$

- Gradient of the target function of partially observable MDP

$$\frac{\partial J(\theta)}{\partial \theta} = \sum_{h_0}\sum_{h}\rho_{h_0}^{\pi_\theta}(h)\sum_{a}\frac{\partial \pi_\theta(h,a)}{\partial \theta}Q^{\pi_\theta}(h,a)$$

$$\frac{\partial J(\theta)}{\partial \theta} = \sum_{h_0}\sum_{h}\rho_{h_0}^{\mu_\theta}(h)\frac{\partial Q^{\theta}(h_t,\mu^{\theta}(h_t))}{\partial a}\frac{\partial \mu^{\theta}(h_t)}{\partial \theta}$$

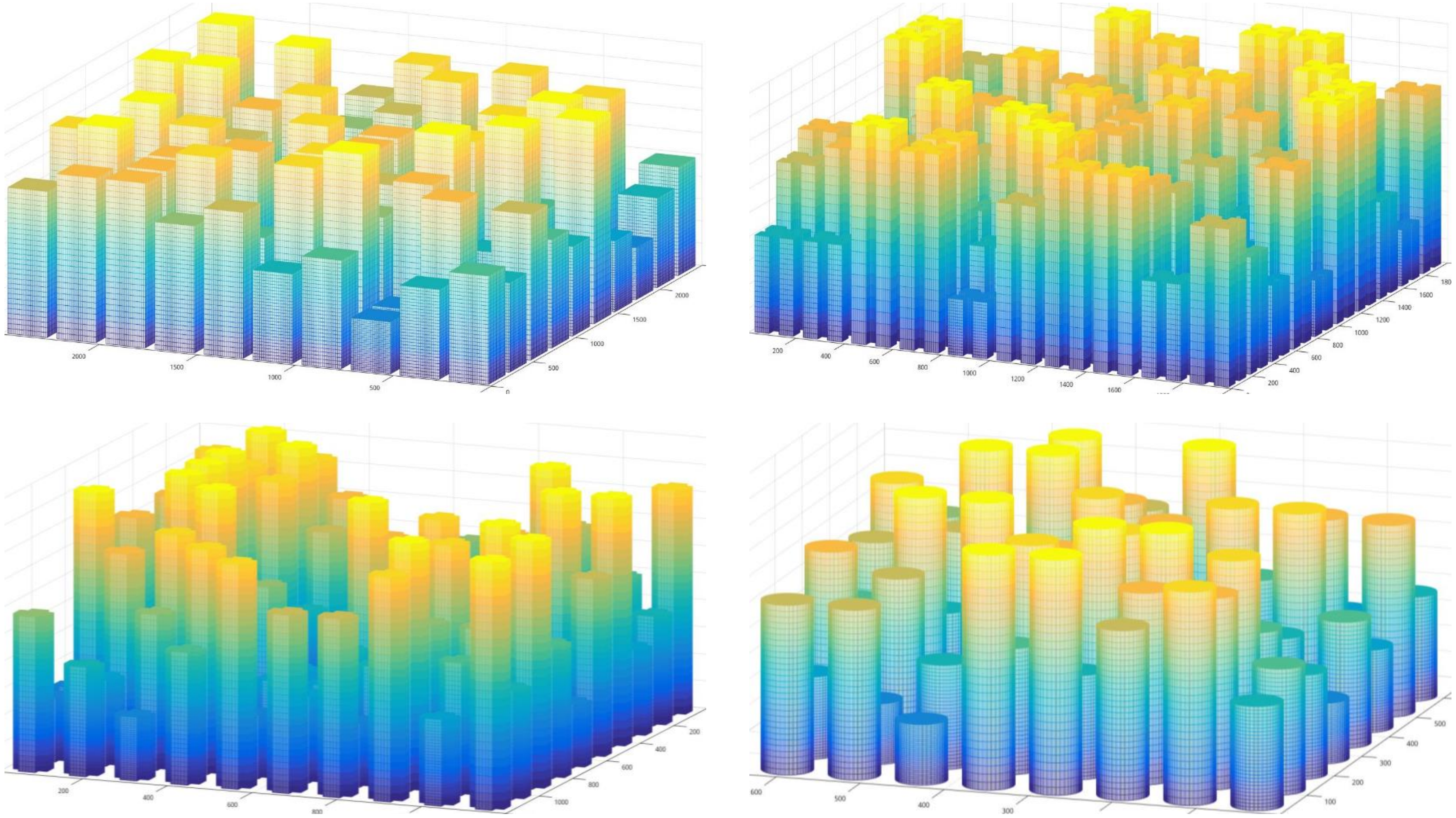
POMDPs can be regarded as MDPs nominally

Algorithm design: Fast-RDPG

- Fast-RDPG: Fast-Recurrent Deterministic Policy Gradient
  - Is based on existing algorithm named RDPG
  - Use Actor-Critic policy gradient architecture
  - Use two LSTMs to approximate $Q(h,a)$ and $\mu(h)$
- RDPG VS Fast-RDPG
  - RDPG lacks of theoretical guarantee
  - Fast-RDPG breaks the temporal correlation of samples

Stochastic gradient of RDPG	Stochastic gradient of Fast-RDPG
$\sum_{t=1}^{T} \gamma^{t-1} \frac{\partial Q^{\mu}(h_t, a)}{\partial a}\Big\vert_{a=\mu^{\theta}(h_t)} \frac{\partial \mu^{\theta}(h_t)}{\partial \theta}$	Actor $\boxed{\frac{\partial Q^{\theta}(h_t, \mu^{\theta}(h_t))}{\partial a}} \boxed{\frac{\partial \mu^{\theta}(h_t)}{\partial \theta}}$ Critic

Generating stochastic environment

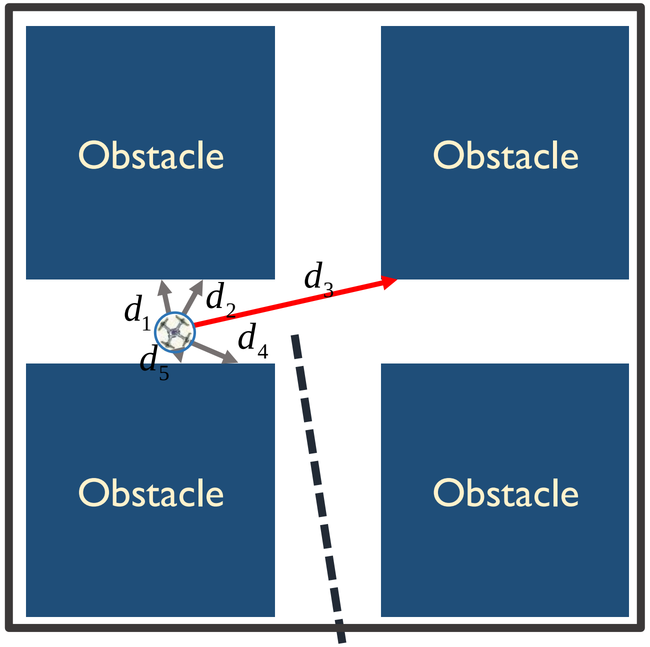


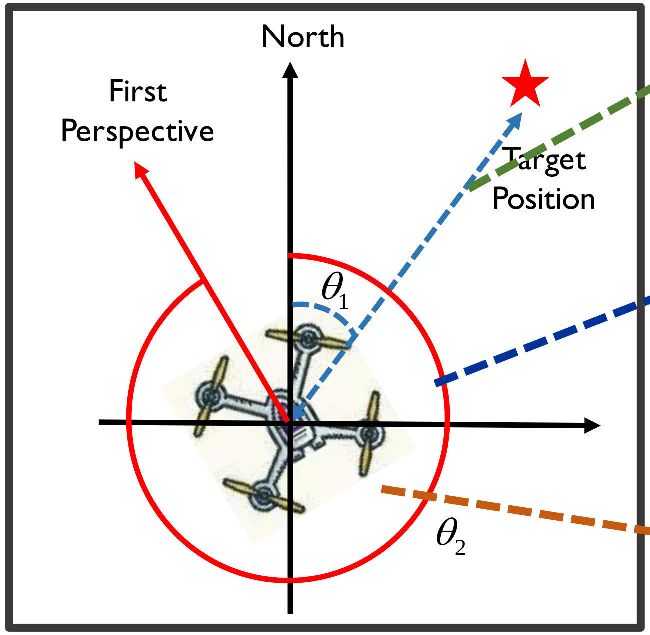
Every time the UAV completes a navigation task, the environment is re-generated randomly

In each environment, the height of the building is random

Sensors deployment

- UAV flies at fixed level and at constant speed
- Observations are composed of four parts





Distances between UAV and target

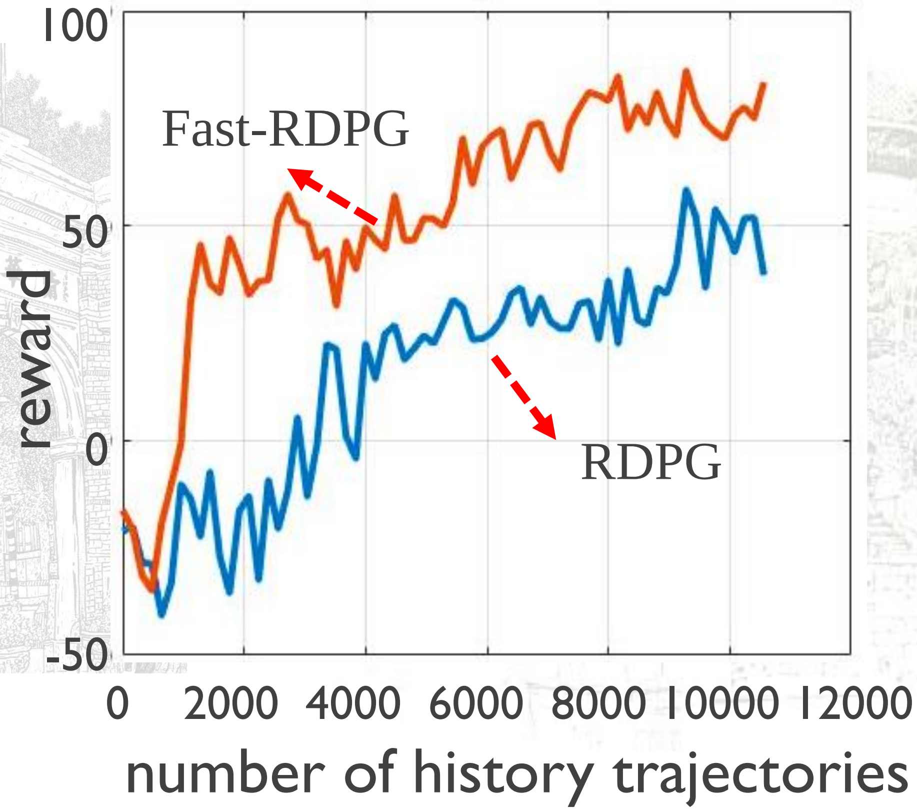
Angle Between UAV's first perspective direction and the North

Angle Between UAV's first perspective direction and the target direction

Distances between UAV and obstacles

Simulation result: RDPG VS Fast-RDPG

- Compared with RDPG, Fast-RDPG breaks the temporal correlation of samples and therefore converges very fast

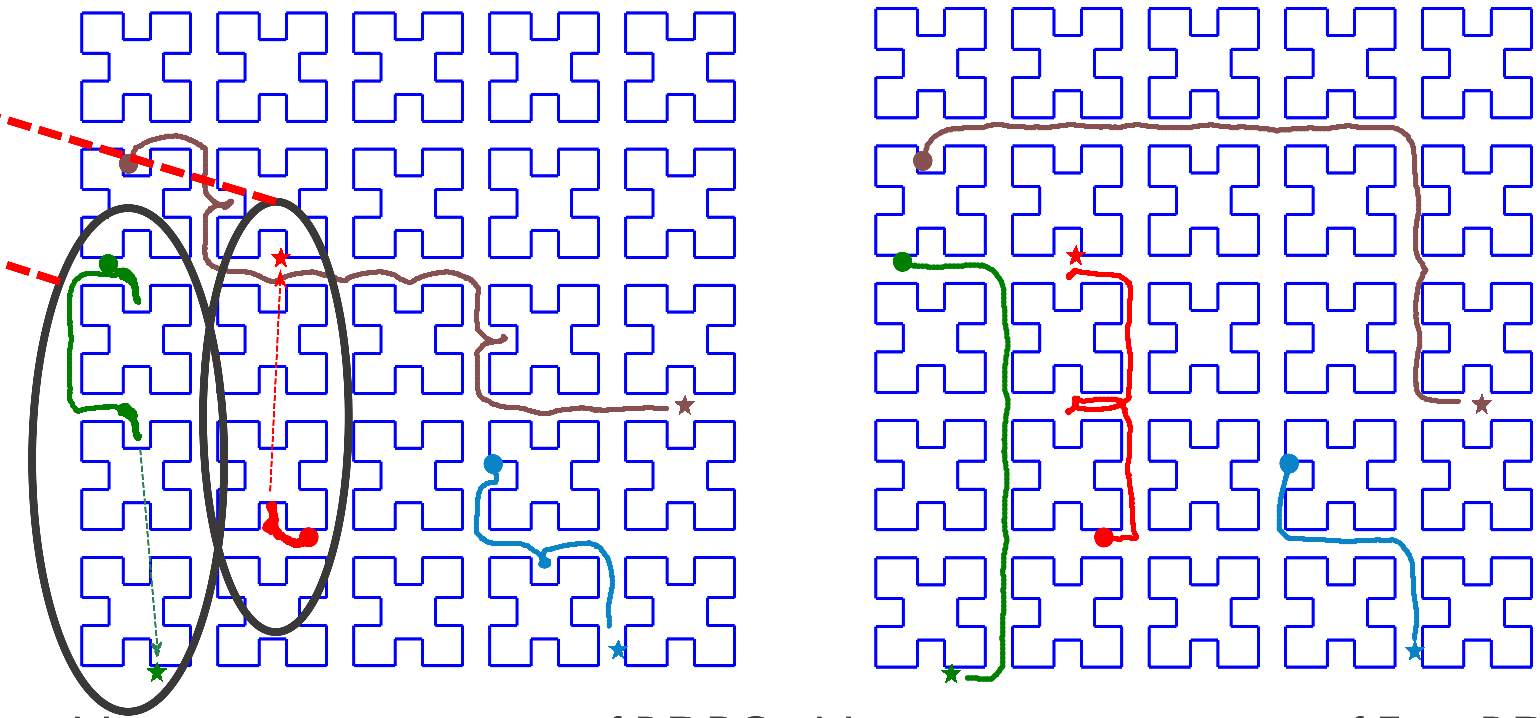


Simulation results: DDPG VS Fast-RDPG

Randomly generate four pairs of starting points and ending points

Navigation mission failed

Limited sensing capacity
Memoryless agent
Stochastic environment



Navigation trajectories of RDPG   Navigation trajectories of Fast-RDPG

Conclusion and future work

Large-scale unknown complex environment brings challenges to UAV navigation

- Highly complex environment disables traditional navigation methods
- Navigation agents need to learn to cope with complex environment

Proposed autonomous navigation of UAV with deep reinforcement learning

- Model UAV navigation as a sequential decision making problem
- Use deep reinforcement learning to solve the decision making problem
- Design Fast-RDPG algorithm to attack Partially observable MDP

Future work

- Test the proposed navigation algorithm in more real environment
- Directly cope with sparse reward

References

1. Mohammed, F., Idries, A., Mohamed, N., Al-Jaroodi, J., & Jawhar, I. (2014, May). UAVs for smart cities: Opportunities and challenges. In Unmanned Aircraft Systems (ICUAS), 2014 International Conference on (pp. 267-273). IEEE.
2. Cui, J. Q., Lai, S., Dong, X., & Chen, B. M. (2016). Autonomous navigation of UAV in foliage environment. Journal of Intelligent & Robotic Systems, 84(1-4), 259-276.
3. Bachrach, A., Prentice, S., He, R., & Roy, N. (2011). RANGE-Robust autonomous navigation in GPS-denied environments. Journal of Field Robotics, 28(5), 644-666.
4. Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on robotics and automation, 17(3), 229-241.
5. Israelsen, J., Beall, M., Bareiss, D., Stuart, D., Keeney, E., & van den Berg, J. (2014, May). Automatic collision avoidance for manually tele-operated unmanned aerial vehicles. In Robotics and Automation (ICRA), 2014 IEEE International Conference on (pp. 6638-6643). IEEE.
6. Zhang, A. M., & Kleeman, L. (2009). Robust appearance based visual route following for navigation in large-scale outdoor environments. The International Journal of Robotics Research, 28(3), 331-356.

References

7. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.
8. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.
9. Konda, V. R., & Tsitsiklis, J. N. (2000). Actor-critic algorithms. In Advances in neural information processing systems (pp. 1008-1014).
10. Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. International Conference on Machine Learning, 2014:387-395.
11. Heess, N., Hunt, J. J., Lillicrap, T. P., & Silver, D. (2015). Memory-based control with recurrent neural networks. arXiv preprint arXiv:1512.04455.
12. Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In Advances in neural information processing systems (pp. 1057-1063).
13. Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Q&A

- Thank you very much!