Reinforcement Learning Approach to Fly Quadrotors with a Faulted Rotor

Anonymous Author(s)
Affiliation
Address
email

Abstract

As applications of quadrotors increase, the development of robust and reliable algorithms to control quadrotors are becoming more meaningful. In this work, we deal with the problem of losing one rotor in the quadrotor. Previous works rely on modeling the complex dynamics of quadrotor and apply some of the existing modern control techniques. In this work we propose to solve this problem using a model-based reinforcement learning framework in conjunction with a meta-learning approach, our main aim is to study solutions of complex dynamics and fast adaptation in challenging robot tasks such as flying a quadrotor only using three rotors.

1 Introduction

Quadrotors are underactuated robots, that use four propellers to fly. Their applications have increased due to their useful features such as versatile displacement, reduction of time costs, ease of use, to name a few. While Hexacopters and Octacopters are more stable and robust to perturbations, quadrotors have higher efficiency and maneuverability concerning to their counterparts. However one of the most important disadvantages of quadrotors is that the system becomes highly unstable if one of the rotors fails.

Flying of quadrotors with only three propellers represents a challenging task since the system is really unstable, usual solutions sacrifice one degree of freedom (vertical axis) spinning-up at a certain angular velocity in the yaw axis [1], [2], [3], [4]. So these solutions require high knowledge in modeling robots and designing appropriate controllers, in contrast, our method learns unknown and complex behaviors to solve this task directly from sensor input.

2 Related Work

Currently, existing techniques deal with this problem by explicitly calculating the dynamics of the system and relying on this, to calculate the controllers using optimal control techniques. In [1] and [2] a quadrotor is modeled taking into account a specific failed rotor i.e. 3 rotors mechanism, this means that a different controller must be implemented for each possible failed rotor. In [5] goes an step further providing a method for planning a safe landing using a (rapidly-exploring random tree) RRT algorithm. Our method make adaptations online and just based of a set of previous states, while the methods mentioned before needs an extra module that must provide a Fault Detection [6], [7], [8],

Recent advances in model-based reinforcement learning shows that these methods can achieve good results in simulated complex tasks [9], [10] or real robots tasks [11]. To the best of our knowledge there are few works that uses model-based reinforcement learning for teaching a quadrotor. In [12] a low level model-based RL is designed, however this method just tackle the hovering problem, needing an extra module to control the position.

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

In [13], a model-based RL is combined with a meta-learning process [14], shown fast adaptation in the Mujoco setting and in a real legged-robot. We take [13] as a base for our proposed of research, however the adaptation and the designing of a model-based method in the quadrotor setting is more challenging as is shown in [12] where the maximum hovering time was of six seconds, thus this method is not comparable to a simpler proportional–integral–derivative (PID) controller even in fault-free case.

3 Approach and Current Progress

We use a model-based RL because is highly sampling efficient, i.e., it promotes a fast adaption [13]. This model-based is constructed using a neural network (f_θ) and is trained in a deterministic way. The network takes as input the concatenation of current state and action, giving as outputs, the difference between the next and current observation, as shown in the equation 1. We use a low computational cost model predictive control (MPC) to take optimal actions.

$$s_{t+1} = s_t + f_\theta(s_t, a_t) \tag{1}$$

In order to control the position and hovering behavior, as observation space we use: the matrix of rotation, position, angular velocity and linear velocity, this kind of observation space was used successfully to train a model-free policy to controls a quadrotor [15].

The reward function considers the euclidean distance between the target and current position. However, we will perform an ablation study so as to also consider the roll and pitch quadratic penalization. By doing so, we encourage the hovering conditions due to those angles are near to zero in such conditions. A quadrotor uses four motors defined as M_1, M_2, M_3, M_4, in the meta-training process we define a task as randomly select a M_i rotor or select any (fault-free), and power off that rotor, where i is sampled uniformly from $\{1, 2, 3, 4\}$. By doing this, we train an optimal model that is able to generalize over the different dynamics of the quadrotor (fault-free, fault-case), providing a rapid adaptation [13].

In order to develop a policy that is capable to fly the quadrotor in fault-free case and be able to adapt in case of rotor failure. One initial milestone is to develop a good policy with a model-based RL that is capable to fly the quadrotor in fault-free case. Thus, if the policy is unable to control the quadrotor in fault-free case, the policy will not be able to control the quadrotor in the fault case due to its more complex dynamics. In Figure 1 results of the milestone mentioned before is shown. The target point represented by the dashed curve $(0, 0, 0)$, we can see two problems. First, despite the quadrotor try to fly around the target point, the curve of the fly is too sharp. Second, the policy can not maintain the quadrotor in around the desired point in the long term. We have some ideas to solve these problem, first, improve the reward function taking into account the penalization of roll and pitch angles as was mentioned before.

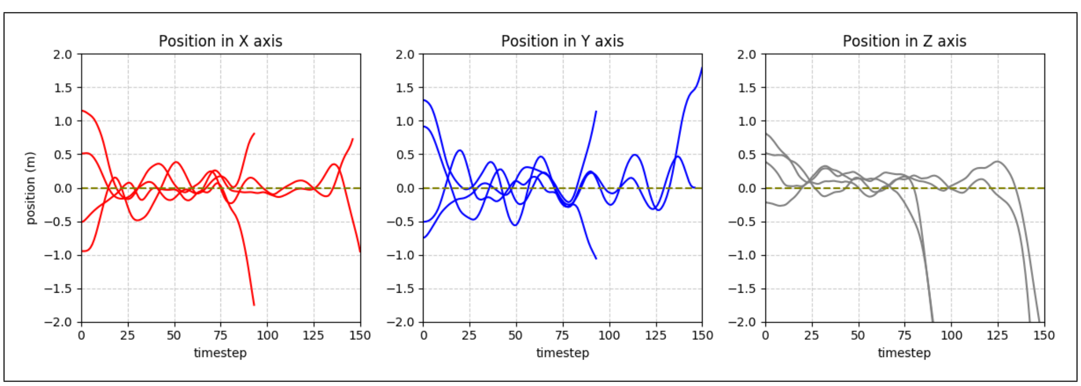

Figure 1: Position of quadrotor over time in X (left), Y (center) and Z (right) axis. Plots was obtained in fault-free case with the Model-based policy reinforcement learning

References

[1] A. Akhtar, S. L. Waslander, and C. Nielsen. Fault tolerant path following for a quadrotor. In *52nd IEEE Conference on Decision and Control*, pages 847–852, Dec 2013.

[2] Alexander Lanzon, Alessandro Freddi, and Sauro Longhi. Flight control of a quadrotor vehicle subsequent to a rotor failure. *Journal of Guidance, Control and Dynamics*, 37(2):580–591, March 2014.

[3] A. Merheb, H. Noura, and F. Bateman. Emergency control of ar drone quadrotor uav suffering a total loss of one rotor. *IEEE/ASME Transactions on Mechatronics*, 22(2):961–971, April 2017.

[4] S. Sun, L. Sijbers, X. Wang, and C. de Visser. High-speed flight of quadrotor despite loss of single rotor. *IEEE Robotics and Automation Letters*, 3(4):3201–3207, Oct 2018.

[5] Mojtaba Hedayatpour, Mehran Mehrandezh, and Farrokh Janabi-Sharifi. Path planning and controlled crash landing of a quadcopter in case of a rotor failure. *CoRR*, abs/1809.09334, 2018.

[6] Alessandro Freddi, Sauro Longhi, and Andrea Monteriù. A diagnostic thau observer for a class of unmanned vehicles. *Journal of Intelligent & Robotic Systems*, 67(1):61–73, Jul 2012.

[7] C. Zhaohui, H. Noura, T. B. Susilo, and Y. A. Younes. Engineering implementation on fault diagnosis for quadrotors based on nonlinear observer. In *25th Chinese Control and Decision Conference*, pages 2971–2975, May 2013.

[8] Zhaohui Cen, Hassan Noura, Tri Bagus Susilo, and Younes Al Younes. Robust fault diagnosis for quadrotor uavs using adaptive thau observer. *Journal of Intelligent & Robotic Systems*, 73(1):573–588, Jan 2014.

[9] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE International Conference on Robotics and Automation*, pages 7559–7566, May 2018.

[10] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model-based reinforcement learning for atari. *CoRR*, abs/1903.00374, 2019.

[11] Pieter Abbeel, Morgan Quigley, and Andrew Y Ng. Using inaccurate models in reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 1–8. ACM, 2006.

[12] Nathan O Lambert, Daniel S Drew, Joseph Yaconelli, Sergey Levine, Roberto Calandra, and Kristofer SJ Pister. Low-level control of a quadrotor with deep model-based reinforcement learning. *IEEE Robotics and Automation Letters*, 4(4):4224–4230, 2019.

[13] Ignasi Clavera, Anusha Nagabandi, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt: Meta-learning for model-based control. *CoRR*, abs/1803.11347, 2018.

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1126–1135. JMLR.org, 2017.

[15] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter. Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, 2(4):2096–2103, Oct 2017.