Modeling and Control of Multi-Rotor UAVs Paper Proposal

Michele Cipriano

January 15, 2019

1 Introduction

The papers I chose to propose cover different aspects of UAVs research. The first one, "Real-Time Planning with Multi-Fidelity Models for Agile Flights in Unknown Environments" [1], introduces a real time planning algorithm for UAVs in unknown environments, developing three different models of the UAV in order to guarantee computational tractability. The second one, "Vision-based Autonomous Landing in Catastrophe-Struck Environments" [2], is a practical use case of UAVs and it introduces a framework that allows a drone to land in a catastrophic environment, creating in the meanwhile a precise reconstrution of the world in order to make a human operator perform more complex analysis. The last paper, "Neural Lander: Stable Drone Landing Control using Learned Dynamics" [3], introduces a nonlinear controller that models the ground effects caused by the interaction between multi-rotor airflow and the environment with a DNN, formally proving its stability and presenting results on a real quadrotor.

2 Real-Time Planning with Multi-Fidelity Models for Agile Flights in Unknown Environments

This paper [1] introduces a real time planning algorithm for UAVs in unknown environments.

The approach is to develop three multi-fidelity models of the UAV in order to make sure the hierarchical planning architecture (composed by a global and a local planner) is behaving correctly. In particular, the Jump Point Search (JPS) algorithm has been used as a global planner to find the shortest path to the goal from the current position of the UAV, guaranteeing completeness and optimality. The local planner monitors the JPS solution looking for changes between each replan. If a change is detected, a new trajectory composed by a high (jerk-controlled), a medium (velocity-controlled) and a low fidelity model (JPS) is computed.

The environment is represented using a sliding map in order to make it possible to compute collision check for each of the three fidelity models. In particular, a depth map is obtained from the UAV and fused into an occupancy

grid using a k-d tree data structure, ensuring that the local planner is using the most recent data of the map.

The experiments have been performed in simulation considering a random cluttered forest, a bugtrap and a cluttered office scenario, making comparison with other methods. The experiments on hardware have used a UAV with all the perception, planning and control algorithms running onboard.

This method guarantees computational tractability keeping replanning times in the order of 5-40 ms in simulation.

Personal Comments: I have found this paper interesting because of the state of the art approach in real time planning when dealing with unknown environments. In particular, I liked the idea of considering multi-fidelity models to make sure the UAV is behaving correctly, always achieving the goal while keeping the computation real-time. The authors managed to describe in detail their approach, considering multiple scenarios and comparing their methods with several different algorithms. A YouTube video with some of the experiments is available at `https://www.youtube.com/watch?v=XVaT_c8uSTA`.

3 Vision-based Autonomous Landing in Catastrophe-Struck Environments

This paper [2] introduces an autonomous landing method based on vision algorithms when dealing with catastrophic environments.

The approach aims at making a UAV equipped with a bioradar to autonomously land on debris piles in order to locate the survivors in a natural disaster. In particular, the authors developed a multi-sensor fusion method to correctly estimate the pose of the UAV in two different 3D representations of the environment. The first one, which uses Octomap, is used as an internal map to make trajectory planning computationally tractable. The second one, which uses Voxblox, is a mesh reconstruction which is transmitted to the ground station in order to be analyzed by a human operator. In this way the UAV will be able to do both planned and emergency landings.

The landing site is evaluated in three steps. Initially a costmap is computed considering the terrain flatness, steepness, the depth accuracy and the energy consumption. Then, a set of candidates is chosen from the costmap. In the end, a clustering algorithm is used to reduce the number of unique landing sites. The landing site is chosen by solving an unconstrained nonlinear optimization problem and a minimum-jerk trajectory is computed using a variant of RRT*.

The experiments have been performed on a virtual environment, which simulates a small city, and in Training Center of Rescue (Germany), where a real quadrotor has been used.

This method, which makes use of a backend to use Octomap and Voxblox, makes it possible to compute the entire landing site detection algorithm in 167.5 ms, allowing it to be used on a real UAV.

Personal Comments: I have found this paper interesting because of its practical application. In particular, I liked how the authors decided to structure the pipeline of their algorithm so that it is possible to make it run on a real UAV. The idea of using two different 3D maps not only makes the UAV able to take decisions on its own, but also allows a human expert to understand

the real situtation of the environment, enabling more complex decisions and reducing the gap between research and real-world applications.

4 Neural Lander: Stable Drone Landing Control using Learned Dynamics

This paper [3] introduces a deep learning based nonlinear controller that controls the quadrotor landing, modeling the ground effects caused by the interaction between multi-rotor airflow and the environment.

Considering the dynamics of the quadrotor:

$$m\dot{\boldsymbol{v}} = m\boldsymbol{g} + R\boldsymbol{f}_u + \boldsymbol{f}_a \tag{1}$$

$$J\dot{\boldsymbol{\omega}} = J\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau}_u + \boldsymbol{\tau}_a \tag{2}$$

with $\dot{\boldsymbol{p}} = \boldsymbol{v}$, $\boldsymbol{p} \in \mathbb{R}^3$ global position, $\boldsymbol{v} \in \mathbb{R}^3$ linear velocity, $\boldsymbol{R} \in SO(3)$ attitude rotation matrix, $\boldsymbol{\omega} \in \mathbb{R}^3$ body angular velocity, $\boldsymbol{g} = [0, 0, -g]^T$ gravity vector, $\boldsymbol{f}_u = [0, 0, T]^T$ total thrust and $\boldsymbol{\tau}_u = [\tau_x, \tau_y, \tau_z]^T$ body torques, the approach aims at developing a DNN that models the unknown disturbances forces $\boldsymbol{f}_a$ and torques $\boldsymbol{\tau}_a$ which originates from complex aerodynamics interactions between the quadrotor and the environment. In particular, since the $\boldsymbol{\tau}_a$ is bounded during landing and take-off, only $\boldsymbol{f}_a$ needs to be studied.

The authors implemented a DNN with spectral normalization to guarantee the stability of the output. In particular, they exploited this kind on normalization to formally prove that the nonlinear controller used to determine the force exerted by the rotors is stable. The proof is built upon the assumption that $\boldsymbol{p}_d(t)$, $\dot{\boldsymbol{p}}_d(t)$ and $\ddot{\boldsymbol{p}}_d(t)$ are bounded, the inputs $\boldsymbol{u}$ of the system update much faster than the position controller and the approximation error on $\hat{\boldsymbol{f}}_a$ is upper bounded.

The experiments have been performed using the Intel Aero quadrotor. First, bench tests have been done in order to estimate the mass, the diameter of the rotors, the thrust coefficient, the air density and the gravity. Then, data has been collected by making the UAV perform some maneuvers in order to create a dataset. In the end, a DNN has been trained and the resulting model has been compared to a PD controller for take-off and landing tasks in both 1D and 3D.

This method allows the drone to precisely land on the ground surface in both 1D and 3D cases, reducing drifts in x, y directions and learning about non-dominant aerodynamics such as air drag. It is also interesting to notice that the lack of spectral normalization can even result in crashes, meaning that it is important to have a rigorous theoretical analysis to guarantee the stability of the controller used on the drone.

Personal Comments: I have found this paper interesting because of its theoretical approach to the field. Using a neural network as it is often results in undesired behaviours, going in the opposite direction of what it needs to be done in robotics. The authors managed to properly model the problem, formally proving the stability of their new approach, allowing it to be used on a real UAV and introducing a new research approach that integrates learning methods with control theory. A YouTube video with some of the experiments is available at `https://www.youtube.com/watch?v=C_K8MkC_SSQ`.

References

[1] J. Tordesillas, B. T. Lopez, J. Carter, J. Ware, and J. P. How, "Real-time planning with multi-fidelity models for agile flights in unknown environments," *CoRR*, vol. abs/1810.01035, 2018.

[2] M. Mittal, A. Valada, and W. Burgard, "Vision-based autonomous landing in catastrophe-struck environments," *CoRR*, vol. abs/1809.05700, 2018.

[3] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S. Chung, "Neural lander: Stable drone landing control using learned dynamics," *CoRR*, vol. abs/1811.08027, 2018.