Artificial Intelligence ••• (••••) •••–•••

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Automated aerial suspended cargo delivery through reinforcement learning

Aleksandra Faust [a], Ivana Palunko [b], Patricio Cruz [c], Rafael Fierro [c], Lydia Tapia [a,*]

[a] Department of Computer Science, University of New Mexico, Albuquerque, NM 87131, USA
[b] Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia
[c] Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131, USA

ARTICLE INFO

Article history:
Received 7 September 2013
Received in revised form 21 September 2014
Accepted 23 November 2014
Available online xxxx

Keywords:
Reinforcement learning
UAVs
Aerial cargo delivery
Probabilistic roadmaps
Motion planning
Trajectory planning
Robotics
Rotorcraft

ABSTRACT

Cargo-bearing unmanned aerial vehicles (UAVs) have tremendous potential to assist humans by delivering food, medicine, and other supplies. For time-critical cargo delivery tasks, UAVs need to be able to quickly navigate their environments and deliver suspended payloads with bounded load displacement. As a constraint balancing task for joint UAV-suspended load system dynamics, this task poses a challenge. This article presents a reinforcement learning approach for aerial cargo delivery tasks in environments with static obstacles. We first learn a minimal residual oscillations task policy in obstacle-free environments using a specifically designed feature vector for value function approximation that allows generalization beyond the training domain. The method works in continuous state and discrete action spaces. Since planning for aerial cargo requires very large action space (over 10^6 actions) that is impractical for learning, we define formal conditions for a class of robotics problems where learning can occur in a simplified problem space and successfully transfer to a broader problem space. Exploiting these guarantees and relying on the discrete action space, we learn the swing-free policy in a subspace several orders of magnitude smaller, and later develop a method for swing-free trajectory planning along a path. As an extension to tasks in environments with static obstacles where the load displacement needs to be bounded throughout the trajectory, sampling-based motion planning generates collision-free paths. Next, a reinforcement learning agent transforms these paths into trajectories that maintain the bound on the load displacement while following the collision-free path in a timely manner. We verify the approach both in simulation and in experiments on a quadrotor with suspended load and verify the method's safety and feasibility through a demonstration where a quadrotor delivers an open container of liquid to a human subject. The contributions of this work are two-fold. First, this article presents a solution to a challenging, and vital problem of planning a constraint-balancing task for an inherently unstable non-linear system in the presence of obstacles. Second, AI and robotics researchers can both benefit from the provided theoretical guarantees of system stability on a class of constraint-balancing tasks that occur in very large action spaces.

* Corresponding author.
E-mail addresses: afaust@cs.unm.edu (A. Faust), ivana.palunko@fer.hr (I. Palunko), pcruzec@unm.edu (P. Cruz), rfierro@unm.edu (R. Fierro), tapia@cs.unm.edu (L. Tapia).

http://dx.doi.org/10.1016/j.artint.2014.11.009
0004-3702/

1. Introduction

Unmanned aerial vehicles (UAVs) show potential for use in remote sensing, transportation, and search and rescue missions [1]. One such UAV, the quadrotor, is an ideal candidate for autonomous cargo delivery due to its high maneuverability, vertical takeoff and landing, single-point hover, and ability to carry loads 50% to 100% of their body weight. For example, cargoes may consist of food and supply delivery in disaster struck areas, patient transport, or spacecraft landing. The four rotor blades of a quadrotors make them easier to maneuver than helicopters. However, they are still inherently unstable systems with complicated non-linear dynamics. The addition of a suspended load further complicates the system's dynamics, posing a significant control challenge. Planning motions that control the load position is difficult, so automated learning methods are necessary for mission safety and success.

Recent research has begun to develop control policies for mini UAVs, including approaches that incorporate learning [1]. Learning system dynamics model parametrization has been successful for adaptation to changes in aerodynamics conditions and system calibration [2]. Other approaches, such as iterative learning methods for policy development, have been shown to be effective for aggressive maneuvers [3]. Another learning method, expectation–maximization, has been applied to the problem of quadrotor trajectory tracking with a linear model [4]. Even the task of suspended load delivery has been addressed for UAVs [5–9]. Our recent work used reinforcement learning (RL) in discrete action spaces to generate swing-free trajectories for quadrotors between two pre-defined, obstacle-free waypoints [5]. RL provided several advantages over previous work with dynamic programming [9]: a single learning phase leading to the generation of multiple trajectories, better compensation for accumulated error resulting from model approximation, and lack of knowledge of the detailed system dynamics. In this article, we extend the work in [5] through the development of an autonomous agent for aerial cargo delivery that works in environments with static obstacles. This method plans and generates trajectories through sampling of the environment and the system dynamics. In contrast to work currently under submission [10], which is concerned with computationally efficient continuous action selection and reinforcement learning for systems with high-dimensional continuous inputs in obstacle-free spaces, the work presented here delivers fully-automated planning agent for aerial cargo delivery in environments with static obstacles that relies on the discrete action space to perform trajectory generation along a collision-free path. The distinction between discrete and continuous action spaces is important because it is a basis for aerial cargo delivery that is both collision-free and with bounded load displacement. Manipulation of a suspended load has been performed with both RL [6] and model predictive control [7]. However, both of these methods require a pre-planned trajectory for the load to track and generate input that controls the UAV. They have an implicit assumption of the obstacle-free space in the area around the reference trajectory.

To learn control policy for minimizing the load displacement, we rely on approximate value iteration [11] with a specifically designed feature vector for value function approximation. There are two challenges we face. First, the approximate value iteration requires a state-space sampling domain to learn the policy. The learning must have sufficient sample-density to be successful. To provide sufficient sample-density while learning, we sample in the small state subspace around the goal but choose a feature vector that is defined beyond the sampling subspace. The second challenge we face, is that the planning for swing-free motion consists of a large action space (over 10^6 actions). Such a large action space is impractical for learning, thus we learn in an action subspace, and plan with the larger action space. Relying on Lyapunov stability theory [12], the article contributes sufficient conditions that the Markov decision process (MDP) formulation (system dynamics, action space, and state-value function approximation) must meet to ensure the cargo delivery with minimal residual oscillations. Within the conditions we are free to change MDP properties as it suits our needs and to transfer learning to compatible MDPs. For example, for practicality we learn in a 2-dimensional action subspace and transfer the state-value function approximation to MDP with a 3-dimensional action space to plan altitude changing trajectories. As another example, we develop a novel path-following agent that minimizes the load displacement by action-space adaptation at every planning step. In the context of related work in the obstacle-free case [13,14], we transfer to MDPs with state and action supersets and noisy dynamics using a behavior transfer function that transfers directly the learned value function approximation to the new domain with no further learning. Another approach examined action transfer between the tasks, learned the optimal policy and transferred only the most relevant actions from the optimal policy [15]. In obstacle-free spaces, we take the opposite approach; to save computational time, we learn a sub-optimal policy on a subset of actions, and transfer it to the expanded action space to produce a more refined plan. When planning a path-following trajectory, we work with a most relevant subset of the expanded action space. Partial policy learning for fixed start and goal states, manages state space complexity by focusing on states that are more likely to be encountered [16]. We are interested in finding minimal residual oscillations trajectories from different start states, but we do have a single goal state. Thus, all trajectories will pass near the goal state, and we learn the partial policy only in the vicinity of the goal state. Then, we apply the learned policy to an arbitrary start state.

Motion planning methods, which define a valid, collision-free path for a robot, are often solved in *configuration space (C_{space})*, the space of all configurations. The valid, collision-free path lies in the collision-free portion of C_{space}, C_{free}. Many planning methods work by either learning and approximating the topology of C_{space}, e.g., Probabilistic Roadmap Methods (PRMs) [17], or by traversing a continuous path in C_{free}, e.g., Rapidly-Exploring Random Tree (RRT) and Expansive Space Tree (EST) methods [18–20]. One primary difference between PRMs and RRT/EST methods is that PRMs were designed to learn C_{free} topology once and then use this knowledge to solve multiple planning queries, where RRTs/ESTs expand from a single start and/or goal position for a single planning query. PRMs have been modified to work well for complex robotic problems, including moving obstacles [21,22], noisily modeled environments (e.g., with sensor) [23], and localization errors [24,25]. Also, PRMs have been previously integrated with RL, e.g., a manipulator moving in a dynamic space [26]. In contrast, we use the RL agent as a local planner to plan trajectories along a collision-free path obtained with PRMs.