Reconfigurable Path Planning for an Autonomous Unmanned Aerial Vehicle

Mariusz Wzorek and Patrick Doherty
Department of Computer and Information Science
Linköping University, SE-58183 Linköping, Sweden
{marwz,patdo}@ida.liu.se

Abstract

In this paper, we present a motion planning framework for a fully deployed autonomous unmanned aerial vehicle which integrates two sample-based motion planning techniques, Probabilistic Roadmaps and Rapidly Exploring Random Trees. Additionally, we incorporate dynamic reconfigurability into the framework by integrating the motion planners with the control kernel of the UAV in a novel manner with little modification to the original algorithms. The framework has been verified through simulation and in actual flight. Empirical results show that these techniques used with such a framework offer a surprisingly efficient method for dynamically reconfiguring a motion plan based on unforeseen contingencies which may arise during the execution of a plan.

1 Introduction

The use of Unmanned Aerial Vehicles (UAVs) which can operate autonomously in dynamic and complex operational environments is becoming increasingly more common. While the application domains in which they are currently used are still predominantly military in nature, in the future we can expect widespread usage in the civil and commercial sectors. In order to insert such vehicles into commercial airspace, it is inherently important that these vehicles can generate collision-free motion plans and also be able to modify such plans during their execution in order to deal with contingencies which arise during the course of operation. Motion planners capable of dynamic reconfiguration will be an essential functionality in any high-level autonomous UAV system. The motion planning problem, that of generating a collision-free path from an initial to goal waypoint, is inherently intractable for vehicles with many degrees of freedom. Recently, a number of sample-based motion planning techniques [2, 4] have been proposed which tradeoff completeness in the planning algorithm for tractability and efficiency in most cases. The purpose of this paper is to show how one can incorporate dynamic reconfigurability in such motion planners on a deployed and fully operational UAV by integrating the motion planner with the control kernel of the UAV in a novel manner with little modification of the original algorithms. Integrating both high- and low-end functionality seamlessly in autonomous architectures is currently one of the major open problems in robotics research. UAV platforms offer an especially difficult challenge in comparison with ground robotic systems due to the often tight time constraints present in the plan generation, execution and reconfiguration stages in many complex mission scenarios. It is the intent of this paper to show how one can leverage sample-based motion planning techniques in this respect, first by describing how such integration would be done and then empirically testing the results in a fully deployed system. The techniques and solutions described are generic in nature and suitable for platforms other than the one used in this experimentation. An important point to note is that to our knowledge we are the first to use these sample-based motion planning techniques with fully deployed UAVs.

2 The Path Planning algorithms

The problem of finding optimal paths between two configurations in a high-dimensional configuration space such as a helicopter is intractible in general. Sample-based approaches often make the path planning problem solvable in practice by sacrificing completeness and optimality. Two implementations of such algorithms are used in the WITAS [1]system: probabilistic roadmaps (PRM) and rapidly exploring random trees (RRT).

The PRM planner is an extended version of the standard algorithm proposed in [2]. It creates a roadmap in the offline stage based on the 3D model of the environment which is used later during the query phase. Our extentions to the original algorithm deal with problems of nonholonomic constraints and delayed constraints handling.

The use of rapidly exploring random trees (RRT) provides an efficient motion planning algorithm [4] that constructs a roadmap online rather than offline (PRM). After the roadmap is created, the remaining steps in the algorithm are the same as with PRMs.

The mean planning time in the current implementation for both planners is below 1000 ms and the use of runtime constraints do not noticeably influence the mean [5].

---

[1] WITAS is an acronym for the Wallenberg Information Technology and Autonomous Systems Lab which hosted a long term UAV research project (1997-2004).

3 Path execution mechanism

The standard path execution scheme in our architecture [3] for static operational environments is depicted in Fig. 1. A UAV mission is specified via a task procedure (TP) in the reactive layer of our architecture, (perhaps after calling a task-based planner). A TP is a high-level procedural execution component which provides a computational mechanism for achieving different robotic behaviors.

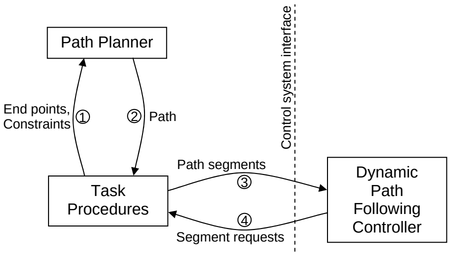

Figure 1: Plan execution scheme

For the case of flying to a waypoint, an instance of a navigation TP is created. First it calls the path planner service (step 1) with the following parameters: initial position, goal position, desired velocity and additional constraints.

If successful, the path planner (step 2) generates a segmented cubic polynomial curve. Each segment is defined by start and end points, start and end directions, target velocity and end velocity. The TP sends the first segment (step 3) of the trajectory via the control system interface and waits for the *Request Segment* event that is generated by the controller. At the control level, the path is executed using a dynamic path following controller [1] which is a reference controller that can follow cubic splines. When a *Request Segment* event arrives (step 4) the TP sends the next segment. This procedure is repeated (step 3-4) until the last segment is sent. However, because the high-level system is not implemented in hard real-time it may happen that the next segment does not arrive to the control kernel on time. In this case, the controller has a timeout limit after which it goes into safety braking mode in order to stop and hover at the end of the current segment. The timeout is determined by a velocity profile, current position and current velocity. In practice the time between receiving *Request segment* event and the controller timeout is large enough to reconfigure the path using the standard path planner. The updated segments are then sent to the DFP controller transparently.

Reconfiguration is triggered by the event created when new forbidden regions are added or deleted. There are several different policies that can be used during the reconfiguration step (Fig. 2):

Policy 1

Reconfiguration is done from the next waypoint (start point of the next segment) to the end point. This implies longer planning times and eventual replacement of collision-free segments.

Policy 2

Segments up to the colliding one are left intact and reconfiguration is done from the last collision-free waypoint to the end point.

Policy 3

Replanning is done only for colliding segments. The helicopter will stay as close to the initial path as possible. Note that each of these policies progressively re-uses more of the plan originally generated, thus cutting down on planning times. We are currently experimenting with all three policies.

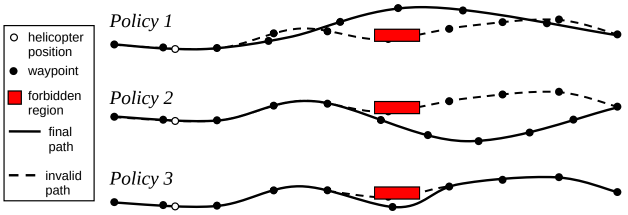

Figure 2: Replanning policies.

4 Experimental results

In our experiments we have used both the PRM and the RRT planner with a TP that implements the first policy. Forbidden regions were randomly added by the ground operator during the flight. Typical helicopter velocity was up to $7m/s$ and the total path length up to $500m$. The results of the experiments shown that the time window between sending two successive segments is generally greater than four times the amount of time required to generate full plans using either the PRM or RRT planners.

References

[1] G. Conte, S. Duranti, and T. Merz. Dynamic 3D Path Following for an Autonomous Helicopter. In *Proc. of the IFAC Symp. on Intelligent Autonomous Vehicles*, 2004.

[2] L. Kavraki *et al.* Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces. *Proc. of the IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[3] P. Doherty *et al.* A Distributed Architecture for Autonomous Unmanned Aerial Vehicle Experimentation. In *Proc. of the Int. Symp. on Distributed Autonomous Robotic Systems*, pages 221–230, 2004.

[4] J. J. Kuffner and S. M. LaValle. RRT-connect: An Efficient Approach to Single-Query Path Planning. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 995–1001, 2000.

[5] P-O Pettersson. Using Randomized Algorithms for Helicopter Path Planning. *Lic. Thesis Linköping University.* To appear, 2005.