

Flexible Trajectory Planning Framework using Optimal Control for Rotary and Fixed-Wing Aircrafts Mission Planning and Target-Pursuit

Rafael Fernandes de Oliveira, Falk Schubert, and Viktor Potyka
EADS

ABSTRACT

Recent advances in small unmanned air vehicles (UAV) lead to robust on-board stabilized platforms ready to use for real-world problems. As a result, many different autonomy functions have been demonstrated, which allow controlling the UAVs at high level. However, the great variety of platforms also poses new challenges when adapting these autonomy functions to new platforms. For instance modifying a trajectory planning algorithm, which was designed for a rotary-aircraft with a moving camera, to work on a fixed-wing aircraft with a static camera is not a trivial task. Often such algorithmic solutions are tailored so specifically to a certain platform that it becomes very complicated to reuse algorithms. This results in a variety of many different approaches trying to solve the same task. We therefore encourage designing algorithms for UAVs autonomy function to be more generic. As an example, we focus on the task to autonomously follow a moving ground object using an UAV. We present two trajectory planning methods: a baseline method suitable that has been developed just for one specific platform, and our proposed flexible optimal-control-based method suitable for a variety of platforms. The original waypoint-based flight planning approach has been evaluated in real world experiments, and while it was capable of dealing with the missions it was designed for, its limitations impeded further development into other UAV platforms. Therefore, in this work, we evaluate the new optimal control based approach in simulated scenarios similar to those used in the original missions, demonstrating the superiority and greater flexibility of the later one in a set of different scenarios and two distinct UAVs models, one being a fixed wing aircraft and the other a helicopter.

CITATION: Fernandes de Oliveira, R., Schubert, F., and Potyka, V., "Flexible Trajectory Planning Framework using Optimal Control for Rotary and Fixed-Wing Aircrafts Mission Planning and Target-Pursuit," *SAE Int. J. Aerosp.* 6(2):2013, doi:10.4271/2013-01-2264.

INTRODUCTION

Unmanned aerial vehicles (UAVs) are becoming more robust and readily available. In order for these platforms to be of practical use, they typically have a high degree of autonomy. For instance, flight-controls are based on waypoints or semantic tasks like "fly loiter pattern". Also, as the demand for cooperative missions between UAVs and other vehicles (e.g. unmanned ground vehicles, humans or other UAVs) is increasing. For example, UAVs can be used to provide an aerial view of an unmanned ground vehicle (UGV) to give the operator of the UGV a better overview of the mission area [32]. Furthermore, UGVs often are often surrounded by landscape (e.g. hills, trees) prohibiting a stable line-of-sight for communication with the ground control station. UAVs can be used in such cases as a communication

relay point. Similarly, GPS signals are often degraded (e.g. due to multi-path effects) by landscape as well. UGVs therefore often have to maneuver locally without geo-coordinates. In order to constantly provide geo-locations to the UGVs, UAVs can be used to visually localize the UGV (via geo-referencing) and transmit this geo-position to the UGV in case it has lost its GPS signal. In all these applications the UAVs have to be able to automatically follow a ground-vehicle based on vision.

Several challenges have to be met to realize such a high level of autonomy. First, the UAV has to be able to visually detect and track the target. Second, the target has to be geo-referenced in order to provide geo-locations which are input to path planning algorithms. Third, a path planning module is needed which calculates flyable trajectories in real-time that allow the pursuit of the UGV. In this paper we present the

originally implemented solution to such a follow mission and demonstrate the functionality in simulation as well as real flight tests.

However, like most research in the field of autonomous path planning, this basic follow mission planning is tailored to a specific platform (i.e. rotary aircraft with waypoint-based autopilots). Little research genuinely proposes path planning algorithms that are generic enough to be easily adaptable to different platforms (e.g. rotary and fixed-wing aircrafts) and various autopilots (e.g. waypoint-based and vector-based), while still taking into account platform limitations and optimality of the solution. We therefore propose a new path planning algorithm which is adaptable to fixed-wing and rotary-wing platforms and show the superiority compared to the basic solution in simulation.

We first describe the two solutions for such a follow mission. Starting with a brief summary of the hardware platform, we outline next the marker detection employed to generate geo-coordinates of the target. This image processing module is used for both, the basic path planning and the optimization-based one. After, we present details of the basic path planning to follow a moving object. We highlight limitations of this basic solution and propose a more advanced approach using optimal control. Finally, we present simulation results, which show the greater flexibility of the optimization-based approach, followed by our conclusions.

Platform Description

The UAV platform is based on a Vario Acrobatic combustion engine powered helicopter. In [Figure 1](#) the most important facts about the platform are summarized. For automatic flight control we employ a commercial autopilot of weControl which allows only purely waypoint-based control. The platform also contains a USB camera (uEye UI 2220SE C CCD) which delivers images with a resolution of 768×576 pixels at a rate of 30 Hz. The camera was chosen mainly because it does not have the very popular rolling-shutter, which would introduce a considerable amount of noise into the captured image due to strong vibrations and motion of the camera. The camera was setup to have the lowest possible exposure time to further reduce the influence of motion blur.

Furthermore the camera is mounted on a pan-tilt unit as shown in [Figure 2](#). The platform also contains an onboard-computer with a dual-core Intel Atom D510 processor (see [Figure 3](#)), which is suitable to run basic autonomy functions like waypoint-planning, simple image processing like marker detection and core communication like operating the command-control datalink. Finally the platform has a number of sensors (see [Figure 2](#)) to estimate the position and orientation of the helicopter as well as the camera platform which allow to geo-reference objects detected in the camera image. With this information at hand it is possible to compute new waypoints like for following a moving target.

All state information of the UAV and mission, like position, orientation, detected markers, etc. are reported to the operator and logged in a ground control station as shown in

[Figure 4](#). This tool was used to evaluate the performance of the path planning.

Vario Benzin Acrobatic	
Weight	7.5 kg
Cruise Speed	10 m/s
Max Speed	20 m/s
Payload	5 kg
Max. Flight Time	1 h
Flying Since	Oktober 2010



Figure 1. Facts about the UAV

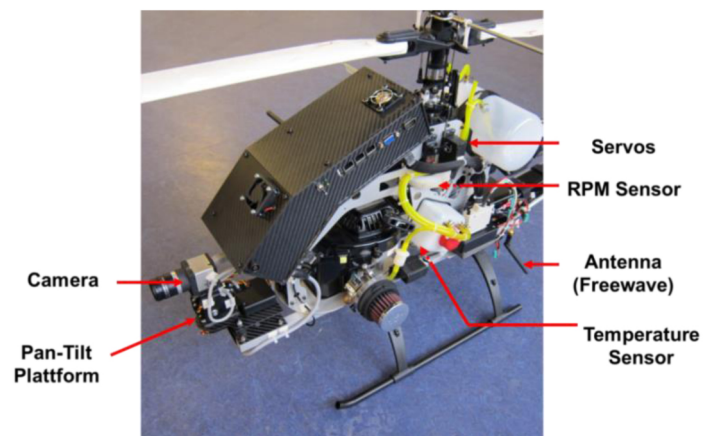


Figure 2. Hardware-Setup of the platform

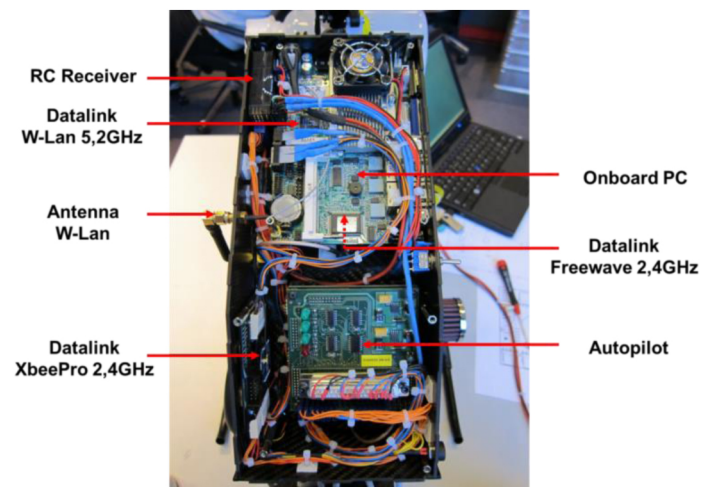


Figure 3. Onboard-Computer

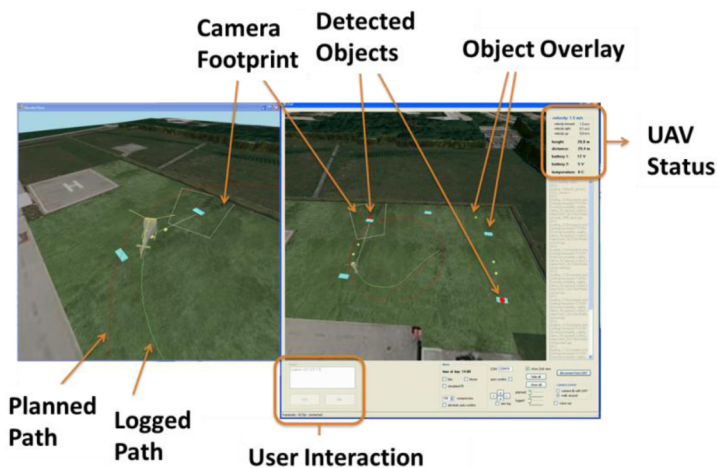


Figure 4. Ground-Control Station

Marker Detection

Generic learning-based object detection [33] allows detecting object instances of generic object categories (e.g. a car) within images. However, despite the great effort and recent advances in this research field, the detection performance (e.g. detection rates) is still limited to a small range of viewpoints (e.g. only 360°-range azimuth and no variation in elevation). This is sufficient to provide good detections for ground-based cameras. However, for aerial imagery, variations in elevation (e.g. 90°-range elevation) also have to be considered. Furthermore, the computational resources are still very demanding (e.g. powerful multi-core processors are required) and limit the processing speed. Typically, only a few frames (e.g. 1-10 frames per second) can be processed per second. For autonomous UAV missions, path planning requires very frequent updates about the position of the objects (e.g. 20-30 frames per second). Furthermore these planning algorithms have to run onboard to allow true autonomous navigation. However, these onboard computational resources are typically not very powerful to fulfill the requirements for power consumption, temperature and size. All these factors prohibit the use of generic learning-based object detection.

In order to still use purely vision-based object localization, we use cooperative targets. In other words, we place markers on the objects to be followed. These markers are then detected from a wide variation of viewing angles and distances. We use a publicly available detection library called ARToolkitPlus [34]. Figure 10 shows some samples of the markers that the toolbox can detect. The markers display a binary pattern, which store a unique identification number. This allows filtering out false detections as the chance that a false marker is confused with a true identification number is very low. In order to estimate the pose correctly the camera needs to be calibrated and its intrinsic parameters (e.g. lens distortions) need be estimated. Every detected marker consists of three elements which are used for the path

planning: ID of the marker, detection confidence and the estimated relative pose to the camera. In Figure 6, examples of real detections are shown. Since the marker detection is computationally not as demanding as the generic learning-based object detection, it runs at 20 Hz on the not so powerful onboard.

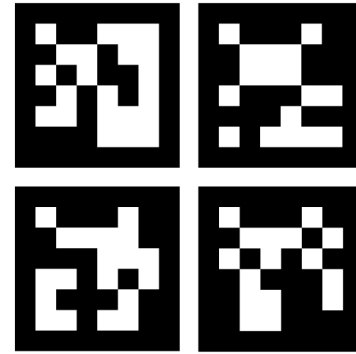


Figure 5. Marker samples of the AR Toolkit Plus.

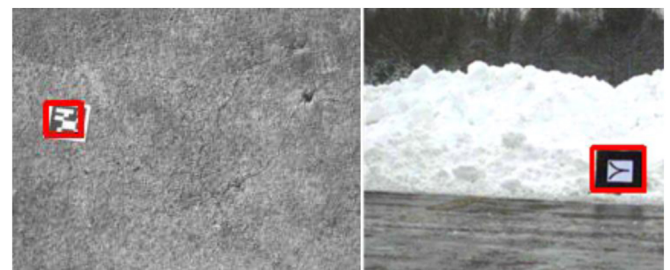


Figure 6. Example detections

Baseline Waypoint-based flight planning

The original trajectory planning method, suitable for waypoint-based autopilots, has been implemented using experimental data to shape the possible trajectories for the UAV platform. This choice of algorithm was suitable for the original goal of this flight planner implementation, which was to provide optimal trajectories for marker tracking and exploring an unknown region. The first problem found when using this waypoint-based autopilot was that there is no guarantee of orientation (see Figure 7) when flying to a given waypoint. This caused problems when implementing the image-based target tracking. Therefore, the first improvement over the waypoint-based flight planning was to add a way of guaranteeing orientation when reaching a given waypoint, this made possible through the use of the Dublins path method, in which two main waypoints are connected by a series of intermediate waypoints, as shown in Figure 8, so that the UAV reaches the second waypoint in the desired attitude. This approach has been used successfully in our missions.

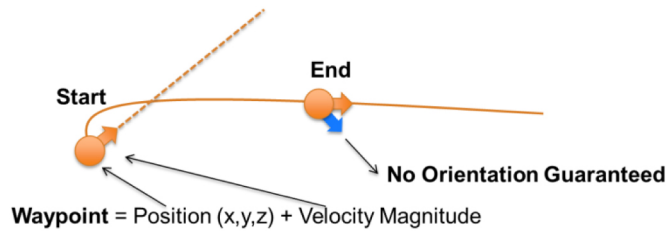


Figure 7. Available autopilot on the platform does not guarantee orientation

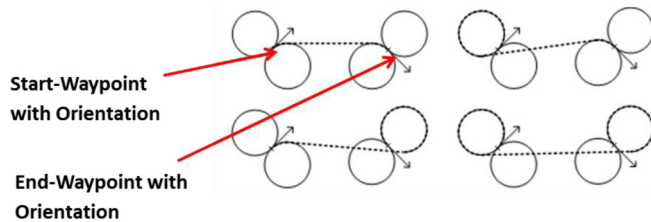


Figure 8. Waypoint-based flight planning with guarantee of exit orientation

Follow Mission

In the previous section, an extension was presented which guarantees a specified heading for a waypoint, since the standard waypoint of the autopilot of the platform does not allow specifying this final heading. With the extended waypoints a basic path planning solution for following a moving object, is to constantly monitor the movement of this object, extrapolate its motion based on a motion model and place an oriented target waypoint at the end of the extrapolated trajectory. While flying towards the current target waypoint, the extrapolation has to be constantly updated with the current motion of the object to adapt to changes of the true trajectory compared to the extrapolated one. We extrapolate the motion of the ground vehicle assuming a linear motion model with a constant velocity. In [Figure 9](#), this planning is illustrated. On the top, the UAV analyzes the motion of the object (the red dots indicate the geo-referenced positions of the marker on the ground). After an analyzing phase (e.g. 20 detections) the linear motion is fitted to the detections resulting in a target direction (i.e. the heading is indicated by the yellow arrow). A target waypoint is placed at infinity along this linear trajectory (i.e. yellow arrow). Another example for this motion extrapolation is shown on the bottom part of [Figure 9](#). Updating new waypoints and loading them into the autopilot, however requires a brief stop of the UAV during which it is hovering. Hence, very agile movements of the ground vehicle which significantly differs from the simple linear motion model, leads to frequent stops of the UAV. This in turn increases chances that the ground object is lost. An advantage of our platform however is the pan-tilt-camera which allows quickly following the ground object even when hovering.

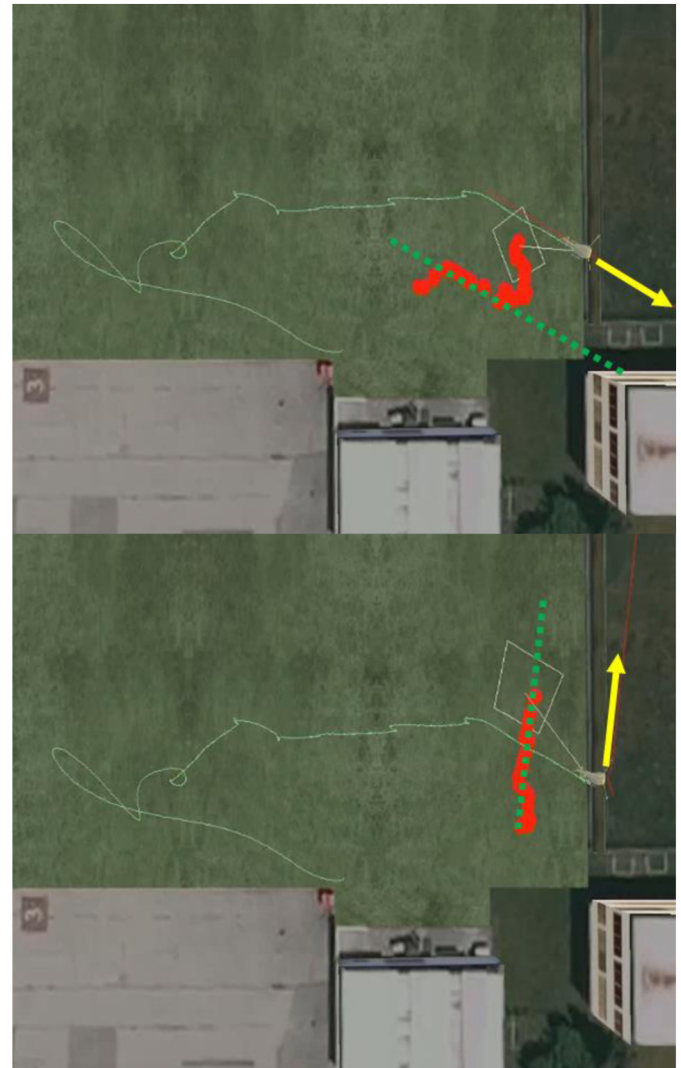


Figure 9. Extrapolation of the marker-tracking motion measured during flight test using a linear motion model

We therefore have a redundancy in the way of how to follow a moving object. While the object is still close to the UAV, we can either follow the object with the UAV or with the camera. Since the UAV is much faster than the ground vehicle, it is easy to catch up even when the ground object has moved away from the UAV. To combine both ways of visually following the object, the basic path planning alternates between following the target using its pan-tilt camera and estimating a pursuit path using motion extrapolation explained above. This follow logic is schematically illustrated in [Figure 10](#). When the object is close to the UAV, we use the camera to follow the object. Only after the viewing angle becomes to oblique (which is equivalent to an increasing distance between the UAV and target object), the motion extrapolation is used to compute a new pursuit trajectory.