Model-Based Automated Flight Path Planning for an Ultralight Aircraft*

Belén Santos León[1], Jane Jean Kiam[2], and Axel Schulte[2]

[1] Technical University of Munich, Germany b.santos@tum.de
[2] Bundeswehr University Munich, Germany {jane.kiam,axel.schulte}@unibw.de

Abstract. Flight guidance is tending towards more automation in order to increase flight safety and efficiency. An important focus is the development and integration of an automated flight path planning system able to assist the pilot in unexpected emergencies. This is challenging due to the dependance on numerous internal and external constraints, but essential for a single-pilot aircraft. We propose the use of a model-based flight path planner in this paper that is based on PDDL+ (Planning Domain Definition Language). We leverage the ability of the language to model hybrid domains, which combine discrete and continuous behavior, enabling hence the modelling of aircraft dynamics as well as discrete events.

Keywords: Flight path planning · Model-based planning · Automated guidance system · 3D trajectory generation · PDDL+.

1 Introduction

Over the last years, the increase of automation in aviation has been driven by several factors, including the necessity to reduce the environmental impact of air transport, to cope with a higher demand and the need to increase safety and efficiency, to decrease considerably the accidents caused by human error, to minimize the pilot's workload and to reduce costs [3, 5].

Unmanned aviation has motivated substantially the inclusion of automation in flight guidance, but this can also be beneficial for other air vehicles, such as ultralight aircraft. Ultralight aviation is commonly practiced for sport or recreational purposes by non-professional pilots, which may not react properly under contrary circumstances such as severe weather or mechanical failure. This fact, together with the single-pilot configuration of many ultralight aircraft, where all the workload relies on a single person, may lead to fatal accidents [6], many of them could be avoided with properly designed automation in flight guidance.

A crucial task in the development of an automated flight guidance system is the integration of a path planning system capable of generating feasible flight trajectories while considering the aircraft's characteristics as well as the external environment. Most techniques in flight path planning are only geometric-based, leading often to flight paths that are non-feasible [11,13]. However, an automated planner requires a more complex planning approach like the offered by PDDL+ planners [7], which are derived mainly for general AI planning, but are capable of considering continuous processes and discrete events. In [10], it is proven for the first time that PDDL+ planners are also capable of performing kinodynamic flight path planning in complex environments.

In this paper, a flight path planning system based on PDDL+ is presented together with the modelling of the planning problem as an extension of the model presented in [10]. The developed planning system is integrated together with a flight control module into a flight guidance system. The trajectories calculated for several scenarios under normal and abnormal conditions are tested with the flight simulator X-Plane [12], a powerful simulation software that considers a realistic model of an ultralight motorized aircraft with a complete global scenery and huge set of customization parameters for the aircraft and its environment. We demonstrate that the calculated flight paths are viable.

2 Flight Path Planning in PDDL+

PDDL+ is an extension of PDDL, a standard language for modelling planning domains [7]. In PDDL+, hybrid domains are supported, i.e., systems which exhibit both continuous and discrete behavior, providing hence an attractive alternative for flight path planning problems.

This work exploits results obtained in [10], in which a flight path planner was developed for the automated 2D trajectory generation of a High-Altitude Pseudo-Satellite (HAPS). Although a different kind of platform, the flight dynamics modelled in the previous work for a fixed-wing aircraft can be exploited, and extended to include 3D flight plans for an ultralight aircraft, exhibiting also the benefit of using a model-based flight path planner.

The planning task in PDDL+ is formulated in two separate files: the domain file and the instance or problem file (see Fig. 2). In the domain file, a series of operators describing the dynamics of any fixed-wing aircraft and how they are affected by internal or external factors. The instance file includes initial and goal state, as well as all variables which uniquely define the target aircraft and all parameters that describe the current planning scenario or environment, i.e. aircraft's current conditions, weather forecast or the positions of obstacles.

The kinematic model of the ultralight airplane to be encoded in PDDL+, includes the influence of the wind speed and it is formulated under the assumption of spherical Earth:

$$\dot{\lambda}(t) = \frac{u_{wind}(t) + v_{TAS}(t)\cos\theta(t)\sin\chi(t)}{(R + h(t))\cos\phi(t)}, \tag{1}$$

$$\dot{\phi}(t) = \frac{v_{wind}(t) + v_{TAS}(t)\cos\theta(t)\cos\chi(t)}{R + h(t)}, \tag{2}$$

$$\dot{h}(t) = v_{TAS}(t) \sin \theta(t), \tag{3}$$

where λ, ϕ and h are the WGS84 coordinates of the aircraft: longitude, latitude and altitude, R is the Earth's mean radius, χ and θ are the yaw and pitch angles, v_{TAS} is the aircraft's True Air Speed and u_{wind}, v_{wind} are the wind components on the horizontal plane.

Fig. 1 depicts some of the included code to generate 3D trajectories. The action to increase pitch rate is done analogously to decrease it; with the calculated pitch rate, the new pitch angle is determined and after that, the altitude is updated. In addition, climb rate is calculated and constrained to be maintained within its minimum and maximum values, which are platform-specific.

```
(:action increase_pitch_rate
 :parameters (?vehicle -vehicle)
 :precondition (< (pitch ?vehicle) (max_pitch ?vehicle))
 :effect (increase (pitch_rate ?vehicle) (delta_pitch_rate ?vehicle))
)

(:process increase_pitch
 :parameters (?vehicle -vehicle)
 :precondition ()
 :effect (increase (pitch ?vehicle) (* #t (pitch_rate ?vehicle)))
)

(:process increase_altitude
 :parameters (?vehicle -vehicle)
 :precondition ()
 :effect (increase (altitude ?vehicle) (* #t (* (speed ?vehicle) (sin (pitch ?vehicle)))))
)

(:process calc_climbrate
 :parameters  (?vehicle -vehicle)
 :precondition ()
 :effect (assign (climb_rate ?vehicle)(* (speed ?vehicle)(sin(pitch ?vehicle))))
)

(:constraint climb
 :parameters (?vehicle -vehicle)
 :condition (and (<= (climb_rate ?vehicle) (max_climb_rate ?vehicle))
                 (>= (climb_rate ?vehicle) (min_climb_rate ?vehicle)))
)
```

Fig. 1: PDDL+ Formulation to calculate pitch angle, altitude and constraint climb rate.

3 System Description, Preliminary Test and Validation

In this section, the flight guidance system developed for the validation of the proposed flight path planner is presented. Preliminary tests using simple scenarios such as a climb phase, a cruise phase, etc., are performed to provide an overview of the validity and the quality of the plan computed by the planner. Moreover, the results of the test will help to detect system faults and areas for future improvement. The system architecture is depicted in Fig. 2, with the following subsection providing more details on each subsystem.

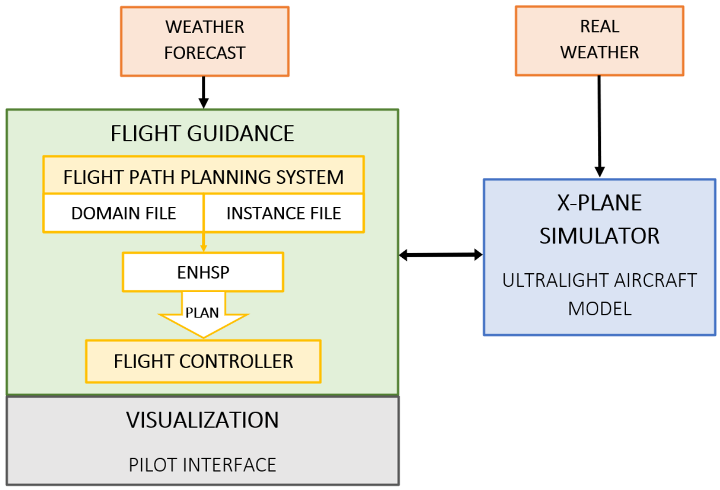


Fig. 2: Validation system architecture

3.1 System Architecture

As shown in 2, a weather plug-in is included to extract weather forecast information online from the National Oceanic and Atmospheric Administration (NOAA) [1]. The forecast defines 3-dimensional weather conditions at different altitudes and divides the airspace horizontally into a square grid of 0.25x0.25 degrees in longitude and latitude. The key weather parameters to consider are 1) the wind condition, relevant to determine the translational movement of the aircraft, and 2) the cloud coverage, so that areas with expected high coverage clouds are defined in the planner as obstacles which must be avoided to comply with Visual Flight Rules (VFR). While weather forecast information is used for planning the flight paths, weather nowcast information is used for simulating the planned paths.

An independent planner is required to solve the planning problem modelled in PDDL+ as described in Section 2. In [10], ENHSP (Expressive Numeric Heuristic Search Planner) is tested with favorable results at solving flight path planning problems for a fixed-wing UAV. ENHSP supports PDDL+ semantics and hybrid domains, while building its heuristics on additive interval-based relaxation (AIBR) to accelerate the search. AIBR is a pruning-safe relaxation, i.e. the relaxed problem has no solution only if the non-relaxed problem is not solvable either [14]. This relaxation simplifies numeric planning, especially problematic for complex behaviors, approximating reachable values with an upper and lower bound. ENHSP, compared to other domain-independent planners which also support PDDL+, is capable of solving more efficiently hybrid problems with non-linear numeric operations, as demonstrated in [8]. These properties make ENHSP a suitable planner for our need.

A compact map visualization to display the planned trajectory, the aircraft's current position, and the weather map is also integrated in the system, serving as an interface between the flight guidance system and the pilot in the cockpit.

A proportional-integral-derivative (PID) controller for lateral, longitudinal and velocity control developed in Matlab/Simulink environment which communicates with the flight simulator (X-Plane) via UDP (User datagram protocol) has been developed for automatically guiding the aircraft to follow the planned path [4]. The controller receives the flight plan position and airspeed at each time step and commands the required control surface deflections and throttle input to follow the desired trajectory. The integration step of the controller is set to 0.01 s to be aligned with the data transfer rate from X-Plane, while the execution step set for ENHSP used to solve the planning problem is 1 s. The single-cockpit ultralight aircraft in X-Plane built according to the characteristics of Aerolite 103 is chosen to validate the system. The characteristics of the aircraft used for modelling the planning problem are adopted from from the aircraft's specifications [2] and the aircraft's model designed in X-Plane.

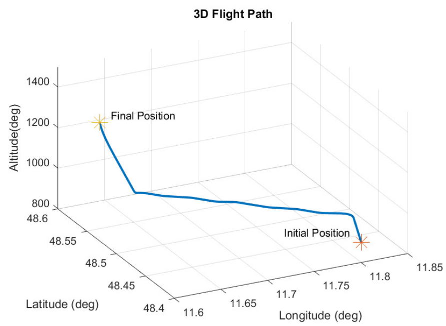
3.2 Results of the Preliminary Tests

In the analyzed first scenario, the planner computes a trajectory for the ultralight aircraft to reach a certain position, requiring to perform several common maneuvers, such as climb, turn, speed variation or fly at constant altitude. External environment consists of constant mild wind and no obstacles. However, more scenarios under less favorable conditions, i.e. with stronger and time-varying wind, forbidden areas or faulty aircraft, will be tested in the future for the system validation. The calculated flight plan for this initial scenario, depicted in Fig. 3a, is then compared with the simulated trajectory. Three key aspects from the simulation are analyzed here:

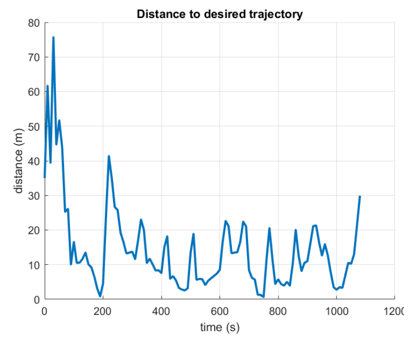
- *Position control*: Figs. 3b and 3c show the lateral distance and altitude discrepancy with respect to the desired trajectory at each time instant. The discrepancy does not exceed 20 m at most segments of the flight path, and reaches values over 40 m at some points.
- *Speed control*: It can be observed in Fig. 3d that the desired velocity is reached and maintained with high accuracy at most time along the trajectory. Only some fluctuations are present at the beginning of the flight path as well as at the end, when the climbing segment starts.
- *Plan duration*: The obtained plan from ENHSP requires a total time of 1071 s to reach the goal position. In the simulation, it takes 1081 s to reach the closest position to the goal state.

3.3 Discussions

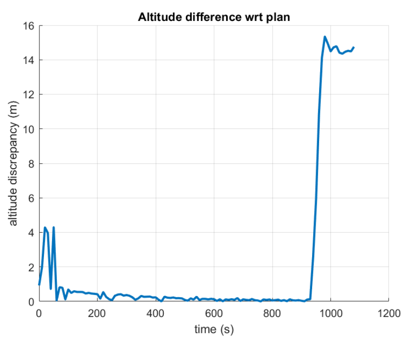
The presented results, exhibit that the obtained PDDL+ plan is feasible as the simulated plan is close to the desired trajectory. Mostly importantly, this



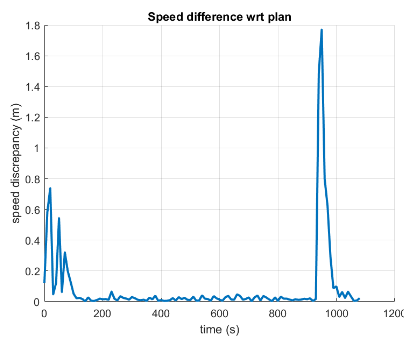
(a) 3D Flight Path



(b) Lateral distance to flight plan



(c) Altitude discrepancy



(d) Speed Discrepancy

Fig. 3: Test Results: Discrepancies between computed plans and simulated results (using a realistic flight simulator).

confirms one of the benefits postulated in [10] on the reusability of a model-based path planning method, i.e. the system description encoded in the domain file for a high-altitude unmanned aircraft can be reused for another fixed-wing piloted aerial platform flying at much lower altitudes, while the platform-specific parameters are defined in the instance file. However, some insufficiencies of the system were identified in the preliminary tests.

The peak of 1.8 m/s observed in the speed discrepancy plot (Fig. 3d) is the result of a sudden speed reduction required for a climbing phase. This overshoot and the initial disturbances, indicate a flaw in the speed controller. This fault affects the capacity of the controller to follow the desired trajectory, leading to worse distance and altitude discrepancy values on those phases, where the speed diverges more from the desired value. These parameters are also influenced by the longitudinal and lateral control subsystems, because of the lateral-longitudinal coupling of the aircraft's dynamics. Note that these discrepancies do not indicate a lack of plan quality with respect to the plan feasibility.

4 Conclusion and Future Work

The presented flight planner is an extension of a 2D trajectory planner which adds a third dimension in the trajectory generation. The independent formulation of PDDL+ signifies that, if properly defined, the domain file could be invariant and transferable to any other aircraft with equivalent dynamics. We demonstrated, with this work, this aspect by exploiting the domain file encoding the system behavior of a fixed-wing high-altitude unmanned aircraft for an ultralight single-pilot fixed-wing aircraft. Hence, the task of implementing the flight planner is considerably simplified, i.e. it is only necessary to encode the platform-specific parameters in the instance file, e.g. cruise speed, yaw rate, pitch angle, etc.

Further development of the planner will focus on the augmentation of fault-tolerance capabilities, so that the planner will be able to identify abnormal conditions and act accordingly, providing a flight path adapted to the unexpected circumstances, such as extreme weather conditions, pilot's incapacity or mechanical failure. For this purpose, unsupervised learning will be exploited to classify the behavior of the platform in each adverse situation, divising hence also situation-specific parameters for the platform to be included in the instance files, e.g. airspeed limit during a motor failure, yaw rate in the case of a broken wind, etc. The application of such AI techniques will deliver more suitable constraints and aircraft parameters to be defined in PDDL+ and consequently, the calculated trajectories will adapt to each specific flight scenario and aircraft requirements.

Particle Swarm Optimization (PSO) algorithm [9] will be implemented to find optimal gain values. PSO varies the gain values iteratively and reduces the Integral Time Absolute Error until the maximum number of iterations is reached or the error is lower than a predefined value. This optimization algorithm will enhance the robustness and reliability of the controller and therefore, the discrepancies observed between the simulated flight and the calculated flight plan will be substantially reduced.

References

1. NOMADS-NOAA Operational Model Archive and Distribution System, https://nomads.ncep.noaa.gov/
2. Specifications - Aerolite 103 Sales, https://www.flyaerolite.com/specifications/
3. Billings, C.E.: Aviation Automation: The Search for a Human-Centered Approach. CRC Press (2018)
4. Bittar, A., Figuereido, H.V., Avelar Guimaraes, P., Correa Mendes, A.: Guidance Software-in-the-Loop Simulation Using X-Plane and Simulink for UAVs. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 993–1002. IEEE (2014)
5. Clarke, J.P.: The Role of Advanced Air Traffic Management in Reducing the Impact of Aircraft Noise and Enabling Aviation Growth. Journal of Air Transport Management **9**(3), 161–165 (2003)

6. De Voogt, A., Chaves, F., Harden, E., Silvestre, M., Gamboa, P.: Ultralight Accidents in the US, UK, and Portugal. Safety **4**(2), 23 (2018)
7. Fox, M., Long, D.: Modelling Mixed Discrete-Continuous Domains for Planning. Journal of Artificial Intelligence Research **27**, 235–297 (2006)
8. Franco, S., Vallati, M., Lindsay, A., McCluskey, T.L.: Improving Planning Performance in PDDL+ Domains via Automated Predicate Reformulation. In: International Conference on Computational Science. pp. 491–498. Springer (2019)
9. Giriraj Kumar, S., Jayaraj, D., Kishan, A.R.: PSO Based Tuning of a PID Controller for a High Performance Drilling Machine. International Journal of Computer Applications **1**(19), 12–18 (2010)
10. Kiam, J.J., Scala, E., Ramirez Javega, M., Schulte, A.: An AI-Based Planning Framework for HAPS in a Time-Varying Environment. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 30, pp. 412–420 (2020)
11. Lekkas, A., Dahl, A.R., Breivik, M., Fossen, T.I.: Continuous-Curvature Path Generation Using Fermat's Spiral. Journal of Modeling, Identification and Control **34**(4), 183–198 (2013)
12. Meyer, A., Laminar Research: X-Plane, https://www.x-plane.com/
13. Piprek, P.: Clothoid Development for a Trajectory System. Master's thesis, Technical University of Munich (2014)
14. Scala, E., Haslum, P., Thiébaux, S., Ramirez Javega, M.: Interval-Based Relaxation for General Numeric Planning. In: Proceedings of the Twenty-second European Conference on Artificial Intelligence. pp. 655–663 (2016)