2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Rapid Trajectory Time Reduction for Unmanned Rotorcraft Navigating in Unknown Terrain

Simon Schopferer, Florian-Michael Adolf

***Abstract*—Onboard and online flight path planning for small-scale unmanned rotorcraft requires very efficient algorithms in order to meet runtime constraints. When flying through a priori unknown environment, rapid replanning is necessary in order to maintain a safe obstacle clearance. The complexity of the planning problem rises vastly when trying to minimize flight time. This is because the helicopter's flight dynamics must be accounted for in order to produce paths which the helicopter can follow safely at high speeds. Building on the success of sampling-based planning algorithms, decoupled approaches that separate the planning of collision free and dynamically feasible flight paths into sequential steps have been shown to solve the problem very efficiently. However, these approaches may produce suboptimal results or may even fail to provide valid flight paths at all. Addressing these problems, we present an approach for planning safe, dynamically feasible and time efficient flight paths using cubic splines. Trying to reduce trajectory time, we consider the influence of path smoothing and refinement measures on the quality of the resulting flight paths and on the reliability of the planning procedure. We present comparative results in a range of benchmark scenarios, including flight through urban terrain, to evaluate the overall mission performance.**

***Index Terms*—UAS, Unmanned Rotorcraft, Trajectory Planning, Online Navigation, Cubic Splines**

NOMENCLATURE

ARTIS	Autonomous Rotorcraft Testbed for Intelligent Systems
MiPlEx	Mission Planning and Execution Framework
a	Acceleration
d_{clear}	Obstacle clearance
d_s	Spline curve deviation
d_{sample}	Maximum sampling distance
d_{sense}	Sensor range
d_{track}	Path tracking error
h	Discretization error
l	Chord length
r	Yaw rate
$\vec{r}$	Position vector (parametric curve)
s	Arc length
S	Cubic spline
V_k	Flight velocity
$V_{k,xy}$	Horizontal component of flight velocity
$V_{k,z}$	Vertical component of flight velocity
X	State space
α_{fov}	Sensor field of view opening angle
γ	Flight path angle
κ	Curvature
τ	Curve parameter

I. INTRODUCTION

At the DLR Institute of Flight Systems the experimental flying platform ARTIS (Autonomous Rotorcraft Testbed for Intelligent Systems) is developed aiming to push the limits of today's autonomous helicopters. The project involves the development of intelligent functions for rotorcraft including vision-based perception of the environment and high level mission planning integrated on a range of small-scale unmanned helicopters. Fig. 1 shows the midiARTIS helicopter, equipped with stereo vision used for collision detection and avoidance, for which the path planning approach presented in this paper has been implemented.

For online navigation through a priori unknown environment, unmanned rotorcraft must be able to react instantaneously to the detection of obstacles. While reactive obstacle avoidance algorithms by themselves are inherently suboptimal and incomplete, it is hard to meet runtime constraints with more complex motion planning algorithms. Decoupling the planning of collision free and dynamically feasible trajectories into sequential planning steps is a practical solution to make the complex planning problem efficiently solvable in order to meet runtime constraints. Furthermore it promotes the flexibility and scalability of the software regarding computational costs. A high degree of modularity may also ease the application of various verification and validation techniques in the development process. However optimality of the resulting flight paths cannot be guaranteed as different constraints and quality criteria might be considered during each planning step. The planner's property of completeness may also be compromised due to interferences that occur, when different constraints are incorporated sequentially.

Setting the focus on these problems, we present a highly decoupled flight path planning approach for unmanned rotorcraft navigating in a priori unknown environment. We combine sampling-based path planning, cubic spline interpolation and path refinement techniques in order to plan safe and time efficient flight paths, that account for the rotorcraft's kinematic constraints. While maintaining the efficiency and flexibility of this decoupled planning approach, we overcome its deficiencies through carefully separating each processing step and its effect on the flight path geometry. The presented approach utilizes the computational efficiency of sampling-based path planning to avoid obstacles and cubic spline interpolation to generate smooth paths. Iterative path refinement is performed in order to maintain the required obstacle clearance of the smoothed path and to account for the current

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

flight velocity at the time of replanning. For this purpose, the path is refined within a local region in order to meet the constraints posed by a simplified kinematic model of the rotorcraft. We present comparative results obtained through high-fidelity simulation of the midiARTIS helicopter's closed-loop dynamics and a generic LIDAR obstacle sensor in a set of benchmark scenarios that were presented in [1]. These benchmark scenarios include avoidance of simple obstacles as well as flight through complex urban terrain.

Figure 1. The unmanned helicopter ARTIS equipped with realtime 3D obstacle detection based on stereo vision.

In the subsequent section related work concerning the problem of efficient flight path planning is presented. Section III gives an overview of previous work integrated with the mission planning framework that we have built upon for the evaluation of our flight path planning approach. In section IV we discuss the path evaluation and refinement techniques that are used in the planning procedure and constitute our approach for trajectory time reduction. Comparative results, obtained through simulation of the benchmark scenarios mentioned above, are presented and discussed in section V. A summary and final conclusions can be found in section VI.

II. Related Work

Various approaches for planning feasible and efficient flight paths and trajectories for unmanned rotorcraft have been presented. The techniques used to tackle the complexity of this planning problem span from conventional path planning and reactive collision avoidance to more complex sampling-based motion planning and model predictive control [2]. The approaches differ especially in the way that the rotorcraft's dynamics are modeled and accounted for in the planning process.

Recently, a lot of research has been done on extending sampling-based motion planning algorithms to consider dynamic constraints. In [3] the well-known RRT*-algorithm is modified to achieve this capability by defining locally optimal steering procedures for specific vehicles to optimally connect state space samples while respecting differential constraints. This is extended in [4] for any controllable system with linear dynamics. In [5] cubic Bézier splines are used to connect two states that were sampled using an approximation of the reachable set and thus guaranteeing an upper bound on the curvature of the spline. These approaches produce trajectories that asymptotically converge to the quality of the local planner that is used to connect each pair of neighboring states. However, the computational complexity limits the applicability of these approaches for small-scale helicopters with their limited on-board resources. The computational cost stem from either having to sample a high dimensional state space and solving two-point boundary value problems to connect each state to the search graph or having to integrate the system's dynamics with sampled control input. Another disadvantage comes directly from accounting for both, obstacle clearance and dynamic feasibility. Such a tightly coupled approach does not allow for efficient multi-query planning using persistent search graphs.

To achieve a better decoupling and reduce computational cost, decreasing the dimensionality of the search space, often sampling only the euclidean space, and accounting for the rotorcraft's dynamics in a sequential smoothing procedure has proven to be a viable approach to obtain high quality solutions [6], [7], [5], [8]. Although most rotorcraft are able to come to a complete stop and therefore can follow flight paths constructed from linear segments, this is unfavorable in terms of mission performance. Therefore, smoothing techniques are necessary to generate flight paths with continuous curvature to allow an energy and time efficient flight. Three problems arise with this hierarchical approach:

1) A trajectory must be generated for a given smoothed path taking into account dynamic constraints. This may be solved through an iterative relaxation of the velocity profile until a feasible trajectory is found as described in [9]. In [10] an algorithm generating minimum snap trajectories for quad-rotors is presented allowing constraints on position, velocity, acceleration and control inputs while minimizing a cost functional. In [11] a feasible velocity profile is obtained for cubic spline curves through integration of a simplified reference model of the midiARTIS helicopter's dynamics after identifying velocity minima based on the curvature and acceleration constraints. This approach was used for velocity profile generation in our planning framework.

2) If the smoothed path deviates too much from the linear path segments, the obstacle clearance may be violated compromising the global planner's property of completeness. This may be solved through iteratively performing collision checks and locally restricting the smoothed path's deviation by inserting support points or adjusting parameter used during interpolation as described in [12], [13], [9]. While this practical approach has been shown to work in most cases, no further considerations on runtime restrictions have been carried out, but instead using linear segments as a fallback mechanism was proposed to achieve realtime capability. Also, the effects of restricting the smoothed path's deviation from the linear path on the mission performance were not examined.

3) When replanning during flight, non-holonomic constraints may arise due to the current flight velocity that limits the set of reachable configurations until the helicopter can

come to a complete stop. This will be referred to as the online planning problem. In [9] this problem is approached by moving the start point of the planned path away from the vehicle to a point beyond the minimum stop distance. Additionally, waypoints within 1.5 times of the minimum stopping distance that are in the general direction of travel are either dropped or moved out of this region. The transition to the new trajectory within this local region is not explicitly planned but left to the path-tracking controller. A different approach is presented in [14]. At flight velocities of up to $10\,\mathrm{m/s}$ obstacle avoidance is performed in a reactive manner. The nominal flight path, which is replanned at fixed intervals, serves only as a coarse orientation for the reactive obstacle avoidance system. In [12] an extrapolated vertex ahead of the helicopter is used as an additional support point to control the shape of the cubic spline curve used to interpolate a linear path. The vertex is placed in a distance equal to the minimum stopping distance of the helicopter and within its dynamically reachable set to ensure a smooth transition from the current state into the general direction of the linear path. This approach provides a way to explicitly plan flight paths without relying on reactive algorithms. However, it cannot guarantee that these flight paths comply with dynamic constraints because it relies on heuristically determined parameters to account for the helicopter's dynamics.

III. Background

The work presented in this paper builds upon and extends the mission planning and execution framework (MiPlEx) which has been developed within the ARTIS project. It is a flexible mission planning framework that is capable of handling many different mission scenarios. Since different hardware platforms are used on a number of helicopters with different payloads within the ARTIS project, scalability of the software is an important requirement, which motivates us to focus on decoupled planning approaches that allow a modular integration of new algorithms. The procedure of planning a mission with MiPlEx includes combinatorial task planning, sampling-based path planning, path refinement and velocity profile generation. The resulting trajectory is executed using a path tracking controller that feeds velocity commands to a low level flight control system. In this section an overview is given of the modules and algorithms that were integrated in previous work and are relevant to our approach for trajectory time reduction presented in subsequent sections.

A. Sampling-based Global Planner

As a global planner that is capable of finding collision free paths from the helicopter's current position to the goal point in the presence of obstacles, we use the roadmap-based planner presented in [15]. To improve efficiency, the helicopter is approximated by a sphere of fixed radius for collision checking, which allows to perform the path search in the euclidean space considering only a subset of the helicopter's higher dimensional state space X. This is a particularly useful simplification as any straight line in the euclidean space with a large enough obstacle clearance d_{clear} can already be considered a valid flight path for a rotorcraft that flies at very low speeds close to the state of hovering. During an offline initialization phase, the search space is sampled using a quasi-random sampling technique and collision checks for the edges between neighboring nodes are performed. The search graph is persistent allowing for efficient incremental updates and multi-query path searching as the helicopter flies through initially unknown terrain. Fig. 2 shows an example of a roadmap search graph in urban terrain.

The graph search algorithm D*-lite is used to produce optimal paths in respect to a cost function that must be specified in order to determine the cost of traversal of any of the search graph's edges. As the focus of this work is set on minimizing trajectory times, the euclidean distance is used as a cost function approximating the traversal time without having to consider system dynamics. An additional penalty on vertical maneuvers is added to the edge costs in order to account for the vertical velocity limit that is used to avoid dangerous flight conditions. Another important feature in respect to motion safety is the limitation of the edge slope and thus the absolute value of the flight path angle γ, which is done during the initialization phase of the roadmap to ensure that the flight path ahead of the helicopter always lies within the current sensor field of view.

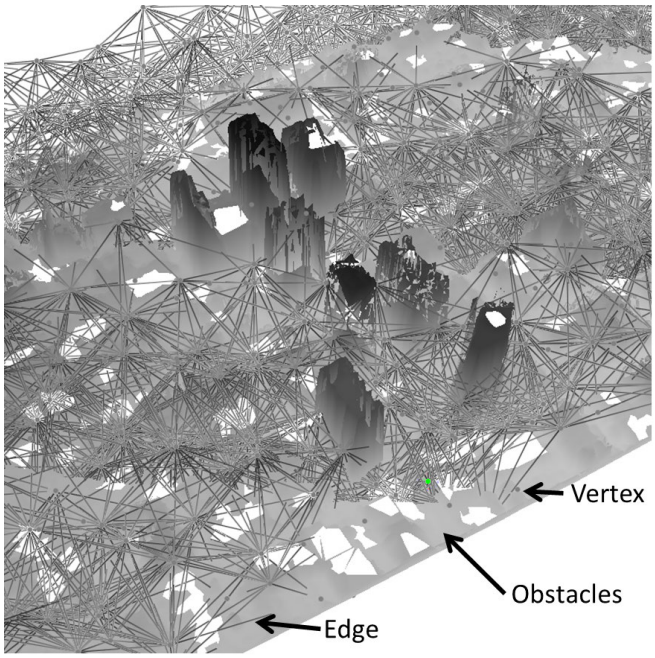

Figure 2. Example of a quasi-random roadmap in urban terrain

B. Parametric Cubic Spline Interpolation

In [11] parametric cubic splines were proposed for the interpolation of linear paths in order to minimize trajectory time. Cubic splines are two times continuously differentiable and of all functions $G \in C^2\,[a, b]$ that share this property on a given interval $[a, b]$ and interpolate a given set of support points, the cubic spline $S\,(\tau)$ minimizes the integral of the second derivative [16]:

$$\int_a^b G''(\tau)^2 \, \mathrm{d}\tau > \int_a^b S''(\tau)^2 \, \mathrm{d}\tau \, . \tag{1}$$

For the interpolation of a set of three dimensional support points with cubic splines, a parametrization must be chosen to map each segment between two sequential points to an interval of the dimensionless curve parameter τ. A continuous space curve

$$\vec{r}(\tau) = S_x(\tau)\,\vec{e}_x + S_y(\tau)\,\vec{e}_y + S_z(\tau)\,\vec{e}_z \tag{2}$$

is then obtained with a cubic spline for each dimension. The choice of parametrization has a great influence on the shape of the resulting curve. For an arc length parametrized curve the curvature κ equals the norm of the second derivative. Therefore, (1) can be considered as a minimum total curvature property for arc length parametrized cubic splines. Because the arc length is initially unknown, we use a chord length parametrization, which yields a good approximation in order to obtain a close-to-minimum curvature spline.

C. Velocity Profile Generation and Tracking

In order to track a nonlinear flight path with an upper bound on the total body fixed acceleration, a_{max}, the total velocity along the flight path, $V_{k,max}$, has to be restricted in order to be able to follow the curved path. As described in [11], the maximum admissible velocity in relation to the curvature and the acceleration tangential to the path, $\dot{V}_k$, at any point of a given flight path can be evaluated by

$$V_{k,max} = \sqrt[4]{\frac{a_{max}^2 - \dot{V}_k^2}{\kappa^2 - (\cos^4\gamma + 1)}} \, . \tag{3}$$

In order to find a feasible velocity profile with a given maximum total acceleration, minima of the admissible velocities are found with (3) using a locally bounded gradient search. An optimal and feasible velocity profile between each pair of consecutive velocity minima is then found through numerical integration using a bang-bang control policy on the reference dynamic model and tangential acceleration as the sole input parameter. At each velocity minimum a forward and a backward integration is started. In case the velocity profiles of two consecutive minima do not have an intersection point, the velocity at one of these minima is iteratively lowered until an intersection point can be found as shown in Fig. 3.

The tracking of a trajectory is performed using a time-shifted control algorithm presented in [17], which significantly improves tracking accuracy compared to conventional dynamic path tracking as described in [18]. This is achieved through moving a reference position ahead from the current position on the trajectory by a constant time delta. This reference position is then used to retrieve the velocity command, which serves as input to the low level flight control system.

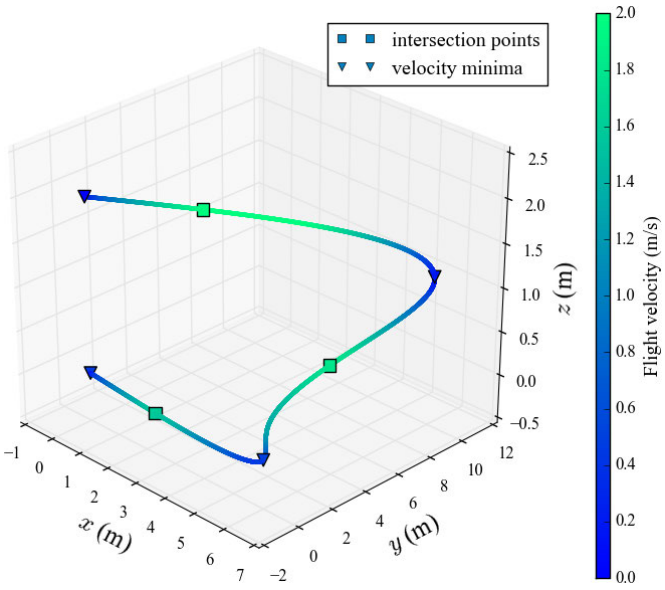

Figure 3. Velocity profile for a 3D spline curve

IV. Online Feasible Trajectory Planning

In the following sections we present our approaches to tackle the typical problems identified for decoupled trajectory planning. First we describe the discretization algorithm used to evaluate the parametric cubic splines. It allows for very efficient discretization with variable step-size controlled by a maximum admissible discretization error. It forms the basis for iteratively evaluating and transforming the trajectories and performing rapid and reliable collision checking. In section IV-B we describe our approach to maintain obstacle clearance and control the curvature profile of the smoothed path through inserting support points. In section IV-C the procedure of adjusting the initial boundary condition in order to obtain feasible flight paths is discussed and in the last section we describe our approach to spatially separate these refinement measures in order to avoid interferences that would compromise the planning completeness.

A. Variable Step-Size Discretization

In order to efficiently and accurately assess obstacle clearance of many possible flight paths in a short amount of time, as needed with any iterative planning approach, we rely on analytic calculation of the minimum distances between geometric primitives. In order to apply these algorithms to cubic spline curves, an efficient discretization algorithm is needed, so that the obstacle clearance may be calculated for linear segments approximating the cubic spline curve locally. However, the discretization error must be considered here as it must be subtracted from the obstacle clearance of each linear segment. Through calculating an upper bound on the discretization error, as shown below, we are able to perform a variable step-size discretization which significantly reduces the number of points and therefore computation time needed to assess obstacle clearance compared to a fixed step-size discretization. With this approach, flight paths that actually

have sufficient obstacle clearance may be rendered unsafe due to the discretization error. However, by selecting an upper bound on the discretization error, based on available computation time and the width of narrow passages between obstacles that the rotorcraft should fly through, a tradeoff between runtime efficiency and planning completeness can be chosen.

To estimate the error of a linear approximation of the i-th curve segment, the arc length Δs_i of this segment may be used, which for an arbitrarily parametrized space curve can be obtained through integration:

$$\Delta s_i = s_{i+1} - s_i \,, \tag{4}$$

$$\Delta s_i = \int_{\tau_i}^{\tau_{i+1}} |\vec{r}\,'(\tau)|\mathrm{d}\tau \,. \tag{5}$$

In general (5) cannot be solved analytically. However, an upper bound on the norm of the curve's derivative may be formulated for cubic spline curve's, as shown below.

For a given point $\vec{r}(\tau + \triangle\tau)$ on a parametric cubic spline curve, the norm of the first and second derivative may be calculated through integrating

$$|\vec{r}\,'(\tau+\triangle\tau)| = |\vec{r}\,'(\tau)| + \int_{\tau}^{\tau+\triangle\tau} \frac{\vec{r}\,''(\tau)\cdot\vec{r}\,'(\tau)}{|\vec{r}\,'(\tau)|}\mathrm{d}\tau \,, \tag{6}$$

$$|\vec{r}\,''(\tau+\triangle\tau)| = |\vec{r}\,''(\tau)| + \int_{\tau}^{\tau+\triangle\tau} \frac{\vec{r}\,'''(\tau)\cdot\vec{r}\,''(\tau)}{|\vec{r}\,''(\tau)|}\mathrm{d}\tau \,. \tag{7}$$

The integrands represent the portion of the higher order derivative that contributes to the norm of the lower order derivative, so that upper bounds can be formulated:

$$|\vec{r}\,'(\tau+\triangle\tau)| \leq |\vec{r}\,'(\tau)| + \int_{\tau}^{\tau+\triangle\tau} |\vec{r}\,''(\tau)|\mathrm{d}\tau \,, \tag{8}$$

$$|\vec{r}\,''(\tau+\triangle\tau)| \leq |\vec{r}\,''(\tau)| + \int_{\tau}^{\tau+\triangle\tau} |\vec{r}\,'''(\tau)|\mathrm{d}\tau \,. \tag{9}$$

The third derivative of a parametric cubic spline curve,

$$\vec{r}\,'''(\tau) = \left(S_x''', S_y''', S_z'''\right) \,, \tag{10}$$

is constant over each segment, so that the integrals of (8) and (9) respectively can be solved analytically. An upper bound on the first derivative and on the arc length of the curve segment can therefore be obtained by formulating a taylor series:

$$|\vec{r}\,''(\tau+\triangle\tau)| \leq |\vec{r}\,''(\tau)| + |\vec{r}\,'''|\triangle\tau \,, \tag{11}$$

$$|\vec{r}\,'(\tau+\triangle\tau)| \leq |\vec{r}\,'(\tau)| + |\vec{r}\,''(\tau)|\triangle\tau + |\vec{r}\,'''|\frac{\triangle\tau^2}{2} \,, \tag{12}$$

$$\triangle s \leq |\vec{r}\,'(\tau)|\triangle\tau + |\vec{r}\,''(\tau)|\frac{\triangle\tau^2}{2} + |\vec{r}\,'''|\frac{\triangle\tau^3}{6} \,. \tag{13}$$

In order to obtain an upper bound on the discretization error h for a linear segment approximating the cubic spline curve, we use two different approaches with different assumptions made about the curve.

1) For a linear segment of length l that approximates a curves segment of the arc length $\triangle s$ an upper bound on the discretization error may be obtained by calculating the height of an isosceles triangle. The sides of equal length amount to the arc length and the length of the bottom side equals the length of the linear segment:

$$h \leq \sqrt{\left(\frac{\Delta s}{2}\right)^2 - \left(\frac{l}{2}\right)^2} \,. \tag{14}$$

This is a rather conservative approximation, however no further assumptions are made about the curve segment.

2) Assuming a curve segment of monotonically increasing or decreasing curvature, an upper bound on the discretization error may be obtained by calculating the chord height of a circular segment that approximates the curve at the point of maximum curvature within the considered interval. For a segment of the osculating circle this gives an upper bound

$$h \leq \frac{1}{\kappa_{max}} - \frac{1}{2}\sqrt{\frac{4}{\kappa_{max}^2} - l^2} \,, \tag{15}$$

depending on the chord length l, which equals the length of the linear segment approximating the curve. It must therefore be smaller than the osculating circle's diameter

$$l \leq \frac{2}{\kappa_{max}} \,, \tag{16}$$

in order for this error approximation to be valid.

We use binary search to efficiently bisect each segment of a cubic spline curve until a maximum admissible discretization error h_{max} and the maximum segment length l_{max} are undercut. As the course of the curvature along a curve segment is initially unknown, the assumption of monotonically increasing or decreasing curvature that is required in order for (15) to be valid cannot be evaluated at first. To overcome this problem, we use a two-stage discretization algorithm that uses (15) to approximate the discretization error during the first run and identifies intervals that may contain curvature maxima numerically. In a second run these intervals are then rediscretized using (14) so that a maximum discretization error can be guaranteed.

B. Restricting the Smoothed Path's Deviation

The interpolation of linear path segments using cubic splines is well suited for generating low curvature paths, which allow high flight velocities. However, there are two aspects that have to be considered with this approach:

1) The resulting curvature profile is closely coupled to the length of linear segments and the angles between these segments. For sampling-based planners using cost functions that consider the path length or trajectory time, it can be assumed that deviations from a straight line to the goal point are either caused by the limited sampling resolution or search graph connectivity of the planner, or the need to avoid obstacles. For a properly configured sampling-based planner, which generates close to optimal paths, it is reasonable to assume that most deviations from this straight line are necessary in order to avoid obstacles. While cubic spline interpolation is great for reducing the curvature maxima resulting from these deviations, it also lengthens the flight path and might lead to unnecessarily large curve radii. If a given curve is feasible to fly with the maximum flight velocity, then any further reduction of curvature that results in a longer flight path length is likely to increase trajectory time.

2) The deviation d_{spl} of the smoothed path from the linear path segments may lead to a violation of the obstacle clearance d_{clear}, which was used by the sampling-based planner. The obstacle clearance is assessed using the discretization technique described in section IV-A checking every linear segment approximating the curve locally. The nominal obstacle clearance, which is assessed during this process, has to be extended by the maximum discretization error to apply the result to the cubic spline curve.

Both of these problems can be solved by locally restricting the smoothed path's deviation from the linear path. As stated in section II, the insertion of support points has been used before as a measure for restricting cubic spline curves. However, the approaches presented in [12] and [13] may fail in certain situations and provide fallback solutions using linear segments. Furthermore, the effects of inserting support points on the curvature of the spline curve are not considered in detail.

In [19] bounds are given for the deviation of periodic cubic splines. For a chord length parametrization the maximum deviation is limited in respect to the linear segment's length

$$d_{spl} \leq \frac{3}{4} l \,. \tag{17}$$

Within this work, clamped boundary conditions, for which (17) may not generally hold true, are used for interpolation. However, it will be used as a reference on the oscillation behavior of cubic splines.

As a maximum deviation $d_{spl,max}$ that adapts to the helicopter's dynamic constraints, we use the minimum curve radius at maximum flight speed

$$d_{spl,max} = \frac{V_{k,max}^2}{a_{max}} \,. \tag{18}$$

This deviation limit is initially used for any given path. However, if a violation of the obstacle clearance has been detected, $d_{spl,max}$ is reduced for the affected linear path segment until the required obstacle clearance can be guaranteed for any curve within this limit. If a spline curve is found to violate the deviation limit in an interval $[\tau_i, \tau_{i+1}]$, support points are inserted iteratively in this interval, following a strategy described below, until the violation is resolved.

When support points are inserted to limit the spline curve's deviation from the linear path, an increase in curvature is to be expected. To reduce this increase, we try to minimize the change of the curve's shape by inserting one support point at a time and placing it along the shortest distance from the linear segment at the the point of maximum deviation within the violating interval, as shown in Fig. 4.

In order to achieve a given deviation limit with a limited number of inserted support points, we place these points in a distance from the linear path segment, d_{sup}, smaller than the deviation limit. This distance depends on the expected deviation of the curve between neighboring support points, which we assume to be bounded by (17) and therefore by the longitudinal distance l_{ins} between support points in respect to the linear path segment:

$$d_{sup} = d_{spl,max} - \frac{3}{4} l_{ins} \,. \tag{19}$$

Assuming that (17) holds true for clamped cubic splines, an upper limit

$$n_{max} = \left\lceil \frac{l}{\frac{4}{3} d_{spl,max}} \right\rceil \tag{20}$$

can be given on the number of support points that have to be inserted along a linear path segment of length l in order to achieve a given deviation limit for the spline curve. Accordingly, the minimum longitudinal distance

$$l_{ins} = \frac{4}{3} d_{spl,max} \tag{21}$$

between inserted support points can be specified for which the cubic spline curve will not violate the maximum deviation when the maximum number of support points have been inserted.

C. Planning Feasible Trajectories

A feasible velocity profile can be found for any C^0-continuous path if the current flight velocity is below a threshold for which the helicopter can be considered a holonomic vehicle. However, when flying through initially unknown terrain with flight velocities above this threshold, the rotorcraft's dynamics must be accounted for when re-planning the flight path. To do this, we consider the following simple kinematic constraints on total acceleration, total flight velocity and yaw rate r :

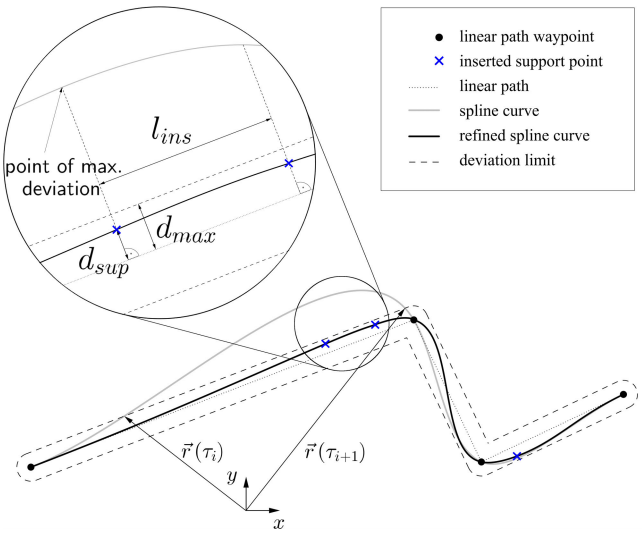

Figure 4. Restricting the path deviation through insertion of support points

$$a \leq a_{max} \ ,$$
$$V_k \leq V_{k,max} \ ,$$
$$r \leq r_{max} \ .$$

In this work, we consider the satisfaction of these constraints as a sufficient requirement for a feasible trajectory which the rotorcraft is able to follow accurately. The constraints on velocity and acceleration take into account the rotorcraft's limited thrust as well as limited bank and pitch angles. For a safe flight through unknown terrain, it is required that the field of view of obstacle sensors mounted onboard the rotorcraft is aligned with the direction of movement at all times. The yaw rate limit must be considered to achieve this, especially when flying turns at low speeds close to hovering.

To assess if the limit on the total acceleration can be met with a replanned path, the following constraint on the curvature of this path can be considered:

$$\kappa \leq \frac{a_{max}}{V_k^2} \ . \tag{22}$$

At very low speeds, close to hovering, rotorcraft are capable of rapidly changing the direction of movement. For these situations we assume the rotorcraft's bank and pitch angles to be negligible in order to account for the yaw rate limit. With this assumption, which may also be justified by choosing a conservative yaw rate limit, the following inequality limiting the curvature of the horizontal projection κ_{xy} of the flight path in respect to the maximum yaw rate and the total horizontal velocity $V_{k,xy}$ must be satisfied:

$$\kappa_{xy} \leq \frac{r_{max}}{V_{k,xy}} \ . \tag{23}$$

For a given flight path and an initial flight velocity, (22) and (23) are assessed through integrating the simplified kinematic model with maximum available deceleration until a sufficiently small velocity $V_{k,hol}$ is reached, for which the helicopter may be considered as a holonomic vehicle. We define this threshold as the velocity for which the helicopter is able to maneuver safely within a trajectory tracking tolerance, d_{track}, in particular where it can come to a complete stop and perform arbitrary turns limited by the maximum yaw rate

$$V_{k,hol} = \min\left(\sqrt{2 d_{track} a_{max}},\ r_{max}\frac{d_{track}}{2}\right) \ . \tag{24}$$

If a violation of (22) or (23) is found before this velocity threshold is reached, an adjustment of the curvature is necessary in order to obtain a feasible flight path. This refinement is performed through scaling the clamped boundary conditions at the beginning of the spline curve, which determine the curve's first derivative $\vec{r}_0{}'$ at this point. For cubic spline curves parametrized by chord length, there is a relation between the magnitude of the first derivative and the curvature κ_0 at the start point of the curve of the form

$$\kappa_0\left(|\vec{r}\,'|\right) = \frac{a}{|\vec{r}\,'|^b} \ , \tag{25}$$

which was identified experimentally and has yet to be formally proven. The constants a and b can be obtained through substitution of (25) for two spline curves that interpolate an identical set of points but with differently scaled start point boundary conditions. Once the values of these constants are known, the magnitude of the first derivative which is necessary to obtain a given maximum admissible curvature at the start point of this cubic spline curve can be calculated with

$$|\vec{r}\,'(\tau_0)| = \left(\frac{a}{\kappa_{max}}\right)^{\frac{1}{b}} \ . \tag{26}$$

Using (26) to obtain the magnitude of the first derivative results in a curvature low enough to be able to meet the kinematic constraints posed by the initial velocity at the start point of the curve. Using these values as a starting point, we iteratively decrease the start point curvature, until a feasible velocity profile can be found. As shown in Fig. 5, the first curvature maximum is moved further from the start point with each iteration. Although we cannot provide an upper bound on the magnitude of the first derivative necessary to generate a feasible flight path, the necessary start point curvature could be found in the range $\frac{\kappa_{max}}{2} \leq \kappa_0 \leq \kappa_{max}$ in all replanning procedures within the simulated scenarios presented in this paper.

D. Local Feasible Planning Region

The two approaches presented in sections IV-B and IV-C to refine flight paths cannot simply be combined to achieve both, obstacle clearance and feasibility, because they could work against each other refining the path without ever achieving both or even one of the required properties. In this section we present an approach to decouple the path refinements through a spatial separation that allows to iteratively solve this problem.

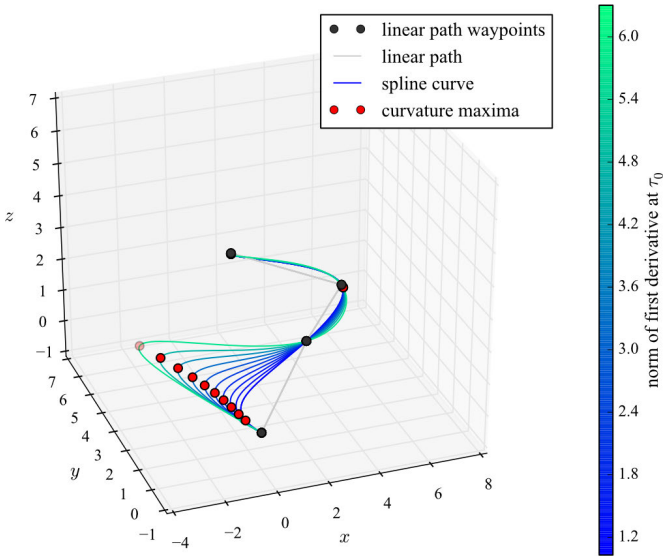

Figure 5. Refinement of a cubic spline curve through scaling of the boundary condition at the start point

In particular, we define a local planning region where the kinematic constraints will be treated with priority and the smoothed path's deviation from linear segments will not be restricted. The extent of this region is defined as a sphere centered at the current position of the helicopter with a radius d_{local}, which will be referred to as the local feasible planning distance. Within this region, no support points will be added to restrict the smoothed path's deviation as this might interfere with the scaling of the boundary condition in order to obtain a feasible flight path.

As shown in Fig. 6, this approach yields low curvature smooth paths that follow the linear path globally and thus maintain its optimality but allow for local refinement in order to achieve feasibility. Furthermore, the insertion of support points does not interfere with the scaling of the boundary condition in a way that would produce a suboptimal path. However, as no support points may be inserted within the feasible planning distance, the obstacle clearance of the flight path may not be maintained. To solve this problem, we iteratively scale the extend of the feasible planning region, as described below.

As mentioned in section IV-C, the feasibility of a flight path can be assessed through calculating the earliest possible stopping point along the path. The shortest feasible flight path is always a straight path which allows maximum deceleration. Therefore the minimum stopping distance

$$d_{stop} = \frac{1}{2}\frac{V_k^2}{a_{max}} \tag{27}$$

is used as a lower bound for the local feasible planning distance, as it will always be feasible to execute a maximum deceleration stopping maneuver within this distance. For the definition of an upper bound, time optimality is considered, which demands high velocity trajectories. A reasonable upper bound on the radius of the local feasible planning region may therefore be defined as twice the minimum curve radius at maximum flight speed

$$R_{max} = \max\left(\frac{V_{k,max}^2}{a_{max}}, \frac{V_{k,max}}{r_{max}}\right) . \tag{28}$$

Within a region that extends over $2R_{max}$, the helicopter is able to perform an arbitrary turn at maximum speed without leaving this region. The radius of the local feasible planning region can therefore be restricted to

$$d_{stop} \leq d_{local} \leq 2R_{max} . \tag{29}$$

Starting at an initial guess, which is purely heuristic and takes the general direction of the linear path into account, we choose a value for d_{local} in the range defined by (29). If the resulting smoothed path's obstacle clearance is not sufficient, we iteratively reduce the feasible planning distance and replan the flight path. While approaching the minimum stopping distance, the flight path is more and more restricted to a shape that may be less optimal in respect to trajectory time but allows higher deceleration rates. At the same time, the deviation from the linear path is restricted for a larger fraction of the smoothed path eliminating violations of the required obstacle clearance. The simple spherical shape that we chose for the local feasible planning region does not retract to the maximum deceleration trajectory, and therefore it cannot be guaranteed that a feasible and safe path will be found. However, this may only limit the planning completeness in situations where the rotorcraft flies very close to obstacles at high speeds. Throughout all of the simulated scenarios presented in the subsequent section, this has not been shown to be problematic as all planning procedures were successfully conducted.

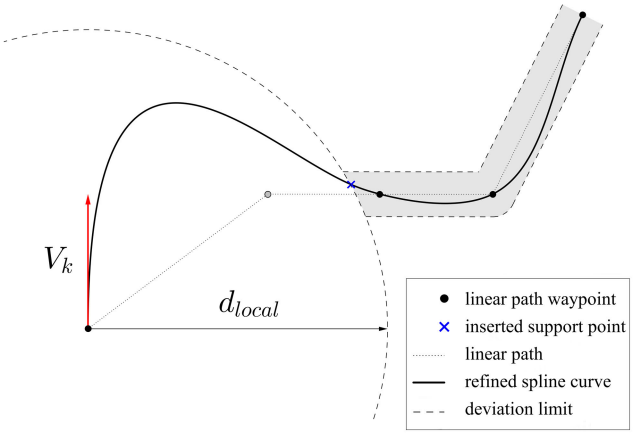

Figure 6. Deviation limits are not enforced within the local feasible planning region

V. Results and Discussion

In this section simulation results of the online trajectory time reduction approach are presented. The main performance aspects to be considered are trajectory time, obstacle clearance and computation runtime. Results are compared

against approaches using linear paths as well as against the results presented in [1], which include numerically optimized baseline trajectories and results from a sampling-based obstacle field navigation algorithm that will be referred to as libOFN.

A. Experimental Setup

The results presented in subsequent sections were obtained using a high-fidelity simulation of the midiARTIS helicopter's closed-loop dynamics. Just like the flight control system integrated with this helicopter, velocity commands are used as input for underlying control loops. Table I shows the parameter values used as kinematic constraints in the flight control system as well as in the planning module.

The virtual obstacle sensor is a LIDAR sensor model rotating a virtual laser scan plane with $\alpha_{fov} = 160\,^\circ$ similarly to the sensor used in [1]. It's maximum range is set to $d_{sense} = 50\,\mathrm{m}$, which is $20\,\mathrm{m}$ less than the sensor model used in [1] but accounts for our experiences with the LIDAR sensor we plan to integrate onboard of the midiARTIS.

The roadmap-based planner, briefly described in section III-A, is used to replan the global path to the current goal point whenever new obstacles have been detected. The roadmap is initialized with an expected maximum sample distance of $d_{sample} = 10\,\mathrm{m}$ for the simple and $d_{sample} = 15\,\mathrm{m}$ for the urban terrain scenarios. Flight path planning was performed with a minimum obstacle clearance of $d_{clear} = 8\,\mathrm{m}$, which accounts for the expected trajectory tracking error of $d_{track} = 1.5\,\mathrm{m}$, the rotor diameter of slightly less than $2\,\mathrm{m}$ and an additional safety margin of $4.5\,\mathrm{m}$, which equals half the minimum stopping distance at maximum flight speed.

The planning of spline trajectories was performed incrementally with each step smoothing the linear path segments within the maximum sensor range. This is done in order to limit computation time used on optimizing path segments which may be replanned later on when new obstacles are detected.

Table I
SIMULATION PARAMETERS USED FOR BENCHMARKING

Parameter	Value
a_{max}	$0.5\,\mathrm{m/s^2}$
r_{max}	$180\,^\circ/\mathrm{s}$
$V_{k,xy,max}$	$3\,\mathrm{m/s}$
$V_{k,z,max}$	$1.5\,\mathrm{m/s}$

The computer hardware used is an Intel Core i7 at $2.8\,\mathrm{GHz}$ with $4\,\mathrm{GB}$ RAM running Windows 7 64bit. The simulation is run without multi-threading and thus with time synchronized planning, control and sensing steps.

The benchmark scenarios used for this evaluation were presented in [1]. They are split into simple scenarios, which include basic obstacle avoidance tasks, and urban scenarios, which consist of navigation tasks in urban terrain. All significant parameters, the terrain data and the mapping process used for the closed loop simulation are identical to our previous evaluations [12], [15] so that the results can be compared against each other. However, for the comparison against libOFN benchmark results and the baseline, which can be found in [1], differences in the order of the commanded goal points and the mapping process have to be considered. While the baseline solution was heavily optimized taking advantage of complete terrain knowledge, the libOFN evaluation started out with no terrain knowledge, but detected obstacles were gathered over both, outward and return flight to each start point (e.g. A-A1-A-A2-A-A3...). For the evaluation of our framework, we also continuously accumulate terrain data, but skipped the outward flight to each start point except for the first scenario. As the start points are far apart from each other, no knowledge of the surrounding area was available at the beginning of each scenario, but terrain knowledge about the area around the center point accumulated over the multiple scenarios.

B. Simulation Results

Tables II and III show the trajectory times achieved in simple and urban benchmark scenarios. The corresponding flight paths for the simple scenarios are shown in Fig. 7 and the urban scenario flight paths are shown altogether in Fig. 8. The flight paths are shown as a top view with the terrain in the background. Altitude and terrain height are indicated by the brightness of the corresponding color.

Table II
TRAJECTORY TIMES OF SIMPLE SCENARIOS IN SECONDS

Scenario	Baseline	libOFN[1] (2010)	MiPlEx (linear)[12]	MiPlEx (spline)
Out and Back	78.8	84.5	83.8	84.3
Point Obstacle	39.3	49.2	50.4	41.3
Wall	39.3	54.1	57.8	48.1
Cube	42.1	52.2	53.9	43.9
Wall Baffle	41.7	52.5	59.5	46.7
Cube Baffle	39.8	51.9	79.7	43.4

Table III
TRAJECTORY TIMES OF URBAN SCENARIOS IN SECONDS

Scenario	Baseline	libOFN[1] (2010)	MiPlEx (linear, [12])	MiPlEx (spline)
A1	92.7	97.9	100.2	88.1
A2	76.2	79.7	110.5	74.4
A3	74.6	81.7	105.8	75.1
A4	70.9	78.3	78.9	69.8
A5	87.4	99.0	140.4	125.8
A6	81.7	84.9	102.1	81.1

The trajectory time could be significantly reduced compared to the linear path planner for all scenarios. Compared to the baseline, the results for simple scenarios are quite diverse. Considering the horizontal and vertical velocity components in scenario "Wall", which are shown in Fig. 9, it can be seen that the impact of vertical maneuvers on the horizontal velocity is significant. The top view for this scenario in Fig. (7), second row on the right, shows another reason for the loss of time in this scenario. The wall obstacle is detected incrementally due to the limited sensor range which leads to the rotorcraft deviating from the optimal baseline trajectory and flying to the left allegedly around the detected obstacle.

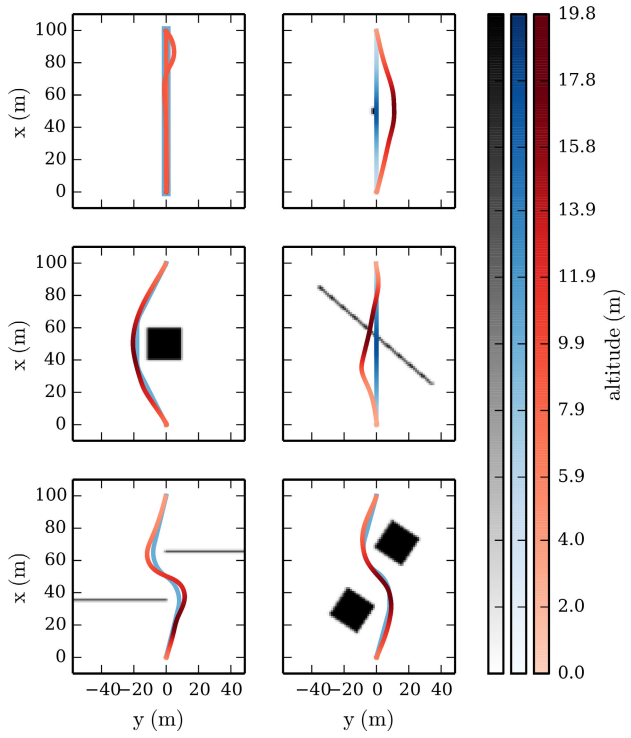

Figure 7. Baseline (blue) and MiPlEx (red) flight paths in simple benchmark scenarios

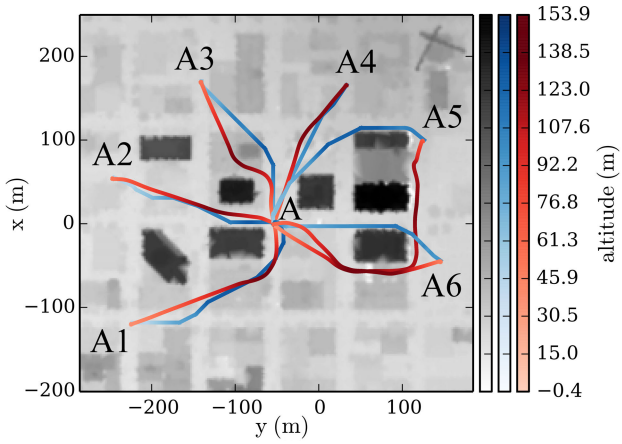

Figure 8. Baseline (blue) and MiPlEx (red) flight paths in urban benchmark scenarios

Not until a large enough portion of the wall is detected, a path across the wall is planned. The multiple replan procedures also explain the changes in the vertical velocity at the beginning of this trajectory, which can be seen in Fig. 9. Each replanned path, that accounted for the next detected portion of the wall, passed through a different sample on the left avoiding the already known part of the wall. Because of the quasi-random sampling procedure, the samples are not aligned vertically which leads to the slight variations in the path slope. The relative difference in trajectory time of this scenario compared to the baseline is 22 %, whereas all other simple scenarios with much less change in altitude lie within approximately 10 % relative difference. In order to overcome

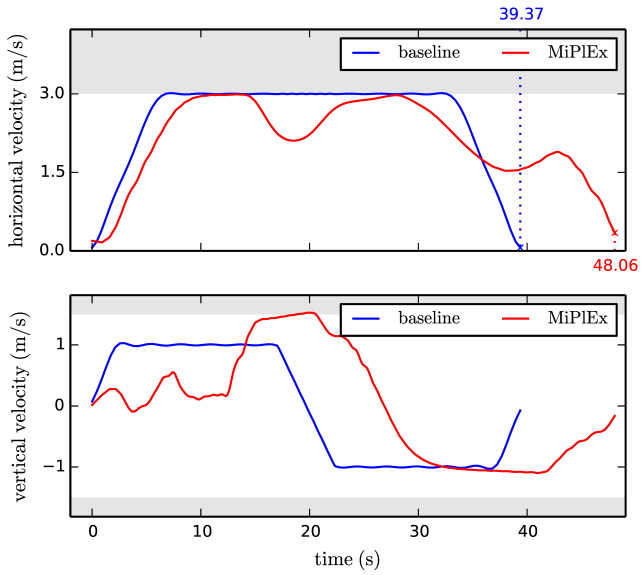

Figure 9. Horizontal and vertical velocity components in scenario “Wall”

this performance difference, further optimization of vertical maneuvers and a larger sensor range would be necessary.

In all but one of the urban scenarios the trajectory times are within ±5 % to the baseline, sometimes even beating the baseline by a few seconds. Looking at the flight path of that one exception, scenario A5, it can be seen that the increase in trajectory time stems from the incomplete knowledge of the environment at the starting point. The global planner chooses a suboptimal path around several tall buildings without ever finding a shortcut passage. The significant loss of time compared to the baseline and libOFN trajectories in this scenario can be explained with the differences in terrain knowledge. However, in Fig. (8) it can also be seen that the path planner did not account for the much better terrain knowledge of the previously flown path from A4 to the center point when it started at A5. Planning a path through already known terrain would have been likely to result in a better mission performance in this situation. Taking into account uncertainties about the terrain knowledge along the flight path would be a promising subject for future research.

While the completeness of the planning procedure including all path refinements have not been formally proven, the simulation results indicate the reliability of the presented approach. Feasible and smooth trajectories were found in all replanning procedures throughout the benchmark scenarios. However, during more thorough testing, we were able to produce situations with very limited planning options especially at speeds close to the limit within narrow passages for which the presented planning approach failed to find safe and feasible trajectories. If the obstacle clearance is below the minimum stopping distance, the conflict between the kinematic constraints and the obstacle clearance cannot always be resolved although a path may exist that satisfies both constraints. However, as we have shown in simulation, this lack of completeness can be compensated through

choosing conservative safety distances without significant loss of mission performance.

Fig. 10 shows the cumulative distribution function of computation time needed for the path smoothing procedures performed throughout all benchmark scenarios. With an average duration of $0.23\,\mathrm{s}$ and staying below $0.31\,\mathrm{s}$ in $95\,\%$ of all cases, the presented approach proves to be very efficient, considering the increase in mission performance. Note that a comparison of computation times against the libOFN results would have required to run both benchmarks on the same hardware and can therefore not be provided. Otherwise, a more complete and hardware independent scaling study would be required, which was not within the scope of this work.

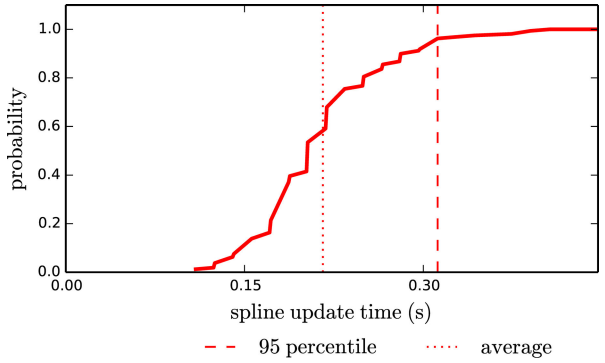

Figure 10. CDF of computation times for path smoothing

In Fig. 11 plots of the cumulative distribution function and histograms for various performance metrics summarized over all benchmark scenarios are shown. It can be seen that the horizontal speed was close to the limit for most of the flight time, with minor violations of $V_k > 3.1\,\mathrm{m/s}$ at $5\,\%$ of the flight time. The acceleration limit has been satisfied in $92.4\,\%$ of the flight time. These results show that the simple kinematic constraints, which were used by the mission planning framework to plan feasible flight paths and are derived from the reference model used by the flight control system, can be satisfied for the majority of the flight time in the tested benchmark scenarios. Violations of these constraints may leave the rotorcraft unable to follow an allegedly feasible path, which leads to a degradation of the path following performance and may therefore pose a safety risk. For our evaluation, we chose an admissible path tracking error of $1.5\,\mathrm{m}$, which from our flight test experience is quite optimistic and would have to be increased significantly for real flight tests due to the effect of wind and even greater model uncertainties. However, in the simulated scenarios the required obstacle clearance was satisfied in $99.9\,\%$ of the flight time, with a minimum of $7.89\,\mathrm{m}$ slightly below the limit but within a tolerance well below the expected sensor uncertainty in real flight conditions. This shows how well the path planning and the flight control system can be configured to match the requirements of a given scenario. Considering an average height above ground of only $21.3\,\mathrm{m}$ and the complexity of the benchmark scenario's obstacle fields, the tradeoff that was chosen between mission performance and safety requirements was successfully put into practice.

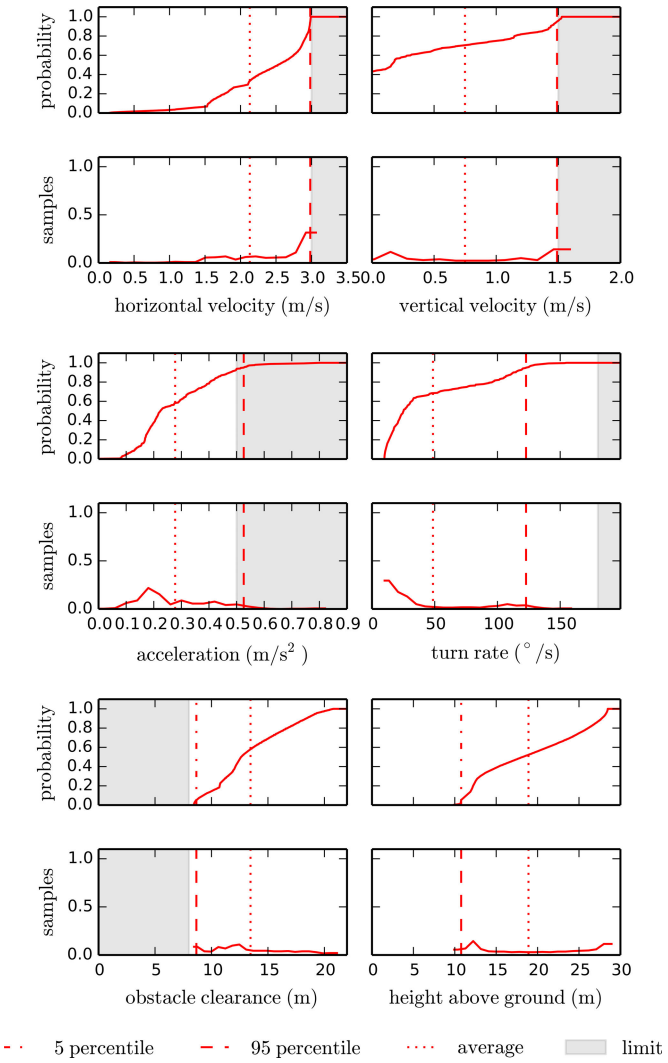

Figure 11. CDF and histogram plots for various performance metrics over all benchmark scenarios

VI. Summary

An approach for runtime efficient en-route trajectory time reduction is presented that allows to achieve close-to-optimal mission performance for unmanned rotorcraft navigating in a priori unknown environment. A roadmap-based path planner is used for global navigation to the current goal point. Cubic spline interpolation is combined with various refinement techniques and an efficient discretization algorithm in order to transform the flight paths into safe, dynamically feasible and time efficient trajectories. We especially focus on problems that commonly arise with such decoupled planning approaches. In particular, we consider the loss of planning completeness due to conflicts between flight path safety and dynamic feasibility as well as suboptimality introduced through the sequential incorporation of constraints. For this purpose, the effects of the different measures taken to refine flight paths are carefully considered. A feasible planning

region is defined within which smoothed paths are refined in order to satisfy kinematic constraints derived from the reference model used by the flight control system. Outside of this region the flight paths are restricted to follow linear paths to the current goal point utilizing the runtime efficiency of the sampling-based global path planner.

Experimental results, obtained with high fidelity simulation of the midiARTIS helicopter's closed-loop dynamics, show that on average the mission performance is within $10\,\%$ of the heavily optimized baseline results presented in [1] for a set of online navigation benchmark scenarios. Although no formal guarantee on the completeness of the planning procedure can be given, it has proven to work very reliable in various mission scenarios. With computation times in the order of tenths of a second, the flight path refinement is very efficient allowing to meet runtime constraints even on the limited onboard hardware of small-scale helicopters.

Several shortcomings have yet to be addressed with the presented approach for planning time-efficient flight paths. Further evaluation of particularly critical situations that may compromise the planning completeness has to be carried out. This includes situations with very limited planning options as they occur when flying very fast and close to obstacles. Another problem identified with our simulation results is the major effect of incomplete terrain knowledge on the mission performance. Accounting for uncertainties about the environment during path planning seems a promising approach to further decrease trajectory time especially in complex urban scenarios. With the benchmark results presented in this paper, we have demonstrated that our path planning and flight control system can be configured for specific scenarios in order to achieve safe and time-efficient flight. However, in order to deal with real flight conditions, including wind and larger model uncertainties, future work should include an adaptive mission planning system that closes the loop between flight control and path planning and accounts for varying tracking and sensing performance. In order to gain experiences and to confirm our simulation results, we are already in the progress of integrating the presented approach onboard our midiARTIS helicopter.

REFERENCES

[1] B. Mettler, Z. Kong, C. Goerzen, and M. Whalley, "Benchmarking of Obstacle Field Navigation Algorithms for Autonomous Helicopters," *Proceedings of the American Helicopter Society 66th Annual Forum, Phoenix, AZ, USA*, 2010.

[2] C. Goerzen, Z. Kong, and B. Mettler, "A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 65–100, 2009. [Online]. Available: http://link.springer.com/10.1007/s10846-009-9383-1

[3] S. Karaman and E. Frazzoli, "Optimal Kinodynamic Motion Planning Using Incremental Sampling-based Methods," *Decision and Control (CDC), 49th IEEE Conference on*, pp. 7681 – 7687, 2010. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5717430

[4] D. Webb and J. Berg, "Kinodynamic RRT*: Optimal Motion Planning for Systems with Linear Differential Constraints," *CoRR*, vol. abs/1205.5088, 2012. [Online]. Available: http://arxiv.org/abs/1205.5088

[5] K. Yang, "An Efficient Spline-based RRT Path Planner for Non-Holonomic Robots in Cluttered Environments," *International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 288–297, May 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6564701

[6] S. M. LaValle, *Planning Algorithms*, S. M. Lavalle, Ed. Cambridge University Press, 2006. [Online]. Available: http://ebooks.cambridge.org/ref/id/CBO9780511546877

[7] S. K. Kannan, W. M. Sisson, D. A. Ginsberg, J. C. Derenick, X. C. Ding, T. A. Frewen, and H. Sane, "Close Proximity Obstacle Avoidance Using Sampling-Based Planners," in *AHS Specialists' Meeting on Unmanned Rotorcraft and Network-Centric Operations*, 2013.

[8] P. Pettersson and P. Doherty, "Probabilistic Roadmap Based Path Planning for an Autonomous Unmanned Helicopter," *Journal of Intelligent and Fuzzy Systems*, vol. 17, pp. 395–405, 2006. [Online]. Available: http://iospress.metapress.com/index/8dc7fpb8taqly4pg.pdf

[9] P. Tsenkov, J. K. Howlett, and M. Whalley, "A System for 3D Autonomous Rotorcraft Navigation in Urban Environments," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008. [Online]. Available: http://uarc.ucsc.edu/flight-control/libofn/papers/Tsenkov_AIAA08.pdf

[10] D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 2520 – 2525, 2011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5980409

[11] S. Lorenz and F. Adolf, "A Decoupled Approach for Trajectory Generation for an Unmanned Rotorcraft," *Advances in Aerospace Guidance, Navigation and Control*, pp. 3–14, 2011. [Online]. Available: http://www.springerlink.com/index/T601V3822H07483P.pdf

[12] F.-M. Adolf, M. Abou-Hussein, and C. Goerzen, "Trajectory Time Reduction using Field of View-based Smoothing of Roadmap-based Paths," in *AHS 69th Annual Forum, 21.-23. May, Phoenix, AZ, USA*, 2013. [Online]. Available: http://elib.dlr.de/82379/

[13] J. Pan and L. Zhang, "Collision-free and curvature-continuous path smoothing in cluttered environments," in *Robotics: Science and Systems VII*, 2012, pp. 233–240. [Online]. Available: http://roboticsproceedings.org/rss07/p32.pdf

[14] S. Scherer, S. Singh, L. Chamberlain, and S. Saripalli, "Flying Fast and Low Among Obstacles," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2023–2029, Apr. 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4209383

[15] F. Adolf, "Multi-Query Path Planning for Exploration Tasks with an Unmanned Rotorcraft," in *AIAA Infotech@Aerospace, 19.-20. Juni, Garden Grove, CA, USA*, 2012.

[16] D. Pollock, *Smoothing with Cubic Splines*. London University, Queen Mary and Westfield College, Department of Economics, 1993. [Online]. Available: http://r.789695.n4.nabble.com/file/n905996/SPLINES.PDF

[17] S. Lorenz and J. C. Dauer, "Evaluation of Time-Shifted Feedforward Control for Unmanned Helicopter Path Tracking," in *AIAA Guidance, Navigation and Control Conference*, Minneapolis, Minnesota, USA, 2012. [Online]. Available: http://www.researchgate.net/publication/230725896_Evaluation_of_Time-Shifted_Feedforward_Control_for_Unmanned_Helicopter_Path_Tracking/file/9fcfd503791dfb2842.pdf

[18] N. Sarkar, X. Yun, and V. Kumar, "Dynamic Path Following: A New Control Algorithm for Mobile Robots," *Decision and Control, Proceedings of the 32nd IEEE Conference on*, vol. 3, pp. 2670 – 2675, 1993. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=325681

[19] M. S. Floater, "On the Deviation of a Parametric Cubic Spline Interpolant from Its Data Polygon," *Computer Aided Geometric Design*, vol. 25, no. 3, pp. 148–156, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167839607000908