Finding 3D Dubins Paths with Pitch Angle Constraint Using Non-linear Optimization

Jáchym Herynek Petr Váňa Jan Faigl

***Abstract*— This paper presents a novel non-linear programming formulation to find the shortest 3D Dubins path with a limited pitch angle. Such a path is suitable for fix-wing aircraft because it satisfies both the minimum turning radius and pitch angle constraints, and thus it is a feasible and smooth path in the 3D space. The proposed method utilizes the existing decoupled approach as an initial solution and improves its quality by dividing the path into small segments with constant curvature. The proposed formulation encodes the path using the direction vectors that significantly reduce the needed optimization variables. Therefore, a path with 100 segments can be optimized in about one second using conventional computational resources. Although the decoupled paths are usually within $2\,\%$ from the lower bound, the proposed approach further reduces the gap by about $30\,\%$.**

I. Introduction

Finding the shortest 3D Dubins path is a natural extension of the well-studied Dubins path planning problem [1] in 2D. However, the 3D extension does not have a closed-form solution. In addition to the minimum turning radius and constant forward speed, the vehicle's movement is also constrained by the maximum climb/dive angle to ensure path feasibility for a real aircraft. Several heuristics have been proposed in the literature to find a feasible solution close to the optimum. Currently, the best existing method to find the 3D Dubins path (to the best of the authors' knowledge) is the decoupled approach proposed in [2]. It solves the horizontal and vertical parts of the path separately, and the minimum turning radius for each segment is determined such that the overall curvature is guaranteed to be within the limit. Therefore, such a path can be further optimized to use general 3D turns that are neither horizontal nor vertical, which is the motivation of the presented work.

We propose a novel Non-Linear Programming (NLP) formulation of the 3D Dubins path problem, which can improve the quality of existing solutions. The path is split into several segments, where each segment is a circular arc in the 3D space (or a straight line). The proposed formulation effectively represents any feasible 3D path if the number of segments is sufficiently high. An example of the difference between the initial 3D path found by the decoupled approach [2] and the optimized path provided by the proposed optimization is depicted in Fig. 1.

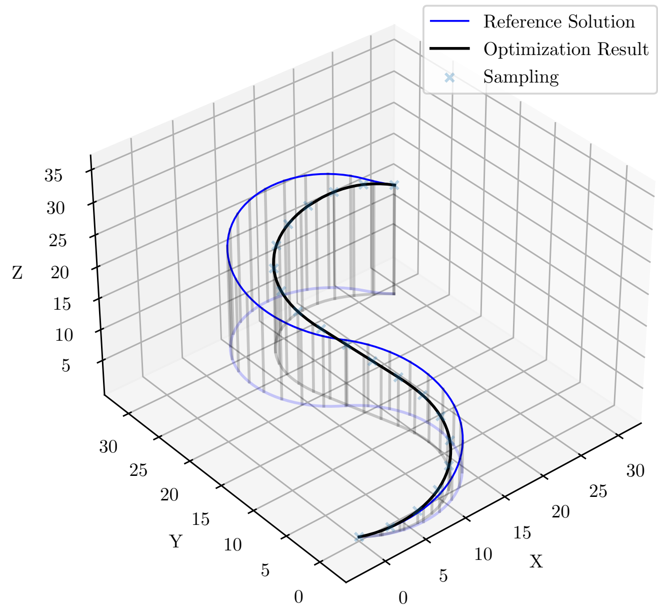

Fig. 1. Example of the 3D Dubins path found by the decoupled approach [2] (blue) and the path optimized by the proposed NLP formulation (black). The crosses represent samples on the boundary of the path segments used in the proposed formalization of the optimization problem.

Although the decoupled approach already provides 3D Dubins paths close to the optimum, based on the empirical evaluation, the proposed NLP-based optimization can further reduce the relative optimality gap to a lower bound estimate of the optimal solution value by about $30\,\%$ without significant computational requirements. The proposed formulation encodes the path by direction vectors instead of intermediate configurations, and a computation of the path with 100 segments takes about one second using conventional computational resources.

The rest of the paper is organized as follows. The related work with a description of the existing decoupled approach is summarized in Section II. The problem of finding the 3D Dubins path is formally introduced in Section III, and the proposed NLP-based optimization is described in Section IV. The evaluation results are reported and discussed in Section V. Finally, Section VI summarizes the paper.

II. Related Work

The problem of finding the shortest path with bounded curvature in the plane was posted by Andrey Markov at the end of the nineteenth century. Later on, the optimal solution

Authors are with the Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic.{herynjac|vanapet1|faiglj}@fel.cvut.cz

The presented work has been supported by the Czech Science Foundation (GAČR) under the research project No. 19-20238S and by the Ministry of Education Youth and Sports (MEYS) of the Czech Republic under project No. LTAIZ19013. The access to the computational infrastructure of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" is also gratefully acknowledged.
978-1-6654-1213-1/21/$31.00

to the problem was found in 1957 by L. E. Dubins [1]. He showed that for any input configurations, the optimal path consists of three segments. Each of those segments is either a straight line or a circular arc. Moreover, the optimal path can be only one of two types: CCC or CSC, where C stands for curve segment, and S stands for a straight segment. It is also proved that each curved segment has a turning radius equal to the minimal turning radius. Hence, the problem is called the Markov-Dubins problem in the literature.

An alternative proof of the optimal solution to the Markov-Dubins problem is presented in [3] using optimal control theory. Besides, relatively recently, the authors of [4] reformulate the path segments so that both the straight segments and the circular arcs can be represented using the same formula, enabling the formulation of the problem as an optimization problem in a unifying way. The authors also present a transformation of the problem to exploit its symmetries.

The Markov-Dubins problem in 3D has already been studied in the literature. The optimal solution is addressed in [5] using geometrical and numerical approaches that address the curvature constraint. The optimal solution in the 3D is studied in [6]. However, real aerial vehicles usually have a limited pitch angle. The pitch angle constraint is considered in the numerical solution [7] that is claimed to provide the optimal solution, but the convergence of the method is not formally supported. The authors of [8] propose to use the minimal 2D path between the configurations and interpolate the altitude linearly. Although the methods presented in [5], [7], [8] provides some 3D Dubins paths, none of them limits the pitch angle, and thus the resulting 3D path may not be feasible for a real aircraft.

The constraint on the limited pitch angle is addressed by the Dubins airplane model proposed in [9]. A helix curve can address a high altitude difference between path endpoints that allows the aircraft to reach the requested altitude. A helical path added at the beginning or end of the path is used in [10] to satisfy the pitch angle constraint. The helical path is also added in the RDDH approach presented in [11] and the final path is a union of three subpaths. The helix curve is used to mitigate the altitude difference of the two points, and a simple 2D Dubins curve can then be used for the rest of the path. The same approach is utilized in [12], where the final path consists of two semicircles and Dubins helices. A geometric approach for the path containing straight segment is considered in [13] for a hypersonic glider.

A different approach based on Bézier curves is presented in [14], [15], and [16]. It provides feasible paths considering the limited pitch angle based on a variation of the quintic Pythagorean hodographs curves for generating smooth paths.

One of the most recent heuristic methods is the decoupled approach [2]. A more detailed description is provided here because the decoupled method is utilized as both an initial solution in the proposed NLP-based optimization and a reference solution for empirical evaluations.

1) Decoupled approach: The general idea of the decoupled method [2] is to divide the problem into horizontal and vertical parts. Both these sub-problems are then solved separately using 2D Dubins paths. The only influence between horizontal and vertical parts is via turning radius because the maximum curvature constraint would be violated if both horizontal and vertical turns would be executed simultaneously. Therefore, the decoupled approach selects a larger turning radius for each part to meet the curvature constraint. Also, the horizontal turning radius is increased if the pitch angle constraint is violated to enlarge the path and decrease the climb/dive angle under the given limit.

The decoupled approach [2] provides lower and upper bounds for the 3D Dubins path based on the same idea. The bounds are quite tight and are utilized in Section V-D to estimate the quality of paths optimized by the proposed NLP-based approach.

III. Problem Statement

The studied problem is to find the shortest curvature-constrained 3D Dubins path between two configurations. In addition to the minimal turning radius of Dubins path, the inclination of the 3D path is limited according to the specific aircraft type. Thus, the state of the vehicle $\boldsymbol{q}$ is given by its position $(x, y, z) \in \mathbb{R}^3$, heading angle $\phi \in \mathbb{S}$, and pitch angle $\psi \in \mathbb{S}$. Hence, the configuration space is $\mathcal{C} = \mathbb{R}^3 \times \mathbb{S}^2$ and $\boldsymbol{q} \in \mathcal{C}$. The pitch angle is limited by its minimum ψ_{min} and maximum ψ_{max} value that constrains the inclination of the vehicle. The roll angle is not considered in the model as it assumed it does not influence the movement directly. The vehicle motion for a constant forward speed v can be described by motion equation

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = v \begin{bmatrix} \cos(\phi)\cos(\psi) \\ \sin(\phi)\cos(\psi) \\ \sin(\psi) \\ u_1 \\ u_2 \end{bmatrix}, \tag{1}$$

$$\psi \in [\psi_{min}, \psi_{max}], \tag{2}$$

where u_1 and u_2 are the control inputs for heading and pitch angles. These control inputs are limited by the maximum curvature κ_{max} as, according to [11],

$$u_1^2 \cos^2(\psi) + u_2^2 \leq \kappa_{max}^2. \tag{3}$$

The problem is to determine the shortest possible 3D Dubins path $\boldsymbol{\Gamma} : [0, \mathcal{L}] \to \mathcal{C}$ from the initial $\boldsymbol{q}_{\mathrm{I}}$ to the final $\boldsymbol{q}_{\mathrm{F}}$ configuration, where $\mathcal{L}$ stands to the length of the path given the fixed speed v. Formally, the problem can be defined as the optimization Problem 1.

Problem 1 (Shortest 3D Dubins Path):

$$\min_{\Gamma} \quad \mathcal{L} \tag{4}$$

subject to

$$\Gamma(0) = \boldsymbol{q}_{\mathrm{I}}, \tag{5}$$

$$\Gamma(\mathcal{L}) = \boldsymbol{q}_{\mathrm{F}}, \tag{6}$$

$$\text{Equations (1), (2), and (3) are met.} \tag{7}$$

IV. Proposed Non-linear Optimization Formulation

The proposed method formulates the 3D Dubins path problem as a Non-linear Programming (NLP) optimization. The path is split into s similarly long segments with a fixed curvature and fixed origin of the turn, if not a straight line. The path connects $s+1$ intermediate configurations $Q = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_{i+1}\}$, where $\boldsymbol{q}_1 = \boldsymbol{q}_I$ and $\boldsymbol{q}_{s+1} = \boldsymbol{q}_F$. The configurations Q are optimized in the NLP formulation to find the shortest 3D Dubins path.

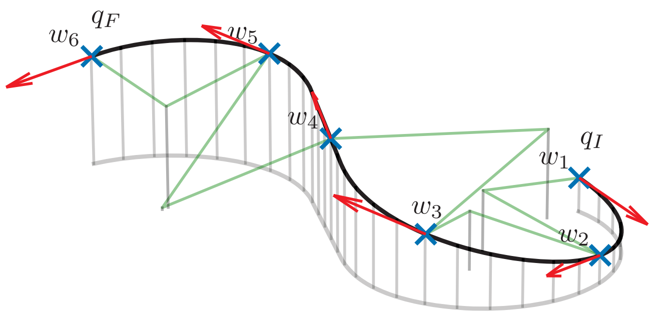

Fig. 2. Example of 3D Dubins path (black) between initial the q_I and final q_F configuration. The path is represented as s turn segments defined by $W = \{w_1, \ldots, w_{s+1}\}$ direction vectors (red) and length multiplicators $D = \{d_1, \ldots, d_s\}$. The green lines indicate the origin of the turn segment. Notice the turn segments are not in the same plane.

The proposed approach encodes the i-th path segment by the initial direction vector $\boldsymbol{w}_i$, final direction vector $\boldsymbol{w}_{i+1}$, and multiplicator d_i that determines the length of the segment. The vectors $W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{s+1}\}$ can be computed based on the ψ_i and ϕ_i angles at the i-th configuration $\boldsymbol{q}_i$ as

$$\boldsymbol{w}_i = \begin{bmatrix} \cos(\psi_i)\cos(\phi_i) \\ \cos(\psi_i)\sin(\phi_i) \\ \sin(\psi_i) \end{bmatrix}. \tag{8}$$

Note that the direction vector is always a unit vector, i.e., $\|\boldsymbol{w}_i\| = 1$. The vehicle's movement in the segment is determined based on the multiplicator d_i and the two end directions encoded by $\boldsymbol{w}_i$ and $\boldsymbol{w}_{i+1}$ allowing an arbitrary length of segments. Thus

$$\boldsymbol{q}_{i+1}^{xyz} = \boldsymbol{q}_i^{xyz} + d_i(\boldsymbol{w}_i + \boldsymbol{w}_{i+1}), \tag{9}$$

where $\boldsymbol{q}_{i+1}^{xyz}$ stands for the 3D position of the initial configuration $\boldsymbol{q}_i$ of the i-th segment. An example of the path defined by the directions W and multiplicators $D = \{d_1, \ldots, d_s\}$ is depicted in Fig. 2.

The problem is to find a curvature-constrained path, and thus it is necessary to determine the actual curvature of each segment. First, the turn angle α_i is determined based on the dot product of the two consecutive directions

$$\alpha_i = \arccos(\boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1}). \tag{10}$$

Then, the curvature κ_i of the i-th segment depends on the angle α_i as

$$|\kappa_i| = \frac{\tan\left(\frac{\alpha_i}{2}\right)}{d_i} = \frac{\sqrt{1 - \boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1}}}{d_i\sqrt{1 + \boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1}}}. \tag{11}$$

Having the introduced preliminaries, the 3D Dubins path problem can be reformulated as an NLP optimization problem with the W and D vectors representing the whole path consisting of s arc (or straight) segments as Problem 2.

Problem 2 (NLP formulation of 3D Dubins path):

$$\min_{W,D} \sum_{i=1}^{s} \mathcal{L}(d_i, \boldsymbol{w}_i, \boldsymbol{w}_{i+1}), \tag{12}$$

subject to

$$\boldsymbol{w}_1 = \boldsymbol{w}_I, \quad \boldsymbol{w}_{s+1} = \boldsymbol{w}_F; \tag{13}$$

$$\boldsymbol{q}_{\mathrm{I}}^{xyz} + \sum_{i=1}^{s} (d_i(\boldsymbol{w}_i + \boldsymbol{w}_{i+1})) = \boldsymbol{q}_{\mathrm{F}}^{xyz}; \tag{14}$$

$$\|\boldsymbol{w}_i\|^2 = 1, \quad i = 1, \ldots, s+1; \tag{15}$$

$$1 - \boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1} \leq d_i^2\, \kappa_{max}^2 (1 + \boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1}), \quad i = 1, \ldots, s; \tag{16}$$

$$\boldsymbol{w}_i^z \in [\sin(\psi_{min}), \sin(\psi_{max})], \quad i = 1, \ldots, s+1; \tag{17}$$

$$d_i = \rho_i\, d_{i-1}, \quad i = 1, \ldots, s-1. \tag{18}$$

The objective function (12) sums the lengths $\mathcal{L}_i$ of all segments determined by the following equation based on the direction vectors and the multiplicator corresponding to the specific segment

$$\begin{aligned} \mathcal{L}_i(d_i, \boldsymbol{w}_i, \boldsymbol{w}_{i+1}) &= \\ &= \frac{\arccos(\boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1})}{\kappa_i} \\ &= d_i \arccos(\boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1}) \frac{\sqrt{1 + \boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1}}}{\sqrt{1 - \boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1}}}. \end{aligned} \tag{19}$$

The proposed non-linear formulation contains $4s + 3$ variables for s segments constrained by a set of equations. First, the initial and final directories $\boldsymbol{w}_I$ and $\boldsymbol{w}_F$ in (13) are determined based on (8), and the direction vectors $\boldsymbol{w}_i$ are ensured to be unit vectors in (15). The final position $\boldsymbol{q}_{\mathrm{F}}^{xyz}$ is constrained in (14) based on the initial position $\boldsymbol{q}_{\mathrm{I}}^{xyz}$ and accumulates all the movements determined by (9). The curvature of each segment determined by (11) is constrained by κ_{max} in (16). The pitch angle is constrained in (17).

The last constraint (18) permits multiplicators of $\boldsymbol{w}_i$ are not equal; so, the formulation is more general. It is beneficial when an existing path initializes the solution. Once the initial path is sampled uniformly, the multiplicators are calculated according to the sampled configurations. Then, the ratios ρ_i can be fixed. Alternatively, the approach allows less dense sampling for a long straight line segment to reduce the computational burden while preserving a high precision.

A. Approximation of the Objective Function

If the number of segments s is high enough, the objective function can be approximated directly using the multiplicators d_i. The turn angle of segments becomes very small, and it goes to zero for an infinite number of segments

$$\lim_{s\to\infty} \boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1} = 1. \tag{20}$$

Then, the length of a single segment from (19) can be approximated with a substitution $p = \boldsymbol{w}_i \cdot \boldsymbol{w}_{i+1}$ and assuming p goes to 1, the equation become

$$\mathcal{L}(d_i, \boldsymbol{w}_i, \boldsymbol{w}_{i+1}) \approx \lim_{p \to 1} d_i \arccos(p) \frac{\sqrt{1+p}}{\sqrt{1-p}} \tag{21}$$

that can be reduced to

$$\mathcal{L}(d_i, \boldsymbol{w}_i, \boldsymbol{w}_{i+1}) \approx 2\, d_i\, . \tag{22}$$

Hence, the objective function can be approximated as a linear combination of D.

B. Initialization of the proposed NLP Optimization

A proper initialization is crucial for an efficient path optimization because, otherwise, the solver might converge to a sub-optimal solution or might not converge at all. One of the benefits of the proposed formulation is that it can be initialized by any path using sampling that is not necessarily uniform. The initialization works as follows.

Given the configuration samples, the initial values of the direction vectors W can be directly computed using (8). Then, the multiplicators D are determined to minimize the error of the movement equation (9). The error can be caused by changing the curvature during a single segment that is not allowed in the proposed formulation. The issue can be reduced by using more segments, if necessary. Once the multiplicators are initialized, the ratios ρ_i are fixed to meet (18) and remain constant during the whole optimization.

Two different initialization methods have been examined. The first method is to initialize the proposed optimization by the decoupled approach [2]. The reported results indicate the proposed optimization improves the paths founds by the decoupled approach, albeit they are close to the optimum, which is measured as the relative gap to lower bound values of the optimal solution. It is because there is still a space for path improvement as not all paths have both horizontal and vertical parts consisting of Dubins path with the same curvatures at both ends, which is assumed in [2]. Here, it is worth noting that for the cases where the initialized segments contain both turn and straight parts, it might not be possible to determine the respective multiplicator d_i exactly, i.e., determine a single curvature for each segment.

The second initialization method is to used 2D Dubins path in xy-plane and interpolate the altitude (z coordinate) linearly between the initial and final altitude as

$$z_i = (\boldsymbol{q}_F^z - \boldsymbol{q}_I^z)\frac{i-1}{s}, \quad i = 1 \ldots s+1. \tag{23}$$

Although this approach is straightforward, the initial solution is not feasible because it does not account for the initial and final directions. However, the pitch angle is limited by its maximum values, and thus it might be close to the final solution. This simple initialization works well in cases where the altitude difference between the initial and final configurations is low.

V. Results

The proposed NLP-based solution of the 3D Dubins path has been empirically evaluated in several scenarios with randomly generated instances and compared to the currently best-performing method [2]. The proposed method is implemented in Julia programming language [17] using interior-point filter line-search algorithm from the `Ipopt` library [18] for solving the proposed non-linear optimization problem. All the results were computed using a single core of the Intel Xeon Scalable Gold 6146 processor.

For the empirical evaluations, 1250 random instances have been generated by selecting a fixed distance between the two configurations, and randomly selecting the height difference and the angles: $\boldsymbol{q}_{\mathrm{I}} = (0, 0, 0, \phi_{\mathrm{I}}, \psi_{\mathrm{I}})$ and $\boldsymbol{q}_{\mathrm{F}} = (x_{\mathrm{F}}, 0, z_{\mathrm{F}}, \phi_{\mathrm{F}}, \psi_{\mathrm{F}})$. The maximum curvature is set to $\kappa_{max} = 1$, and the pitch angle is limited to the interval $\left[-\frac{\pi}{10}, \frac{\pi}{10}\right]$. The directions are selected by the uniform distribution. Since the location of q_I is fixed, x_{F} also determines the distance between endpoints. The value of z_{F} is selected randomly from the interval $[0, 6z_{max}]$, where z_{max} is the maximum altitude achievable on a path between the two configurations disregarding the angles

$$z_{max} = \tan(\psi_{max})\, x_{\mathrm{F}}, \tag{24}$$

where x_{F} is the distance between the two configurations. The high difference z_{max} is almost always achievable, but it becomes a limiting factor at about $3z_{max}$, depending on the overall path length.

A. Path Length Improvement

The first empirical evaluation is focused on the examination of the solution quality. The proposed NLP-based solver is initialized by the decoupled method [2] also used as a baseline solution. Fig. 3 shows the relative path lengths to

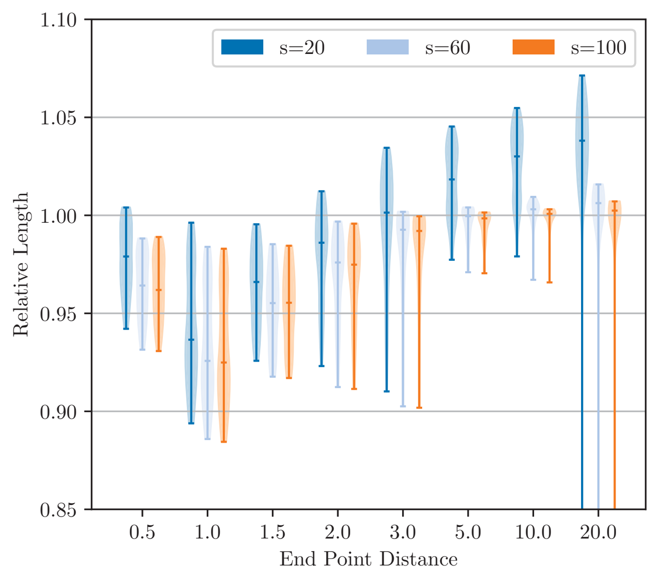

Fig. 3. Relative path lengths normalized to the length of the initialization path provided by [2]. The middle mark corresponds to the median and 10 % of best and worst solutions are removed to estimate 80 % non-parametric confidence interval.

the initial solutions. The results are computed for various distances between the endpoints x_F and the various segment counts $s \in [20, 60, 100]$. The mean values are marked for each of the cases. Further, $10\,\%$ of the best and worst results are discarded for visualization improvement, i.e., the minimum and maximum values estimate the $80\,\%$ non-parametric confidence interval.

The results indicate that for distances of the endpoints $x_F = 1$, the length improvement provided by the proposed NLP-based approach is up to about $10\,\%$, and the mean improvement is about $7\,\%$. It is also noticeable that a low number of segments ($s = 20$) is not sufficient for larger endpoint distances. It is probably influenced by an insufficient number of segments approximating the end turns because a long straight segment in the middle is expected.

Notice that the `Ipopt` solver may not found a solution because the algorithm does not converge in the selected limit of 500 iterations. Usually, the maximum number is not reached, but it prevents the algorithm from being stuck for cases with poor convergence. Such cases are not included in the presented results, and the fail rate is discussed in the following section.

B. Fail Rate

The employed `Ipopt` solver is not able to always converge to a feasible solution. There are various reasons for that, but the optimization algorithm reaches a point where it is impossible to converge to an optimum. First, we examined the influences of the endpoint distance and the number of samples. Fail rates are depicted in Fig. 4 for paths initialized

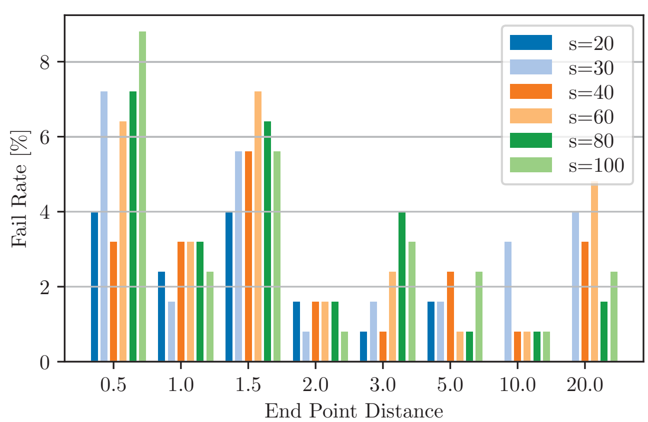

Fig. 4. Fail rate of the `Ipopt` solver to find a solution of the NLP-based optimization initialized by [2] according to distance between the endpoints and the number of segments n.

by the decoupled approach [2]. The results suggest that the failure rates are slightly higher for shorter paths, while an influence of the number of segments is not visible.

Secondly, the influence of the initialization using 2D Dubins paths has been studied. Thus, the NLP-solver is initialized by the decoupled method and using the 2D Dubins path. Besides, both initialization methods are examined, and the best solution is reported. The results are reported in Fig. 5 from which it can be observed that the initialization using the decoupled approach yields a lower failure rate than using

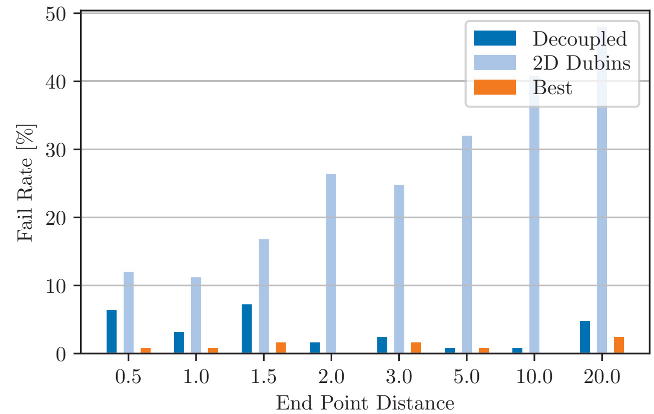

Fig. 5. Fail rate of the `Ipopt` solver to find a solution of the NLP-based optimization with $s = 60$ segments and initialization by the decoupled approach [2] and 2D Dubins path. The “Combined” tries both initialization and select the better one.

the 2D Dubins path. That is why the decoupled approach is considered a more suitable option, and it is utilized in all other results presented in this paper. However, the 2D Dubins path initialization can provide better convergence in specific cases. Therefore, the overall reliability is slightly improved if both initialization methods are combined, denoted “Combined” in the plot. Besides, the initialization using 2D Dubins paths might be preferable for instances with low altitude differences because it is easier to compute.

C. Computational Time

The computational time of the proposed NLP-based approach is reported in Fig. 6 for the increasing number of segments s. The time to get the initial decoupled path is included, but it is negligible compared to the optimization time. The computational requirements seem to depend exponentially on the number of segments s. Notice that the plot is semi-logarithmic.

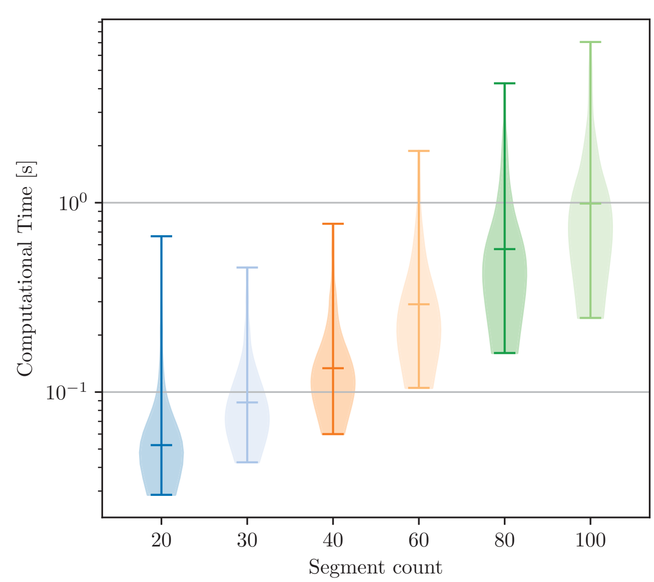

Fig. 6. Required computational times of the proposed NLP-based approach depending on the number of segments s.

D. Comparing Results to the Lower Bound

Based on the reported results, the path improvement using the proposed NLP-based approach may seem small compared to the initialization paths provided by the decoupled method [2]. The decoupled approach can provide paths within $2\,\%$ from the lower bound estimation of the optimal solution value that is also provided in [2]. Since some improvement can be observed in Fig. 3, we examine how much the proposed optimization can reduce the gap from the lower bound.

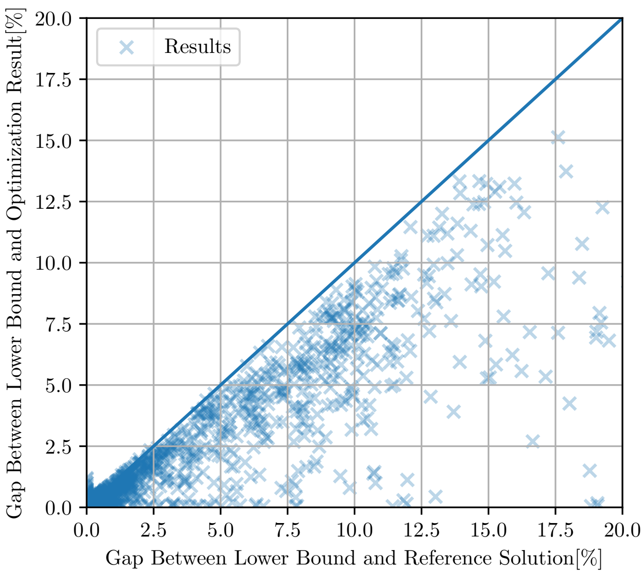

Fig. 7. Relative gap of the proposed NLP-based solution to the lower bound estimate of the optimal solution value and the initial decoupled path [2]. The line separates paths that are improved during the proposed optimization and unimproved cases. The number of path segments is $s = 100$, and the endpoint distance is randomly selected from the previously computed results. Note that there are several results with the original gap greater than $20\,\%$ that do not fit into the plot, but all of them are significantly improved by the proposed optimization.

The relative gap to the optimization solution and initial reference path found by the decoupled method [2] is shown in Fig. 7. The results show that the proposed method can improve the path if the initial gap is larger than 0.5% in almost all cases. Contrarily, if the initial gap is close to zero, there is no space for improvement, and the NLP-based method suffers from convergence issues. Thus the initial solution can be utilized as the final solution in such a case.

VI. Conclusion

In this paper, we introduce a novel NLP-based formulation of the 3D Dubins path problem with a limited pitch angle. The proposed approach splits the 3D path into several segments, each with constant curvature. The 3D path is encoded by intermediate direction vectors (instead of vehicles' states directly) that reduce the number of variables and support faster convergence of the used NLP solver. The proposed optimization can improve initial solutions noticeably. Although the existing decoupled method provides solutions within $2\,\%$ from the lower bound, the proposed approach further reduces the gap by about $30\,\%$. Furthermore, if the distance of the endpoints is close to the length of the turning radius, the median total improvement is about $7\,\%$. The computational requirements are relatively low, and solutions are found in less than one second, which motivates to employ the proposed solution in more complex planning problems such as multi-goal planning. In our future work, we plan to optimize paths connecting multiple locations as the first step towards combinatorial multi-goal planning, where the sequencing needs to be addressed.

References

[1] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[2] P. Váňa, A. Alves Neto, J. Faigl, and D. G. Macharet, "Minimal 3d dubins path with bounded curvature and pitch angle," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8497–8503.

[3] J.-D. Boissonnat, A. Cérézo, and J. Leblond, "Shortest paths of bounded curvature in the plane," *Journal of Intelligent and Robotic Systems*, vol. 11, pp. 5–20, 1994.

[4] P. Bevilacqua, M. Frego, D. Fontanelli, and L. Palopoli, "A novel formalisation of the markov-dubins problem," in *European Control Conference (ECC)*, 2020, pp. 1987–1992.

[5] S. Hota and D. Ghose, "Optimal geometrical path in 3d with curvature constraint," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 113–118.

[6] D. Mittenhuber, "Dubins' problem is intrinsically three-dimensional," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 3, pp. 1–22, 1998.

[7] S. Hota and D. Ghose, "Optimal path planning for an aerial vehicle in 3d space," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 4902–4907.

[8] Y. Lin and S. Saripalli, "Path planning using 3d dubins curve for unmanned aerial vehicles," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 296–304.

[9] H. Chitsaz and S. M. LaValle, "Time-optimal paths for a dubins airplane," in *IEEE Conference on Decision and Control*, 2007, pp. 2379–2384.

[10] M. Owen, R. W. Beard, and T. W. McLain, *Implementing Dubins Airplane Paths on Fixed-Wing UAVs**. Dordrecht: Springer Netherlands, 2015, pp. 1677–1701.

[11] Y. Wang, S. Wang, M. Tan, C. Zhou, and Q. Wei, "Real-time dynamic dubins-helix method for 3-d trajectory smoothing," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 730–736, 2015.

[12] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, E. De Lellis, and A. Pironti, "Path generation and tracking in 3-d for uavs," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 980–988, 2009.

[13] P. Pharpatara, B. Hérissé, and Y. Bestaoui, "3d-shortest paths for a hypersonic glider in a heterogeneous environment," *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 186–191, 2015.

[14] A. A. Neto, D. G. Macharet, and M. F. M. Campos, "3d path planning with continuous bounded curvature and pitch angle profiles using 7th order curves," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4923–4928.

[15] A. A. Neto and M. F. Campos, "On the generation of feasible paths for aerial robots with limited climb angle," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 2827–2877.

[16] A. Askari, M. Mortazavi, H. Talebi, and A. Motamedi, "A new approach in uav path planning using bezier–dubins continuous curvature path," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 230, no. 6, pp. 1103–1113, 2016.

[17] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.

[18] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.