CEAS Aeronautical Journal
https://doi.org/10.1007/s13272-023-00691-w

ORIGINAL PAPER

Bio-inspired altitude changing extension to the 3DVFH* local obstacle avoidance algorithm

Karolin Thomessen[1] · Andreas Thoma[1,2] · Carsten Braun[1]

Received: 28 February 2023 / Revised: 22 September 2023 / Accepted: 5 October 2023

Abstract
Obstacle avoidance is critical for unmanned aerial vehicles (UAVs) operating autonomously. Obstacle avoidance algorithms either rely on global environment data or local sensor data. Local path planners react to unforeseen objects and plan purely on local sensor information. Similarly, animals need to find feasible paths based on local information about their surroundings. Therefore, their behavior is a valuable source of inspiration for path planning. Bumblebees tend to fly vertically over far-away obstacles and horizontally around close ones, implying two zones for different flight strategies depending on the distance to obstacles. This work enhances the local path planner 3DVFH* with this bio-inspired strategy. The algorithm alters the goal-driven function of the 3DVFH* to climb-preferring if obstacles are far away. Prior experiments with bumblebees led to two definitions of flight zone limits depending on the distance to obstacles, leading to two algorithm variants. Both variants reduce the probability of not reaching the goal of a 3DVFH* implementation in Matlab/Simulink. The best variant, 3DVFH*b-b, reduces this probability from 70.7 to 18.6% in city-like worlds using a strong vertical evasion strategy. Energy consumption is higher, and flight paths are longer compared to the algorithm version with pronounced horizontal evasion tendency. A parameter study analyzes the effect of different weighting factors in the cost function. The best parameter combination shows a failure probability of 6.9% in city-like worlds and reduces energy consumption by 28%. Our findings demonstrate the potential of bio-inspired approaches for improving the performance of local path planning algorithms for UAV.

Keywords UAV · Obstacle avoidance · Autonomy · Local path planning

1 Introduction

Due to technical advances in the field of unmanned aerial vehicles (UAVs), the number of use cases for UAV missions increased over the last few years. These use cases include maintenance, surveillance, and delivery [1–3]. UAV steering can be done remotely or autonomously. The higher the level of autonomy becomes, the higher the need for a safe and reliable obstacle avoidance algorithm to avoid crashes and injuries will be.

Obstacle avoidance covers obstacle detection, path planning, and maneuvering. Sensor systems have already been widely researched and developed over the past years. The path planning aspect of obstacle avoidance has the greatest potential for improvement. Path planning aims to find an optimal and efficient connecting path from a starting point to a goal point. Path planning either follows a global or local approach. A global path planning algorithm relies on global information about the UAV's environment. Therefore, the quality of the information about the environment limits global path planning algorithms. On the contrary, local path planning uses only the UAV's local sensor information. Its goal is to make the best progress based on local information to reach the goal. However, the limited information about the environment restricts the quality of the local path planning algorithm [4]. An advantage of the local approach is that unforeseen obstacles, for example, newly built houses that do not occur on a map, can be avoided. Therefore, local path planning algorithms often represent a backup strategy for evading globally unknown obstacles [4]. Because the sensor range limits the local planner, it can plan into dead ends, which is problematic.

A popular local path planning algorithm is the 3DVFH* [5]. This algorithm is also used in the px4 flight controller, a

✉ Karolin Thomessen
karolin.thomessen@alumni.fh-aachen.de

[1] FH Aachen, Aachen, Germany

[2] RMIT University, Melbourne, Australia

Published online: 18 November 2023

widely used flight controller for small UAVs. It combines a purely local approach to find the best local direction with the A* global path-finding algorithm. The general idea behind the 3DVFH* is to detect obstacle-free directions, choose the best direction based on a cost function, and evaluate this chosen direction by applying the A* search algorithm. Therefore, the algorithm also examines what would happen if the UAV moved in a specified direction. It also has a memory strategy to remember previously seen points. Examining possible directions and the memory strategy are advantages of the 3DVFH*. However, the algorithm performs poorly in cluttered environments, since its cost function prefers a horizontal goal-driven evasion strategy [6].

Animals usually show optimal and efficient behavior, because they have had thousands of years of evolution to learn from. Therefore, observing animals or plants and adapting their strategies to technological systems is widely used in developing new technology strategies [7]. In several optimization problems, bio-inspired algorithms are better suited than the theoretical mathematical approaches, for example, the Genetic Algorithm or the Particle Swarm Optimization [8]. Flight strategies are also often implemented using inspiration from flying insects, for example, honeybees [9]. Bumblebees also often serve as a source of inspiration since they are big and heavy and, therefore, must have an efficient obstacle avoidance strategy. Experiments with bumblebees have revealed that they tend to evade close obstacles horizontally, whereas far-away obstacles are evaded vertically [10]. This knowledge builds the foundation for a bio-inspired adaption of the 3DVFH*, in which obstacles should be vertically evaded if they are far away.

This paper presents the main functionalities of the 3DVFH* algorithm, the bio-inspired adaptions to this foundational algorithm, and how this novel algorithm improved different UAV missions.

2 Background

In robotics, a path usually describes the connection between a start and goal point through an environment known as the configuration space. Path planning can be divided into global and local path planning [11]. Figure 1 shows the classification used in this work. Bio-inspired path planning is a growing field of interest, and is also discussed in this chapter.

2.1 Global path planning

In global path planning, the global environment is known to a full extent [12]. Therefore, the global path's quality first depends on the quality of the environmental data and the algorithm's performance. Global path planning algorithms divide into sample-based and graph-based approaches, which Fig. 1 visualizes. The sample-based strategies search for the best path by sampling the configuration space while also performing collision detection [13]. Examples of sample-based path planning approaches are RRT and RRT* [14]. Comparing the RRT* with the RRT, the RRT* produces more optimal paths, but is more time-consuming [15]. The idea behind graph-based path planning algorithms is that they explore the configuration space by searching for the best possible path in a graph [14]. Examples of graph-based path planning algorithms are A*, Theta*, and D* [14]. Although the A* was developed over 30 years ago, it is still a widely used path planning algorithm due to its simplicity [12, 16]. The A* searches a configuration space by expanding nodes from a starting node and applying a cost function to analyze the new nodes [17]. The cost function contains a general term, the distance to the newly expanded node, and a heuristic term, the distance toward the goal. The best node with the lowest cost is then expanded again. Repeating these steps leads to a path toward the goal.

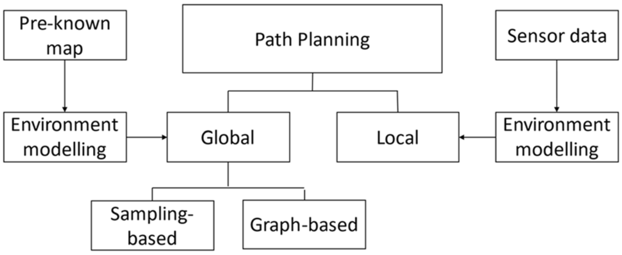

Fig. 1 Classification of path planning

2.2 Local path planning

A local path planning algorithm, also called reactive path planning, calculates the path based on the current sensor information and has no information on the environment behind that sensor range [12]. Therefore, the sensor range limits the local path planning algorithm [4]. Still, using local information enables the local path planning algorithm to operate in unknown environments, which makes reliable local path planning one of the most important steps toward fully autonomous flight.

In the group of local path planning algorithms, the 3DVFH* plays a major role. It is one of the most popular local path planning algorithms, which T. Baumann introduced in [5]. As it is integrated into the flight controller px4's avoidance software [18], it can be seen as state-of-the-art for local path planning in a three-dimensional space. The 3DVFH* combines the two-dimensional local path planning algorithm VFH, which was introduced in [19], and the global path planning algorithm A*, and adapts it to a 3D space. For environmental modeling, the 3DVFH* uses a grid-based representation in the form of a two-dimensional polar histogram. The 3DVFH* also has a memory strategy, which uses a two-dimensional polar histogram, called a memory histogram, containing the points seen in several previous timesteps. The general idea is to find the best possible flight direction in this histogram representation and evaluate this choice by applying the A* search algorithm.

As shown in Fig. 2, the first step of the algorithm is building the histogram, which the path planner uses later. Therefore, the current sensor information is transferred into the histogram. Every histogram cell displays a certain area in the spherical surroundings of the drone. Histogram cells can either be free or occupied, while also holding information about the distance to the closest obstacle in that area. The memory histogram contains information about the age of the information in the histogram cells. Combining this histogram with the memory histogram, containing the information from the last several timesteps, leads to the combined histogram. The A* search algorithm, which presents the next step of the algorithm in Fig. 2, uses this combined histogram to calculate the best path, which is defined by the cost function. Each node that the A* algorithm evaluates displays one histogram cell. After every expansion of the best nodes in the A* algorithm, safety measures are applied. These safety measures include ground avoidance and sphere avoidance. The sphere avoidance assures that a spherical region around an occupied node is blocked. The A* search tree expands until reaching the maximum number of allowed nodes or until the best node lies within the goal region. The flight controller then receives the best path.

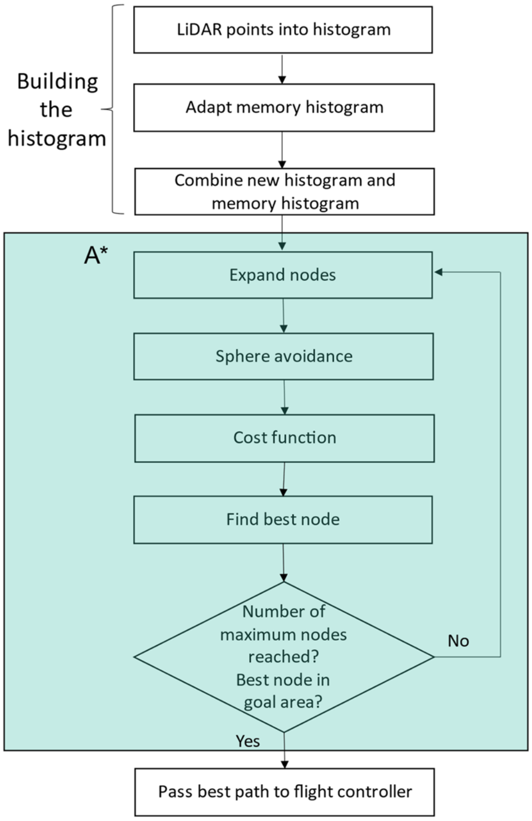
Building the histogram
LiDAR points into histogram
Adapt memory histogram
Combine new histogram and memory histogram
A*
Expand nodes
Sphere avoidance
Cost function
Find best node
Number of maximum nodes reached? Best node in goal area?
No
Yes
Pass best path to flight controller

Fig. 2 Structure of the implemented 3DVFH* algorithm

The core of the 3DVFH* is the cost function, which evaluates possible paths and attempts to find the best suitable path. The cost function is the third step of the A* algorithm as visualized in Fig. 2. The cost function described here is derived from the px4 avoidance algorithm [18], which uses the 3DVFH*. It consists of four general cost parameters and the heuristic cost. The four general cost parameters include yaw cost, pitch cost, velocity cost, and obstacle cost. The heuristic cost describes the distance from the considered node to the goal. All cost parts have weighting factors to vary the weight of the different costs, which are indicated by the variable k in the equations describing the cost terms. The cost parameters are calculated for every node not blocked by the safety measures. Each node has a pitch angle and a yaw angle. The pitch angle is the elevation angle, and the yaw angle is the azimuth angle, which were assigned to the nodes in the expanding nodes' process. The angles are in a local coordinate system.

The yaw cost considers the difference between the yaw of the currently considered node yaw_{node} and the yaw of the goal yaw_{goal}. The less difference between the two angles, the smaller the value for yaw cost. The following equation gives the yaw cost $cost_{yaw}$:

$$cost_{yaw} = k_{yaw} * \left(yaw_{node} - yaw_{goal}\right)^2. \tag{1}$$

The pitch cost $cost_{pitch}$ represents the difference between the pitch of the currently considered node $pitch_{node}$ and the pitch of the goal $pitch_{goal}$ and is calculated by the following equation. Large differences in the pitch angles lead to a high pitch cost

$$cost_{pitch} = k_{pitch} * \left(pitch_{node} - pitch_{goal}\right)^2. \tag{2}$$

The velocity cost term $cost_{velocity}$ evaluates the chosen paths' smoothness by determining the straightness between two branches of the tree of waypoints. It considers the dot product of the normalized position of the next node p_{node}, and the velocity of the UAV to the current node, v. Since the dot product can also be described as the multiplication of the magnitudes of the two vectors and the cosine of the angle between them, the velocity cost term characterizes the influence of the direction of motion. Large differences in the angle between the position vector and the velocity vector lead to high values for the velocity cost term. The following equation gives the velocity cost:

$$cost_{velocity} = k_{vel} * \left(|v| - \frac{p_{node}}{|p_{node}|} \cdot v \right). \tag{3}$$

The following equation describes the obstacle cost $cost_{obst}$. $d_{histogram}$ describes the distance of the node to the next obstacle in that particular histogram field. The weighting factor k_{obst} is a distance value and serves as an inflection point. If a distance is closer than this value, the cost value for this node increases. The cost value $cost_{obst}$ utilizes the difference of k_{obst} and $d_{histogram}$ in an empirical function. The obstacle cost parameter only exists for occupied histogram cells

$$cost_{obst} = 5000 * \left(1 + \frac{d}{\sqrt{1 + d * d}} \right)$$

$$d = k_{obst} - d_{histogram}. \tag{4}$$

The following equation summarizes the calculation of the cost of a node:

$$cost_{total} = cost_{yaw} + cost_{pitch} + \mathrm{cost}_{velocity} + cost_{obst}. \tag{5}$$

After applying the cost function, the heuristic cost term is added to the nodes with the lowest cost.

2.3 Bio-inspired path planning

Bio-inspired algorithms utilize the behavior of animals, plants, or evolutional aspects to find the best path. Path planning usually consists of different available paths and a cost function to evaluate them and decide on the best available path. The minimization of the cost function also poses an optimization problem. In the scope of optimization problems, several bio-inspired algorithms exist, which can be classified into evolutionary algorithms, swarm intelligence-based algorithms, behavior-based algorithms, and neuronal network inspired algorithms [20]. As a great number of these algorithms exist, this work will only highlight a few important algorithms. Evolutionary algorithms, for example the Genetic Algorithm (GA), introduced by Holland [21], use aspects of evolution as selection, cross-overs, and mutations for the evaluation of possible paths. The genetic algorithms starts with a random initial starting population, evaluates them with a cost function, and then creates a new population using mutations and cross-overs. Using mutations and cross-overs ensure a higher probability of avoiding local minima. Various implementations and variations of evolutionary algorithms in UAV applications exist [22, 23].

Another category of bio-inspired path optimization algorithms is based on swarm intelligence, as for example the Particle-Swarm-Optimization (PSO), or Ant Colony Optimization (ACO) [24]. The ACO is inspired by the pheromone communication of ants, which helps them to find the shortest path between their home and a feeding source [14].

Other algorithms use the bio-inspiration in the obstacle avoidance part of the path planning problem. They adapt obstacle avoidance behavior of animals or plants to find the best path. This usually implies the adaption of the visual system of animals, as for example in [9], where honeybees serve as an inspiration. The behavior of plants can also serve as bio-inspiration. In [25], the authors propose a plant-inspired path planning approach for obstacle avoidance in a UAV scenario. They tested their algorithm in simple simulations and flight test, but have not analyzed the behavior in a cluttered environment.

3 Method

Experiments with bumblebees performed by A.Thoma et al. showed that they tend to evade an obstacle vertically if detected as far away and horizontally if the obstacle is close [10], which suggests that a vertical evasion strategy is more efficient for far-off obstacles than a horizontal evasion strategy. A. Thoma et al. implemented this knowledge by changing the yaw and pitch cost weighting factors of the 3DVFH* [26]. They concluded that this adaption of the yaw and pitch cost weighting factors was not very useful for complex city-like environments, because the weighting factors' adaptions are very sensitive. However, the general results that altering the weighting cost parameters for a vertical evasion strategy can improve the success rate indicate that the strategy of vertically evading obstacles is promising to improve the success rate of flight missions. However, it must also be further developed to work in city scenarios. Two steps must be followed to implement the findings from the bumblebee experiments into the 3DVFH*: First, it must be defined when to follow different evasion strategies. Second, it must be defined how these strategies, focusing on the vertical evasion strategy, can be introduced in the 3DVFH*.

3.1 Defining behavior change distances

The bumblebee flights in [10] showed that bumblebees use a vertical evasion strategy for far-away obstacles and a horizontal evasion strategy for close obstacles. In between, they tend to mix horizontal and vertical evasion strategies. Different evasion zones can be determined from these experiments, in which different evasion strategies apply. The bumblebee flights are further analyzed to derive the boundaries for these evasion zones.

Resulting from the bumblebee experiments in [10], A. Thoma et al. defined the relationship between distance to the obstacle and evasion strategy Bumblebees had to fly through a tunnel containing one obstacle. The dimension and placement of the obstacle in the tunnel varied. The bumblebees evaded the obstacle of a size of 20 cm*25 cm (width*height) horizontally if their distance to the obstacle was 15 cm. If their distance to the obstacle was 40 cm, the bumblebees vertically evaded it. In another situation, the bumblebees had to avoid an obstacle with a dimension of 25 cm*25 cm. The bumblebees also evaded this obstacle vertically if the distance to the obstacle was 40 cm, but they evaded it horizontally if it was about 5 cm away. These findings must be transferred to the UAV scenario. Therefore, the sizes of the bumblebees and the UAV are considered to approximate the decision distance, at which the UAV needs to decide which strategy it applies. The wingspan of a bumblebee, *bombus terrestris*, is about 2.8 cm [27]. The diameter of the UAV is about 50 cm. Considering the distance for vertical evasion for the bumblebees results in a rounded decision distance $d_{vertical}$ for the vertical evasion strategy of 7 m, which can be calculated with the following equation:

$$d_{vertical} = 50cm * \frac{40cm}{2.8cm} = 714cm \approx 700\boldsymbol{cm}. \tag{6}$$

This is done analogically for the horizontal evasion strategy, which results in a decision distance for horizontal evasion $d_{horizontal}$ of 3 m. This can be traced with the following equation:

$$d_{horizontal} = 50cm * \frac{15cm}{2.8cm} = 268 \approx 300\boldsymbol{cm}. \tag{7}$$

The 3 m resulted from the obstacle with the dimensions 20 cm*25 cm (w*h). This represents an obstacle that is rather high than wide. For the other obstacle with the dimension 25 cm*25 cm equation 7 gives a distance of 1 m for horizontal evasion.

Therefore, two categories for the decision distances defining the zones in which different evasion strategies are applied exist, summarized in Table 1. These two categories lead to two algorithms, now called 3DVFH*b-a and 3DVFH*b-b.

If an obstacle is more than 7 m away, the UAV should evade vertically; if an obstacle is closer than 3 m or 1 m (depending on the algorithm), the UAV should evade horizontally. A mixture of horizontal and vertical evasion strategies is applied between those zones, performed by linear interpolation.

Table 1 Categories for the decision distances

Algorithm	$d_{horizontal}$	$d_{vertical}$
3DVFH*b-a	3 m	7 m
3DVFH*b-b	1 m	7 m

3.2 Implementing the climb strategy

After defining the flight behavior zones, the next step is determining how the UAV will adapt its flight behavior to them. The pitch cost parameter is modified to get a strategy that enables the UAV to climb in different areas and adjust its pitch angle. The pitch cost function in Eq. (2) considers the difference between the pitch of a node and the goal direction, leading to a constant draw of the UAV toward the goal level. Specifically, the pitch cost function in Eq. (2), which calculates the difference between the pitch of a node and the goal direction, is adjusted to prevent the UAV from being constantly drawn toward the goal level. Instead, the UAV should be guided toward a different pitch angle that allows it to navigate over obstacles. The optimal pitch angle should direct the UAV to a point above the obstacle, allowing it to continue its flight safely. The pitch angle$pitch_{goal}$, in the pitch cost function, is therefore replaced by an angle, $pitch_{optimal}$, that lies above the detected obstacle

$$cost_{pitch} = k_{pitch} \cdot \left(pitch_{node} - pitch_{optimal}\right)^2. \tag{8}$$

The maximum angle the UAV can detect obstacles in is 22° up and 22° down due to its sensor limits in this work. A maximum of 35° is blocked in the histogram with the sphere avoidance. As this would result in a low climb rate, an offset of 40° is added. This ensures a better avoidance of obstacles with bigger dimensions.

In the transition area where a combination of vertical and horizontal evasion is applied, the optimal pitch angle is altered to be on a linear function between the optimal angle above an obstacle and the goal direction. Whether the UAV is closer to the vertical or horizontal evasion area, the optimal angle is closer to the optimal pitch angle above the obstacle or to the goal pitch angle. If no obstacle is detected, the UAV applies the horizontal evasion strategy because climbing would be inefficient if no obstacle is seen. The algorithm decides which strategy to follow for each timestep based on the distance to the obstacle in goal direction.

The weighting factors for the cost parameters are also significant for the success of the path planning algorithm. The default values of the 3DVFH* are displayed in Table 2. For the novel algorithm, the pitch cost weighting factor remains at 25, supporting the novel strategy of following a

Table 2 Values of 3DVFH* and 3DVFH*b

Parameter	3DVFH*–default	3DVFH*b
Pitch cost	25	25
Yaw cost	3	10–3
Velocity cost	6000	6000
Obstacle cost	8.5	7

new angle. The obstacle cost weighting factor is changed to 7, as a prior study evaluated this value as better suited than the previous value of 8.5. The yaw cost weighting factor is changed to 10 in the vertical evasion zone and remains at 3 for the horizontal evasion zone. The velocity cost weighting factor also remains at 6000.

3.3 Parameter study

The parameter setting greatly influences the failure probability of the path planning algorithm [28]. Therefore, a parameter study is conducted to analyze the effect of the parameters yaw cost, obstacle cost, and velocity cost on the failure probability. The choice of the parameter values was derived from the work in [28]. The values for the obstacle cost parameter are 5 and 8.5. The velocity cost values are 6000 and 18,000. As the yaw cost directly influences flight behavior, the values are changed separately for the flight strategies. The first value describes the vertical flight strategies value, and the second value describes the horizontal flight strategies value. The values for the yaw cost are: 10–3, 10–1, 7–3, and 7–1. Combining all parameter values for the three chosen cost parameters leads to 16 combinations. Table 3 presents all parameter combinations. Additionally, a seventeenth parameter combination resulting from averaging the cost parameter values completes a full factorial surface Design of Experiments (DoE). Six further center points of the edges of the DoE are chosen to analyze the influence of the parameter combinations more deeply (Combinations 18 to 23).

Table 3 Parameter combinations for the parameter study

No	Yaw cost	Obstacle cost	Velocity cost
1	10–3	5	6000
2	10–3	5	18,000
3	10–3	8.5	6000
4	10–3	8.5	18,000
5	10–1	5	6000
6	10–1	5	18,000
7	10–1	8.5	6000
8	10–1	8.5	18,000
9	7–3	5	6000
10	7–3	5	18,000
11	7–3	8.5	6000
12	7–3	8.5	18,000
13	7–1	5	6000
14	7–1	5	18,000
15	7–1	8.5	6000
16	7–1	8.5	18,000
17	8.5–2	7	12,000
18	10–3	7	6000
19	10–3	7	12,000
20	10–3	7	18,000
21	7–3	7	6000
22	7–3	7	12,000
23	7–3	7	18,000

3.4 Experimental setup

Flight missions in two different environments, here referred to as “worlds”, are simulated in a Software-In-The-Loop simulation in Matlab/Simulink using the novel algorithms. S. Seker created 1408 city-like worlds, and 900 simple worlds [29]. The simple worlds consist of thin cuboids representing walls, whereas city worlds contain larger cuboids that represent houses in towns or blocks in larger cities. Figure 3 shows an example of a simple and city world. Different characteristics of the worlds vary, for example, the number of cuboids and the size of the obstacle field. The city worlds contain a random number of cuboids, which are lined up, so that they represent streets. In the city worlds, the height of the obstacles and the number of streets vary, for example. Figure 4 shows an example of a flight path in a city world.

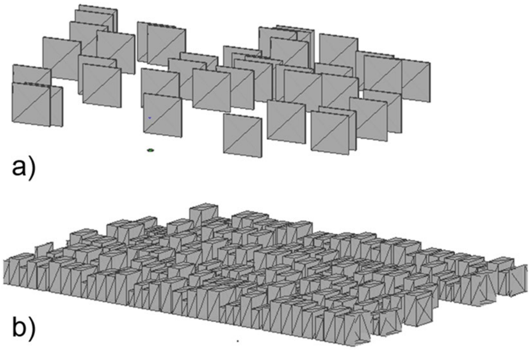

Fig. 3 Examples of two worlds: **a** represents a simple world, and **b** represents a sector of a city world

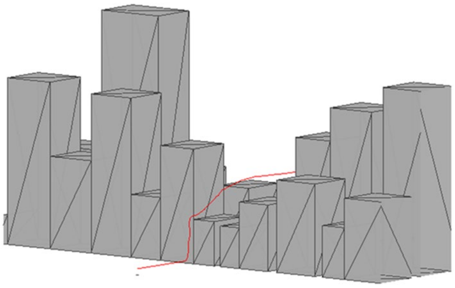

Fig. 4 Example of city world with flight path

3.5 Evaluation criteria

Each world is simulated once for each algorithm or parameter combination. The failure probability serves as a measurement to evaluate the different algorithms. A flight fails if the UAV does not reach the goal, which means that the path planning algorithm cannot find a path to the goal in that specific world. Therefore, a flight either failed or was successful. In a group of flights n, there is a group of failed flights n_f, which represents the flights that did not reach the goal or crashed. The following equation defines the failure probability f of a set of flights n as

$$f = \frac{n_f}{n}. \tag{9}$$

The baseline for the evaluation is the implementation of the original 3DVFH* in MATLAB/Simulink, as described and validated in [28]. The group of flights in the simple worlds has a failure probability of 18.1% and a failure probability of 70.7% in the city worlds.

4 Results and discussion

For evaluating the novel algorithm 3DVFH*b in its two variants, flights were simulated using the experimental setup described in 3.4. The best of these two versions builds the foundational algorithm for the parameter study. The parameter study only performs in the city worlds, because these worlds present higher obstacles that challenge the algorithm.

4.1 3DVFH*b

For the first evaluation of the novel 3DVFH*b algorithm, the failure probabilities of the two algorithm variants in the simple worlds are displayed in Fig. 5. The baseline has a failure probability of 18.1% in the simple worlds. Thirty-three worlds failed using the 3DVFH*b-a, leading to a failure probability of 3.67%. The 3DVFH*b-b has a failure probability of 1.4%, which equals 13 failed worlds. Both variants of the 3DVFH*b algorithm greatly reduced the failure probability of the 3DVFH* in the simple worlds. The 3DVFH*b-b showed a greater effect on reducing the failure probability than the 3DVFH*b-a, which implies the superiority of a shorter horizontal evasion zone and, therefore, an enhanced vertical evasion strategy in the in-between zone.

Figure 6 shows the failure probabilities of the three algorithms in the city worlds. The baseline has a failure probability of 70.7%. The 3DVFH*b-a reaches a failure probability of 19.6% and the 3DVFH*b-b has a failure probability of 18.6%. The low failure probabilities show that both variants of the 3DVFH*b algorithm notably reduced the failure probability of the 3DVFH* baseline algorithm.

The analysis of the failure probabilities of the 3DVFH*b variants shows that the 3DVFH*b-b performed better in both world types. This shows that a smaller horizontal evasion zone, and, therefore, following the vertical evasion slightly longer, yields better results in terms of failure probability.

Furthermore, the traveled distance and the energy content of the two variants of the new algorithm are essential factors when analyzing the efficiency of a flight path. These two values are compared in the worlds where both algorithms could find a path to the goal. Both traveled distance and energy are considered better for one algorithm, lower than 5% of the respective value. Both algorithms perform equally in terms of traveled distance in the simple worlds. 76% of the successful flights have the same traveled distance and each algorithm has shorter flight paths in 12% of the flights. In the city worlds, the 3DVFH*b- a produces shorter flight paths in 40% of the flights, compared to 2% for the 3DVFH*b-b. In 74% of the shorter flights of the 3DVFH*b-a, the maximum flight height was also lower for the 3DVFH*b-a. This shows that the 3DVFH*b-b tends to fly higher than the 3DVFH*b-a, which then produces longer flight paths.

Comparing the energy also underlines the statement that the paths of the 3DVFH*b-a are more efficient. Flights in 773 out of the 900 worlds were successful using both algorithms. Here, the 3DVFH*b-a has a lower energy consumption in 27% of the successful flights, compared to 11% of the

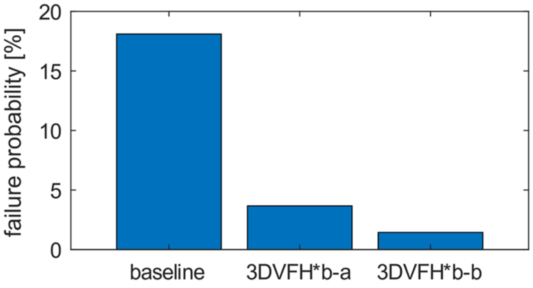

Fig. 5 Failure probabilities of the baseline, the 3DVFH*b-a, and 3DVFH*b-b in the simple worlds

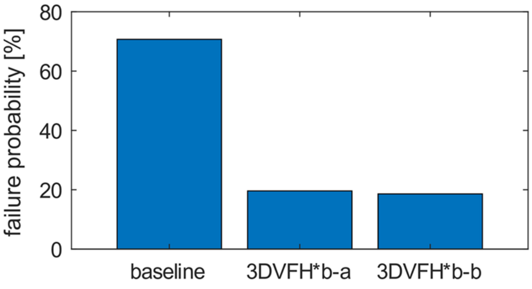

Fig. 6 Failure probabilities of the baseline, the 3DVFH*b-a, and 3DVFH*b-b in the city worlds

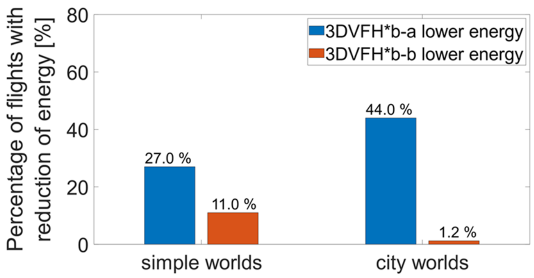

Fig. 7 Comparison of energy consumption using 3DVFH*b-a and 3DVFH*b-b in simple and city worlds. The blue bar represents the percentage of successful worlds, in which the 3DVFH*b-a leads to flight paths with lower energy consumption, while the red bar represents the percentage, in which the 3DVFH*b-b has a lower energy consumption

flights in which the 3DVFH*b- b performs better energy-wise, which is displayed by Fig. 7. The other 52% of flights do not show a significant difference in energy consumption. In 98% of cases where the 3DVFHb-a has lower energy than the 3DVFHb-b, it also has a lower flight height, indicating a relationship between maximum flight height and energy consumption. A comparison of energy levels for the city worlds also confirms these findings. Figure 7 shows the comparison of energy consumption for both simple and city worlds. In 44% of the flights, the 3DVFH*b-a produces more energy-efficient paths, while 98% of these flights were shorter, and 73% of the flights reached a lower maximum flight height. The 3DVFH*b-b only produced more energy-efficient paths in 1.2% of the flights. This shows that the tendency to go back to a horizontal evasion maneuver earlier, as done by the 3DVFH*b-a, produces paths with a lower energy consumption than an evasion strategy focusing more on vertical evasion.

4.2 Parameter study

As the 3DVFH*b-b shows a more prominent effect on reducing the failure probabilities, it builds the foundational algorithm for the parameter study.

The seventeenth combination represents the center point of the DoE and is simulated twice to analyze the significance of the simulations. In the first run, 135 worlds failed, and 136 in the second run. The failure probabilities of these two runs lead to a standard deviation of 0,000355. As the standard deviation is very low, it can be concluded nearly any difference in failure probability is significant without further testing.

Figure 8 displays the failure probabilities of the 16 combinations from Table 3. It visualizes the influence of different yaw-cost settings depending on the weighting factors of obstacle and velocity costs. On the x-axis is the difference between the yaw-cost setting in the vertical and horizontal zones. The data points with a k_{yaw} difference of four, for example, represent the combinations with a yaw cost value of 7 in the vertical zone and a value of 3 in the horizontal zone. The connected lines have the same value in the vertical evasion zone. The dashed lines represent combinations with a velocity cost weighting factor of 18,000, whereas the solid lines represent combinations with a velocity cost weighting factor of 6000. The blue lines represent an obstacle cost weighting factor of 8.5, and the red lines represent an obstacle cost weighting factor of 5. For the combinations using a k_{yaw} of 10 in the vertical zone (k_{yaw} difference of 7 and 9), the influence of the k_{yaw} value in the horizontal zone is minimal, because the failure probabilities do not change much with a higher k_{yaw} difference. For the combinations using a c_{yaw} of 7 in the vertical zone (k_{yaw} difference of 4 and 6), the influence of the k_{yaw} value in the horizontal zone is higher for most combinations, because the failure probabilities change more with a higher k_{yaw} difference. In Fig. 8, it becomes evident that the higher velocity cost weighting factor 18,000 produces lower failure probabilities for most combinations than the combinations using 6000. A higher velocity cost weighting factor indicates that a change of direction has a higher cost. As the higher velocity cost weighting factor reduces the failure probability, a higher velocity cost value supports the vertical evasion strategy as it produces smoother paths.

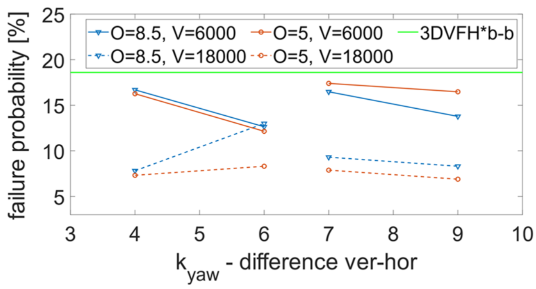

Fig. 8 Failure probabilities for combinations 1–16 of parameter study. O = obstacle cost; V = velocity cost. Markers indicate data points. The green line represents the failure probability of the 3DVFH*b-b

Figure 8 also shows that a lower obstacle cost weighting factor of 5 produces slightly lower failure probabilities in most combinations, except for combinations using a k_{yaw} of 10 in the vertical evasion zone and a velocity cost weighting factor of 6000, where the higher k_{obst} of 8.5 produced lower failure probabilities. The obstacle cost weighting factor indicates the distance at which the obstacle costs increase. Therefore, a lower obstacle cost weighting factor indicates that the planner can plan closer to obstacles. As the failure probability is lower for most combinations with the obstacle cost weighting factor of 5, a lower obstacle cost value is favorable. The best parameter combination is combination

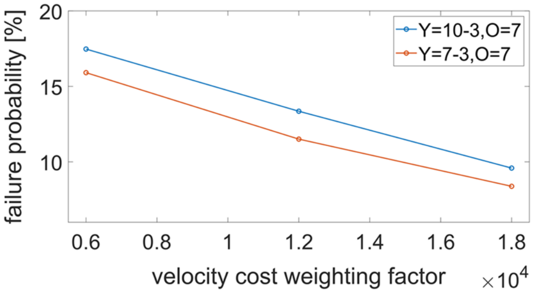

Fig. 9 Failure probabilities of combinations 18–23 of parameter study. Markers indicate data points. Y = yaw cost; O = obstacle cost

eight and has a failure probability of 6.9%. The weighting factor setting for this combination is $k_{yaw} = 10 - 1$, $k_{obst} = 5, k_{vel} = 18000$.

Combinations 18 to 23 were chosen to analyze the influence of the velocity cost weighting factor. Their failure probabilities are displayed in Fig. 9. With a constant obstacle cost of 7 and a yaw cost combination of 10–3 (combinations 18 to 20), the failure probability increases with decreasing velocity cost weighting factor, showing a linear behavior. The combinations with a constant obstacle cost of 7 and a yaw cost combination of 7–3 (combinations 21 to 23) show similar behavior (Fig. 9), which shows that higher values for the velocity cost weighting factor produce lower failure probabilities. The planner follows a chosen direction longer with higher velocity costs, which creates smoother paths. Higher velocity cost weighting factors are, therefore, favorable.

Additionally, the traveled distance and energy consumption are assessed for the parameter combination with the lowest failure probability, combination eight (PC8), and the baseline 3DVFH*b-b algorithm. 1127 worlds were successful in both algorithm adaptions, which serve as a baseline for traveled distance and energy comparisons. Considering the traveled distance, PC8 produces significantly more shorter flight paths in the same worlds–68.7% in the improved version (PC8), whereas only 1.7% were shorter using the baseline version of the 3DVFH*b-b. About 55% of the shorter flights using PC8 also had a lower maximum flight height, which shows that height has no great influence in this case.

Comparing the consumed energy, PC8 produced paths with a lower energy consumption compared to the baseline 3DVFH*b-b in 75% of cases. This highlights that varying the parameters yaw cost, obstacle cost, and velocity cost greatly influences the energy consumption of a flight path. In 98.5% of the worlds with lower energy consumption, the traveled distance is also lower, underlining the correlation between energy and traveled distance. The parameters of PC8—a high difference in yaw cost in the different evasion zones, low obstacle cost, and high-velocity cost—imply a strong support of the vertical evasion maneuver in the vertical evasion zone and a strong focus on following a chosen direction, also in the horizontal zone. Therefore, smoother paths are more favorable regarding energy consumption. The average reduction of energy consumption using PC8 is 28%.

5 Conclusion

This work presented a novel, bio-inspired local path planning algorithm based on the 3DVFH*. From bumblebee experiments, zones depending on the distance to the obstacle in the goal direction were developed, in which either a vertical or a horizontal evasion strategy is applied. Two definitions of the distances defining these zones exist, resulting in two algorithms, the 3DVFH*b-a and 3DVFH*b-b. The vertical evasion strategy alters the favorable pitch direction in the cost function from goal-driven to climb-driven. The 3DVFH*b-b with a shorter horizontal evasion zone reduced the failure probability in simple worlds from 18.1% to 1.4% and in city worlds from 70 to 18.6% with mediocre parameters. Comparing the traveled distance and energy consumption in the worlds that were successful in both algorithm variants—3DVFH*b-a and 3DVFH*b-b—showed that the focus on the vertical evasion strategy leads to longer flight paths and higher energy consumption, but with the strongly positive aspect of reaching the goal more often.

A parameter study was conducted to optimize the 3DVFH*b-b further and identify the best parameter setting for the cost function's weighting factors. Higher velocity cost weighting factors produce lower failure probabilities, because they support a chosen direction and produce smoother paths. Lower obstacles cost weighting factors produce lower failure probabilities, because they indicate that the planner can plan closer to obstacles. The weighting factor setting for the best combination is $k_{yaw} = 10 - 1$, $k = 5, k_{vel} = 18000$, which reduces the failure probability in city worlds to 6.9%. This parameter combination also reduces energy consumption in 75% of successful flights, highlighting that a better parameter combination significantly reduces consumed energy and failure probability.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability Please contact the corresponding author for any data and material requests.

Code availability Please contact the corresponding author for any code requests.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Chiang, W.-C., Li, Y., Shang, J., Urban, T.L.: Impact of drone delivery on sustainability and cost: realizing the UAV potential through vehicle routing optimization. Appl. Energy **242**, 1164–1175 (2019). https://doi.org/10.1016/j.apenergy.2019.03.117
2. Jalil, B., Leone, G.R., Martinelli, M., Moroni, D., Pascali, M.A., Berton, A.: "Fault detection in power equipment via an unmanned aerial system using multi modal data. Sensors (2019). https://doi.org/10.3390/s19133014
3. E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous UAV Surveillance in Complex Urban Environments," in (2009) IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT): WI-IAT 2009 ; Milano, Italy, 15–18 September 2009 ; [proceedings ; including workshop papers, Milan, Italy, 2009, pp. 82–85
4. W. Khaksar, S. Vivekananthen, K. S. M. Saharia, M. Yousefi, and F. B. Ismail, "A review on mobile robots motion path planning in unknown environments," in 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Langkawi, Malaysia, 2015, pp. 295–300.
5. T. Baumann, "Obstacle Avoidance for Drones Using a 3DVFH* Algorithm," Masterarbeit, ETH Zürich, Zürich, 2018. Accessed: Sep. 18 2023. [Online]. Available: https://drive.google.com/file/d/1EhjTvv1QyUxfdnQnBQcPc-T3Zg3OMnlD/edit
6. A. Thoma, A. Fisher, A. Gardi, and C. Braun, "Investigating the performance dependency on the weight of different cost functions for the 3DVFH*," 2021.
7. Gorb, S.N.: Insect-Inspired Technologies: Insects as a Source for Biomimetics. In: Vilcinskas, A. (ed.) Insect Biotechnology, pp. 241–264. Springer, Netherlands, Dordrecht (2011)
8. S. K. Debnath, R. Omar, and N. B. A. Latip, "A Review on Energy Efficient Path Planning Algorithms for Unmanned Air Vehicles," in Lecture Notes in Electrical Engineering, Computational Science and Technology, R. Alfred, Y. Lim, A. A. A. Ibrahim, and P. Anthony, Eds., Singapore: Springer Singapore, 2019, pp. 523–532.
9. Srinivasan, M.V.: Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics. Physiol. Rev. **91**(2), 413–460 (2011). https://doi.org/10.1152/physrev.00005.2010
10. Thoma, A., Fisher, A., Bertrand, O., Braun, C.: Evaluation of Possible Flight Strategies for Close Object Evasion from Bumblebee Experiments. In: Mura, A., Tauber, F., Speck, T., Prescott, T.J., Verschure, P.F.M.J. (eds.) Lecture Notes in Computer Science, Biomimetic and Biohybrid Systems, vol. Vouloutsi, pp. 354–365. Springer International Publishing, Cham (2020)
11. Zhang, H., Lin, W., Chen, A.: Path planning for the mobile robot: a review. Symmetry **10**(10), 450 (2018). https://doi.org/10.3390/sym10100450
12. M. Pittnera, M. Hillera, F. Partickea, L. Patiño-Studenckia, and J. Thielecke, "Systematic Analysis of Global and Local Planners for Optimal Trajectory Planning," 2018.
13. S. M. LaValle, "Planning algorithms," Booklet, Cambridge University Press, 2006. [Online]. Available: https://dblp.org/rec/books/daglib/0016830
14. Yang, L., Qi, J., Song, D., Xiao, J., Han, J., Xia, Y.: Survey of robot 3D Path planning algorithms. J Cont Sci Eng **2016**, 1–22 (2016). https://doi.org/10.1155/2016/7426913
15. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Int J Robotics Res **30**(7), 846–894 (2011). https://doi.org/10.1177/0278364911406761
16. C. Zammit and E.-J. van Kampen, "Comparison between A* and RRT Algorithms for UAV Path Planning," in 2018 AIAA Guidance Navigation and Control Conference Kissimmee Florida. January 08–12 2018, Kissimmee, Florida, 2018.
17. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cyber. **4**(2), 100–107 (1968). https://doi.org/10.1109/TSSC.1968.300136
18. Px4 Autopilot, *Px4 Avoidance Code.* [Online]. Available: https://github.com/PX4/PX4-Avoidance (Accessed: Jul. 31 2022).
19. J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots in cluttered environments," in Proceedings., IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 1990, pp. 572–577.
20. S. Poudel, M. Y. Arafat, and S. Moh, (2023) "Bio-inspired optimization-based path planning algorithms in unmanned aerial Vehicles: A Survey https://doi.org/10.3390/s23063051.
21. J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence,* 1st ed., 1992. [Online]. Available: https://permalink.obvsg.at/
22. Roberge, V., Tarbouchi, M., Labonte, G.: Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans. Industr. Inf. **9**(1), 132–141 (2013). https://doi.org/10.1109/TII.2012.2198665
23. Da Silva Arantes, J., Da Silva Arantes, M., Toledo, C.F.M., Júnior, O.T., Williams, B.C.: Heuristic and genetic algorithm approaches for UAV Path planning under critical situation. Int J Artificial Intellig Tool (2017). https://doi.org/10.1142/S0218213017600089
24. S. Konatowski and P. Pawlowski, "Ant colony optimization algorithm for UAV path planning," in 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET): Conference proceedings : Lviv-Slavske, Ukraine, February 20–24, 2018, Lviv-Slavske, Ukraine, 2018, pp. 177–182.
25. Zhou, Y., Su, Y., Xie, A., Kong, L.: A newly bio-inspired path planning algorithm for autonomous obstacle avoidance of UAV. Chin. J. Aeronaut. **34**(9), 199–209 (2021). https://doi.org/10.1016/j.cja.2020.12.018
26. A. Thoma, C. Braun, and A. Gardi, Improving UAV Obstacle Avoidance by Environment Dependent Cost for Path Planning, 2022.
27. Westphal, C., Steffan-Dewenter, I., Tscharntke, T.: Bumblebees experience landscapes at different spatial scales: possible implications for coexistence. Oecologia **149**(2), 289–300 (2006). https://doi.org/10.1007/s00442-006-0448-6

28. K. Thomessen, “A bio-inspired local path planning algorithm based on the 3DVFH*,” Bachelor’s Thesis, FH Aachen, Aachen, 2022.
29. S. Seker, “Obstacle Avoidance mit dem px4 Eine Parameterstudie,” Bachelorarbeit, Fachhochschule Aachen, Aachen, 2022.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.