

Integrated Global and Local Path Planning for Quadrotor Using Particle Swarm Optimization

Youkyung Hong * Suseong Kim ** Jihun Cha ***

* *Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea, (e-mail: youkyh1@etri.re.kr).*

** *Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea, (e-mail: suseongkim@etri.re.kr).*

*** *Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea, (e-mail: jihun@etri.re.kr).*

Abstract: This study proposes a new path planning method for quadrotors to determine a set of waypoints by considering both geometric constraints to avoid collisions with obstacles and dynamic constraints to reflect the dynamic characteristics of the quadrotor. The proposed path planning method can be formulated as a non-linear optimization problem that minimizes the Euclidean distance between waypoints while satisfying the geometric and dynamic constraints. Particle swarm optimization is utilized to solve the non-linear optimization problem efficiently. By utilizing the Gazebo simulator, the performance of the proposed path planning method is validated for a quadrotor.

Keywords: Quadrotor, Path Planning, Particle Swarm Optimization, Minimum Snap Trajectory

1. INTRODUCTION

Multicopter unmanned aerial vehicles, commonly known as quadrotors or drones, are rapidly expanding in a wide range of civil applications, such as disaster management, maintenance inspection and precision agriculture. In order for a quadrotor to autonomously fly as a fully automatic system, various technologies (such as positioning through navigation, state recognition by fusing multi-sensor information, and environmental perception) are required. Among them, this study focuses on the quadrotor path planning problem, inspired by recent advances (Richter et al. (2016)).

In general, the path planning problem consists of global path planning and local path planning. The global path planning plans trajectories that only satisfy geometric constraints to avoid collisions with surrounding obstacles. The output of the global path planning is a set of waypoints, and the dynamics of the quadrotor is not taken into account in the global path planning. On the other hand, the local path planning takes into account the dynamic characteristics of the quadrotor, creating a smooth and appropriate trajectory that the quadrotor can follow.

In this study, instead of maintaining the current paradigm of the quadrotor path planning where the global path planning and the local path planning are designed separately, a new integrated global and local path planning is proposed. The proposed method aims to determine the coordinates of a set of waypoints (like the conventional global path planning). However, the waypoints are determined by considering that they are connected by the minimum snap trajectory (unlike the conventional global path planning). The proposed method can be formulated as an optimization problem consisting of the cost function that minimizes the Euclidean distance between waypoints, and constraints that avoid collisions with obstacles and reflect dynamic characteristics. Determining waypoints, which can be regarded as the conventional global path planning, is more challenging

to solve due to the nonlinearity in the constraints. To handle the complexity introduced by the simultaneous consideration of both global and local path planning, Particle Swarm Optimization (PSO), which is a population-based stochastic optimization technique, is adopted in this study (Eberhart and Kennedy (1995)). The features of PSO can be summarized as follows. PSO is easier to implement with a less computational load than other stochastic optimization schemes and empirically demonstrated that the performance of PSO is not sensitive to population size (Dong et al. (2005)). The solution quality of PSO does not depend on the initial population as long as particles exist within the search space (Abido (2002)). Furthermore, PSO has the flexibility to control the balance between global and local experiences in the search space. This property improves the search capabilities of PSO and avoids premature convergence to the local optimal solution.

The remainder of this study is organized as follows. Section 2 explains the characteristics of the conventional path planning method and describes the concept of operation of the proposed method. In Section 3, the proposed integrated path planning method is presented in detail. Section 4, simulation results are described to demonstrate the performance of the proposed integrated path planning method. In Section 5, the conclusions of this study and some directions for future research are provided.

2. PROBLEM STATEMENT

2.1 Disadvantage of the conventional path planning method

As shown in Fig. 1, the quadrotor path planning problem can be expressed as a two-level hierarchical architecture consisting of the global path planning and the local path planning. In the higher level, the global path planning determines a set of waypoints to guide the quadrotor from the start point to the goal point, where the connection between waypoints is considered as a straight line. In the lower level, the local path plan-

ning generates a smooth trajectory passing through waypoints, which are determined in the global path planning. However, as shown in Fig. 2(a) and 2(b), the conventional method can cause new collisions with surrounding obstacles in the process of smoothly connecting two adjacent waypoints in the local path planning. The reason is that in the global path planning, the connection between waypoints is a straight line, not a smooth trajectory. Richter et al. mentioned that after the conventional local path planning, a particular trajectory segment might intersect obstacles, but adding an intermediate waypoint between two waypoints solves the problem (Richter et al. (2016)), as shown in Fig. 2(c). However, this method does not always guarantee a collision-free trajectory. In some cases, the number of additional waypoints required to repair collisions can be more than one or infinite. Also, as auxiliary waypoints are added to connect waypoints artificially, the local path planning might not be able to create a feasible trajectory that takes into account the quadrotor dynamics. Furthermore, adding an artificial waypoint changes the trajectories determined by the local path planning in the adjacent segments because the continuity constraints at the boundary that should be satisfied in the adjacent segment changes. As a result, the newly added waypoint might cause collisions in other adjacent segments that previously did not have collisions, as described in Fig. 2(d). For these reasons, adding auxiliary waypoints is not the ultimate solution. We need to consider other ways to solve the disadvantage inherent in the conventional method.

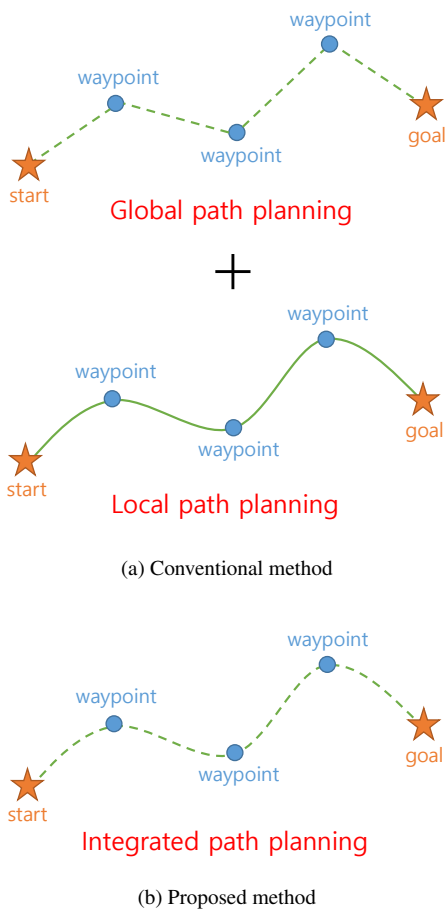


Fig. 1. Concept of operation.

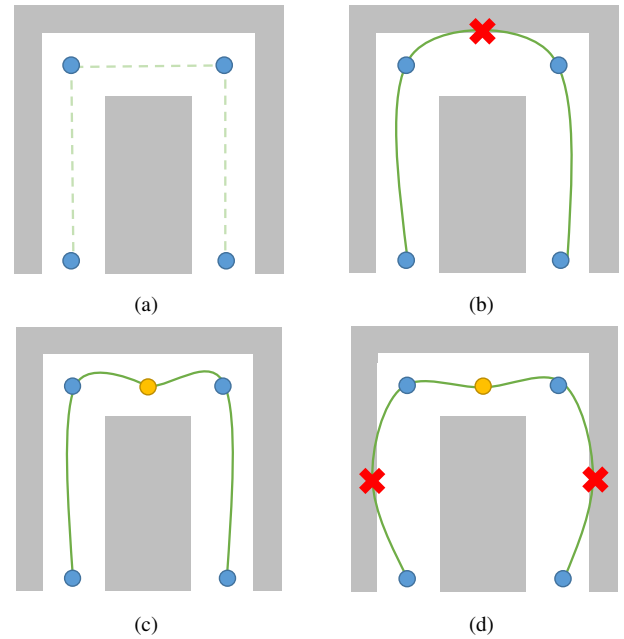


Fig. 2. Example of the disadvantage of the conventional method. (a) Global path planning without collision, (b) local path planning with collision, (c) desired performance, and (d) undesired performance of the additional waypoint.

2.2 Differentiated strategy

In this study, a new integrated path planning method is proposed that takes into account the smooth trajectories that the local path planning will generate when the global path planning determines the waypoints, as illustrated in Fig. 1. In this new approach, the global path planning becomes more complicated by considering smooth trajectories rather than straight lines, but it can prevent the disadvantage inherent in the conventional path planning method mentioned above. Now, the problem we are trying to solve can be thought of as an optimization problem to generate the shortest collision-free path from the start point to the goal point, while satisfying the constraints required for flight. Also, the decision variable of the optimization problem is a set of waypoints determined by global path planning. In this study, PSO is adopted to solve the proposed optimization problem efficiently. Note that the local path planning in this study is designed by using the minimum snap trajectory, which proved to be very useful as quadrotor trajectories because the motor torque is related to the snap (Mellinger and Kumar (2011)).

2.3 Minimum snap trajectory

The basic concept of the minimum snap trajectory is briefly described as follows. In general, the state vector \mathbf{x} to describe the dynamics of the quadrotor is chosen as:

$$\mathbf{x} = [x, y, z, \phi, \theta, \psi, u, v, w, p, q, r]^T \quad (1)$$

where (x, y, z) , (ϕ, θ, ψ) , (u, v, w) , and (p, q, r) denote positions, velocities, Euler angles, and the angular rates, respectively. Additionally, the control input vector \mathbf{u} consists of the total lift force and roll, pitch, and yaw moments as follows.

$$\mathbf{u} = [u_1, u_2, u_3, u_4]^T \quad (2)$$

Although the full dynamics of the quadrotor is nonlinear and under-actuated, Mellinger and Kumar claimed that the quadrotor dynamics is differentially flat when positions and yaw angle are considered as the flat output vector σ as follows (Mellinger and Kumar (2011)).

$$\sigma = [x, y, z, \psi]^T \quad (3)$$

In other words, by defining positions and yaw angle as the flat output vector, the state and control input vectors of quadrotors can be represented as algebraic functions of the flat output vector and its derivatives. In summary, because the quadrotor is a differentially flat system, the feasible control input vector can be generated by mapping the state and control input vectors to the path represented by the positions and yaw angle of the quadrotor.

$$(\mathbf{x}, \mathbf{u}) = \Phi(\sigma, \dot{\sigma}, \ddot{\sigma}, \ddot{\sigma}, \ddot{\sigma}) \quad (4)$$

3. ALGORITHM DESCRIPTION

In this section, the new quadrotor path planning for determining the waypoint coordinates, taking into account the global and local path planning simultaneously, is formulated as an optimization problem and then solved by using PSO.

3.1 Decision variables

To be formulated as the optimization problem, decision variables are considered the coordinate of the set of waypoints and can be encoded as a matrix type as follows:

$$\begin{pmatrix} w_{x_1} & w_{y_1} & w_{z_1} \\ w_{x_2} & w_{y_2} & w_{z_2} \\ \vdots & \vdots & \vdots \\ w_{x_N} & w_{y_N} & w_{z_N} \end{pmatrix} \quad (5)$$

where each row represents the $(w_{x_i}, w_{y_i}, w_{z_i})$ coordinates of the i th waypoint, and N is the number of waypoints between the start and goal points.

3.2 Constraints and performance index

In the optimization problem, two sets of fundamental constraints are considered, which are intended to avoid obstacles and generate dynamically executable paths geometrically. The first set of constraints is that each waypoint is inside the map boundaries and outside the obstacles. The second set of constraints is that each minimum snap trajectory connecting two adjacent waypoints is inside the map boundaries and outside the obstacles.

Also, the performance index PI is defined to minimize the Euclidean distance between waypoints, which can be represented as follows:

$$\begin{aligned} \text{PI} = & \| (x_{start} - w_{x_1}, y_{start} - w_{y_1}, z_{start} - w_{z_1}) \| \\ & + \sum_{i=1}^{N-1} \| (w_{x_{i+1}} - w_{x_i}, w_{y_{i+1}} - w_{y_i}, w_{z_{i+1}} - w_{z_i}) \| \\ & + \| (x_{goal} - w_{x_N}, y_{goal} - w_{y_N}, z_{goal} - w_{z_N}) \| \end{aligned} \quad (6)$$

where $(x_{start}, y_{start}, z_{start})$ and $(x_{goal}, y_{goal}, z_{goal})$ denote the coordinates of the start and goal points, respectively.

Finally, before applying PSO, all constraints are incorporated in the fitness function J as follows:

$$J = \text{PI} + \sum_{k=1}^K C_k \{1 - \mathbb{I}(g_k)\} \quad (7)$$

where the second term is the penalty function to satisfy the inequality constraints; g_k is the k th inequality constraint, C_k is the weighting parameter, and $\mathbb{I}(\cdot)$ is the indicator function, which is unity if the argument is true and zero otherwise.

3.3 PSO implementation

In the original PSO algorithm, the velocity vector $u_s \in \mathbb{R}^{2N}$ and position vector $x_s \in \mathbb{R}^{2N}$ of particle s ($s = 1, \dots, S$), where S is the swarm size, are randomly generated in the beginning, and updates are performed as follows:

$$u_s^{iter} = K u_s^{iter-1} + c_1 r_1 (x_{s_p}^* - x_s^{iter-1}) + c_2 r_2 (x_g^* - x_s^{iter-1}) \quad (8)$$

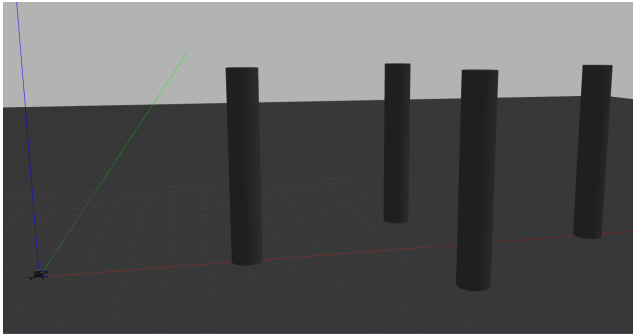
$$x_s^{iter} = x_s^{iter-1} + u_s^{iter} \quad (9)$$

where $x_{s_p}^* \in \mathbb{R}^{3N}$ and $x_g^* \in \mathbb{R}^{3N}$ are the remembered best previous position of particle s and the position of the best particle among all the particles, respectively, c_1 and c_2 are the acceleration constants, r_1 and r_2 are uniform random values between 0 and 1, and K is the inertia weight which ensures the convergence of PSO.

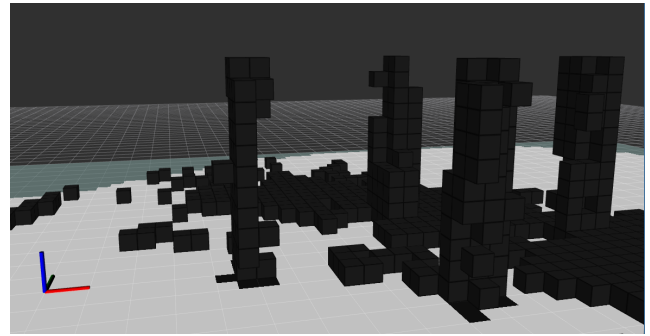
In this study, the position vector x_s corresponds to the decision variable, i.e., each element of the matrix in Eq. (5). Each particle evaluates its current position x_s by calculating its cost function J_s^{iter} based on Eq. (7). If the current cost function of a particle J_s^{iter} is less than its previous cost function J_s^{iter-1} , then it updates its personal best $x_{s_p}^*$. After all particles' cost functions are calculated, the values are compared with each other, and then the best of the personal bests $x_{s_p}^*$ is set to the global best x_g^* . Additionally, each particle updates its position and velocity using Eqs. (8) and (9), and thereby they move closer to the global optimum with the personal best $x_{s_p}^*$ and the global best x_g^* .

4. SIMULATION RESULTS

In this section, the performance of the proposed integrated path planning method is evaluated. Realistic simulation is carried out in the virtual physical simulator Gazebo with the open robotic environment ROS (Robot Operating System). We generate $25 m \times 10 m \times 3 m$ map where four cylinder obstacles of length $5 m$ and radius $0.4 m$ are positioned as shown in Fig. 3(a). To evaluate the proposed algorithm on an autonomous system, in addition to a path planning module, we need i) an environmental perception module, ii) a quadrotor model (virtual control object), and iii) a control system module. First, in this simulation, the OctoMap library is adopted to build a 3D occupancy grid map representing obstacle information on the map. More specifically, the resolution of the occupancy grid map is set to $1 m$ and obstacles are detected by using a Hokuyo laser rangefinder. The maximum range of the sensor is set to $10 m$, and the output of the sensor gives a value between 0 and 1 to represent whether there is an obstacle at a point in space. In this simulation, if the probability value is greater than 0.5, then the point is considered occupied. At the current level of this study, obstacle information is pre-mapped and stored in



(a) Test scene in Gazebo



(b) Environmental perception through OctoMap library

Fig. 3. Simulation setup for evaluations.

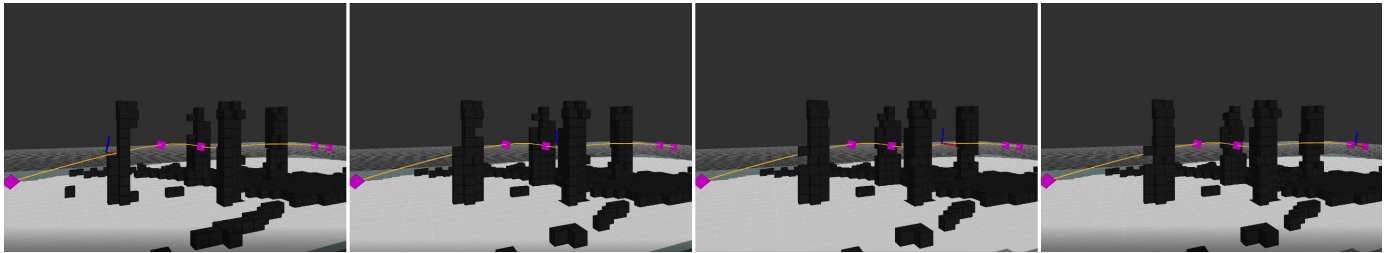


Fig. 4. Snapshots of simulation taken every two seconds.

the occupancy grid map in the off-line operation. Afterward, in the on-line operation, the smooth and collision-free trajectory connecting the start point and the goal point is determined by using the proposed algorithm from a fully known environment described in Fig. 3(b). Second, 3DR Iris Quadcopter is adopted as a quadrotor model, where the first order drag effects are included. Lastly, the control system is implemented based on the PX4 firmware v1.9.0. All software is implemented based on ROS, and the open-source project MAVROS is used as the interface between PX4 and ROS.

The parameters related to the proposed path planning method are described in the following. The coordinates of the start point and the goal point are set to $(0, 0, 1.5)$ and $(19, -1, 2.5)$, respectively, and three waypoints are determined by the proposed algorithm. It means that the entire trajectory from the start point to the goal point consists of four segments. In order to determine the coordinates of waypoints, the swarm size is set to three times the number of decision variables. On initialization, particles are randomly positioned so that the coordinates of the waypoints are distributed evenly across the entire map. The stopping criterion is applied to the last 20 iterations, and the desired accuracy is set to 10^{-3} . In Eq. (7), the weighting parameters C_k are set to 10^5 . Additionally, we use 8 order polynomial to generate the minimum snap trajectory, and its constraint checking is performed every 0.1 seconds.

Figure 4 shows the snapshots taken every two seconds during the simulation. In Fig. 4, the solid yellow line, and the magenta square points indicate the desired trajectory and the waypoints, respectively, determined by the proposed path planning method. The number of iteration until PSO converges was 52, and the resulted fitness value in Eq. (7) was $38.85m$. Also, the coordinates of the determined three waypoints were $(7.07, 1.32, 2.94)$, $(9.36, 0.42, 2.81)$, and $(17.48, -1.13, 2.69)$. The computation time to determine the optimal set of waypoints was 0.42 seconds, and the computation is performed using a laptop PC

with a 2.60 GHz Intel i7 CPU processor. Figure 5 shows the time histories of the states, including positions, velocities, Euler angles, and angular rates. In Fig. 5, the red dashed lines indicate the desired trajectory generated by the proposed algorithm, and the solid blue lines indicate the actual trajectory traveled by the quadrotor. Figure 6 shows the desired trajectory and the actual trajectory in three dimensions with the mapped obstacles, and ‘wp’ stands for ‘waypoint.’ From the results, it was verified that the proposed algorithm could generate a feasible trajectory within low computational time (less than 1 second), which avoids collisions with obstacles and is suitable for the quadrotor to follow.

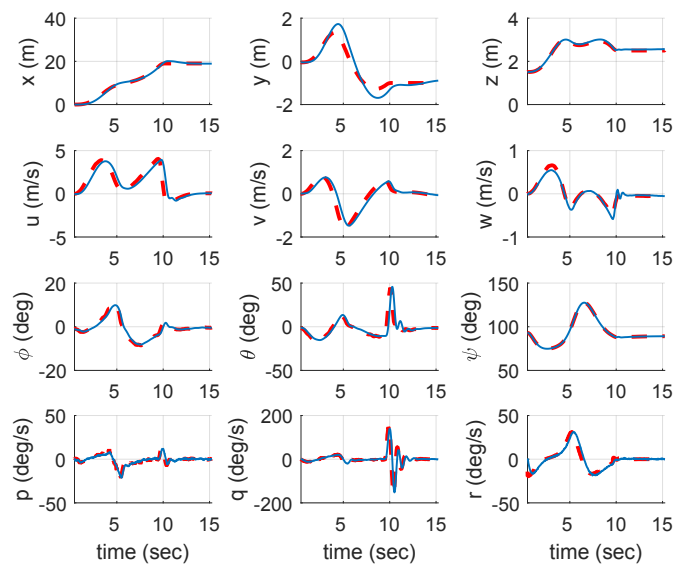


Fig. 5. Time histories of states.

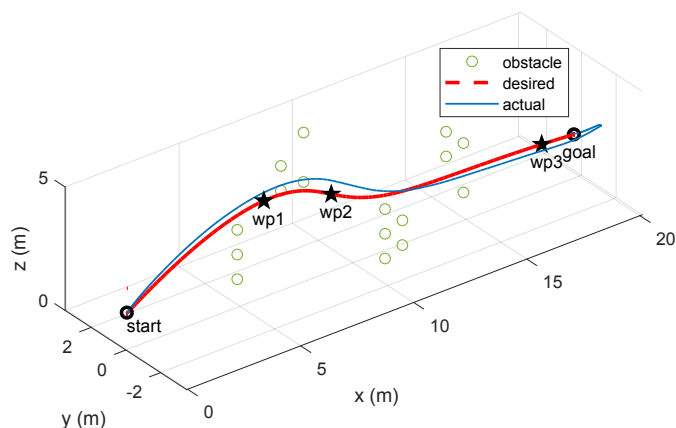


Fig. 6. Resulting trajectories in three dimensions recorded from simulation.

5. CONCLUSION

In this study, a new integrated path planning method is formulated as an optimization problem and is solved by using particle swarm optimization. Unlike the conventional method, one of the key strengths of the proposed method is that a set of waypoints are determined by taking into account the smooth trajectory between two adjacent waypoints. The optimization problem is designed to minimize the flight distance while satisfying the geometric constraints to avoid collisions with obstacles and minimizing snap to consider the dynamic capabilities of the quadrotor. The performance of the proposed path planning method was validated through a realistic simulation. As a result, the proposed method was sufficient to generate the collision-free trajectory connecting the start and goal points within a short computation time.

For future works, this study will be extended to deal with more cluttered environments. As more complex environments are considered, more waypoints are needed. Then, the proposed path planning method might be a more difficult problem because several optimal solutions should be determined within a limited time. Additionally, at the current level of this study, obstacle information is mapped to the occupancy grid map in advance, and the path planning is performed based on this information. However, in the future, this study will be improved to a more fully autonomous system that recognizes obstacles in real-time and simultaneously plans the trajectory, without a pre-mapping procedure.

ACKNOWLEDGEMENTS

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [2017-0-00067, Development of ICT Core Technologies for Safe Unmanned Vehicles]

REFERENCES

- Abido, M. (2002). Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems*, 24(7), 563–571.
- Dong, Y., Tang, J., Xu, B., and Wang, D. (2005). An application of swarm optimization to nonlinear programming. *Computers & Mathematics with Applications*, 49(11-12), 1655–1668.

- Eberhart, R. and Kennedy, J. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, 1942–1948. Citeseer.
- Mellinger, D. and Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE International Conference on Robotics and Automation*, 2520–2525. IEEE.
- Richter, C., Bry, A., and Roy, N. (2016). Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. 649–666.