Optimal Obstacle Avoidance of Stationary and Dynamic Objects using a Quadrotor UAV

Wouter Van Gijseghem
Stanford University
wvg@stanford.edu

Ben Moore
Stanford University
benmoore@stanford.edu

Bernard Lange
Stanford University
blange@stanford.edu

***Abstract*—This paper presents a multi-agent path planning algorithm based on Sequential Convex Programming (SCP) that finds optimal collision free trajectories. It employs a centralized SCP algorithm which is able to navigate drones through cluttered environment, navigating between static and dynamic obstacles. Optimal trajectories are defined as trajectories with minimal control effort, minimizing the accelerations of the UAVs. The method can be easily extended to include other vehicles or to optimize other objectives. All non-convex collision constraints are linearized to satisfy the convex requirement and the feasibility of the trajectory is guaranteed by setting limits on the thrust input and jerk of the UAVs. The collision free path is iteratively generated using SCP, limiting the change in solution to guarantee the correctness of the linearization. Since the optimization is centralized, it is possible to guide a swarm of drones collision free to a pre-determined formation. It is shown that the SCP algorithm scales with $\mathcal{O}(n^3)$, where n is the number of vehicles.**

***Index Terms*—Obstacle Avoidance, Sequential Convex Programming, UAVs**

I. Introduction

The significance of unmanned aerial vehicles (UAVs) has been steadily increasing over the last decade. UAVs can be used in numerous important applications, such as goods delivery, maintenance, inspection and surveillance. Integrating them into the airspace requires the ability to detect and avoid both static and moving obstacles which poses a challenge when identifying the optimal formation trajectories without violating the collision free requirement.

This paper focuses on the path planning and collision avoidance for quadrocopters in a multi-agent setting. Perfect knowledge of the environment has been assumed which laid foundation for further derivation of the control and optimization method in this paper. Trajectory identification with collision avoidance requirement has been posed as a Sequential Convex Programming (SCP) problem which has been demonstrated to provide good balance between optimality of the trajectory, computational tractability and flexibility to adjust the optimization environment to different settings, e.g. vehicles with different dynamics, different shapes of the obstacles or various objective functions [1].

Collision avoidance is a widely researched topic. In general, the collision avoidance algorithms are divided into two categories: planning and reacting algorithms. Planning algorithms involve an offline approach which generates collision free trajectories for all vehicles a priori. The reacting counterpart is an online approach in which the vehicles adapts its trajectory mid run depending on the risk of potential collision and the motion of other agents. This paper falls in the planning approach, hence, an offline algorithm approach predetermining the trajectories of all agents simultaneously is implemented. An advantage of SCP is that it converges quicker than both Covariance Hamiltonian Optimization for Motion Planning (CHOMP) and Stochastic Trajectory Optimization for Motion Planning (STOMP), but does require perfect a priori knowledge of the map [2]. SCP is part of direct methods family of algorithms which are not as sensitive to the initialization and generally also less computationally expensive than an indirect method approach. For more complete information the reader is encouraged to refer to a survey paper [3] with an comprehensive overview of the most common collision avoidance algorithms.

By solving the collision avoidance problem using SCP, an optimal collision free trajectory is generated and optimized to the predefined cost function. In this paper, minimum thrust trajectories are used, also referred to as minSnap trajectories [4]. It is assumed that the drones are able to track the trajectory by setting feasibility constraints on the abilities of the UAVs. Using these collision free optimal trajectories, groups of drones can move together in formation to a predetermined final location, avoiding both other drones and any obstacle encountered. It must be noted that if there is no perfect a priori knowledge of the environment, this SCP algorithm can be easily formulated as a Model Predictive Controller (MPC) program, solving for n-steps while updating the knowledge of the world.

The structure of this paper is as follows. In section II, the collision free trajectory generation problem is formulated as an non-convex problem and the approximation of these non-convex constraints as convex constraints is discussed. The complete algorithm and the choice of hyperparmeters is detailed in section III. In section IV, the simulation results are presented and discussed.

II. Problem Formulation

A. Trajectory Dynamics

The trajectory dynamics of the drone are modelled using a linearized dynamics model [5], with $p_i[k] \in R^3$ the position of drone i at time k and h is the discretization time step.

$v_i[k]$ and $a_i[k]$ are the velocity and acceleration of the UAV, respectively.

$$v_i[k+1] = v_i[k] + ha_i[k] \tag{1}$$

,

$$p_i[k+1] = p_i[k] + hv_{\rm i}[k] + \frac{h^2}{2}a_{\rm i}[k] \tag{2}$$

This simple linear model of UAV dynamics is proven sufficiently accurate given bounds on jerk, velocity and acceleration and given a constant yaw angle [6]. Jerk is defined as the rate of change of acceleration. By bounding jerk one ensures continuity in the UAV's attitude.

B. Objective Function

The objective of this problem is to find the minimizer of Equation 3, for a system of N UAVs in D dimensions with K discretization times steps.

$$\arg\min \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K} w_{ij}\mathrm{a}_i^T[k]\mathrm{a}_j^T[k]\mathrm{a}_i[k] \tag{3}$$

The minimization goal can be altered by adjusting the weights w, which allows for minimization over any linear function combination of acceleration. Possible minimization goals include path length or control effort. In this paper control effort is minimized which is equivalent to minimizing thrust over the trajectory.

$$f_0 = \sum_{i=1}^{N}\sum_{k=1}^{K} \|a_i[k]\|_2^2 \tag{4}$$

The objective function can be easily modified to include gravitational effects, which would also require adjustment of the vertical component of the thrust vector of the UAV to account for gravitational effects.

C. Constraints

Corner, wall and floor constraints are applied to bound the drones during simulation and avoid drones becoming trapped in a corner. The corner avoidance constraints are imposed using an exponential loss function (Equation 5) linearized around the previous solution of the SCP equivalently to collision avoidance constraint in subsection II-D.

$$e^{(c(p_i^x[k]-x_{cor}))} + e^{(c(p_i^y[k]-y_{cor}))} \geq 2 \quad \forall i,\ k \tag{5}$$

Increasing the magnitude of constant c leads to improved approximation of the boundary of the corner [1].

The position bounds on the UAVs are given by Equation 6.

$$p_{min,l} \leq p_{i,l}[k] \leq p_{max,l}, \qquad l \in \{x,y,z\}, \quad \forall i,k \tag{6}$$

Maximum acceleration constraints are also applied in order to enforce the actuator saturation limits.

$$a_{min,l} \leq a_{i,l}[k] \leq a_{max,l} \qquad l \in \{x,y,z\}, \quad \forall i,k \tag{7}$$

Constraints are applied to the enforce a maximum jerk magnitude, to ensure continuous UAV trajectories.

$$j_{min,l} \leq j_{i,l}[k] \leq j_{max,l} \quad l \in \{x,y,z\}, \quad \forall i,k \tag{8}$$

The linearized dynamics in Equation 1 and Equation 2 are applied as constraints to enforce the the model dynamics.

D. Collision Avoidance

Dynamic and static obstacle avoidance constraints are used to ensure the trajectory is not obstructed at any time during the simulation. At all times, the UAVs must keep an offset distance away from each other and from the static obstacles. This requirement is described by the non-convex constraint in Equation 9

$$\|p_i[k] - p_j[k]\|_2 \geq R, \quad \forall i,j, \quad i \neq j, \quad \forall k \tag{9}$$

This constraint is however not applicable to convex programming and is therefore linearized. The collision avoidance constraint in Equation 9 is linearized using a first order Taylor expansion at iteration $(q+1)$ around the previous solution found at q.

$$\begin{aligned}\|p_i^q[k] - p_j^q[k]\|_2 +\\ \eta^T[(p_i[k]-p_j[k]) - (p_i^q[k]-p_j^q[k])] \geq R\end{aligned} \tag{10}$$

where

$$\eta = \frac{p_i^q[k]-p_j^q[k]}{\|p_i^q[k]-p_j^q[k]\|_2} \tag{11}$$

In order to linearize the constraint about a previous solution, the SCP algorithm finds an initial solution without this constraint and then enforces this linearized constraint for all future iterations. This linear approximation of the non-convex constraint is a valid approximation given a small time increment and a small trust region ϵ.

III. The Algorithm

The algorithm consists of 3 different stages:

- Initialization
- Solve SCP
- Post-processing

A. Initialization

First the side length of the box (b), the number of pillars ($n_{pillars}$) and the radius of the pillars (r_{pillar}) have to be chosen. Secondly the user defines the number of collocation points (N), together with the simulation time (T) and the avoidance radius for dynamic obstacles. Notice that $\frac{T}{N}$ defines the time step of the simulation. Small values of T are likely to lead to a violation of the physical limits while large T leads to a very high-dimensional optimization problem that leads to slow computations. The location of the pillars can be generated at random within the box or in a deterministic grid. When the environment is built, the initial and final states of the drones are chosen at random, however they can be prespecified by the user to enable formation flight of UAVs. The initial conditions are verified to satisfy all constraints before the simulation is started. All initial paths of the drones are straight lines between the starting and end point of the drone, with the thrust set to half the maximum input value.

B. Solving SCP

An optimal collision free path is iteratively generated by solving the SCP and updating the paths of the drones. The simulation is stopped when the norm between the previous solution and the current solution is smaller than Υ.

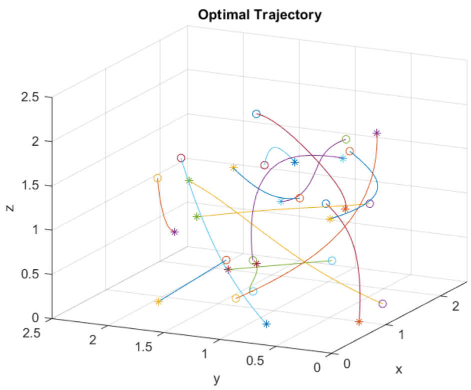

Fig. 1. Optimal collision free constraints for 15 drones in the absence of static constraints

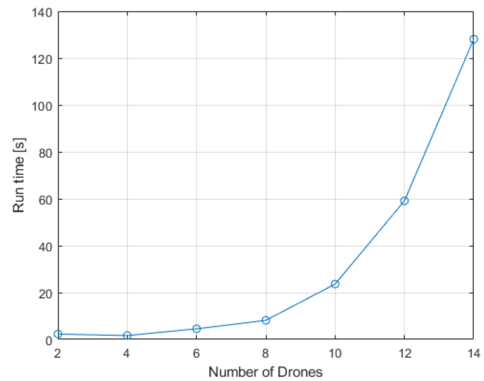

Fig. 2. Average run time versus number of drones

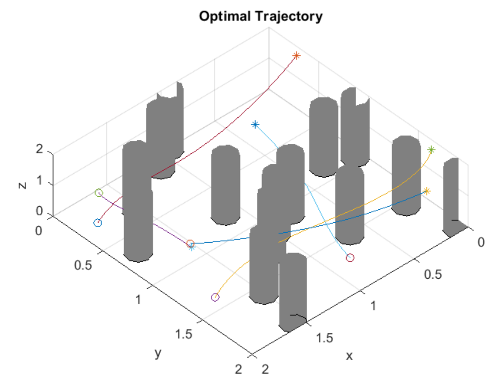

Fig. 3. Five drones avoiding 15 pillars

C. Post-processing

The final state vector is used to generate all trajectories of all drones, including plots of the velocity and thrust input of every drone.

IV. Results

All results presented in the following section are run with 50 collocation points over $T = 5$, which results in $h = 0.1$. The SCP is converged with $\Upsilon = 10^{-3}$. $j_{max} = 23m/s^3$, similar to [6]. $a_{max} = 12m/s^2$ [6]. The linearization constraints ϵ is set to 0.5. All drones are bounded to a cube with side length of 2 meters. All simulations are run on a laptop with an Intel Core i7-8750H CPU @ 2.2 GHZ, equipped with 16 GB of RAM.

A. Dynamic Collision Constraints

1) Example Optimal Route: In Figure 1, 15 drones have to move from random initial conditions and random final conditions while maintaining at least a 0.5 meter radius from every drone. All drones start with zero initial velocity and have to end with zero velocity. The trajectory minimizes the control effort, resulting in a cost of 1008 m/s^2.

2) Computational Effort: In Figure 2, the average run time of the SCP is plotted against the number of drones (averaged over 10 runs). The SCP algorithms scales with $\mathcal{O}(n^3)$ with n the number of drones.

B. Dynamic and Static Collision Constraints

1) Random Cluttered Environment: The algorithm is able to determine the optimal trajectories for a number of drones in a cluttered random environment, constrained to a room, avoiding the corners and avoiding each other. Figure 3 shows five drones avoiding 15 pillars in isometric view. The minimized control effort equals 46.988 m/s^2, which is considerably lower than the control effort to navigate 15 drones but without pillars

2) Deterministic Environment: Next to cluttered random environment, the algorithm is also able to find optimal routes in artificial environments such as grids of pillars. In Figure 4, eight drones move collision free in a grid of 25 pillars with a total control effort cost of 66.7 m/s^2. The drones are avoiding each other with an R of 0.2 m. Any environment in real-life can be discretized in a set of cylinders and SCP is able to find, most of the time, feasible optimal collision free trajectories.

It must be noted that as soon as the shapes become more complex, the SCP algorithm struggled to find a feasible solution. It could not find any feasible solution when the starting points where defined in a half circle and the drones had to find their way out of the circle.

3) Computational Effort: In Figure 5, the average run time of the SCP is plotted against the number of static obstacles (averaged over 10 runs). It is clear that static obstacles are a lot less computationally expensive than moving drone obstacles. The algorithm scales with $\mathcal{O}(n)$ where n is the amount of static obstacles. The run time is highly dependent on the initialization of the drones.

C. Formation Flight

Since the SCP algorithm presented in this paper, is centralized, meaning it optimizes the routes of all drones simultane-

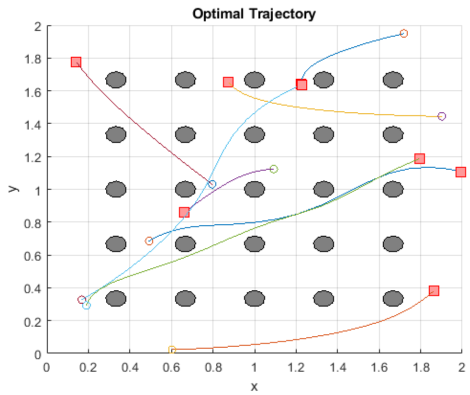

Fig. 4. Eight drones navigating a grid of 25 pillars

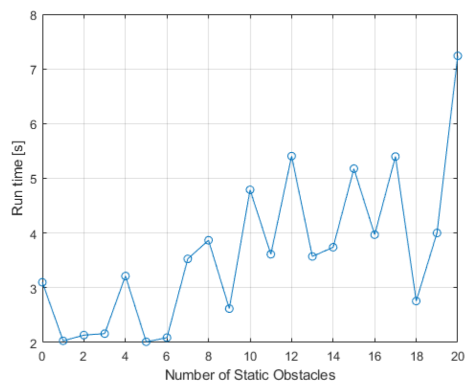

Fig. 5. Average run time versus number of static obstacles

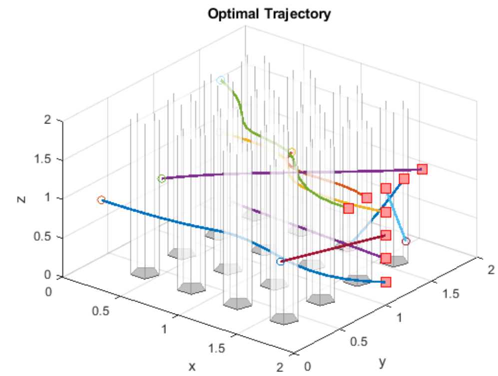

Fig. 6. Example of formation flight, 9 drones form collision free the letter T, 16 pillars

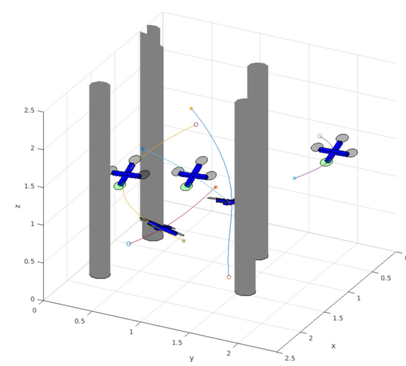

Fig. 7. Visual Inspection with 5 drones and 4 pillars

ously, it is trivial to extend the method to include formation flight of UAVs. With formation flight is meant that since the drones are aware of each other locations, they can collision free transition to any desirable shape. Figure 6 gives an example where 9 drones after successfully navigating a grid of 25 pillars form the letter T in the air. Additionally, the results of the simulation were visually inspected in the simulator with the size of the drone matching collision avoidance constraint parameters (Figure 7).

V. Conclusion

Optimal and collision-free UAV navigation is an essential requirement for the realization of future UAV technologies and capabilities, in cluttered and dynamic environments. Such technology has the potential to revolutionize industries such as goods transportation, equipment inspection and search and rescue missions.

Using the work of [5] and [1], the collision avoidance and corner avoidance capabilities were merged. Building on this, static obstacle avoidance and formation flight was implemented. Thus allowing any map to be modelled using a cylindrical volume as a discretization unit and any desired final formation to be achieved, respectively. In this paper, a three-dimensional multi-UAV dynamic and static obstacle avoidance problem is formulated as an SCP problem, minimizing control effort, which is expressed as thrust delivered by the propellers. The method presented in this paper generates collision-free trajectories for multiple agents and multiple obstacles, which was achieved by linearizing the non-convex constraints. The accuracy of the linear approximation of the the UAV dynamics was ensured by the addition of bounding constraints on the vehicle, in accordance with [6]. These physical vehicle constraints also ensured feasible and continuous trajectories. The performance of the algorithm was investigated for both increasing number of drones and static obstacles, it was found that the run time grows proportional to $\mathcal{O}(n^3)$ and $\mathcal{O}(n)$, respectively. Future studies, include alleviating the necessity of full a-priori knowledge of the map by implementing a finite horizon MPC and adding noise to the control of the UAVs to include stochasticity in the model.

References

[1] Y. Chen, M. Cutler, and J. P. How, "Decoupled multiagent path planning via incremental sequential convex programming," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5954–5961.

[2] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online uav replan-

ning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5332–5339.

[3] H. Pham, S. A. Smolka, S. D. Stoller, D. Phan, and J. Yang, "A survey on unmanned aerial vehicle collision avoidance systems," *CoRR*, vol. abs/1508.07723, 2015. [Online]. Available: http://arxiv.org/abs/1508.07723

[4] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 3, May 2002, pp. 1936–1941 vol.3.

[5] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 1917–1922.

[6] F. Augugliaro, "Dancing quadrocopters: Trajectory generation, feasibility, and user interface," Master's thesis, ETH Zurich, 2011, master Thesis ETH Zurich, 2011.