

Tracking a Moving Target in Cluttered Environments Using a Quadrotor

Jing Chen, Tianbo Liu, and Shaojie Shen

Abstract— We address the challenging problem of tracking a moving target in cluttered environments using a quadrotor. Our online trajectory planning method generates smooth, dynamically feasible, and collision-free polynomial trajectories that follow a visually-tracked moving target. As visual observations of the target are obtained, the target trajectory can be estimated and used to predict the target motion for a short time horizon. We propose a formulation to embed both limited horizon tracking error and quadrotor control costs in the cost function for a quadratic programming (QP), while encoding both collision avoidance and dynamical feasibility as linear inequality constraints for the QP. Our method generates tracking trajectories in the order of milliseconds and is therefore suitable for online target tracking with a limited sensing range. We implement our approach on-board a quadrotor testbed equipped with cameras, a laser range finder, an IMU, and onboard computing. Statistical analysis, simulation, and real-world experiments are conducted to demonstrate the effectiveness of our approach.

I. INTRODUCTION

The maturity of estimation [1], control [2], and planning [3] technologies for autonomous aerial robots enables autonomy in complex indoor and outdoor 3D environments, thus opening up a promising market with applications in aerial photography, monitoring, inspection, and search and rescue. Many of these applications require an aerial robot to autonomously follow a moving target. However, the problem of autonomous target following/tracking in a cluttered environment with guaranteed collision avoidance remains open due to the mixture of multiple, possibly conflicting constraints. For instance, “stiff” tracking of a target may result in collision into obstacles due to the inertia of the aerial robot, lower control authorities of the aerial robot compared to the target, and different obstacle structures at different flight levels. In the case of naive following, a quick turn or stopping of the target can lead to a tracking trajectory that is dynamically infeasible for the aerial robot.

We examine the trade-offs and constraints for target tracking in cluttered environments. We note that a naive, “stiff” tracking is actually not desirable. This is particularly true for visual tracking, where small changes in the distance between the aerial robot and the target, or even occasional occlusion are acceptable, provided that the target remains in the field of view of the visual sensor for most of the time. However, there are hard constraints, such as obstacle avoidance and dynamic

All authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. jchenbr@connect.ust.hk, tliuam@connect.ust.hk, eeshaojie@ust.hk

This work was supported by HKUST project R9341. The authors would also like to thank the research donation and equipment support from DJI.

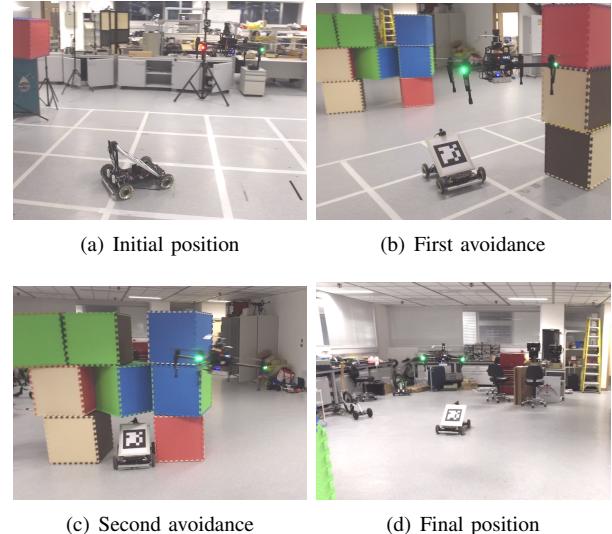


Fig. 1. Snapshots of the real-world experiment as presented in Sect. VI-C. Visualization and statistics of this experiment are shown in Fig. 8. Our experimental testbed is based on the DJI Matrice 100 quadrotor. It is equipped with an Intel NUC (Intel Core i54250U, 16GB RAM), a downward-facing camera for optical flow, a 45-degree-tilted camera for target detection, and a 2D laser range finder for localization and obstacle detection. The moving target is a ground vehicle equipped with an artificial visual marker for easy detection. A video of the experiment can be found at <http://www.ece.ust.hk/~eeshaojie/iros2016jing.mp4>.

feasibility, that must be satisfied or a crash will occur. We highlight the feasibility issue as the generated trajectory should never require the quadrotor to go beyond its maximum velocity and acceleration profile. Otherwise, the quadrotor may fail to follow the trajectory, which eventually result in collision. We are therefore willing to downgrade tracking quality to ensure safety and feasibility of the motion. We propose a formulation to embed the tracking error over the entire tracking horizon, as well as the quadrotor control cost, in the cost function for a quadratic programming (QP), while encoding both collision avoidance and platform dynamics as linear inequality constraints for the QP.

Promising results have been presented to address the problem of estimating and predicting target motion with onboard sensors. Using an RGB-D camera, [4] proposed a *FollowMe* system which can segment the target from the background and then generate waypoints for tracking flight. Using advanced computer vision techniques, estimation systems that use only a monocular camera have been proposed to calculate the target motion relative to the aerial platform [5]–[8]. In particular, the method in [8] is able to precisely reconstruct the target trajectory when the target height is known. [9] presented a method where the target position in the road-

constrained scenario is modeled as a multi-modal Gaussian distribution and is updated using Bayesian estimation. The quadrotor can take macro actions accordingly, even if the target is out of sensor range. In this work, however, we focus on the trajectory planning and control for the tracking problem and use artificial markers to obtain the 3D target's relative position to the aerial robot.

In the previous literature, the trajectory planning and control problem for target-tracking is often treated in a local control setting. Many recent works have adopted vision-based control approaches [10] since cameras are becoming a standard equipment for all kinds of robots. [11]–[13] developed vision-based control methods to enable quadrotors to track a selected moving ground target utilizing the online video stream. However, as they do not consider the existence of obstacles in the flight space, their methods are limited to a few applications in wide open areas. A simple potential field-based method assisted by an external motion capture system was presented in [14]. To ensure flight safety, Jeremy H. Gillula *et al.* introduced guaranteed safe online learning via reachability (GSOLR) [15, 16], in which quadrotors avoid picking up control signals that lead to collision by computing a keep-out set in the state space. The control signal is chosen by minimizing a conditional entropy with the safety constraint. Nevertheless, the computing cost of this method quickly becomes intractable with anything more than a few simple obstacles. It is also unable to take the global information into account and is prone to becoming stuck in local minimums.

The key contribution of this work is an algorithm to generate smooth trajectories for target tracking with guaranteed safety and dynamics feasibility. We formulate the trajectory as a spline (piecewise polynomials), as proposed in [3], thanks to the differential flatness property of quadrotors. Collision avoidance and dynamical feasibility are ensured by developing from our previous methods [17, 18] that focus on trajectory planning for fixed destinations. Starting from noisy visual observations of a target, we estimate and predict the target motion using polynomial fitting. We then formulate the error of the entire predicted horizon, together with the quadrotor control costs (integral of the jerk) into a cost function that is to be minimized in a QP. With a global 3D map that is built online, collision avoidance and dynamics feasibility are formulated as linear inequality constraints for the QP. This way, we encode tracking error as “soft constraints” over the entire time horizon, while ensuring safety as “hard constraints”.

In contrast to local control-based methods, our trajectory generation-based method incorporates global obstacle information and avoids being trapped in local minimums. In other words, as long as the location of obstacles is obtained by onboard sensors, the quadrotor can find a go-around trajectory, if one exists, that is guaranteed to reach the target location. Thanks to the predicted short-horizon target motion, we are even able to recover after short-term loss of visual measurements, as shown in Fig. 7(a) and Fig. 8(a). We optimize to all constraints in one go and provide a simple

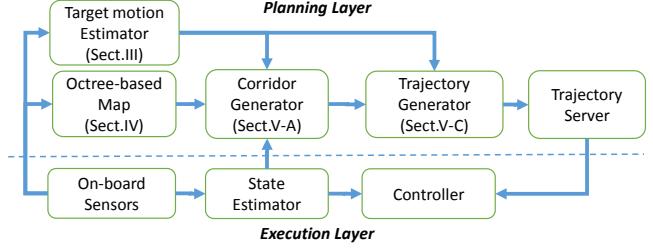


Fig. 2. A diagram of the system architecture. Blocks marked with section numbers are parts of the trajectory planning algorithm described in this paper. Target detection is done using artificial markers [19]. State estimation and control modules are adopted from a state-of-the-art aerial navigation system [20].

parameter tuning interface to adjust to different dynamical properties between different platforms.

We deploy our system on-board a quadrotor testbed equipped with multiple cameras, a laser range finder, an IMU, and an onboard computer. Our online experiments show that, to the best of our knowledge, we are the first to achieve simultaneous aerial tracking of a moving target and collision avoidance in unknown, cluttered environments.

II. OVERVIEW

A. Problem Formulation

Target tracking is a complex engineering system that involves technical modules in hardware, localization, mapping, planning, and control. In this work, we focus on the trajectory planning part of the system. We use engineering solutions on sensing and target detection [19] and state-of-the-art results on aerial navigation [20] to form a complete aerial tracking system. For trajectory planning, we consider a setting with the following assumptions:

- 1) Obstacles are initially unknown to the aerial robot but can be detected online by the on-board omnidirectional range sensors with a limited sensing range.
- 2) The target is expected to have smooth motion with bounded changes in velocity and acceleration.
- 3) The quadrotor state can be estimated online.
- 4) The noisy relative pose of the target can be observed online.
- 5) The quadrotor is expected to maintain a relative position $\mathbf{d}_s \in \mathbb{R}^3$ to the moving target, and its orientation is controlled separately to face the target.

B. System Architecture

The algorithmic flow for trajectory planning is shown in Fig. 2. To represent the environment efficiently, obstacle information is stored in variable-sized 3D cubic grids that are organized in an octree structure [21]. Exploiting observations of the target position provided by onboard sensors, the target trajectory is approximated using polynomial fitting and then predicted for a limited time horizon (Fig. 3(b)). We then shift the estimated target trajectory in the \mathbf{d}_s direction to get the “stiff” trajectory for the following/tracking task (Fig. 3(c)). Afterwards, the collection of map grids, which the shifted target trajectory passes through, can be found. These grids form a grid path, called the *initial flight corridor* (Fig. 3(d)).

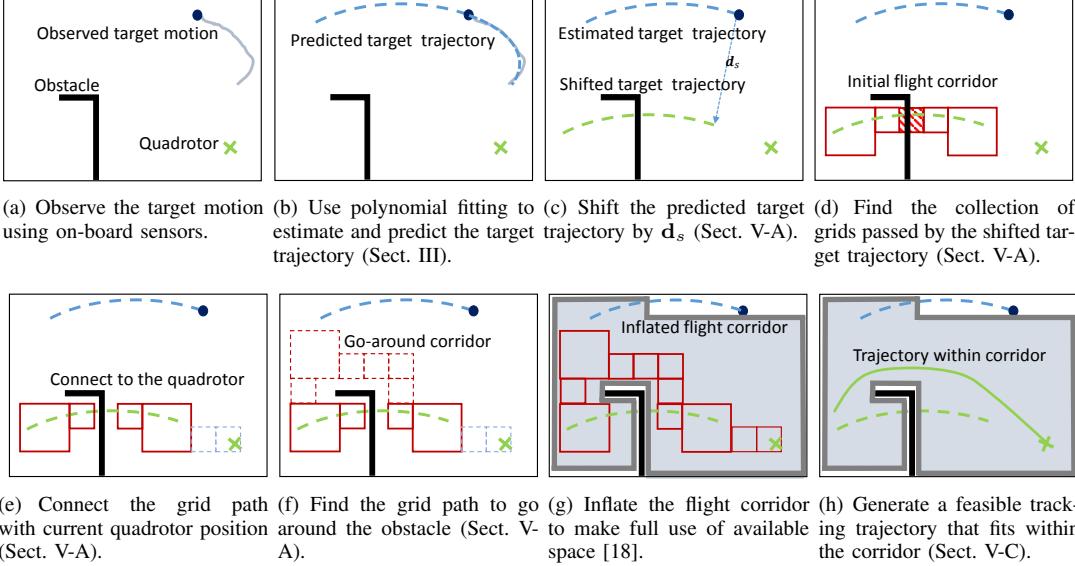


Fig. 3. Graphical illustration of the algorithmic pipeline. The whole pipeline loops at a fixed frequency of 5 Hz.

Portions of this initial flight corridor that intersect with obstacles are adjusted using a multi-start A^* method to find a go-around corridor that is collision-free (Fig. 3(f)). This flight corridor is further connected with the current quadrotor position using A^* (Fig. 3(f)), and then inflated to generate the final collision-free flight corridor for the quadrotor (Fig. 3(g)). Finally, we use our QP-based method to generate a trajectory that fits entirely within the flight corridor while satisfying all vehicle dynamical constraints (Fig. 3(h)). In our implementation, the whole pipeline loops at a fixed frequency of 5 Hz.

The architecture of the overall system is illustrated in Fig. 2. All blocks without a section number are from our prior research on state estimation, mapping, and control [20].

III. TARGET MOTION ESTIMATION AND PREDICTION

We denote the 3D position of the target at time $t \in \mathbb{R}$ as $\mathbf{T}(t) \in \mathbb{R}^3$. Since the target motion is assumed to be smooth, we can rewrite the motion using Taylor expansion and approximate it by omitting terms with degrees higher than n_t :

$$\mathbf{T}(t) = \sum_{i=0}^{\infty} \frac{\mathbf{T}^{(i)}(0)}{i!} \times t^i, \quad (1)$$

$$= \sum_{i=0}^{n_t} \frac{\mathbf{T}^{(i)}(0)}{i!} \times t^i + O(t^{n_t}) \quad (2)$$

$$\approx \begin{bmatrix} \frac{\mathbf{T}^{(0)}(0)}{0!} \\ \frac{\mathbf{T}^{(1)}(0)}{1!} \\ \vdots \\ \frac{\mathbf{T}^{(n_t)}(0)}{n_t!} \end{bmatrix}^T \begin{bmatrix} 1 \\ t \\ \vdots \\ t^{n_t} \end{bmatrix} \quad (3)$$

$$= \hat{\mathbf{T}}(t), \quad (4)$$

The approximated target trajectory $\hat{\mathbf{T}}(t)$ then becomes an n_t degree polynomial of t .

Assuming that the relative position of the target with respect to the quadrotor can be obtained using onboard sensors, we can transform it using the quadrotor position to obtain a sequence of noisy target position observations in the global frame. Let t_0 be the current time, and we have L observations within the time period $[t_l, t_0]$, where t_l is the starting time of a sliding window. The polynomial is learned (III) using classical regression by minimizing the distance error over all L observations:

$$\min_{\hat{\mathbf{T}}(\cdot)} \sum_{i=0}^L \|\hat{\mathbf{T}}(t_i) - \mathbf{p}_i\|_2^2, \quad (5)$$

where \mathbf{p}_i is the i^{th} observed target 3D position in the global frame at time t_i . To avoid over-fitting, we incorporate the prior knowledge that the target motion state does not change drastically by adding an acceleration regulator:

$$\min_{\hat{\mathbf{T}}(\cdot)} \underbrace{\sum_{i=0}^L \|\hat{\mathbf{T}}(t_i) - \mathbf{p}_i\|_2^2}_{\text{estimation residual}} + L\lambda_t \underbrace{\int_{t_l}^{t_m} \|\dot{\hat{\mathbf{T}}}^{(2)}(t)\|_2^2 dt}_{\text{acceleration regulator}}, \quad (6)$$

where λ_t is used to adjust the weighting of the regulator. t_m is the time limit under which we consider the predicted target trajectory. Note here that $[t_l, t_0]$ is the time period that we actually have target observations, and $[t_0, t_m]$ is the short time horizon during which we predict the target motion. This prediction is utilized for the generation of tracking trajectories (Sect. V). Since the integral of the l^2 norm of a polynomial has a quadratic closed form, (6) becomes a constraint-free QP with a closed form solution [22] for $\hat{\mathbf{T}}(\cdot)$.

IV. OCTREE-BASED ENVIRONMENT REPRESENTATION

As in our prior work [18], we use the octree data structure [23] that discretizes the 3D environment into a collection

of variable-sized 3D cubic grids. In this data structure, we initially use a large grid to represent the entire space. If obstacles exist in a grid, we will partition it into eight sub-grids and mark those filled with obstacles. We apply this strategy recursively to each of the sub-grids until the sub-grid is free of obstacles or we reach the finest resolution. This data structure is memory efficient in the sense that it represents large volumes of free space using only a few large grids. It also has a suitable logarithm runtime complexity for volume occupancy inquiry, as shown in [18].

Furthermore, in order to use search-based methods to find a valid flight corridor in an online fashion, a graph that represents the connectivity information between grids is maintained online [18]. This graph is updated locally as obstacles are observed. This saves a significant amount of online computing resources and is proved to be efficient both theoretically and practically.

V. TRAJECTORY GENERATION FOR TARGET TRACKING

In this section, we detail our algorithmic pipeline for generating collision-free tracking trajectories given the predicted target motion. The pipeline is also illustrated in Figs. 3(c)-3(h), with section numbers captioned for easy visualization of our method.

A. Generation of the Flight Corridor

During the tracking task, the quadrotor is expected to keep a fixed distance $\mathbf{d}_s \in \mathbb{R}^3$ from the target. Under the ideal case where no obstacle or platform dynamical constraints exist, the quadrotor is expected to follow the shifted predicted target trajectory $\mathbf{T}_s(t) = \hat{\mathbf{T}}(t) + \mathbf{d}_s$ (Fig. 3(c)). $\mathbf{T}_s(t)$ is also a polynomial with the form $\mathbf{T}_s(t) = \sum_{j=0}^{n_t} [\mathbf{c}_j \cdot t^j]$, where $\{\mathbf{c}_j \in \mathbb{R}^3 | 0 \leq j \leq n_t\}$ are coefficients.

The octree-based map allows us to rapidly find a sequence of connecting grids, called the *flight corridor*, that $\mathbf{T}_s(t)$ passes through. Some of these grids may be blocked by obstacles (Fig. 3(d)). We therefore perform a multi-start A^* search to find a sequence of go-around grids (Fig. 3(f)). This can be done by setting the preceding grid of the blocked grids as the destinations, and by setting all available succeeding grids as starting points. The heuristic function is the Euclidean distance to the destination. The starting grid of this go-around corridor is further connected with the current quadrotor position to form the collision-free flight corridor with m grids (Fig. 3(e)). This search will provide a path that brings the quadrotor close to the shifted target trajectory $\mathbf{T}_s(\cdot)$ as soon as possible, while avoiding obstacles. This collision-free flight corridor will then be inflated into m large overlapping grids, as in [18], to enlarge the configuration space for our optimization-based trajectory generation (Fig. 3(g)).

B. Differential Flatness and Polynomial Trajectories

As shown in [3], a quadrotor enjoys the differential flatness property. This enables us to express the full state of a quadrotor as a function of simple flat outputs, such as the x, y, z position and the yaw angle, and their derivatives.

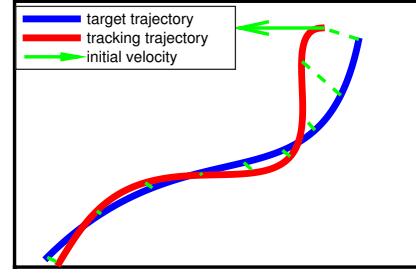


Fig. 4. An illustration of the generated tracking trajectory. Dashed lines represent the time correspondence between two trajectory segments. This illustrates the tracking error minimization over the entire predicted time horizon of the target motion (Fig. 3(b)), which allows the quadrotor to downgrade tracking quality (deviate from the target trajectory) in order to improve safety and reduce dynamic variations.

In this work, we use simple PD control for the yaw angle such that the onboard camera always faces the target. For the 3D position, we utilize the flatness property and express the quadrotor trajectory as three separate piecewise polynomials. For each dimension, we have:

$$\mathbf{f}(t) = \begin{cases} \sum_{j=0}^n [\mathbf{a}_{1j} \cdot (t - t_0)^j] & t_0 \leq t \leq t_1 \\ \vdots \\ \sum_{j=0}^n [\mathbf{a}_{ij} \cdot (t - t_{i-1})^j] & t_{i-1} \leq t \leq t_i \\ \vdots \\ \sum_{j=0}^n [\mathbf{a}_{Mj} \cdot (t - t_{m-1})^j] & t_{m-1} \leq t \leq t_m \end{cases}, \quad (7)$$

Each segment of the trajectory is an n^{th} degree polynomial function on a specified time interval $[t_{i-1}, t_i]$. $\mathbf{a}_{ij} \in \mathbb{R}^3$ is the j^{th} polynomial coefficient for the i^{th} segment. We have the number of polynomial segments equal to the number of grids (m) in the flight corridor.

C. Optimization-Based Trajectory Generation

We now introduce a novel QP based method to generate polynomials, as illustrated in Fig. 3(h), that follow the shifted target trajectory (Fig. 3(c)), while still fitting entirely within the flight corridor, and avoid collision.

Ideally, the shifted target trajectory $\mathbf{T}_s(\cdot)$, as shown in Fig. 3(c), should be followed by the quadrotor to achieve the best tracking result. However, it is often impossible to use $\mathbf{T}_s(\cdot)$ as the actual flight trajectory as $\mathbf{T}_s(\cdot)$ may intersect with obstacles. Moreover, the dynamical capability of the target (such as a human) may be superior to that of the quadrotor with sudden stops and sharp turns, thus the target motion may be impossible for the quadrotor to follow precisely.

It is better to trade off some of the tracking precision to get a graceful flight, with control commands that fit well within the dynamical capability of the quadrotor. To this end, we choose to minimize a cost function that considers jointly the effect of two terms: 1) the tracking error over the entire predicted horizon of target motion; and 2) the integral of the jerk over the entire tracking horizon. The first term brings the quadrotor close to the predicted target trajectory, while

the second term minimizes the control effort for tracking. the dynamic capability of the quadrotor is considered as linear inequality constraints to bound the trajectory acceleration and jerk of the platform. We transform collision avoidance into corridor boundary constraints, which can be formulated as linear inequalities. This is, sufficient as a trajectory that fits entirely within the flight corridor is known to be collision-free. The inflated corridor is able to cover a large volumes of the free space, thus providing the quadrotor with flexible and safe flight areas for target tracking. The overall formulation of the optimization-based trajectory generation can be written as follows:

$$\min_{\mathbf{f}(\cdot)} \underbrace{\int_{t_0}^{t_m} \|\mathbf{f}(t) - \mathbf{T}_s(t)\|_2^2 dt}_{\text{tracking error}} + \lambda \underbrace{\int_{t_0}^{t_m} \|\mathbf{f}^{(3)}(t)\|_2^2 dt}_{\text{jerk regulator}} \quad (8)$$

$$\text{s.t. } \underbrace{\mathbf{f}^{(j)}(t_0) = \mathbf{f}_0^j}_{\text{initial state constraint, } \forall j \in \{0, 1, 2\}} \quad (9)$$

$$\underbrace{\mathbf{f}^{(j)}(t_m) = 0}_{\text{stopping policy constraint, } \forall j = 0 \dots 2} \quad (10)$$

$$\underbrace{\lim_{t \rightarrow t_i-} \mathbf{f}^{(j)}(t) = \lim_{t \rightarrow t_i+} \mathbf{f}^{(j)}(t)}_{\text{smoothness constraint, } \forall t=2 \dots m, j=0 \dots 2} \quad (11)$$

$$\underbrace{\mathbf{f}(t) \in \Omega_p(t)}_{\text{safety constraint}} \quad (12)$$

$$\underbrace{\mathbf{f}^{(1)}(t) \in \Omega_v(t)}_{\text{velocity constraint}} \quad (13)$$

$$\underbrace{\mathbf{f}^{(2)}(t) \in \Omega_a(t)}, \quad (14)$$

acceleration constraint

where $\Omega_p(t)$, $\Omega_v(t)$ and $\Omega_a(t)$ represent, at time t , the flight corridor, reachable velocities, and reachable accelerations, respectively. λ is the weighting term to adjust how “stiff” the tracking should be. Smaller λ results in trajectories that closely track the target motion, while large λ leads to smoother trajectories with downgraded tracking performance.

For the i^{th} segment, both the shifted target trajectory $\mathbf{T}_s(\cdot)$ and the tracking trajectory $\mathbf{f}(t)$ are polynomials, and the tracking error term in (8) is formulated as the integral of the square of the Euclidean distance between the target and the UAV along the entire trajectory. The square-form is chosen not only because it is smooth within the domain, but it also allows a closed-form solution for the integration. Within the time interval $[t_{i-1}, t_i]$, which corresponds to the i^{th} segment, we rewrite the shifted target trajectory $\mathbf{T}_s(\cdot)$ as

$$\mathbf{T}_s(t) = \underbrace{\begin{bmatrix} 1 \\ t - t_{i-1} \\ \vdots \\ (t - t_{i-1})^n \end{bmatrix}}_{\mathbf{t}_i^r}^T \underbrace{\begin{bmatrix} \sum_{k=0}^{n_t} \binom{k}{0} \cdot t_i^{k-0} \cdot \mathbf{c}_{ik}^T \\ \sum_{k=1}^{n_t} \binom{k}{1} \cdot t_i^{k-1} \cdot \mathbf{c}_{ik}^T \\ \vdots \\ \sum_{k=n}^{n_t} \binom{k}{n} \cdot t_i^{k-n} \cdot \mathbf{c}_{ik}^T \end{bmatrix}}_{\mathbf{C}_i} = \mathbf{t}_i^r \mathbf{C}_i, \quad (15)$$

Similarly, we can write the tracking trajectory for the i^{th} segment as

$$\mathbf{f}(t) = \underbrace{\begin{bmatrix} 1 \\ t - t_i \\ \vdots \\ (t - t_{i-1})^n \end{bmatrix}}_{\mathbf{t}_{i-1}^r}^T \underbrace{\begin{bmatrix} \mathbf{a}_{i0}^T \\ \mathbf{a}_{i1}^T \\ \vdots \\ \mathbf{a}_{in}^T \end{bmatrix}}_{\mathbf{A}_i} = \mathbf{t}_i^r \mathbf{A}_i, \quad (16)$$

Now both the target and tracking trajectories are in the same form. Using $l \in \{x, y, z\}$ to abstract the dimension, we can rewrite the integrated tracking error term in (8) as:

$$\int_{t_{i-1}}^{t_i} \|\mathbf{f}(t) - \mathbf{T}_s(t)\|_2^2 dt \quad (17)$$

$$= \int_{t_{i-1}}^{t_i} \sum_{l \in \{x, y, z\}} [\mathbf{t}_i^r (\mathbf{A}_i^l - \mathbf{C}_i^l)]^T [\mathbf{t}_i^r (\mathbf{A}_i^l - \mathbf{C}_i^l)] dt \quad (18)$$

$$= \sum_{l \in \{x, y, z\}} (\mathbf{A}_i^l - \mathbf{C}_i^l)^T \underbrace{\int_{t_{i-1}}^{t_i} (\mathbf{t}_i^r \mathbf{t}_i^r)^T dt}_{\mathbf{Q}_i} (\mathbf{A}_i^l - \mathbf{C}_i^l) \quad (19)$$

$$= \sum_{l \in \{x, y, z\}} \mathbf{A}_i^{lT} \mathbf{Q}_i \mathbf{A}_i^l - 2\mathbf{b}_i^{lT} \mathbf{A}_i^l + \mathbf{C}_i^{lT} \mathbf{Q}_i \mathbf{C}_i^l, \quad (20)$$

where $\mathbf{b}_i^l = \mathbf{Q}_i \mathbf{C}_i^l$, and \mathbf{Q}_i are defined inline (19). Similarly, the jerk regulator term of the i^{th} segment can be written in the quadratic form

$$\int_{t_{i-1}}^{t_i} \|\mathbf{f}^{(3)}(t)\|_2^2 dt \quad (21)$$

$$= \int_{t_{i-1}}^{t_i} \left\| \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{3!}{0!} \cdot (t - t_{i-1})^0 \\ \frac{4!}{1!} \cdot (t - t_{i-1})^1 \\ \vdots \\ \frac{n!}{(n-3)!} \cdot (t - t_{i-1})^{n-3} \end{bmatrix}^T \underbrace{\begin{bmatrix} \mathbf{a}_{i1}^T \\ \mathbf{a}_{i2}^T \\ \mathbf{a}_{i3}^T \\ \mathbf{a}_{i4}^T \\ \mathbf{a}_{i5}^T \\ \vdots \\ \mathbf{a}_{in}^T \end{bmatrix}}_{\mathbf{A}_i} \right\|_2^2 dt \quad (22)$$

$$= \sum_{l \in \{x, y, z\}} \mathbf{A}_i^{lT} \underbrace{\int_{t_{i-1}}^{t_i} \mathbf{t}_i^r \mathbf{t}_i^r dt}_{\mathbf{H}_i} \mathbf{A}_i^l \quad (23)$$

$$= \sum_{l \in \{x, y, z\}} \mathbf{A}_i^{lT} \mathbf{H}_i \mathbf{A}_i^l, \quad (24)$$

where \mathbf{H}_i is defined as above (23). \mathbf{A}_i^l and \mathbf{C}_i^l are the l^{th} column of \mathbf{A}_i and \mathbf{C}_i , respectively, where \mathbf{A}_i and \mathbf{C}_i are defined in (15) and (16). Combining the tracking error and regularization terms together, we can formulate (8) as a standard cost function for the QP:

$$\min_{\mathbf{A}_i} \sum_{i=1}^m \sum_{l \in \{x, y, z\}} [\mathbf{A}_i^{lT} (\mathbf{Q}_i + \lambda \mathbf{H}_i) \mathbf{A}_i^l - 2\mathbf{b}_i^{lT} \mathbf{A}_i^l]. \quad (25)$$

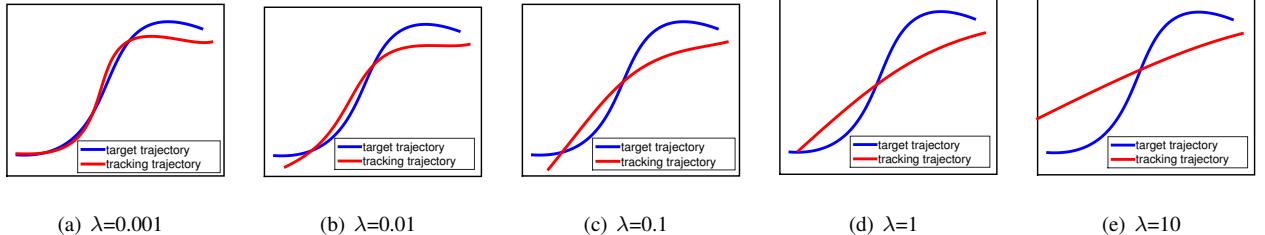


Fig. 5. The generated tracking trajectory with different regulator λ . Higher λ results in trajectories that are smoother but offer downgraded tracking quality (Sect. VI-A).

Since $f(\cdot)$ is a piecewise polynomial function, the initial state (9), final state (10), and the smoothness requirement (11) can be straightforwardly formulated as linear equality constraints. To enforce the safety (12), velocity (13) and acceleration (14) constraints over the entire tracking trajectory, we utilize the proven method proposed in our previous works [17, 18]. We first generate the tracking trajectory without considering any of the interval constraints above. Since the safety, velocity, and acceleration constraints are essentially interval bounds for the 0th, 1st, and 2nd derivatives of the polynomial, we can check all extrema points of the generated polynomials and their derivatives, and add extrema points that violate the corridor boundary or exceeds the preset velocity/acceleration limit as point constraints to the QP. We iteratively perform this process until no constraint violation exists in the entire trajectory. Our method has proven a theoretical bound on the number of constraint points that need to be added to the QP. It is also worthwhile to note that the actual number of constraint points required for practical scenarios is usually far smaller than the theoretical maximum. As a matter of fact, our method is extremely fast in practice, as demonstrated in both simulation and real-world experiments (Sect. VI). We suggest readers refer to [17, 18] for the detailed formulation of the QP constraints and theoretical proof of our method.

With the formulation above, the generation of the tracking trajectory can be done by solving a standard QP problem.

Moreover, the QP formulation in our case is sparse, which suggests that the computing time can be further optimized using open-source QP solvers.

VI. EXPERIMENTAL RESULTS

We conduct two simulations and one real-world flight experiments to demonstrate the performance of the proposed method. To achieve real-time operation, we implement the system in C++11 using a g++ compiler with the -O3 optimization option. The popular open-source linear algebra library Eigen and an open-source quadratic programming solver, Object Oriented software for Quadratic Programming(OOQP) [24], are used for constructing and solving the QP for trajectory generation. In all test cases, the quadrotor takes off without any prior knowledge about the environment, and the whole pipeline loops at a fixed frequency of 5 Hz to achieve real-time obstacle detection and trajectory replanning.

A. Adjusting Tracking "Stiffness"

In this simulation, we show the effect of the tracking regulator λ , as defined in (8) in Sect. V-C, on the overall tracking performance. We set different values of λ from 0.001 to 10. As shown in Figs. 5 and 6, smaller λ allows more agile motions, resulting in "stiffer" tracking trajectories with smaller tracking error. On the other hand, larger λ results in smoother trajectories with smaller changes in the velocity profile, and is thus easier for platforms with slower dynamics to follow. As expected, the tracking error increases with less agile trajectories. It can be seen that the target motion is often more agile than the tracking platform. However, we are able to maintain reasonable tracking performance with all values of λ .

λ offers a convenient way to adjust the tracking performance such that the level of agility of the tracking trajectories matches the expectation of the user. For instance, we might want "stiffer" tracking for monitoring tasks where the focus is on the target itself. On the other hand, we might opt for "softer" tracking for aerial photography, where the focus is the surrounding scene. Nevertheless, we stress that no matter how we adjust λ , the resulting tracking trajectories are always safe and dynamically feasible for the flying platform,

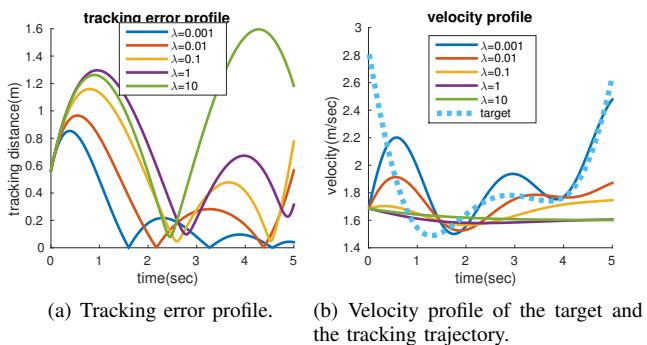


Fig. 6. Comparison between different regulator λ weightings in terms of tracking error and velocity profiles. It can be seen that larger λ results in smoother trajectories but causes larger tracking error (Sect. VI-A).

as these constraints are enforced in our QP formulation (Sect. V-C).

B. Simulated Target Tracking

In this section, we present the simulation results of a quadrotor tracking a moving target in an initially unknown and cluttered environment. We set the range for the target detection sensor as 10 m with a field-of-view of 90 degrees. Target observations are corrupted by Gaussian noise with zero mean and standard deviation of 0.1 m. Obstacles are observed by a simulated laser range finder with a range of 10 m.

In Fig. 7(a), we highlight the tracking trajectory in yellow, and the target motion in red. The quadrotor overshoots a little bit when the target makes sudden velocity changes (A, B, E). Then it takes a slight go-around to avoid obstacles (C), and makes an emergency stop as no dynamically feasible go-around trajectory can be found due to the new obstacles (D). The velocity of the simulated quadrotor changes rapidly when avoiding obstacles that suddenly comes into the sensing range (Fig. 7(b)). Although the quadrotor actively downgrades tracking performance in exchange for flight safety, we can see from Fig. 7(c) that the tracking error is always bounded. We also show the run-time performance of each of the components of our system. The maximum computing time for the whole pipeline is around 60 ms, suggesting real-time capability (Fig. 7(d)).

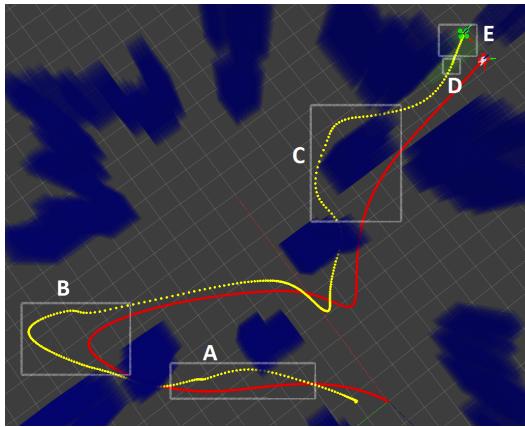
C. Real-World Experiment

We now present a real-world experiment of a quadrotor tracking a moving ground vehicle in an unknown, cluttered indoor environment. The quadrotor testbed, as shown in Fig 1, is based on the DJI Matrice 100 platform and is equipped with an Intel NUC (Intel Core i54250U, 16GB RAM), a downward-facing camera for optical flow, a 45-degree-tilted camera for target detection, and a 2D Hokuyo laser range finder for localization and obstacle detection. The ground vehicle carries an artificial visual marker for easy detection.

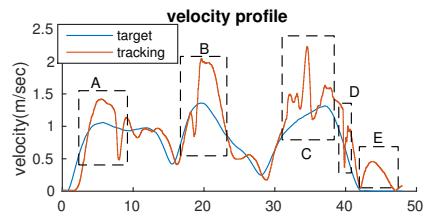
Snapshots and visualization of the experiment can be found in Fig. 1 and Fig. 8(a) respectively. As shown in Fig. 8(b), target observations are extremely noisy. However, the quadrotor is still able to follow the target in a reasonable way due to the polynomial fitting of the target motion and the smooth trajectory generation algorithm. We highlight that during the experiment, the ground vehicle moves into a hole on a block-wall (Fig. 1(c)), resulting in temporary tracking loss, which is also evidenced in Fig. 8(c). However, the quadrotor is still able to finish the avoidance action using the predicted target motion, and re-observe the target on the other side of the wall. The run-time statistics are shown in Fig. 8(d), again demonstrating real-time capability.

VII. CONCLUSION AND FUTURE WORK

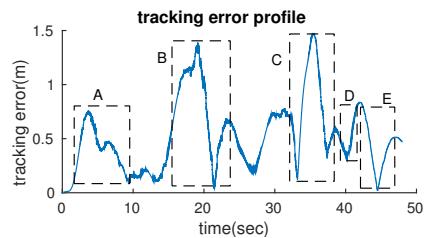
In this paper, we present a novel trajectory planning method for real-time tracking of a moving target using a



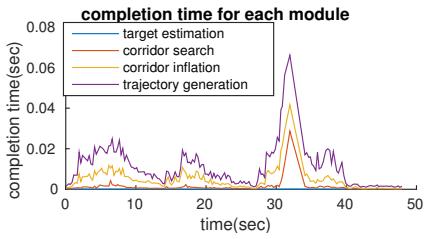
(a) A view of the simulated tracking in a 3D environment. The target trajectory is marked in red. The tracking trajectory of the simulated quadrotor is in yellow. The blue 3D blocks are the generated obstacles with c-space inflation. The trajectory appears to be intersected with obstacles due to the perspective view of the 3D map. The five rectangles are cases that we discuss in detail in the main text (Sect. VI-B).



(b) Velocity profile

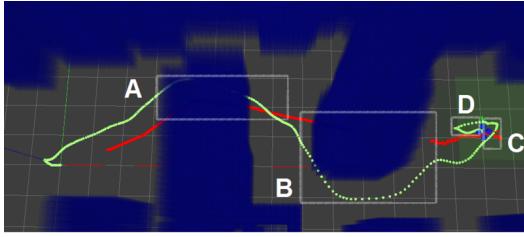


(c) Tracking error profile

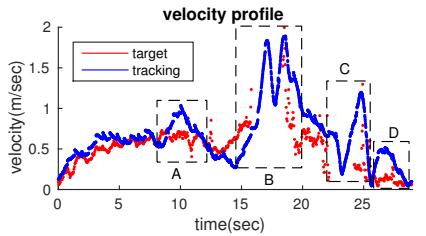


(d) Completion time for each module, as well as the total run-time for the whole system

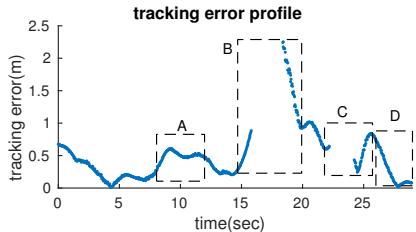
Fig. 7. Results of tracking a moving target in the a simulated unknown, cluttered environment. (Sect. VI-B)



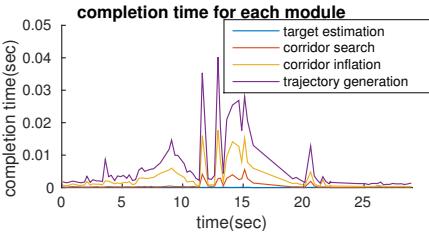
(a) A top-down view of the 3D map during the real-world experiment. The target trajectory is marked in red. The tracking trajectory of the quadrotor is in green. The trajectory appears to be intersected with obstacles due to the perspective view of the 3D map. We highlight rectangles A and B, in which the quadrotor avoids obstacles, resulting in large tracking error. In particular, a temporary tracking loss occurs in B.



(b) Velocity profile



(c) Tracking error profile



(d) Completion time profile

Fig. 8. Visualization and statistics of a real-world target tracking experiment in an unknown, cluttered indoor environment (Sect. VI-C).

quadrotor. Utilizing a novel QP formulation and the previously introduced flight corridor concept, we are able to combine the minimization of tracking error and assurance of safety and dynamics feasibility into a single optimization problem that can be solved efficiently online. To the best of our knowledge, our real-world experiment is the first to demonstrate real-time target tracking in unknown, cluttered environments with a collision avoidance and dynamics feasibility guarantee. In the next stage of our research, we aim to integrate vision-based target detection methods into our system to achieve markerless target tracking.

REFERENCES

- [1] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor," in *Proc. of Robot.: Sci. and Syst.*, Berlin, Germany, June 2013.
- [2] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor UAV on $SE(3)$," in *Proc. of the Intl. Conf. on Decision and Control*, Atlanta, GA, Dec. 2010, pp. 5420–5425.
- [3] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011, pp. 2520–2525.
- [4] T. Naseer, J. Sturm, and D. Cremers, "FollowMe: Person following and gesture recognition with a quadcopter," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Tokyo, Japan, Nov 2013.
- [5] C. Teuliere, L. Eck, E. Marchand, and N. Guenard, "3D model-based tracking for UAV position control," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Taipei, Taiwan, Oct 2010.
- [6] C. Teuliere and L. E. E. Marchand, "Chasing a moving target from a flying MAV," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, San Francisco, CA, US, Sept 2011.
- [7] J. E. Gomez-Balderas, G. Flores, L. R. G. Carrillo, and R. Lozano, "Tracking a ground moving target with a quadrotor using switching control," *J. of Intell. Robot. Syst.*, vol. 70, pp. 65–78, 2013.
- [8] H. Lim and S. N. Sinha, "Monocular localization of a moving person onboard a quadrotor MAV," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Seattle, WA, US, May 2015.
- [9] R. He, A. Bachrach, and N. Roy, "Efficient planning under uncertainty for a target-tracking micro-aerial vehicle," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Anchorage, AK, US, May 2010.
- [10] S. Benhimane and E. Malis, "Homography-based 2D visual tracking and servoing," *Intl. J. Robot. Research*, vol. 26, pp. 661–676, 2007.
- [11] J. W. Kim and D. H. Shim, "A vision-based target tracking control system of a quadrotor by a tablet computer," in *Proc. of the Intl. Conf. on Unm. Air. Syst.*, Atlanta, GA, US, May 2013.
- [12] S. Azrad, F. Kendoul, and K. Nonami, "Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm," *J. of Syst. Design and Dynamic*, vol. 4, pp. 255–265, 2010.
- [13] A. G. Kendall, N. N. Salvapantula, and K. A. Stol, "On-board object tracking control of a quadcopter with monocular vision," in *Proc. of the Intl. Conf. on Unm. Air. Syst.*, Orlando, FL, US, May 2014.
- [14] A. C. Woods and H. M. La, "Dynamic target tracking and obstacle avoidance using a drone," in *The 11th Intel. Sym. on Vis. Comput.*, Las Vegas, NV, USA, Dec 2015.
- [15] J. H. Gillula and C. J. Tomlin, "Guaranteed safe online learning of a bounded system," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, San Francisco, CA, USA, Sept 2011.
- [16] ———, "Guaranteed safe online learning via reachability: Tracking a ground target using a quadrotor," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, San Paul, MN, US, May 2012.
- [17] J. Chen, K. Su, and S. Shen, "Real-time safe trajectory generation for quadrotor flight in cluttered environments," in *Proc. of the IEEE Intl. Conf. on Robot. and Biom.*, Zhuhai, China, Aug. 2015, URL <http://www.ece.ust.hk/~eeshaojie/robio2015jing.pdf>.
- [18] J. Chen, T. Liu, and S. Shen, "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Stockholm, Sweden, May 2016, URL <http://www.ece.ust.hk/~eeshaojie/icra2016jing.pdf>.
- [19] S. G. Jurado, R. M. Salinas, F. J. M. Cuevas, and M. J. M. Jimenez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280 – 2292, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320314000235>
- [20] S. Shen, N. Michael, and V. Kumar, "Obtaining liftoff indoors: autonomous navigation in confined indoor environments," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 40 – 48, 2013.
- [21] M. D. Berg, V. K. Marc, O. Mark, and S. O. Cheong, "Computational geometry," 2000.
- [22] B. Stephen and V. Lieven, *Convex Optimization*. Cambridge University Press, 2004.
- [23] M. Donald, "Geometric modeling using octree encoding," *Comput. graph. and img. proc.*, vol. 19, no. 2, pp. 129–147, 1982.
- [24] E. M. Gertz and S. J. Wright, "Object-oriented software for quadratic programming," *ACM Trans. on Math. Softw.*, vol. 29, pp. 58–81, 2003.