



HAL
open science

Vision-Based Reactive Planning for Aggressive Target Tracking while Avoiding Collisions and Occlusions

Bryan Penin, Paolo Robuffo Giordano, François Chaumette

► **To cite this version:**

Bryan Penin, Paolo Robuffo Giordano, François Chaumette. Vision-Based Reactive Planning for Aggressive Target Tracking while Avoiding Collisions and Occlusions. *IEEE Robotics and Automation Letters*, 2018, 3 (4), pp.3725 - 3732. 10.1109/LRA.2018.2856526 . hal-01836586

HAL Id: hal-01836586

<https://inria.hal.science/hal-01836586>

Submitted on 12 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vision-Based Reactive Planning for Aggressive Target Tracking while Avoiding Collisions and Occlusions

Bryan Penin¹, Paolo Robuffo Giordano², and François Chaumette¹

Abstract—In this paper we investigate the online generation of optimal trajectories for target tracking with a quadrotor while satisfying a set of image-based and actuation constraints. We consider a quadrotor equipped with a camera (either down or front-looking) with limited field of view. The aim is to follow in a smooth but reactive way a moving target while avoiding obstacles in the environment and occlusions in the image space. We propose vision-based approaches based on multi-objective optimization, especially with the occlusion constraint formulation. We design an online replanning strategy inspired from Model Predictive Control (MPC) that successively solves a non-linear optimization problem. The problem is formulated as a Nonlinear Program (NLP) using differential flatness and finite parametrization with B-Splines. This allows a resolution by Sequential Quadratic Programming (SQP) at a rate of 30Hz. The robustness and reactivity of the replanning algorithm are demonstrated through realistic simulation results. Experiments validating the performance with a real quadrotor are also presented.

Index Terms—Reactive and Sensor-Based Planning, Optimization and Optimal Control, Visual-Based Navigation

I. INTRODUCTION

UNMANNED Aerial Vehicles (UAVs) are pushing away the boundaries for applications covering agile navigation, inspection and surveillance to name a few. To be reactive to changes in the environment, the use of on-board cameras has become a fundamental necessity. Indeed, moderately invasive they come with numerous computer vision algorithms for tracking objects, detecting obstacles but also for giving reliable measurements for non-linear observers (e.g., active vision, state estimation). For these reasons, incorporating visual cues in the loop has become a popular approach in robotic control for designing robust and reactive control laws. A historical technique is *visual servoing* [1]. However, this method cannot be directly applied to quadrotors due to their complex dynamics and inherent under-actuation that generates rotations of the camera which may conflict with the main servoing task. Although, dynamic visual servoing schemes have been developed for second order or under-actuated systems (e.g.,

Manuscript received: February, 24, 2018; Revised April, 6, 2018; Accepted July, 9, 2018.

This paper was recommended for publication by Editor Tamim Asfour upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the French Direction Générale de l'Armement (DGA).

¹B. Penin and F. Chaumette are with Inria, Univ Rennes, CNRS, IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France bryan.penin@inria.fr, francois.chaumette@inria.fr

²P. Robuffo Giordano is with CNRS, Univ Rennes, Inria, IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France prg@irisa.fr

Digital Object Identifier (DOI): see top of this page.

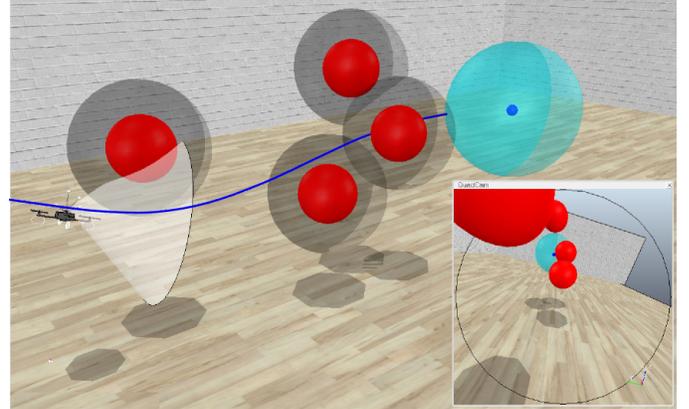


Fig. 1. An optimal trajectory for a quadrotor equipped with a camera is generated for reaching a minimum distance with a target (blue sphere) while avoiding collisions with spherical obstacles (inflated dark spheres) and occlusions of the target from the obstacles (red spheres)

[2], [3] or [4] for quadrotors), the underlying assumptions fail for high speed manoeuvres and in any case, do not take into account possible loss of visibility or occlusions.

Preserving visibility is a substantial concern when vision is in the loop since losing track of features may lead to a failure of the task. Visual servoing original formulation does not prevent critical configurations related to: 1) field of view limits, i.e., the target may become invisible as the camera moves, especially with underactuated robots such as quadrotors; 2) occlusions, i.e., the image features may be occluded due to obstacles or body parts. 3) physical constraints such as feasibility constraints 4) collision with obstacles or self-collision.

When it comes to vision-based control, potential fields are classically used for designing control laws that repel visual cues from the obstacles in the image [5], [6] but this may strongly conflict with the nominal servoing task and increase the chance of falling into local minima. Gradient Projection Methods (GPM) use the system redundancy to mitigate the completion of two tasks [7], [8].

However the redundancy formalism does not appear reasonable when dealing with under-actuated robots such as quadrotors.

In contrast with the aforementioned approaches the present work explicitly considers the full underactuation of the camera which is often problematic for standard visual servoing laws.

Optimization-based planning has motivated a great deal of research with its ability to handle multiple types of constraints and to regulate multiple and possibly conflicting objectives. Nowadays, these numerical approaches have become more and more computationally tractable even for highly non-linear problems. A collection of complex aerial tasks are presented in [9]. When complex motion is sought, optimization is often the chosen approach and has proved its efficiency for generating high speed manoeuvres with quadrotors [10], navigation in heavily cluttered environments [11], passing through narrow gaps [12] and so on.

More recently, self-collision and simple occlusion avoidance tasks for a humanoid were incorporated in a quadratic optimization problem in [13]. Visibility of the feature is handled by using an avoidance task as in [14]. A more precise occlusion avoidance formulation is proposed in [15] but uses a larger set of visual constraints.

In [16] the authors explored a randomized kinodynamic hybrid path planning approach applied to a manipulator for finding a feasible path. It satisfies a great deal of constraints both in the image and in the joint space but takes a few minutes to return a solution.

Finally, optimal problems can be solved online and handled in a continuous control approach similar to MPC [17]. This control framework is very convenient since it merges both reactive control and planning approaches by integrating feedback into the optimal planning. Most relevant works consider optimization for generating effortless collision-free trajectories by predicting the obstacles motion using visual measurements [18] and for more aggressive autonomous target tracking with field of view constraints in terms of acceleration [19]. Along a similar direction, this work considers a multi-objective optimization formulation with extensions for explicit occlusion and collision avoidance with either a down or a front-looking camera (see Fig. 1). In a previous work [20] we presented preliminaries for online minimum-time trajectory replanning under field of view constraints. In this work we will show that reactive manoeuvres can still be achieved without minimizing time and in the presence of more complex constraints. Besides, we improved the accuracy and the numerical stability of the gradients evaluation by using complex-step differentiation to improve the SQP convergence. Our method differs from previous works under three main aspects:

- soft occlusion avoidance formulation in the image space
- efficient and reactive online replanning strategy
- visual constraints independent of the camera/UAV configuration

The rest of this paper is organized as follows. In Section II, we introduce the system and how the problem is formulated as a Nonlinear Program. In Section III, we give the methodology of the vision-based 3D target tracking problem. In Section IV, we present our occlusion and collision avoidance approach. The implementation of the proposed approach is explained in Section V as well as the replanning strategy. Section VII presents the experimental results performed with a quadrotor. Finally, we discuss the results and draw some conclusions in Section VIII.

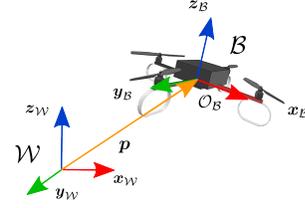


Fig. 2. Quadrotor model

II. PRELIMINARIES

A. Quadrotor dynamics

With reference to Fig. 2, let us define a world frame \mathcal{W} and a body frame \mathcal{B} with origin \mathcal{O}_B attached to the center of mass (COM) and axis z_B parallel to the propeller rotational axes. Let us also assume, without loss of generality, that the robot COM corresponds to the barycentre of the propellers.

The robot state is $\chi = (p, \mathbf{R}, \mathbf{v}, \boldsymbol{\omega}) \in SE(3) \times \mathbb{R}^6$ where $p \in \mathbb{R}^3$ is the position of the robot COM in \mathcal{W} , $\mathbf{R} \in SO(3)$ is the rotation matrix from \mathcal{W} to \mathcal{B} , \mathbf{v} the COM linear velocity expressed in \mathcal{W} and $\boldsymbol{\omega}$ the angular velocity expressed in \mathcal{B} . As well-known (see, e.g., [21]), the quadrotor dynamics can then be modelled as

$$\dot{p} = \mathbf{v} \quad (1a)$$

$$\dot{\mathbf{v}} = \mathbf{g} - \frac{f}{m} z_B \quad (1b)$$

$$\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}]_{\times} \quad (1c)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}([\mathbf{J}\boldsymbol{\omega}]_{\times} \boldsymbol{\omega} + \boldsymbol{\tau}) \quad (1d)$$

where m is the robot mass, $\mathbf{J} \in \mathbb{R}^3$ is the inertia tensor, $\mathbf{g} \in \mathbb{R}^3$ the (constant) gravity acceleration in the world frame, and $[\cdot]_{\times}$ the usual skew-symmetric operator. Finally $(f, \boldsymbol{\tau}) \in \mathbb{R}^4$ are the total thrust and torques applied by the propellers, which can be expressed as a set of control inputs \mathbf{u} in terms of the individual propeller thrusts $\mathbf{u} = (f_1, f_2, f_3, f_4) \in \mathbb{R}^4$ according to the following linear expression

$$\begin{bmatrix} f \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -l & 0 & l \\ -l & 0 & l & 0 \\ b & -b & b & -b \end{bmatrix} \mathbf{u} \quad (2)$$

where l is the distance between the propellers and the COM, and b is the drag constant. The set of admissible control inputs is the box $\mathcal{U} = [f_m, f_M]^4$ with $0 < f_m < f_M$.

Let us also assume the robot to be equipped with an on-board camera whose *fixed* pose w.r.t. \mathcal{B} is known from a preliminary calibration. The spherical projection of a target point in 3D w.r.t. the frame \mathcal{B} is the bearing vector

$$\boldsymbol{\beta} = \frac{\mathbf{R}^T(\mathbf{r} - \mathbf{p})}{\|\mathbf{R}^T(\mathbf{r} - \mathbf{p})\|} = \frac{\mathbf{m}}{\|\mathbf{m}\|} \in \mathbb{S} \quad (3)$$

where $\mathbf{r} \in \mathbb{R}^3$ is the position of the feature in the world frame and \mathbb{S} is the surface of the unit sphere and $\mathbf{m} = (u, v, 1)$ is the image measurement from which $\boldsymbol{\beta}$ is computed. The field of view is then modelled as a cone (see Fig. 1).

B. Differential flatness

A powerful property that can simplify the optimization problem is differential flatness. A non-linear system is termed *flat* if it is possible to find an analytic expression of the system nominal state and control inputs in terms of a finite number of variables (called *flat outputs*) and their derivatives up to a finite order. Then, this transformation gives a linear equivalent description of the system. Introduced by Fliess [22] in the nineties it has been widely used since then in optimization especially for underactuated systems such as unicycles and (more recently) quadrotors (see, e.g., [10], [23], [11] or [24]). Differential flatness can significantly simplify any optimization problem by avoiding any numerical integration of the system dynamics (and the need of introducing the dynamics as differential constraints in the optimization scheme). It is well-known that the quadrotor dynamics (1) are flat with flat outputs $\gamma = (\mathbf{p}, \psi)^T \in \mathbb{R}^4$, where ψ is the yaw angle from the usual roll/pitch/yaw decomposition of the rotation matrix \mathbf{R} .

Indeed, under the assumption $f > 0$, one can find an invertible algebraic mapping of the form:

$$\chi = \phi_\chi(\mathbf{p}, \mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, \psi, \dot{\psi}) \quad (4a)$$

$$(f, \dot{f}, \ddot{f}, \tau) = \phi_u(\dot{\mathbf{v}}, \ddot{\mathbf{v}}, \ddot{\psi}, \dot{\psi}, \ddot{\psi}) \quad (4b)$$

The complete expression of (4) can be found in, e.g., [25], and it is not reported here for lack of space. For simplicity of notation, we indicate with $\sigma = (\mathbf{p}, \mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, \psi, \dot{\psi}, \ddot{\psi})$ the vector of all quantities appearing on the right side of (4), and with $\sigma_\chi = (\mathbf{p}, \mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, \psi, \dot{\psi})$ only those involved in (4a).

Now, any smooth enough trajectory in the flat space is also a feasible trajectory for the system.

C. Trajectory representation

We adopted a finite parametrization of the search space (i.e., of the flat outputs) using piecewise polynomial functions in the B-spline form [26]. Given a vector of *control points* $\theta = (\mathbf{p}_1, \dots, \mathbf{p}_{n_p}, \psi_1, \dots, \psi_{n_\psi}) \in \mathbb{R}^{3n_p+n_\psi}$, and two (fixed) normalized *knot vectors* $\mathbf{s}_p \in [0, 1]^{l_p}$, $\mathbf{s}_\psi \in [0, 1]^{l_\psi}$, the flat output trajectories from generic time t and of duration T can be represented as:

$$\begin{cases} \mathbf{p}(s) = \sum_{k=1}^{n_p} B_k^{d_p} \left(\frac{s-t}{T} \right) \mathbf{p}_k \\ \psi(s) = \sum_{k=1}^{n_\psi} B_k^{d_\psi} \left(\frac{s-t}{T} \right) \psi_k \end{cases}, \forall s \in [t, t+T], \quad (5)$$

where B_k^d is the k -th B-spline positive basis function of degree $d-1$ with minimal compact support where d is the B-spline order. They can be computed recursively as described in [26]. Given (4), in order to ensure state continuity and input boundedness, one has to guarantee Lipschitz continuity of $\ddot{\mathbf{v}}$ and $\ddot{\psi}$ (and continuity of lower order derivatives). This condition can be met by using *open-uniform* distributions of $l = n+d$ ordered knots (i.e. $s_i = 0$, for $i = 1, \dots, d$, $s_i = 1$, for $i = n+1, \dots, l$, and s_d, \dots, s_{n+1} equally spaced in $[0, 1]$) and by taking $d = d_p = 4$ for \mathbf{p} and $d = d_\psi = 2$ for ψ .

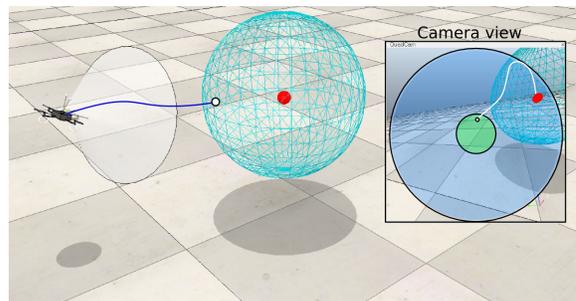


Fig. 3. The quadrotor has to follow the target in red while keeping a safety distance represented by the light blue sphere. The terminal constraint in the image space is represented by the green circle. The blue circle represents the field of view inside which the feature trajectory must lie. Here, the optimal trajectory in blue steers the quadrotor towards a final position (white dot) and the resulting image feature trajectory is the one in white

III. VISION-BASED TARGET TRACKING

In this section, we address the case of tracking a moving 3D target and ensuring final visibility of the target at the unique image center. We assume that the relative pose of the target can be estimated using vision only as done in [19] with some preliminary knowledge of the target model (e.g., the target radius in case of a sphere). We are also interested in allowing the quadrotor undertaking aggressive manoeuvres for reaching the target in near minimum-time conditions. The final goal is to reach a hovering state such that the target appears at the image center while keeping a safety distance from the target (see Fig. 3).

A. Multi-Objective Cost Function

For achieving the aforementioned behaviour we adopt multi-objective programming and minimize the following weighted cost function at running time t with weights w_i

$$\begin{aligned} J = & w_1 \int_t^{t+T} \|\ddot{\mathbf{v}}(t)\|^2 dt + w_2 \int_t^{t+T} \|\ddot{\psi}(t)\|^2 dt \\ & + w_3 \int_t^{t+T} \|d(t) - R_s\|^2 dt + w_4 \int_t^{t+T} \|\mathbf{v}(t)\| dt \end{aligned} \quad (6)$$

The latter is divided into four parts: the first and the second terms minimize the norm of the snap and the norm of the yaw acceleration respectively. They are used for encouraging smoothness while still exploiting the quadrotor aggressiveness [10]. The third term minimizes the error between the distance of the target to the camera and a value R_s defined as a safety distance: the radius of the sphere centred on the target (see Fig. 3). Naturally, to achieve more aggressive trajectories, one can enforce this cost. The fourth term penalizes the path length. Indeed, for a circular symmetric target the quadrotor may fly around the target indefinitely without changing the image feature position, an effect arising from the residual available degrees of freedom and that one can prevent by adding this fourth term.

In [20] we minimized the completion time T to generate aggressive trajectories. Here, we are dealing with more complex constraints and we will show that the optimal trajectories

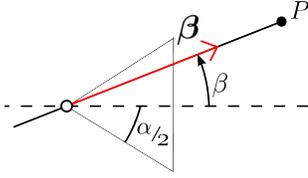


Fig. 4. The visual constraints (7) is equivalent as ensuring that the angular position β of an image point P is lower than the field of view angle $\alpha/2$

can still be aggressive. Moreover, we avoid the penalization of such a complex parameter and the re-evaluation of the B-spline bases at each solver iteration. The completion time T is then a fixed parameter and should be chosen as a rough upper bound of the time required to reach the final pose vicinity. If T is too small, the trajectories might be infeasible. If too high, the system may be less reactive.

Finally, by suitably weighing these costs we are able to trigger the expected behaviour and prevent the objectives from conflicting with each other. Considering a front-looking camera, [27] adopted the minimization of the distance between the target and the camera center which also played the role of limiting visibility loss. However, this approach penalizes the motion aggressiveness by not fully exploiting the image space while, in our case, the target is free to move away from the camera center in order to allow large rotations of the camera and therefore to allow large accelerations of the quadrotor. The adopted visibility constraints are shown below.

B. Visibility constraints

Defining e_c as the camera optical axis in the frame \mathcal{B} , namely, x_B (or y_B) for a front-looking camera and $-z_B$ for a down-looking camera, the visibility constraint is written as

$$\beta^T(t)e_c \geq \cos(\alpha/2), \forall s \in [t, t+T] \quad (7)$$

where α is the angle of view of the camera and β is given by (3). Figure 4 shows that (7) is equivalent as constraining the feature bearing angle β but whose numerical evaluation is more complex.

Instead of imposing equality constraints on the final position in order to guarantee visual convergence of the target to the camera center, we rather define a terminal constraint such that the feature bearing angle has to belong to a (smaller) angular area at the camera centre (see Fig. 3) with

$$\beta^T(t)e_c \geq \cos(\gamma/2), \forall s \in [t+T_v, t+T], \quad 0 \leq T_v \leq T \quad (8)$$

where γ is the angle defining the circular region of convergence in the image. The time T_v is a parameter that defines at what time the feature shall enter the vicinity region. It can be tuned to affect the convergence rate towards this region. Namely, a value closer to 0 will demand a longer activation of the constraint. T_v will also vary depending on the camera orientation due to the quadrotor dynamics.

With the above definitions, one can encapsulate visibility constraints and visual convergence for any camera orientation in the optimal framework defined in the Section V.

IV. VISION-BASED OBSTACLE AND OCCLUSION AVOIDANCE

In this section, we describe the vision-based optimization scheme for avoiding collisions with static spherical obstacles of radius R_{occ} and the occlusions they might generate with the tracked target. Collision-free trajectories with N static spherical obstacles of inflated radius $R_{col} > R_{occ}$ at position r_{obs} are generated using volumetric constraints such as

$$\|p - r_{obs_i}\|^2 > R_{col_i}^2, \quad i = 1, \dots, N \quad (9)$$

The occlusion constraint can be modelled analogously to (9), but in the image space, as follows

$$\|\beta - \beta_{obs_i}\|^2 > a_1^2, \quad i = 1, \dots, N \quad (10)$$

where β and β_{obs} are the spherical projections of the target and the obstacle center respectively, and a_1 is the length of the semi-minor axis of the projected ellipse of the spherical obstacle of radius R_{occ} in the image plane. Assuming knowledge of the size of an obstacle in 3D with coordinates $\mathbf{X} = (X_o, Y_o, Z_o)$ in the camera frame one has [28]

$$a_1^2 = \frac{R_{occ}^2}{4(Z_o^2 - R_{occ}^2)} \quad (11)$$

With this constraint one seeks to prevent the target from *colliding* the obstacles in the image space by keeping a minimum distance a_1 that grows as the depth Z_o of the obstacle in the camera frame decreases. As the quadrotor moves towards the target the occlusion constraints from the obstacles passing behind the camera are of course discarded. However, when dealing with occlusion avoidance with a quadrotor there exist configurations where strict avoidance is not feasible. Indeed, when the target goes exactly below an obstacle (for a down-looking camera, see Fig. 5) the quadrotor may not have sufficient actuation capability or sufficient space for avoiding any occlusions. These situations may occur for any camera orientation. Therefore, in order to avoid such critical situations and always provide a feasible solution, we introduce a *slack variable* λ within the occlusion constraint to authorize partial occlusion if necessary. One has

$$a_1^2 = \frac{(R_{occ} - \lambda)^2}{4(Z_o^2 - (R_{occ} - \lambda)^2)}, \quad s.t. \quad 0 \leq \lambda \leq R_{occ} \quad (12)$$

The λ term plays the role of relaxing a hard constraint when the solver encounters not feasible situations. At the most critical configurations when the target is below an obstacle (Fig. 5), the collision constraint can be reduced or even cancelled by having λ reach the value R_{occ} which is the actual radius of the obstacle (lower than R_{col}). With equation (12) we observed very reactive responses from the system in case of occlusions. This may be due to the fact that small changes of λ induce a strong action on the occlusion constraint (10). Besides, by imposing a straightforward upper bound (R_{col}) for λ , its action will take effect only in case of violations of constraint (10).

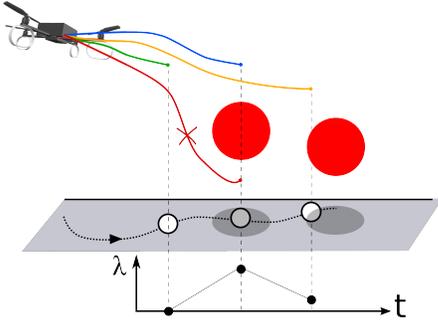


Fig. 5. In case of strong and sudden occlusions when the target is below an obstacle, the red trajectory cannot be a viable solution since it may violate actuation and/or spatial constraints. We instead allow minimal constraint violations to keep the solver efficient. The blue trajectory is then a relaxed solution where λ hits its limit value R_{occ} (obstacle radius). The orange trajectory represents the case of a less relaxed occlusion constraint where λ eventually reaches a smaller value and is zero when there are no occlusions (green trajectory)

It would be of course possible to use the ellipse semi-major axis defined as:

$$a_2^2 = \frac{R_{occ}^2(X_o^2 + Y_o^2 + Z_o^2 - R_{occ}^2)}{4(R_{occ}^2 - Z_o^2)^2} \quad (13)$$

However, in practice one has $X_o^2 + Y_o^2 \ll Z_o^2$ especially if one considers the limited camera field of view. This is why we consider the semi-minor axis (11) which is also less complex. Of course, the occlusion constraint and the slack variable λ introduce conservatism to some extent. However, the main objective of λ is to improve stability and continuity of the solution in case of critical configurations.

As it is common in MPC, the trajectories are sampled. The discretization step should be chosen regarding the obstacles resolution and the trajectories smoothness since the motor thrusts can be very sharp with the considered B-Spline minimal orders.

V. OPTIMIZATION PROBLEM DEFINITION

Finally, the current optimization problem can be stated as the following static NLP

Problem 1: find θ, λ such that:

$$\min_{\theta, \lambda} J + w_5 \|\lambda\|^2 \quad (14a)$$

$$\text{s.t. } \sigma_{\chi}(t) = \sigma_{\chi_t}, \quad (14b)$$

$$\sigma^{(i)}(t+T) = \sigma_{\chi^*}, i = 1, \dots, 3, \quad (14c)$$

$$\mathbf{u}(s) \in \mathcal{U}, \forall s \in [t, t+T], \quad (14d)$$

$$\beta^T(t)e_c \geq \cos(\alpha/2), \forall s \in [t, t+T], \quad (14e)$$

$$\beta^T(t)e_c \geq \cos(\gamma/2), \forall s \in [t+T_v, t+T] \quad (14f)$$

$$\|\mathbf{p}(s) - \mathbf{r}_{obs}\|^2 > R_{col}^2, \forall s \in [t, t+T], \quad (14g)$$

$$\|\beta(s) - \beta_{obs}\|^2 > a_1^2, \forall s \in [t, t+T], \quad (14h)$$

$$0 \leq \lambda \leq R_{occ} \quad (14i)$$

where $\sigma_{\chi_t} = \phi_{\chi}^{-1}(\chi_t)$ and $\sigma_{\chi^*} = \phi_{\chi}^{-1}(\chi^*)$. For a final hovering state χ^* one has of course $\sigma^{(i)}(t+T) = 0, i = 1, \dots, 3$. We choose not to explicitly constrain the final cartesian position which is considered as a free parameter to be determined

by the optimization algorithm. We seek feasible trajectories with constraint (14d) but, due to unknown target motion, the quadrotor might not have sufficient actuation to always satisfy the visibility constraint (14e).

The costs and constraints gradients in the flat space are computed analytically. However, it is more efficient to estimate the gradients of the visual and inputs constraints due to their complex and heavy analytic formulation. More especially, we use complex-step differentiation [29]. By using the following first-order approximation

$$\frac{\partial f}{\partial x} = \frac{\text{Im}\{f(x + ih)\}}{h} + \mathcal{O}(h^2), \quad h \in \mathbb{R} \quad (15)$$

This technique is attractive to determine first derivatives since it only requires a single evaluation of the function and avoids the problem of subtractive cancellation of classic finite approximation (round-off errors). Therefore, it is known to have superior accuracy (close to analytic accuracy) and numerical stability as analysed in [30]. It is also less intrusive in terms of program transformation than automatic differentiation which can sometimes require large and deep source overloads. On the other hand, complex differentiation requires some mathematical adaptations to be used with complex values (the square root or the absolute value functions for instance) which increases the computational cost. The choice between these two methods then hinges on a trade-off between ease of implementation and execution efficiency and is further discussed in [31].

At this point, any off-the-shelf optimization solver can be used to find a numerical solution to Problem 1. Unfortunately, due to the non trivial non-linearities in its formulation (e.g., (14d), (14h)), Problem 1 cannot be proven to be convex. The optimization will thus, in general, return a local minimum.

A. Trajectory replanning Strategy

In order to be reactive to the target unknown motion and increase robustness to the different sources of disturbance (e.g., noise, miscalibrations, neglected dynamics, and so on) that will make the robot to diverge from the planned trajectory, we take inspiration from MPC. Namely, we perform an *on-line* re-planning of an optimal trajectory by solving Problem 1 to incorporate a *feedback* action.

SQP can solve non-convex problems relatively fast by iterating from a *good* initial guess. For that, we use the same replanning process detailed in our previous work [20] where we discussed how to instantly generate an initial guess by operations on the B-Splines only. This method exploits two important properties of B-Splines. The *local support* property implies that the shape of the curve in a knot span (s_k, s_{k+1}) is only determined by a subset of the B-spline control points. The *convex hull* property guarantees, instead, that in each knot span, the spline curve is locally contained in the convex hull of the same subset of control points.

Using the proposed B-Spline adaptation, we achieve faster convergence to a local optimum and provide more stability and continuity of the initial guess than using random search techniques for example.

B. Down-looking case

In this section, we express the replanning strategy by solving Problem 1 for tracking a target on the ground and considering a down-looking camera with a (more realistic) pyramid-shaped field of view as in [20]. The *perspective* projection of the target is considered

$$\beta = \frac{\mathbf{R}^T(\mathbf{r} - \mathbf{p})}{z_B^T \mathbf{R}^T(\mathbf{r} - \mathbf{p})} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \in \mathbb{P}^2 \quad (16)$$

where \mathbb{P}^2 is the space of 3-D homogeneous vectors.

We adopt here the minimization of the distance error by the penalization of the position error with the target in the XY plane $\mathbf{e}_{xy} \in \mathbb{R}^2$.

Moreover, we let the final height z_T free but bounded for limiting the motion along $z_{\mathcal{V}}$. This lighter formulation will basically compute trajectories for steering the quadrotor to a final hovering position such that the target appears at the camera center.

$$J_2 = w_1 \int_t^T \|\ddot{\mathbf{v}}(t)\|^2 dt + w_2 \int_t^T \|\ddot{\psi}(t)\|^2 dt + w_3 \int_t^T \|\mathbf{e}_{xy}(t)\|^2 dt + w_4 z_T + w_5 \|\lambda\|^2 \quad (17)$$

We define the following optimal problem

Problem 2: find θ, λ such that:

$$\min_{\theta, \lambda} J_2 \quad (18a)$$

$$s.t. \quad \sigma_{\chi}(t) = \sigma_{\chi_t}, \quad (18b)$$

$$\sigma^{(i)}(t+T) = 0, i = 1, \dots, 3, \quad (18c)$$

$$\mathbf{p}_{xy}(t+T) = \mathbf{p}_{xy}^*, \quad (18d)$$

$$\psi(t+T) = \psi^*, \quad (\text{arbitrary value}) \quad (18e)$$

$$\beta(s) \in \Omega, \forall s \in [t, t+T], \quad (18f)$$

$$\mathbf{u}(s) \in \mathcal{U}, \forall s \in [t, t+T], \quad (18g)$$

$$\|\mathbf{p}(s) - \mathbf{r}_{obs}\|^2 > R_{col}^2, \forall s \in [t, t+T], \quad (18h)$$

$$\|\beta(s) - \beta_{obs}\|^2 > a_1^2, \forall s \in [t, t+T], \quad (18i)$$

$$0 \leq \lambda \leq R_{occ}, \quad (18j)$$

$$Z_{min} \leq z_T \leq Z_{max} \quad (18k)$$

Where \mathbf{p}_{xy}^* is the target position in the XY plane at time t and the (square) image domain $\Omega = \{\beta \in \mathbb{P}^2 \text{ s.t. } \max(|\beta^T \mathbf{x}_B|, |\beta^T \mathbf{y}_B|) \leq \tan(\alpha)\}$. It should be noted that the gradient of the occlusion constraint (18f) gives three possible directions for the decision variables to satisfy this constraint: 1) increase the distance between $\beta(s)$ and β_{obs} , 2) reduce the quantity a_1 by increasing Z_o , 3) relax the constraint by increasing λ . In order to restrain the second direction that steers the solution towards the local minima of infinite height ($a_1 \rightarrow 0$) we put an upper limit to the quadrotor final height z_T with an additional constraint (18k). We also minimize z_T (in θ) to prevent the camera from staying at the maximum height Z_{max} . The constraint also helps reducing unnecessary and tricky motion towards the opposite direction by considering

a minimum height Z_{min} . This formulation allows efficient replanning at the rate of 30Hz. The next section gives more insight on the different system behaviours with respect to the given weights.

VI. SIMULATION RESULTS

The presented approach was validated in a physically realistic environment. The inertial parameters of the quadrotor were slightly biased on purpose to introduce model uncertainties. Then, since the controller cannot perfectly track the desired trajectories the replanning strategy played an important role for re-adapting the reference trajectory online. The quadrotor dynamics were simulated using V-Rep¹ at 150Hz. The replanning strategy described in Sect. V-A was implemented in C++ within ROS and we exploited the SQP optimization routine from NLOPT [32] to compute the optimal solutions. The generated trajectories of the flat outputs were sent to TeleKyb [33] which then computed the actual control inputs using controller [34].

The tracked target shown in Fig. 5 is manually commanded in velocity saturated at 1.0 m/s. We considered three spherical obstacles of radius $R_{occ} = 0.15m$ and of inflated radius $R_{col} = 0.4m$ (see Figure 8).

We invite the reader to refer to the attached video for a better insight of the results and to a higher-resolution video at <https://www.youtube.com/watch?v=mvvF1I72HM8>

Table I shows the solver parameters used for computing the optimal trajectories from Problem 2, respectively: the number of computed trajectories, the duration T_{max} for the stopping criterion, the planning horizon T , the range of SQP iterations and the mean iteration, the step and constraint tolerance and finally the percentage of total optimization instances that failed to return a solution within T_{max} . When it occurs the previous solution is returned.

loops	T_{max}	T	$iter_{range}$	$\overline{SQP_{iter}}$	tol	ratio
2734	33ms	3.5s	[6, 63]	31	10^{-4}	6%

TABLE I

SETTINGS AND PERFORMANCE OF THE SOLVER FOR A 90S FLIGHT

The evolution of λ and the the final height z_T during the simulation is show in Fig. 6 and Fig. 7 respectively. The

In the proposed formulations, the weights in the cost function are free parameters and need to be tuned empirically in order to generate the expected behaviour. Since the optimal problem is complex and highly nonlinear, any set of weights can generate a different response. However, from our experience the tuning triggers consistent responses and does not necessitate a particularly fine analysis. Generally, a higher w_1 will produce smoother trajectories and less aggressive motion. A higher w_3 will increase the convergence rate towards the target and a lower w_5 will tend to produce more relaxed occlusion constraints. Table II shows the numerical parameters used for the simulation. Moreover, we have used five different sets of weights (Table III) an discuss their consistent effect on the generated trajectories shown in Fig. 8.

¹<http://www.coppeliarobotics.com/>

T	w_1	w_2	w_3	w_4	w_5
3.5s	$1e^{-5}$	$5e^{-3}$	$1e^1$	5	$5e^3$

TABLE II
CONSIDERED PARAMETERS FOR THE SIMULATION

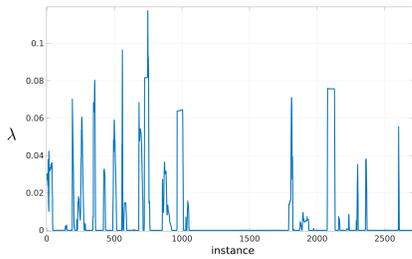


Fig. 6. Evolution of λ for the simulation. The peaks reveal the presence of sudden occlusions. The flat section (from around instance 1100 to instance 1700) represents a flight period far from the obstacles. Therefore, λ is very close to 0. Here λ did not hit the upper limit $R_{occ} = 0.15m$.

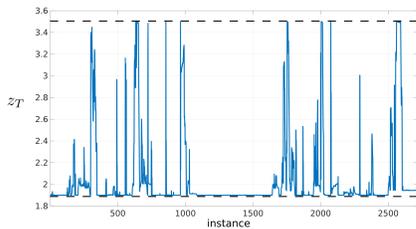


Fig. 7. Evolution of z_T for the simulation. The final height was bounded between 1.9m and 3.5m. Again, z_T reaches its minimum value when no collisions and occlusions occur.

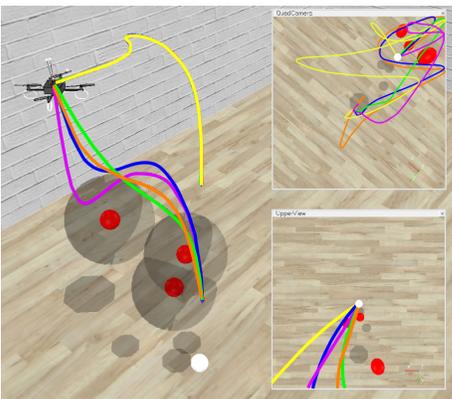


Fig. 8. Different solution trajectories returned with different sets of weights (in Table III) in our simulation environment. The camera view is shown in the upper right corner with the image trajectories and a view from above the target is shown below. The orange trajectory uses the chosen values (in Table II) with a higher w_3 . One can observe a faster convergence towards the image center (with an overshoot). The blue one has a higher w_1 resulting in a smoother trajectory than the orange one. The green one is even smoother and is more compliant to occlusions (less jerky) since w_5 is lower. The yellow trajectory is very sharp (low w_1) because the occlusion constraints are more respected (high w_5) and the final height is less penalized (low w_4). The pink solution gives a different and smoother path (high w_1) that benefits less from the constraints relaxation (since w_5 is high).

VII. EXPERIMENTAL RESULTS

This section reports the experiment meant to validate the proposed replanning strategy applied to the tracking of a target

Trajectory	w_1	w_2	w_3	w_4	w_5
orange	$1e^{-5}$	$5e^{-3}$	$5e^1$	5	$5e^3$
blue	$1e^{-4}$	$5e^{-3}$	$1e^1$	$1e^{-1}$	$5e^3$
green	$1e^{-3}$	$5e^{-3}$	$1e^1$	$1e^1$	$1e^3$
yellow	$1e^{-5}$	$5e^{-3}$	$1e^1$	1	$5e^4$
pink	$1e^{-3}$	$5e^{-3}$	$1e^1$	5	$1e^4$

TABLE III

VALUES OF THE WEIGHTS USED FOR GENERATING THE TRAJECTORIES SHOWN IN FIG. 8

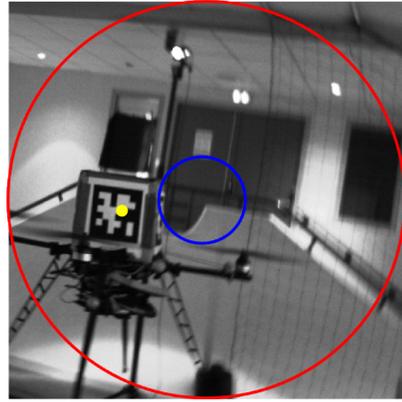


Fig. 9. The center of the box (yellow dot) attached on a second quadrotor and covered by four Apriltags had to remain inside the field of view of the camera (red circle) during the entire flight and appear inside the blue circle in the end of the computed trajectories

with the visibility constraints detailed in Sect. III-B (the reader is also referred to the attached video). For the experiment illustrated in this section we used a MK-Quadro from MiKroKopter equipped with a front-looking camera with a field of view of 90° . The setup included an on-board ODROID-XU4 Linux Computer running ROS and the TeleKyb framework for interfacing the replanning algorithm ran on a standard desktop PC (Quadcore Intel i7 CPU@2.6 GHz). A Vicon motion capture system was employed for giving state measurements of the quadrotor and position measurements of the tracked target. We used Apriltags as a generic target attached on top of a other MK-Quadro controlled remotely (Fig. 9).

VIII. CONCLUSION AND FUTURE WORK

In this work, we adapted our previous approach [20] on vision-based optimal trajectory generation to a wider context by considering reactive target tracking and both occlusion and obstacle avoidance for either a front or a down looking camera (or any other camera orientation).

The quadrotor trajectories are mainly driven by vision while seeking aggressive but smooth trajectories that respect actuation and sensor limits for any camera orientations. Then, starting from a good initial guess the solver was able to return an optimal solution about 94% of the time within 33ms allowing an online replanning strategy capable of absorbing noise, disturbances and any non-modelled effect for long duration flights. The same strategy was applied during an experiment using a real quadrotor for the case of a front-looking camera.

In contrast with the previous literature, we proposed a method that explicitly handles visibility constraints and occlu-

sion avoidance within a fast online replanning strategy. The occlusion constraint is expressed as a single constraint per object in contrast with [15] and could be applied to non-spherical objects to some extent by finding the enclosing sphere. For elongated objects such as bars, one could cover the object main axis with a finite number of spheres as it is done sometimes for collision avoidance with robot manipulators (known as “sphere swept models” see e.g., [35]). Thus, there is a trade-off between conservatism/accuracy and ease of computation. Finally, we coped with the issue that the discontinuity of occlusion constraints can generate by introducing a minimal relaxation in Sect. IV.

To be successful, our path-planning approach requires a complete knowledge of the environment and robot model. These requirements can be limiting in many real applications. The need for such exact knowledge could be relaxed by accounting for modelling and calibration uncertainties at planning stage. Finally, future work includes validating the method with real (and maybe dynamic) obstacles and the use of vision only for estimating the target relative position. In this case the relaxation term will play an even more decisive role.

REFERENCES

- [1] F. Chaumette and S. Hutchinson, “Visual servo control. I. Basic approaches,” *2006 IEEE Robotics & Automation Mag.*, vol. 13, no. 4, pp. 82–90, December 2006.
- [2] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, “Image-Based Visual Servo Control of the Translation Kinematics of a Quadrotor Aerial Vehicle,” *IEEE Trans. on Robotics*, vol. 25, no. 3, pp. 743–749, 2009.
- [3] H. Jabbari and G. Oriolo and H. Bolandi, “An Adaptive Scheme for IBVS of an Underactuated UAV,” *Int. J. of Robotics Research*, vol. 29, no. 1, pp. 92–104, 2014.
- [4] R. Mebarki, V. Lippiello, and B. Siciliano, “Nonlinear visual control of unmanned aerial vehicles in gps-denied environments,” *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 1004–1017, 2015.
- [5] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, “Visual servoing via navigation functions,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 521–533, 2002.
- [6] Y. Mezouar and F. Chaumette, “Path planning for robust image-based control,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.
- [7] E. Marchand and G. D. Hager, “Dynamic sensor planning in visual servoing,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3. IEEE, 1998, pp. 1988–1993.
- [8] N. Mansard and F. Chaumette, “A new redundancy formalism for avoidance in visual servoing,” in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 468–474.
- [9] M. Geisert and N. Mansard, “Trajectory generation for quadrotor based systems using numerical optimal control,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2958–2964.
- [10] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.
- [11] B. Landry, R. Deits, P. R. Florence, and R. Tedrake, “Aggressive quadrotor flight through cluttered environments using mixed integer programming,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1469–1475.
- [12] D. Falanga, E. Mueggler, M. Faessler, and D. Scaramuzza, “Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5774–5781.
- [13] D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette, “Visual servoing in an optimization framework for the whole-body control of humanoid robots,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 608–615, 2017.
- [14] D. Folio and V. Cadenat, “A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment,” in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC’05. 44th IEEE Conference on*. IEEE, 2005, pp. 3898–3903.
- [15] D. Nocolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, “Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots,” *IEEE Robotics and Automation Letters*, 2018.
- [16] M. Kazemi, K. K. Gupta, and M. Mehrandezh, “Randomized kinodynamic planning for robust visual servoing,” *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1197–1211, 2013.
- [17] H. Michalska and D. Q. Mayne, “Robust receding horizon control of constrained nonlinear systems,” *IEEE transactions on automatic control*, vol. 38, no. 11, pp. 1623–1633, 1993.
- [18] F. Gao and S. Shen, “Quadrotor trajectory generation in dynamic environments using semi-definite relaxation on nonconvex qcqp,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 6354–6361.
- [19] J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar, “Autonomous flight for detection, localization, and tracking of moving targets with a small quadrotor,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1762–1769, 2017.
- [20] B. Penin, R. Spica, P. R. Giordano, and F. Chaumette, “Vision-based minimum-time trajectory generation for a quadrotor uav,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS’17*, 2017.
- [21] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *IEEE Robotics & Automation Mag.*, vol. 19, no. 3, pp. 20–32, 2012.
- [22] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, “Flatness and defect of nonlinear systems: Introductory theory and examples,” *International journal of control*, vol. 61, no. 6, pp. 1327–1361, 1995.
- [23] C. Richter, A. Bry, and N. Roy, “Polynomial trajectory planning for quadrotor flight,” in *Proc. of the IEEE Intl. Conf. on Robot. and Autom., Karlsruhe, Germany*, 2013.
- [24] M. Tognon and A. Franchi, “Position tracking control for an aerial robot passively tethered to an independently moving platform,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1069–1074, 2017.
- [25] M. Faessler, A. Franchi, and D. Scaramuzza, “Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018.
- [26] C. D. Boor, *A practical guide to splines*. Springer-Verlag New York, 1978, vol. 27.
- [27] M. Sheckells, G. Garimella, and M. Kobilarov, “Optimal Visual Servoing for Differentially Flat Underactuated Systems,” in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016.
- [28] R. Spica, P. R. Giordano, and F. Chaumette, “Active structure from motion for spherical and cylindrical targets,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5434–5440.
- [29] W. Squire and G. Trapp, “Using complex variables to estimate derivatives of real functions,” *Siam Review*, vol. 40, no. 1, pp. 110–112, 1998.
- [30] R. Abreu, D. Stich, and J. Morales, “On the generalization of the complex step method,” *Journal of Computational and Applied Mathematics*, vol. 241, pp. 84–102, 2013.
- [31] J. Martins, P. Sturdza, and J. Alonso, “The connection between the complex-step derivative approximation and algorithmic differentiation,” in *39th Aerospace Sciences Meeting and Exhibit*, 2001, p. 921.
- [32] S. G. Johnson, “The nlopt nonlinear-optimization package.” <http://ab-initio.mit.edu/nlopt>.
- [33] V. Grabe, M. Riedel, H. H. Bühlhoff, P. R. Giordano, and A. Franchi, “The TeleKyb Framework for a Modular and Extendible ROS-based Quadrotor Control,” in *Mobile Robots (ECMR), 2013 European Conference on*. IEEE, 2013, pp. 19–25.
- [34] T. Lee, M. Leokyand, and N. H. McClamroch, “Geometric tracking control of a quadrotor UAV on SE(3),” in *49th IEEE Conf. on Decision and Control*, Atlanta, GA, Dec. 2010, pp. 5420–5425.
- [35] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, “Real-time collision avoidance with whole body motion control for humanoid robots,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 2053–2058.