2022 International Conference on Unmanned Aircraft Systems (ICUAS)
Dubrovnik, Croatia. June 21-24, 2022

A Perception-Aware NMPC for Vision-Based Target Tracking and Collision Avoidance with a Multi-Rotor UAV

Andriy Dmytruk[1], Giuseppe Silano[1], Davide Bicego[2], Daniel Bonilla Licea[1], and Martin Saska[1]

2022 International Conference on Unmanned Aircraft Systems (ICUAS) | 978-1-6654-0593-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICUAS54217.2022.9836071

***Abstract*— A perception-aware Nonlinear Model Predictive Control (NMPC) strategy aimed at performing vision-based target tracking and collision avoidance with a multi-rotor aerial vehicle is presented in this paper. The proposed control strategy considers both realistic actuation limits at the torque level and visual perception constraints to enforce the visibility coverage of a target while complying with the mission objectives. Furthermore, the approach allows to safely navigate in a workspace area populated by dynamic obstacles with a ballistic motion. The formulation is meant to be generic and set upon a large class of multi-rotor vehicles that covers both coplanar designs like quadrotors as well as fully-actuated platforms with tilted propellers. The feasibility and effectiveness of the control strategy are demonstrated via closed-loop simulations achieved in MATLAB.**

***Index Terms*— Vision Based Navigation and Control, Aerial Systems: Applications, Multi-Rotor UAVs, Nonlinear MPC.**

I. Introduction

In the last decade, the research and commercial interest on Unmanned Aerial Vehicles (UAVs) has exploded as demonstrated by the growing number of applications, such as infrastructure monitoring [1], aerial filming [2], surveillance and search and rescue missions [3] and wireless communications [4]. Most of these applications require UAVs with visual sensors and with certain autonomy that enable them to rapidly react in dynamic environments to successfully accomplish their mission. To achieve this objective, perception constraints must be integrated within the UAVs control framework.

Numerous control strategies have been proposed to perform this integration. Among these, the Model Predictive Control (MPC) scheme has been proven to be a promising solution to control the UAV motion while complying with its dynamics and multiple heterogeneous constraints, such as those coming from the visual part [5]–[7]. Specifically, Nonlinear Model Predictive Control (NMPC) has resulted particularly suitable to control aerial vehicles when their agility is essential for the specific application and must be exploited at the best [8]. In this formulation, nonlinear models and constraints are used to predict the UAV dynamics alongside with the mission objectives. Recent works explored extensions of this formulation using adaptive [9] and data-driven [10] approaches.

Relevant advances in the computational capabilities of modern computers and improvements in the algorithms efficiency [11]–[13] have made it possible to manage the high-computational load and stringent real-time requirements to solve these problems. Several software frameworks [14]–[16] have been released over the years to facilitate modeling, control design and simulation for a broad class of MPC applications.

In this context, various works have investigated NMPC strategies considering perception-objectives [8], [17], [18]. A common problem studied in those works is the tracking of a target subject to certain visibility constraints. In most cases, NMPC is used in the outer loop of a cascaded architecture to provide a reference trajectory to an inner loop pose tracking controller [17]. This approach allows to attain the perception-objectives, but it can cause problems as the NMPC generator does not consider the limitations posed by the low-level controller [8], [18]. As a consequence, the generated trajectory could violate the actuation limitations of the UAV resulting in an unfeasible solution.

To overcome this limitation, NMPC can be used to combine trajectory generation, subject to visibility and collision avoidance constraints, and trajectory tracking, subject to actuation constraints, in a single optimization problem, as shown in [5], [7], [19]. This allows to account for partially target occlusions that may appear as the camera moves, and physical constraints, such as feasibility and collision avoidance constraints, that could jeopardize the mission accomplishment. These approaches leverage on the intrinsic capability of the optimization framework to include inequality and equality constraints of various semantics. Therefore, the so-formulated problem allows to keep tracking of the target feature, preventing critical configurations related to Field of View (FoV) constraints, while taking into account the limits imposed by actuators.

Such single control layer architectures with perception and actuation constraints have been investigated also in [5], [6]. In these works, the authors make use of generic dynamic models which can represent a large class of multi-rotor UAVs [18]. This allows to formulate the problem for both standard coplanar under-actuated vehicles as well as for fully-actuated platforms with tilted propellers characterized by a fixed geometry (i.e., propellers' orientation is fixed). However, collision avoidance tasks with respect to (w.r.t.)

This work was partially funded by the European Union's Horizon 2020 research and innovation programme AERIAL-CORE under grant agreement no. 871479, by Czech Science Foundation (GAČR) under research project no. 20-10280S, by CTU grant no. SGS20/174/OHK3/3T/13, and by OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics".

[1]Andriy Dmytruk, Giuseppe Silano, Daniel Bonilla Licea, and Martin Saska are with the Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic, (email: {dmytrand, silangiu, bonildan, saskam1}@fel.cvut.cz).

[2]Davide Bicego is with the Robotics and Mechatronics group, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, Netherlands (email: d.bicego@utwente.nl).

multiple dynamic obstacles are not explicitly taken into account.

Following this line of research, a NMPC architecture for vision-driven target tracking and collision avoidance that considers both visual perception constraints and physical actuation limitations of a broad class of multi-rotor platforms is proposed. In particular, the problem where a Generically Tilted Multi-Rotor (GTMR) [20], [21] equipped with a visual sensor is required to track the trajectory of a moving target while keeping it in the camera FoV is considered. Meanwhile, the GTMR is required to safely navigate in a workspace area populated by dynamic obstacles characterized by a ballistic motion. The initial position of the obstacles and their motion are assumed to be known in the whole prediction horizon of the NMPC. Such a scenario aims at simulating the motion of balls thrown by a person[1]. The visual perception objectives are implemented as hard constraints, while the obstacle avoidance is enforced by soft constraints. This approach compensates for the lack of knowledge on the target motion by exploiting those of the obstacles to relax the optimization problem. Soft constraints ensure continuity of the solution when constraints may arise unfeasibility issues.

The paper is organized as follows. Section II describes the model of a GTMR and a generic visual sensor considered equipping the vehicle. Section III presents the optimal control problem formulation, including the equality and inequality constraints for the collision avoidance and visual-target tracking. Section IV reports the simulations results achieved in MATLAB, which are used to demonstrate the validity of the proposed approach. Finally, Section V concludes the paper.

II. System Modeling

A. System dynamics

Let us consider a GTMR model [21], composed of a rigid body and $n \in \mathbb{R}_{>0}$ propellers that spin about a generically oriented axis with fixed but arbitrary orientations. The relative propeller orientation, jointly with the number n of rotors, determine whether the GTMR is an under-actuated or a fully-actuated platform [20]. A schematic representation of the system is reported in Fig. 1.

Let us denote with $\mathcal{F}_W$ and $\mathcal{F}_B$ the *world frame* and *body frame* reference systems, respectively. The body frame is attached to the GTMR so that the origin of the frame O_B coincides with the Center of Mass (CoM) of the vehicle. The position of the origin O_B of the body frame $\mathcal{F}_B$ w.r.t. the world frame $\mathcal{F}_W$ is denoted with $\mathbf{p} \in \mathbb{R}^3$, while the unit quaternion representing the rotation from the body frame $\mathcal{F}_B$ to the world frame $\mathcal{F}_W$ is denoted as $\mathbf{q} \in \mathbb{S}^3$. The angular velocity of the GTMR in $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$, expressed in $\mathcal{F}_B$, is denoted with $\boldsymbol{\omega} \in \mathbb{R}^3$, whereas the linear velocity of O_B in $\mathcal{F}_W$ is denoted by $\mathbf{v} = \dot{\mathbf{p}} \in \mathbb{R}^3$.

The i-th propeller spinning velocity $\Omega_i \in \mathbb{R}_{\geq 0}$, with $i = \{1, 2, \dots, n\}$, represents the controllable input variable of the system, i.e., $\mathbf{u} = \boldsymbol{\Omega}$ and $\boldsymbol{\Omega} = [\Omega_1 \cdots \Omega_n]^\top \in \mathbb{R}^n_{\geq 0}$.

[1] https://youtu.be/w2itwFJCgFQ?t=531

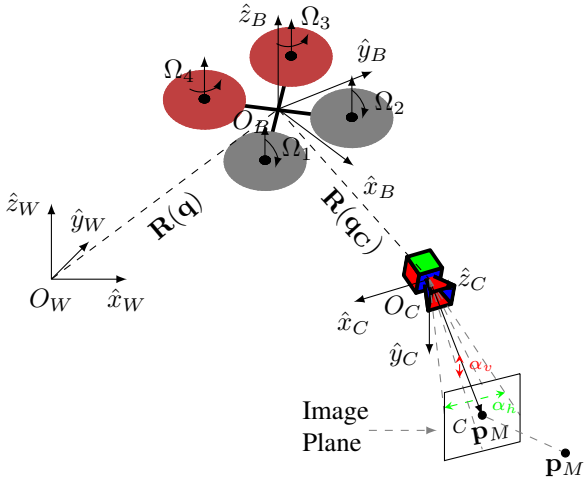


Fig. 1: A schematic representation of a GTMR model equipped with a generic visual sensor in the case of four propellers ($n = 4$) and coplanar orientation.

While rotating, each propeller exerts a thrust force $\mathbf{f}_i \in \mathbb{R}^3$ oriented along the axis perpendicular to the plane spanned by the propeller. Following the right-hand convention, this force generates a drag momentum $\boldsymbol{\tau}_i \in \mathbb{R}^3$ oriented as the angular velocity vector $\boldsymbol{\omega}$ in case of clockwise rotation, and opposite in case of counter-clockwise rotation. The sum of all forces $\mathbf{f}_i$ coincides with the control force $\mathbf{f}_c \in \mathbb{R}^3$ exerted at the platform CoM, while the control momentum $\boldsymbol{\tau}_c \in \mathbb{R}^3$ is the sum of the momentum contributions $\boldsymbol{\tau}_i$ due to both the thrust forces and the drag momenta. Further details on the model derivation are available in [21].

After neglecting second order effects and using the Newton-Euler approach, the GTMR dynamics can be approximated by the set of equations

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \\ m\dot{\mathbf{v}} = -mg\mathbf{e}_3 + \mathbf{R}(\mathbf{q})\mathbf{F}\mathbf{u} \\ \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}\mathbf{u} \end{cases}, \tag{1}$$

where $m \in \mathbb{R}_{>0}$ and $g \in \mathbb{R}_{>0}$ denote the GTMR mass and the gravitational constant, respectively, $\mathbf{e}_m \in \mathbb{R}^3$, with $m \in \{1, 2, 3\}$, represents the m-th column of the identity matrix $\mathbf{I}_3 \in \mathbb{R}^{3\times 3}$, and $\mathbf{J} \in \mathbb{R}^{3\times 3}$ is the positive definite constant GTMR inertia matrix in $\mathcal{F}_B$. The symbols $\circ$ and $\times$ denote the quaternion product and the vector cross product operations, respectively. $\mathbf{R}$ is the rotation matrix from the body $\mathcal{F}_B$ to the world frame $\mathcal{F}_W$. Finally, $\mathbf{F} \in \mathbb{R}^{3\times n}$ and $\mathbf{M} \in \mathbb{R}^{3\times n}$ are the forces and momenta allocation matrices, respectively, mapping the vector of forces produced by each i-th propeller to the total force and momentum acting on the vehicle CoM [21]. The model (1) describes a nonlinear dynamic system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, with state $\mathbf{x} = [\mathbf{p}^\top \mathbf{q}^\top \mathbf{v}^\top \boldsymbol{\omega}^\top]^\top \in \mathbb{R}^3 \times \mathbb{S}^3 \times \mathbb{R}^6$ and control input $\mathbf{u} = \boldsymbol{\Omega}$.

To account for the limited bandwidth of the control action variable $\mathbf{u}$ (i.e., maximum derivative of the propeller spinning

velocity value), smoothness has to be enforced. In this regard, it is convenient to extend model (1) assuming as new control input variable $\bar{\mathbf{u}} = \dot{\mathbf{u}}$, i.e., the time derivative of the propeller spinning velocity $\dot{\boldsymbol{\Omega}} \in \mathbb{R}^n$, including $\mathbf{u}$ among the system state variables $\mathbf{x}$ [18]. Therefore, the system model (1) can be rewritten as $\dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$, where $\bar{\mathbf{x}} = [\mathbf{p}^\top \mathbf{q}^\top \mathbf{v}^\top \boldsymbol{\omega}^\top \mathbf{u}^\top]^\top$ and $\bar{\mathbf{u}} = \dot{\mathbf{u}}$. In this way, realistic physical limitations on the actuators, i.e., lower and upper bounds on the rotor accelerations, which lie at the same differential level of the motor torques, can be considered within the system modeling (1) and in the control problem formulation (Section III). Thus, following the identification process in [18], the actuator bounds can be expressed as

$$\underline{\boldsymbol{\gamma}} \leq \mathbf{u} \leq \bar{\boldsymbol{\gamma}}, \tag{2a}$$

$$\underline{\dot{\boldsymbol{\gamma}}} \leq \bar{\mathbf{u}} \leq \bar{\dot{\boldsymbol{\gamma}}}, \tag{2b}$$

where $\underline{\boldsymbol{\gamma}}$ and $\bar{\boldsymbol{\gamma}}$ represent the minimum and maximum propeller spinning velocity values, respectively, while $\underline{\dot{\boldsymbol{\gamma}}}$ and $\bar{\dot{\boldsymbol{\gamma}}}$ are the minimum and maximum propeller spinning acceleration values, respectively, which can depend on the rotor velocity.

B. Generic sensor model

For the sake of generality of the proposed control strategy, the GTMR is considered to be equipped with a generic visual sensor able to retrieve the position of the target on the camera image plane. Such an assumption allows synthesizing an optimal control problem independent from the type of sensor used to detect the target. The visual sensor reference frame is denoted as $\mathcal{F}_C$ with optical axis $\hat{z}_C$, as depicted in Fig. 1. Also, the visual sensor is assumed to be rigidly attached to the GTMR body and its pose ($\mathbf{p}_C$, $\mathbf{q}_C$) to be known. In other words, the position ($\mathbf{p}_C$) and orientation ($\mathbf{q}_C$) of the origin O_C of the camera frame $\mathcal{F}_C$ w.r.t. the body frame $\mathcal{F}_B$ are assumed to be fixed and known. Hence, the transformation between the visual sensor reference frame $\mathcal{F}_C$ and the GTMR body frame $\mathcal{F}_B$ is also known. Besides, a *pinhole camera model* is used to describe the mathematical relationship between the position of the target $\mathbf{p}_M$ in $\mathcal{F}_W$ and its projection ${}^C\mathbf{p}_M$ onto the image plane $\mathcal{F}_C$ while neglecting possible distortions on the image produced by lenses [7]. Therefore, given a generic point $\mathbf{p}_M = [x_M \, y_M \, z_M]^\top$ in the world frame $\mathcal{F}_W$, its projection onto the camera frame $\mathcal{F}_C$ is denoted as ${}^C\mathbf{p}_M = [{}^Cx_M \; {}^Cy_M \; {}^Cz_M]^\top$. Under the assumptions of a pyramidal FoV (see Fig. 1), the constraints to enforce the visibility coverage of the target position $\mathbf{p}_M$ can be defined as

$$|{}^Cx_M / {}^Cy_M| \leq \tan\alpha_h, \tag{3a}$$

$$|{}^Cy_M / {}^Cz_M| \leq \tan\alpha_v, \tag{3b}$$

where α_h and α_v denote the horizontal and the vertical angles of the pyramidal FoV, respectively, as depicted in Fig. 1.

III. Optimal Control Problem Formulation

A GTMR equipped with a generic visual sensor has to track the motion of a target, whose motion is unknown, and maintain it in the camera image plane. Meanwhile, the multi-rotor is required to avoid dynamic obstacles populating the workspace. The proposed control setup has to comply with the platform physical limitations and vision constraints, while fulfilling the assigned mission at the best. The proposed framework is stated for one sensor and a single target feature. However, such an assumption does not preclude to extend the framework to consider more targets as in [5]. The motion of the target is such as to ensure that the GTMR can track it.

The following sections deal with describing the collision constraints applied to the system, the objective function, and the NMPC optimal formulation.

A. Collision avoidance

As stated in Sec. III, the GTMR is required to maintain the target within the camera FoV while avoiding obstacles in the workspace area. The obstacles' motion is assumed to be known in the whole prediction horizon of the NMPC. Besides, a point-mass model is used to approximate the GTMR body and the obstacle sizes leaving out the actual sizes of the robot and those of the obstacles from the collision avoidance requirements.

Let us denote with $\mathbf{p}_{\mathrm{o}_j} = [x_{\mathrm{o}_j} \, y_{\mathrm{o}_j} \, z_{\mathrm{o}_j}]^\top$ the position of the j-th obstacle in the world frame $\mathcal{F}_W$, with $j \in \{1, 2, \ldots, O\}$ and $O \in \mathbb{N}_{>0}$ denoting the number of obstacles populating the workspace area. Therefore, the collision avoidance constraint is considered to be the square Euclidean distance defined as

$$\|\mathbf{p} - \mathbf{p}_{\mathrm{o}_j}\|^2 \geq \Gamma_j^2, \tag{4}$$

where $\Gamma_j \in \mathbb{R}_{>0}$ is the minimum distance value the multi-rotor has to maintain to avoid crashes with the obstacles.

Remark 1 *(Optimal Solvers)*: Note that for optimization solvers whose algorithm is based on a Newton-type method, such as ACADOS [13], PANOC [14], and CasADi [11], only one system linearization and Quadratic Programming (QP) problem are performed per time. Each QP problem corresponds to a linear approximation of the original Nonlinear Programming (NLP) problem along a time-varying trajectory. In this configuration, squaring the Euclidean norm, as in (4), provides local trajectories that are over-conservative. In a case where the linearization points are far away from the boundaries of the linearized feasible set of the original NLP, the use of (4) leads to linear constraints that are "least" conservative. Further details can be found in [22].

B. Objective function

A motion tracking problem is described as a minimization distance task between the vehicle state $\mathbf{x}$ and a reference motion $\mathbf{x}_{\mathrm{d}}$. Usually, a reference trajectory is expressed in both position and attitude, denoted as $(\mathbf{p}_{\mathrm{d}}, \mathbf{q}_{\mathrm{d}})$, and the corresponding first $(\mathbf{v}_{\mathrm{d}}, \boldsymbol{\omega}_{\mathrm{d}})$ and second $(\dot{\mathbf{v}}_{\mathrm{d}}, \dot{\boldsymbol{\omega}}_{\mathrm{d}})$ order time derivatives. These signals are sampled over the prediction horizon of the optimization problem and retrieved as output of a trajectory planner. Examples of these schemes can be found in [5], [7], [19].

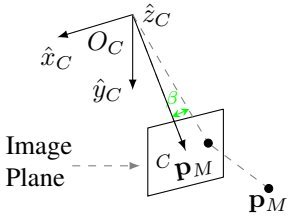


Fig. 2: A schematic representation of the angular position β of an image point ${}^C\mathbf{p}_M$.

The trajectory tracking error is defined as a weighted square Euclidean norm denoted as $\|\cdot\|^2_{\mathbf{Q}}$, where $\mathbf{Q} \in \mathbb{R}_{\geq 0}$ is a diagonal weight matrix used as a tunable controller gain. Note that the Euclidean norm cannot be used to model the differences between the reference attitude $\mathbf{q}_\mathrm{d}$ and the drone attitude $\mathbf{q}$ due to the disambiguities introduced by the unit quaternion representation. Indeed, the $\mathbf{q}$ and $-\mathbf{q}$ unit quaternions represent the same attitude.

To overcome this issue, the geodesic distance described in [6] was considered to model the difference between quaterions. Hence, given the unit quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$, their distance can be calculated as $\|\log(\mathbf{q}_1 \circ \mathbf{q}_2^\star)\|$, where $\mathbf{q}_\bullet^\star$ represents the conjugate of the unit quaternion $\mathbf{q}_\bullet$. Note that, in order to improve the paper readability, with an abuse of notation the weighted Euclidean norm $\|\mathbf{q} - \mathbf{q}_\mathrm{d}\|^2_{\mathbf{Q}}$ is used to denote the weighted attitude error between the unit quaternions describing the drone ($\mathbf{q}$) and reference attitudes ($\mathbf{q}_\mathrm{d}$), instead of referring to the weighted geodesic distance $\|\log(\mathbf{q}_1 \circ \mathbf{q}_2^\star)\|^2_{\mathbf{Q}}$. Such a representation allows to compact the notation of the optimal control problem (Section III-C).

Along with tracking errors, perception errors are also embedded into the objective function of the optimal control problem to maintain the target in the camera FoV. Following [19], the angular distance β between the target position on the image plane (${}^C\mathbf{p}_M$) and the z-axis of the camera frame $\mathcal{F}_C$ is minimized. In addition, to avoid motion blur and smoothly guide the target tracking, the time derivative $\dot{\beta}$ is also accounted in the optimal problem formulation. Figure 2 shows a schematic representation of the perception objectives.

To make the problem resolution more efficient, the difference in angle and its derivative are replaced with the cosine function, i.e., $c\beta$ and $c\dot{\beta}$, where $c\bullet$ refers to the short notation for the cosine function. Besides, a weighted distance term $\|\mathrm{dist}(\mathbf{p}, \mathbf{p}_M) - \Upsilon\|^2_{\mathbf{Q}}$ enforces a constant distance $\Upsilon \in \mathbb{R}_{>0}$ between the drone ($\mathbf{p}$) and the target ($\mathbf{p}_M$) positions. The $\mathrm{dist}(\bullet, \bullet)$ function refers to the Euclidean distance function between two points.

Therefore, the objective function can be written as the weighted difference between the desired $\mathbf{y}_\mathrm{d}$ and the current $\mathbf{y} = \mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \mathbf{p}_M)$ output signals, i.e., $\|\mathbf{y} - \mathbf{y}_\mathrm{d}\|_{\mathbf{Q}}$, as

$$\mathbf{y}_\mathrm{d} = [\mathbf{p}_\mathrm{d}^\top \ \mathbf{q}_\mathrm{d}^\top \ \mathbf{v}_\mathrm{d}^\top \ \boldsymbol{\omega}_\mathrm{d}^\top \ \dot{\mathbf{v}}_\mathrm{d}^\top \ \dot{\boldsymbol{\omega}}_\mathrm{d}^\top 1 \ 0 \ \Upsilon]^\top, \tag{5a}$$

$$\mathbf{y} = [\mathbf{p}^\top \ \mathbf{q}^\top \ \mathbf{v}^\top \ \boldsymbol{\omega}^\top \ \dot{\mathbf{v}}^\top \ \dot{\boldsymbol{\omega}}^\top \mathrm{c}\beta \ \ \mathrm{c}\dot{\beta} \ \ \mathrm{dist}(\mathbf{p}, \mathbf{p}_M)]^\top. \tag{5b}$$

C. Optimal control problem

Thus, the optimal control problem over a prediction horizon of N steps, where $N \in \mathbb{N}_{>0}$, considering the revised system model equations $\dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$, can be formulated as a minimization problem per each time step $\mathbf{t}_k = kTs$, with T_s being the sampling time and $k \in \mathbb{N}_{>0}$, as follows

$$\underset{\bar{\mathbf{x}},\, \bar{\mathbf{u}},\, \mathbf{s}}{\text{minimize}} \ \sum_{k=0}^{N} \|\mathbf{y}_{\mathrm{d},k} - \mathbf{y}_k\|^2_{\mathbf{Q}_1} + \|\mathbf{s}_{j,k}\|^2_{\mathbf{Q}_2} \tag{6a}$$

$$\text{s.t.} \ \ \bar{\mathbf{x}}_0 = \bar{\mathbf{x}}(\mathbf{t}_k), k = 0, \tag{6b}$$

$$\bar{\mathbf{x}}_{k+1} = \mathbf{f}(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k), k \in \{0, N-1\}, \tag{6c}$$

$$\mathbf{y}_k = \mathbf{h}(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k, \mathbf{p}_{M_k}), k \in \{0, N\}, \tag{6d}$$

$$\underline{\gamma} \leq \mathbf{u}_k \leq \bar{\gamma}, k \in \{0, N\}, \tag{6e}$$

$$\dot{\underline{\gamma}} \leq \bar{\mathbf{u}}_k \leq \dot{\bar{\gamma}}, k \in \{0, N-1\}, \tag{6f}$$

$$|{}^C x_{M_k} / {}^C z_{M_k}| \leq \tan \alpha_h, k \in \{0, N\}, \tag{6g}$$

$$|{}^C y_{M_k} / {}^C y_{M_k}| \leq \tan \alpha_v, k \in \{0, N\}, \tag{6h}$$

$$\|\mathbf{p}_k - \mathbf{p}_{\mathrm{o}_j,k}\|^2 + \mathbf{s}^2_{j,k} \geq \Gamma^2_j, \mathbf{s}_{j,k} > 0, \tag{6i}$$

$$k \in \{0, N\}, j \in \{1, O\},$$

where (6a) is the objective function, (6b) sets the initial state conditions, (6c) and (6d) express the discretized dynamic model for the GTMR and the output signals of the system, respectively, while actuator limits are embedded in (6e) and (6f). The hard constraints (6g) and (6h) ensure that the target remains in the camera FoV during the whole trajectory tracking problem. The soft constraint (6i) prevents the drone from colliding with the j-th obstacle by enforcing the vehicle to maintain a safety distance Γ_j. The slack variables $\mathbf{s} = [s_1 \cdots s_O]^\top \in \mathbb{R}^O$, with $\mathbf{s}_j$ denoting the j-th element of the vector $\mathbf{s}$, add a maneuverability room margin to the optimization problem by ensuring continuity of the solution when constraints may arise unfeasibility issues. Finally, the vectors $\bar{\mathbf{u}}_k$, $\bar{\mathbf{x}}_k$, $\mathbf{s}_{j,k}$, $\mathbf{y}_{\mathrm{d},k}$ and $\mathbf{y}_k$ denote the k-th element of vectors $\bar{\mathbf{u}}$, $\bar{\mathbf{x}}$, $\mathbf{s}_j$, $\mathbf{y}_\mathrm{d}$ and $\mathbf{y}$, respectively.

Similarly to [22], [23], this approach allows relaxing the constraint on the obstacle avoidance to guarantee feasible solutions in tight situations by introducing a penalty term in the objective function (6a), i.e., $\|\mathbf{s}_j\|^2_{\mathbf{Q}_2}$. Specifically, when the GTMR gets too close to the obstacles, violating the safety distance constraint (6i), then the slack variable $\mathbf{s}_j$ takes a positive value to satisfy the inequality (6i), but this also adds a penalization term in the cost function (6a).

Remark 2 *(Soft Constraints)*: Even though soft constraints are designed to be possibly violated, such as in (6i), this situation should be minimized. This can be done by properly tuning the weighted matrix $\mathbf{Q}_2$. This approach allows flexible obstacle definition, guaranteeing feasible solutions.

IV. Simulation Results

In this section we provide MATLAB simulation results to validate the proposed control strategy. Specifically, the optimal control problem was coded using the MATMPC framework [16], with a 4^{th} fixed step Runge-Kutta integrator and sampling time $T_s = 15\,\mathrm{ms}$, and qpOASES [12] as solver. All simulations were performed on a laptop with a i7-8565U

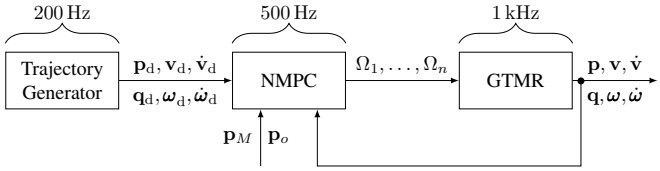


Fig. 3: Block diagram of the proposed optimal control strategy.

Sym.	Value	Sym.	Value	Sym.	Value	Sym.	Value
Q_p	1	Q_q	1	Q_v	0.1	Q_ω	0.1
$Q_{\dot{v}}$	0.01	$Q_{\dot{\omega}}$	0.01	$Q_{c\beta}$	100	$Q_{\dot{c}\beta}$	100
Q_{d_trg}	10	Q_s	1×10^4	$\bar{\gamma}$	40 Hz	$\underline{\gamma}$	90 Hz
$\dot{\bar{\gamma}}$	$200\,\mathrm{Hz\,s^{-1}}$	$\dot{\underline{\gamma}}$	$-110\,\mathrm{Hz\,s^{-1}}$	m	1.042 kg	g	$9.84\,\mathrm{m\,s^{-2}}$
$\mathbf{J}_1$	0.015	$\mathbf{J}_2$	0.015	$\mathbf{J}_3$	0.070	Γ	1 m
α_h	$\pi/2$	α_v	$\pi/2$	Υ	1 m	-	-

TABLE I: List of parameters and their values. Q_k refers to the k-th element of the weighted diagonal matrix $\mathbf{Q}_1$ and $\mathbf{Q}_2$ w.r.t. the output map $\mathbf{y}$ and the slack variables $\mathbf{s}$, respectively.

processor (1.80 GHz) and 32GB of RAM running on Ubuntu 20.04. Animation videos with the results of the numerical simulations are available at `http://mrs.felk.cvut.cz/perception-aware-nmpc`.

The control architecture is shown in Fig. 3. A reference generator working at 200 Hz provides the reference trajectory $(\mathbf{p}_\mathrm{d}^\top, \mathbf{q}_\mathrm{d}^\top, \mathbf{v}_\mathrm{d}^\top, \boldsymbol{\omega}_\mathrm{d}^\top, \dot{\mathbf{v}}_\mathrm{d}^\top, \dot{\boldsymbol{\omega}}_\mathrm{d}^\top)$ to the NMPC which runs at 500 Hz. The output of the NMPC are the propeller spinning velocities $\boldsymbol{\Omega}$ supplied to the aerial vehicle to control its motion. The high frequency at which the control algorithm works simulates the times required in real applications to control the vehicle dynamics. The optimal control strategy (Section III-C) runs considering $N = 50$ shooting points and a prediction horizon of 0.75 s, while the system dynamics (1) are integrated with a sampling time of 1 ms Table I reports the NMPC gains along with the GTMR's parameters values. The parameters provide a balanced trade-off between efficiency, safety and tracking.

The trajectory tracking scenario consists of a target following an ascending ramp trajectory at a constant velocity of $|\mathbf{v}_\mathrm{d}| = 1\,\mathrm{m\,s^{-1}}$ for 10 s. A single integrator model was used to characterize the motion of the target. This model is a canonical example of a first order control system. As described in [24], demonstrating numerical convergence with this model guarantees convergence even with more complex models, as long as the control law is suitably adapted to the various motion constraints.

The GTMR is equipped with $n = 4$ propellers arranged parallel to the xy-plane of the body frame $\mathcal{F}_B$ and having the same direction of its z-axis (i.e., a coplanar under-actuated platform). The initial drone and target positions in the world frame $\mathcal{F}_W$ are $\mathbf{p} = [0, 0, 0]^\top$ and $\mathbf{p}_M = [6, 6, 0]^\top$, respectively. Two dynamic obstacles have been considered populating the workspace area with initial position $\mathbf{p}_{\mathrm{o}_1} = [2, 6, 0]^\top$ and $\mathbf{p}_{\mathrm{o}_2} = [10, 6, 2]^\top$. The origin of the visual generic sensor O_C lies on the vehicle body frame $\mathcal{F}_B$ and is positive translated 0.1 m along its x-axis ($\hat{x}_B$). The

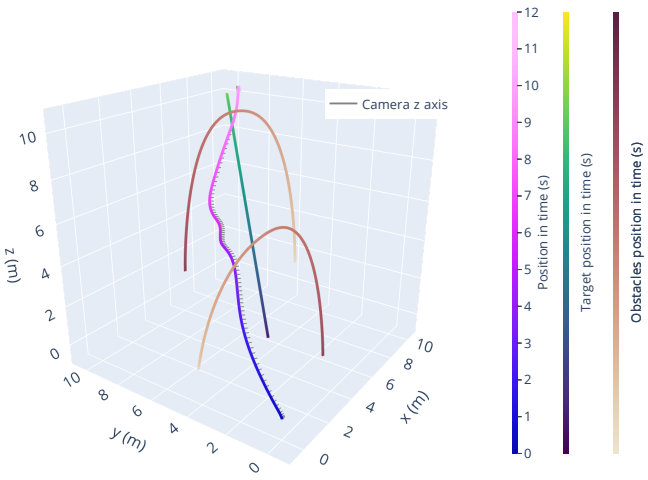


Fig. 4: The vehicle motion in the trajectory tracking and collision avoidance scenario. Gradient colors show the drone, target and obstacles motion during time. A short segment represents the z-axis orientation of the camera ($\hat{z}_c$).

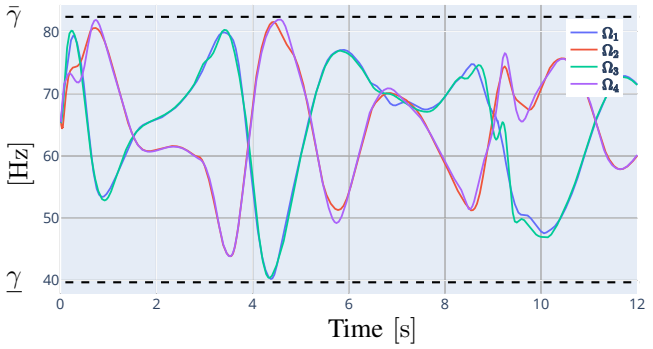
overall scenario is depicted in Fig. 4 along with the vehicle, obstacles, and target motion during time.

Figs 5a and 5b show the evolution of the propeller spinning velocities $\boldsymbol{\Omega}$ and their variation $\dot{\boldsymbol{\Omega}}$ during time. As it can be seen from the figures, both values remain within the boundaries ($\underline{\gamma}$, $\bar{\gamma}$, $\dot{\underline{\gamma}}$, $\dot{\bar{\gamma}}$) as required by the constraints (6e) and (6f). Figure 6 shows the distances between the target ($\mathbf{p}_M$) and the obstacles ($\mathbf{p}_{\mathrm{o}_1}$ and $\mathbf{p}_{\mathrm{o}_2}$) w.r.t. the drone's position ($\mathbf{p}$). As it can be seen from the graph, the collision avoidance avoidance constraints (6i) is always satisfied for both for the obstacles and the target (6a).

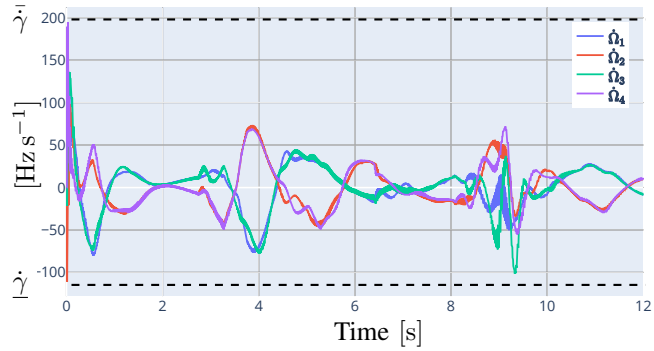
Remark 3 *(Penalties on Soft Constraints)*: It is worth noticing that, the effort required to tune the penalty values $\mathbf{Q}_2$ on the slack variables $\mathbf{s}$ directly affects the amount of constraints violation. Roughly speaking, imposing very strict penalties increases the difficulty in driving the solution towards the optimum while distancing the vehicle from the lower bound. Conversely, if the penalty is not strict enough, then the search will tend to stall outside the feasible region and thereby violate the constraint. Hence, such an approach does not guarantee that the violation will not occur.

V. Conclusions

In this paper, a perception-aware NMPC strategy for vision-based target tracking and obstacle avoidance have been proposed. In particular, an optimization problem was set to control the behavior of a multi-rotor aerial vehicle enforcing the visibility coverage of a target while accounting for real actuation limits and obstacles avoidance. A full nonlinear generic constrained model has been considered for the flight control system design covering both coplanar under-actuated platforms and fully-actuated tilted propellers. Numerical simulations achieved in MATLAB have demonstrated the feasibility of the proposed control approach, aiming towards the fulfillment of real-word tests. Future work



(a) Propeller angular velocities Ω_i, with $i = \{1, 2, 3, 4\}$.



(b) Control input variables $\bar{\mathbf{u}} = [\dot{\Omega}_1\ \dot{\Omega}_2\ \dot{\Omega}_3\ \dot{\Omega}_4]^\top$.

Fig. 5: Propeller spinning velocities and their variation during time along with the corresponding bounds $\bar{\gamma}$, $\underline{\gamma}$, $\dot{\bar{\gamma}}$, and $\dot{\underline{\gamma}}$.

includes relaxing the assumptions on the obstacle motion over the prediction horizon of the optimization problem. In addition, more challenging scenarios will be investigated, such the combination of static and dynamic obstacles, in the direction of field experiments.

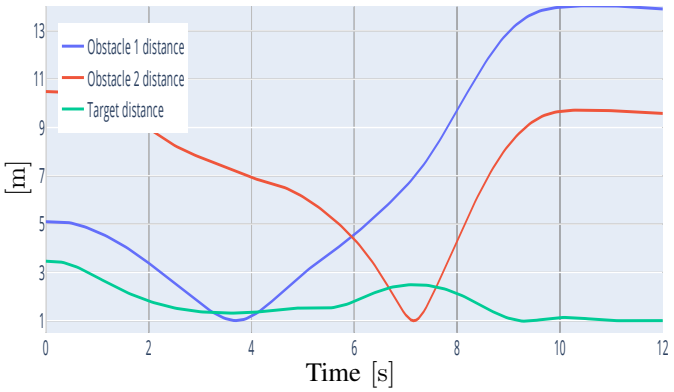


Fig. 6: Distances between the GTMR and the obstacles (in blue and red) and the GTMR and the target (in green) during time.

References

[1] G. Silano *et al.*, "Power Line Inspection Tasks With Multi-Aerial Robot Systems Via Signal Temporal Logic Specifications," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4169–4176, 2021.

[2] V. Krátký *et al.*, "Autonomous Aerial Filming With Distributed Lighting by a Team of Unmanned Aerial Vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7580–7587, 2021.

[3] P. Petráček *et al.*, "Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7596–7603, 2021.

[4] D. Bonilla Licea *et al.*, "Optimum Trajectory Planning for Multi-Rotor UAV Relays with Tilt and Antenna Orientation Variations," in *29th European Signal Processing Conference*, 2021, pp. 1586–1590.

[5] M. Jacquet *et al.*, "Motor-Level N-MPC for Cooperative Active Perception With Multiple Heterogeneous UAVs," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2063–2070, 2022.

[6] ——, "Motor and Perception Constrained NMPC for Torque-Controlled Generic Aerial Vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 518–525, 2021.

[7] D. Falanga *et al.*, "PAMPC: Perception-Aware Model Predictive Control for Quadrotors," in *IEEE International Conference on Intelligent Robots and Systems*, 2018, pp. 1–8.

[8] M. Kamel *et al.*, "Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles," in *IEEE International Conference on Robotics and Automation*, vol. 50, no. 1, 2017, pp. 3463–3469.

[9] D. Hanover *et al.*, "Performance, Precision, and Payloads: Adaptive Nonlinear MPC for Quadrotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 690–697, 2022.

[10] G. Torrente *et al.*, "Data-Driven MPC for Quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.

[11] J. A. E. Andersoon *et al.*, "CasADi - A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[12] H. Ferreau *et al.*, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[13] R. Verschueren *et al.*, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–37, 2021.

[14] A. Sathya *et al.*, "Embedded nonlinear model predictive control for obstacle avoidance using PANOC," in *17th European Control Conference*, 2018, pp. 1523–1528.

[15] S. Pantelis *et al.*, "OpEn: Code Generation for Embedded Nonconvex Optimization," in *IFAC-PapersOnLine*, vol. 53, no. 2, 2020, pp. 6548–6554.

[16] Y. Chen *et al.*, "MATMPC - A MATLAB Based Toolbox for Real-time Nonlinear Model Predictive Control," in *18th European Control Conference*, 2019, pp. 3365–3370.

[17] J. L. Paneque *et al.*, "Perception-Aware Perching on Powerlines With Multirotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3077–3084, 2022.

[18] D. Bicego *et al.*, "Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3, pp. 1213–1247, 2020.

[19] B. Penin *et al.*, "Vision-Based Reactive Planning for Aggressive Target Tracking While Avoiding Collisions and Occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.

[20] M. Ryll *et al.*, "6D interaction control with aerial robots: The flying end-effector paradigm," *International Journal of Robotics Research*, vol. 38, no. 9, pp. 1045–1062, 2019.

[21] G. Michieletto *et al.*, "Fundamental Actuation Properties of Multirotors: Force–Moment Decoupling and Fail–Safe Robustness," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 702–715, 2018.

[22] B. Barros Carlos *et al.*, "Least Conservative Linearized Constraint Formulation for Real-Time Motion Generation," in *IFAC-PapersOnLine*, vol. 53, no. 2, 2020, pp. 9384–9390.

[23] M. Castillo-Lopez *et al.*, "Model Predictive Control for Aerial Collision Avoidance in Dynamic Environments," in *26th Mediterranean Conference on Control and Automation*, 2018, pp. 198–203.

[24] S. Zhao *et al.*, "Defend the practicality of single-integrator models in multi-robot coordination control," in *13th IEEE International Conference on Control Automation*, 2017, pp. 666–671.