Vision-driven NMPC for Autonomous Aerial Navigation in Subterranean Environments [⋆]

Christoforos Kanellakis [*] **Petros Karvelis** [*]
Sina Sharif Mansouri [*] **Ali-akbar Agha-mohammadi** [**]
George Nikolakopoulos [*]

[*] *The authors are with the Robotics Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden (e-mails:{chrkan,petkar,geonik}@ltu.se).*
[**] *The author is with Jet Propulsion Laboratory California Institute of Technology Pasadena, CA, 91109. (e-mail: aliagha@jpl.nasa.gov*

Abstract:
This work establishes a novel robocentric Non-linear Model Predictive Control (NMPC) framework for fast fully autonomous navigation of quadrotors in featureless dark tunnel environments. Additionally, this work leverages the processing of a single camera to generate direction commands along the tunnel axis, while regulating the platform's altitude. The extracted visual dynamics are coupled in the sequel with the NMPC problem, structured around the Proximal Averaged Newton-type method for Optimal Control (PANOC), which is a fast numerical optimization method that is not sensitive to ill conditioning and is suitable for embedded NMPC implementations. Multiple fully realistic simulation results demonstrate the effectiveness of the proposed method in challenging environments.

Keywords: Autonomous Vehicles, Robot Navigation, Non linear model predictive control, MAV

1. INTRODUCTION

The quest for autonomous MAVs that can reliably navigate in partially-known or unknown areas brings these platforms in the forefront of research and technological breakthroughs, while introducing novel approaches for several application areas. Infrastructure inspection, search and rescue, area coverage/surveillanceare fields that already pursue the incorporation of aerial vehicles in their operation cycles. These applications can be part of large scale outdoors environments (e.g. bridges, wind-turbines, power plants), urban environments (e.g. cities) and subterranean operating environments (e.g. tunnels and cave networks). This article approaches the problem of fast navigation in dark and featureless subterranean environments like tunnels, by coupling a Nonlinear Model Predictive Control (NMPC) architecture that plans the robot motion along a subterranean tunnel axis and regulates its altitude with a single image frame processing architecture.

Several works in the existing literature have addressed the control and navigation of MAVs in challenging environments using various sensor configurations. In Özaslan et al. (2017) the fields of estimation, control and mapping for the MAV's autonomous navigation along penstocks, have been studied. In Tan et al. (2019) a range based sensor array approach has been developed to navigate along right-rectangular tunnels and cylindrical shafts. In Mascarich et al. (2018), the authors presented a multi-modal sensor unit for mapping applications, as a means for aerial robots to navigate in dark underground tunnels. In Falanga et al. (2018), a perception-aware Model Predictive Control (MPC) algorithm has been proposed to compute trajectories for quadrotors that maximize the visibility of a desired target, optimizing both the action and perception objectives. In Mcfadyen et al. (2013) a collision avoidance scheme has been presented that guides an aerial vehicle around an object along a conical spiral trajectory using a spherical camera model in a visual predictive control. Other works on vision based control have approached the navigation from other aspects, such in Potena et al. (2017) where the authors proposed a method to guide a multirotor to a desired pose, while simultaneously keeping a target within the field of view of the onboard camera. In this work, a visual servoing scheme is used to generate a trajectory, based on the minimization of the target re-projection error, while a MPC scheme was developed to track the trajectory. In Sheckells et al. (2016) the authors presented a hybrid visual servoing scheme for deferentially flat systems formulated as an optimal control. The method initially computed the final pose of the vehicle using the desired camera view by solving a Perspective-n-Point problem and then used optimal control to compute a feasible trajectory, with a cost function to keep the image features in the view.

Based on the aforementioned state of the art, the aim of this article is to advance the current state by the following four novel contributions: a) establish a general control framework for fast navigation of MAVs in featureless tunnel-like subterranean environments b) incorporate

[⋆] This work has been funded by the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 730302 SIMS. Corresponding Author's email: chrkan@ltu.se

perception states in the MAV dynamics and coupling them with the MAV's states and control inputs, where the perception objectives are part of the cost function during the optimization. c) approach the platform navigation as a free-flying object in the velocity domain of x and y axis, while using single image depth-map estimation to align the heading towards the tunnel axis and regulate the platform height and d) provide extended simulation results to demonstrate the performance.

1.1 *Outline*

The rest of the article is structured as it follows. Section 2 describes the overall framework, the platform and the image dynamics, the proposed centroid extraction method and the proposed mathematical establishment of the novel vision based NMPC control formulation. Section 3 provides an overview of the performed extended realistic simulation results on different scenarios and finally, Section 4 presents the concluding remarks of the developed system.

2. FREE-FLYING MAV VISION-BASED CONTROL FRAMEWORK

This section describes the methodology of all relative components, from the control design to the visual processing scheme. Figure 1 showcases the proposed resulting concept of the method in two different tunnel environments with multiple turns and inclinations, while the MAV follows the vertical and horizontal axes along the tunnel. In the sequel, the proposed free-flying MAV scheme will be established.

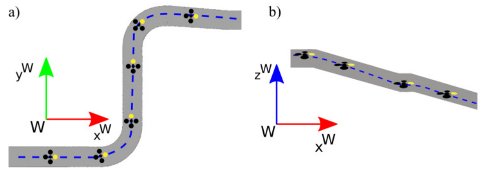

Fig. 1. Tunnel views showing the MAV following the tunnel axis and a corresponding altitude descend, where a) depicts xy view of the tunnel axis and b) depicts xz view of the tunnel altitude descend.

2.1 *Preliminaries*

The world frame $\mathcal{W}$ is fixed with the unit vectors $\{x^{\mathcal{W}}, y^{\mathcal{W}}, z^{\mathcal{W}}\}$ following the North-West-Up (NWU) frame convention. The body frame of the aerial vehicle $\mathcal{B}$ is attached on its base with the unit vectors $\{x^{\mathcal{B}}, y^{\mathcal{B}}, z^{\mathcal{B}}\}$. The $z^{\mathcal{B}}$ is antiparallel to the gravity vector, $x^{\mathcal{B}}$ is looking forward the platform's base and $y^{\mathcal{B}}$ is in the NWU convention. The onboard camera frame $\mathcal{C}$ has unit vectors $\{x^{\mathcal{C}}, y^{\mathcal{C}}, z^{\mathcal{C}}\}$. Furthermore, $y^{\mathcal{C}}$ is parallel to the gravity vector and $z^{\mathcal{C}}$ points in front of the camera. Finally, the image plane is defined as $\mathcal{I}$ with unit vectors $[x^{\mathcal{I}}, y^{\mathcal{I}}]$. Figure 2 depicts an overview of the utilized main coordinate frames of the aerial platform.

2.2 *MAV Dynamics*

The first step for solving the model based control problem considers the description of the robot dynamics. In this work the proposed control scheme is employed alongside a low level flight controller and therefore the modelling does not focus on producing low level motor commands from the sensory input, but stays at a level higher. To this end, the quadrotor model derived in Kamel et al. (2017) is used, providing roll ϕ_{cmd}, pitch $\theta_{cmd} \in [-\pi/2, \pi/2]$ and thrust $T_{cmd} \in \mathbb{R}_+$ commands, that are later handled by a low-level controller mounted onboard the aerial vehicle. The dynamic model considers that the main forces acting on the vehicle are produced from the propellers, while it is enhanced by taking into account the induced drag, another critical effect that is evident while maneuvering, and it is introducing additional forces on the rotors. The drag force introduces an extra force on the xy rotor plane, while the following Equation (1) summarizes the quadrotor model.

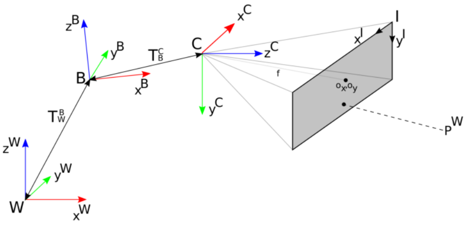

Fig. 2. Coordinate frames, where $^{\mathcal{W}}$, $^{\mathcal{B}}$, $^{\mathcal{C}}$ and $^{\mathcal{I}}$ denote world, body and camera and image coordinate frames respectively.

$$\begin{bmatrix} \dot{v}_x^{\mathcal{B}} \\ \dot{v}_y^{\mathcal{B}} \\ \dot{v}_z^{\mathcal{B}} \end{bmatrix} = R(\theta, \phi) \begin{bmatrix} 0 \\ 0 \\ T_d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} - \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} v_x^{\mathcal{B}} \\ v_y^{\mathcal{B}} \\ v_z^{\mathcal{B}} \end{bmatrix}$$
$$\dot{\phi} = \frac{1}{\tau_\phi}(K_\phi \phi_{cmd} - \phi) \quad (1)$$
$$\dot{\theta} = \frac{1}{\tau_\theta}(K_\theta \theta_{cmd} - \theta)$$

Where $v^{\mathcal{B}} = [v_x^{\mathcal{B}}, v_y^{\mathcal{B}}, v_z^{\mathcal{B}}]^\top \in \mathbb{R}^3$ represents the linear velocities on each axis, ϕ, $\theta \in \mathbb{R}$ are roll and pitch, $R \in SO(3)$ is the rotation matrix without a rotation around the z-axis, since the coordinates are in the body frame, $T_d \in \mathbb{R}$ is the mass normalized thrust, g is the gravitational acceleration, A_x, A_y, and A_z are the normalized mass drag coefficients, τ_ϕ and τ_θ are the time constants, and K_ϕ, K_θ are the roll and pitch angle gains, and finally ϕ_{cmd}, θ_{cmd} are the reference values of the roll and pitch angle for the low level controller. In this model, the drag force is expressed with the diagonal matrix, which associates the linear velocities with the force affecting the robot motion. Following the work presented in Small et al. (2019), the model incorporates the mass normalized thrust T_d that is converted to T_{cmd} using the derived adaptive acceleration control scheme. In this way, the controller is able to adapt to changes regarding the payload, the motor thrust and the thrust loss from the battery depletion over time.

2.3 *Image Dynamics*

Image Based Visual Servoing (IBVS) The visual processing part of the proposed control scheme operates in the 2D image plane, as discussed in Section 2.4.1, without any available metric in the 3D world frame. To follow the desired target in the image plane the coupling between the camera motion, attached on the MAV, and the target movement in the image plane should be studied. Therefore, for deriving proper control actions, the image dynamics should be coupled with the MAV's control input and/or states.

A 3D point $P^{\mathcal{W}} = [X_P^{\mathcal{W}}, Y_P^{\mathcal{W}}, Z_P^{\mathcal{W}}]^\top \in \mathbb{R}^3$ fixed in the world $\mathcal{W}$ is expressed in the camera frame $P^{\mathcal{C}} = [X_P^{\mathcal{C}}, Y_P^{\mathcal{C}}, Z_P^{\mathcal{C}}]^\top \in \mathbb{R}^3$, using the homogeneous transformation $\mathrm{T} : \mathbb{R}^4 \to \mathbb{R}^4$ through matrices $T_{\mathcal{W}}^{\mathcal{B}}$ and $T_{\mathcal{B}}^{\mathcal{C}} \in \mathbb{R}^{4\times4}$. The point is projected into the image plane $\mathcal{I}$ using a pinhole camera model Hartley and Zisserman (2003) and the perspective projection equation. The projection function $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$ results in the 2D point $p^{\mathcal{C}} = [u_p, v_p] \in \mathbb{R}^2$ can be expressed as it follows:

$$u_p = f_x \frac{X_P^{\mathcal{C}}}{Z_P^{\mathcal{C}}}, \quad v_p = f_y \frac{Y_P^{\mathcal{C}}}{Z_P^{\mathcal{C}}} \tag{2}$$

where $u_p,\ v_p \in \mathbb{R}$ are the horizontal and vertical pixel coordinates and $f_x,\ f_y \in \mathbb{R}$ are the camera focal length for the pixel columns and rows respectively. u_p and v_p are then discretized and quantized into pixel coordinates $u^{\mathcal{I}}$, $v^{\mathcal{I}} \in \mathbb{N}$. The normalized pixel coordinates s_x and $s_y \in \mathbb{R}$ can be defined as:

$$s_x = (u^{\mathcal{I}} - o_x)/f_x, \ s_y = (v^{\mathcal{I}} - o_y)/f_y, \tag{3}$$

where $o_x \in \mathbb{N}$ and $o_y \in \mathbb{N}$ are the principal points in pixels.

Following the IBVS theory Chaumette and Hutchinson (2006), calculating the derivative of (2) over time provides the matrix that connects the time derivative of a 2D point in the image plane with the rates of the camera in the 3D plane. More specifically, (4) describes the desired relation in the 2D target motion and the 3D camera motion, with the assumption that the target is static.

$$\begin{bmatrix} \dot{s}_x \\ \dot{s}_y \end{bmatrix} = L \begin{bmatrix} v_x^{\mathcal{C}} & v_y^{\mathcal{C}} & v_z^{\mathcal{C}} & \omega_x^{\mathcal{C}} & \omega_y^{\mathcal{C}} & \omega_z^{\mathcal{C}} \end{bmatrix}^\top \tag{4}$$

where $v^{\mathcal{C}} = [v_x^{\mathcal{C}}, v_y^{\mathcal{C}}, v_z^{\mathcal{C}}]^\top \in \mathbb{R}^3$ and $\omega^{\mathcal{C}} = [\omega_x^{\mathcal{C}}, \omega_y^{\mathcal{C}}, \omega_z^{\mathcal{C}}]^\top \in \mathbb{R}^3$ represent the linear and angular velocities in frame $\mathcal{C}$, while the $L \in \mathbb{R}^{2\times6}$ matrix that describes the coupling is called Interaction matrix and is defined as:

$$L = \begin{bmatrix} -\frac{1}{d(u^{\mathcal{I}},v^{\mathcal{I}})} & 0 & \frac{s_x}{d(u^{\mathcal{I}},v^{\mathcal{I}})} & s_x s_y & -1-s_x^2 & s_y \\ 0 & -\frac{1}{d(u^{\mathcal{I}},v^{\mathcal{I}})} & \frac{s_y}{d(u^{\mathcal{I}},v^{\mathcal{I}})} & 1+s_y^2 & s_x s_y & -s_x \end{bmatrix} \tag{5}$$

where $d(u^{\mathcal{I}}, v^{\mathcal{I}}) = Z_P^{\mathcal{C}}$ and $d(\dot{})$ represents the depth map of the image $I(u^{\mathcal{I}}, v^{\mathcal{I}}), \forall u^{\mathcal{I}} \in columns$ and $v^{\mathcal{I}} \in rows$. The interaction matrix L includes the depth $Z_P^{\mathcal{C}}$, information that is lost when working in the 2D image plane. In this article the 2D processing extracts the dark area, which is assumed to lie in front of the aerial vehicle in a close range. Making this assumption, the system considers the depth measurement as known and fixed.

Perception States The starting point to introduce the perception states in the dynamic model of the MAV and couple them with the control input and the corresponding state is presented in (4). Generally, the aim is to simplify the interaction matrix, keeping the camera motions that have the major impact in the image dynamics. The rate of change of s_x is mainly affected when the camera is undergoing a yaw motion ω_y and a lateral velocity v_x. Similarly, the rate of change of s_y is mainly affected when the camera is undergoing a vertical velocity v_y and a pitch rate ω_x. In both cases the other camera motions have negligible affect on the rate of change of the target position in the image plane. Thus, the simplified image dynamics can be summarized as it follows:

$$\begin{bmatrix} \dot{s}_x \\ \dot{s}_y \end{bmatrix} = \begin{bmatrix} -\frac{1}{d(u^{\mathcal{I}}, v^{\mathcal{I}})} & 0 & 0 & -1 - s_x^2 \\ 0 & -\frac{1}{d(u^{\mathcal{I}}, v^{\mathcal{I}})} & 1 + s_x^2 & 0 \end{bmatrix} \begin{bmatrix} v_x^{\mathcal{C}} \\ v_y^{\mathcal{C}} \\ \omega_x^{\mathcal{C}} \\ \omega_y^{\mathcal{C}} \end{bmatrix}$$

In the case where s_x is closer to the center of the image, $\omega_y^{\mathcal{C}}$ has a greater impact in the rate of change, when compared to the lateral velocity $v_x^{\mathcal{C}}$ of the camera and therefore v_x can be ignored. Similarly, for an aerial vehicle, the $\theta^{\mathcal{B}}$ is commanded to zero and as a consequence the $\dot{\theta}^{\mathcal{B}}$ will be close to zero as well, thus having a negligible effect in the s_y rate of change. To this end, the vertical velocity v_y has the main impact in the target motion in the image plane. Equations (6) and (7) express the simplified image dynamics that consist the perception state.

$$\dot{s}_x = (1 + s_x^2)(\dot{\psi}_{cmd}) \tag{6}$$

$$\dot{s}_y = \frac{1}{d(u^{\mathcal{I}}, v^{\mathcal{I}})} v_z^{\mathcal{B}} \tag{7}$$

As previously discussed, the depth $Z_P^{\mathcal{C}}$ has been assumed known and in the system it has been set at 3 m distance from the MAV. It is also evident that the s_x is directly coupled with the control input command $\dot{\psi}_{cmd}$, while s_y is coupled with the vehicle's state v_z. Finally, using (6) and (7), the perception state vector $S = [s_x, s_y] \in \mathbb{R}^2$ is defined.

2.4 Visual Processing

Depth Estimation Light scattering Cozman and Krotkov (1997) is a well known process, where the light is deflected to other directions and the formation of an image can be defined as follows:

$$I(u, v) = O(u, v) \cdot tr(u, v) + a[1 - tr(u, v)] \tag{8}$$

where $I : [0...M - 1] \times [0...N - 1] \to \mathbb{N}^2$ is the observed image, $O : [0...M - 1] \times [0...N - 1] \to \mathbb{N}^2$ is the original image, a is the color of the atmospheric light, $tr(u, v)$ is the transmission term, (u, v) are the pixel coordinates, where $u = 0, ..., M-1$ and $v = 0, ..., N-1$ with M the width and N the height of the image. The first term $O(u, v) \cdot tr(u, v)$ is called direct attenuation Tan (2008), and the second term $a[1 - tr(u)]$ is called airlight. The transmission term describes the portion of the light that is not scattered and reaches the camera and can be defined as:

$$tr(u, v) = e^{-\beta d(u,v)} \tag{9}$$

where β is the scattering coefficient of the atmosphere and $d(u, v)$ is the depth of the scene for pixel coordinates (u, v).

The (9) could be utilized in order to estimate a depth map from the original image. However, the estimation of the terms $tr(u, v)$ and a is required. Thus, the Dark Channel Prior (DCP) method He et al. (2011) is used in order to estimate these terms.

Centroid Extraction In order to extract the point with the maximum distance from the MAV, we first employ a grey scale morphological operation in the depth map image Soille (2003) and in the sequel employ the *kmeans* Theodoridis and Koutroumbas (2008) algorithm in order to segment the depth image into a number of clusters defined as $C_i, i = 1, ..., 10$. Finally, we compute the average intensity for each cluster and extract the centroid (s_x, s_y) of the cluster with the minimum average intensity C_m from the depth map image Gonzalez and Woods (2006). The centroid of the cluster is the arithmetic mean of all (x, y) pixel coordinates in the cluster defined as:

$$s_x = \frac{1}{|C_m|} \sum_{(x,y)\in C_m} x, \;\; s_y = \frac{1}{|C_m|} \sum_{(x,y)\in C_m} y \tag{10}$$

where $|C_m|$ is the number of pixels of the cluster. The calculated centroid is visualized with a red circle in the sequential frames, while the visual processing architecture provides the centroid with update rates of 1 Hz. The overall concept and impact on the centroid calculation and the overall clustering on the sequential image frames is depicted in Figure 3.

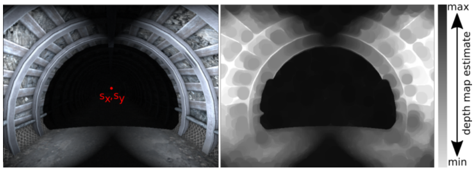

Fig. 3. On the left onboard image frame with the extracted centroid denoted with the red circle, while on the right the estimated depth map from a single image is depicted.

2.5 NMPC Controller

The NMPC considers the non-linear dynamic model of the platform, which is derived by (1), (6), and (7). The state of the system is defined as $x^{\mathcal{B}}(t) = [v_x^{\mathcal{B}}, v_y^{\mathcal{B}}, v_z^{\mathcal{B}}, \phi, \theta] \in \mathbb{R}^5$, while $\hat{x}^{\mathcal{B}}$ is the estimated state obtained from the onboard sensor measurements. More specifically, the linear velocities $[\hat{v}_x^{\mathcal{B}}, \hat{v}_y^{\mathcal{B}}, \hat{v}_z^{\mathcal{B}}]$ are provided from the ground truth simulation environment, while the roll $\hat{\phi}$ and pitch $\hat{\theta}$ states are provided from the onboard IMU. Moreover, the perception state vector $S(t) = [s_x, s_y]$ includes the visual states on the centroid, where $\hat{S}$ is the estimated state and the values $\hat{s}_x$ and $\hat{s}_y$ are provided from the visual processing unit discussed in Section 2.4.1. The NMPC objective is to generate $u^{\mathcal{B}} = [\phi_{cmd}, \theta_{cmd}, \dot{\psi}_{cmd}, T_{cmd}]$ to keep the centroid in the center of the image and by doing this, the MAV navigates along the tunnel, while following the tunnel's altitude changes. Based on the obtained commands $u^{\mathcal{B}}$, the low-level controller generates motor commands for the MAV.

The continuous-time dynamics can be discretized Stetter (1973) (e.g. using an explicit Runge-Kutta method) leading to the overall discrete-time dynamical system of the form in (11). Moreover, $x^{\mathcal{B}}_{k+j|k}$, $u^{\mathcal{B}}_{k+j|k}$ and $S_{k+j|k}$ are the vehicle's state, control action and perception state ahead of $k + j$ steps form the current time k denoted as:

$$x^{\mathcal{B}}_{k+1} = f(x^{\mathcal{B}}_k, u^{\mathcal{B}}_k), \;\; S_{k+1} = g(v^{\mathcal{B}}_{z,k}, S_k, \dot{\psi}_k) \tag{11}$$

For the proposed NMPC, the following finite horizon stage cost function is defined $l : \mathbb{R}^{n_x} \times \mathbb{R}^{n_S} \times \mathbb{R}^{n_u} \to \mathbb{R}_+$ and the terminal cost function $l_f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_S} \to \mathbb{R}_+$.

$$l(x^{\mathcal{B}}, S, u^{\mathcal{B}}) =$$

$$\underbrace{\left\|x^{\mathcal{B}} - x^{\mathcal{B}}{}_{ref}\right\|^2_{Q^{x^{\mathcal{B}}}}}_{\text{tracking vehicle states}} + \underbrace{\left\|S - S_{ref}\right\|^2_{Q^S}}_{\text{tracking the centroid}} + \underbrace{\left\|u^{\mathcal{B}} - u^{\mathcal{B}}{}_{ref}\right\|^2_{R}}_{\text{hovering term}}$$

$$lf\left(x^{\mathcal{B}}, S\right) = \underbrace{\left\|x^{\mathcal{B}} - x^{\mathcal{B}}{}_{ref}\right\|^2_{Q_f^{x^{\mathcal{B}}}}}_{\text{tracking vehicle states}} + \underbrace{\left\|S - S_{ref}\right\|^2_{Q_f^S}}_{\text{tracking the centroid}}$$

The cost function of the optimization involves four terms. The first term tracks the vehicle's states towards the desired velocities and attitudes, the second term is minimizing the position of the centroid, the third term is the hovering term, where $u^{\mathcal{B}}_{ref}$ is $[g, 0, 0]^\top$ that is the hover trust with horizontal angles, and the forth term is the terminal cost. Additionally, $Q^{x^{\mathcal{B}}} \in \mathbb{R}^{n_x \times n_x}$ is the weight matrix for the MAV's states, $Q^S \in \mathbb{R}^{n_S \times n_S}$ is the weight matrix for the perception states, $R \in \mathbb{R}^{n_u \times n_u}$ is the weight matrix for the control input, $Q_f^{x^{\mathcal{B}}} \in \mathbb{R}^{n_x \times n_x}$ is the terminal cost and $Q_f^S \in \mathbb{R}^{n_S \times n_S}$ is the terminal cost for the perception part, while each term value reflects the relative importance of the term in the objective function. Now, the NMPC scheme can be formulated in the following equation for the problem of navigation in unknown subterranean tunnels:

$$\begin{aligned} \min_{\{u^{\mathcal{B}}_{k+j|k}\}_{j=0}^{N-1}} \; & lf_N\left(x^{\mathcal{B}}_{k+N|k}, S_{k+N|k}\right) \\ & + \sum_{j=0}^{N-1} l_{k+j|k}\left(x^{\mathcal{B}}_{k+j|k}, S_{k+j|k}, u^{\mathcal{B}}_{k+j|k}\right) \\ \text{s.t.} \; & \\ & x^{\mathcal{B}}_{k+j+1|k} = f(x^{\mathcal{B}}_{k+j|k}, u^{\mathcal{B}}_{k+j|k}), \;\; j \in \mathbb{N}_{[0,N-1]} \\ & S_{k+j+1|k} = g(v^{\mathcal{B}}_{z,k+j|k}, S_{k+j|k}, \dot{\psi}_{k+j|k}), \;\; j \in \mathbb{N}_{[0,N-1]} \\ & u^{\mathcal{B}}_{k+j|k} \in [u^{\mathcal{B}}_{\min}, u^{\mathcal{B}}_{\max}], \;\; j \in \mathbb{N}_{[0,N-1]} \\ & x^{\mathcal{B}}_{k|k} = \hat{x}^{\mathcal{B}}_k \\ & S_{k|k} = \hat{S}_k \end{aligned} \tag{12}$$

where $N \in \mathbb{N}$ is the control horizon, $u^{\mathcal{B}}_{\min}$ and $u^{\mathcal{B}}_{\max}$ are bounds on control actions. At every time instant k, a finite-horizon optimal problem is solved with a user defined interval, while a corresponding optimal sequence of control actions $u^{\star\mathcal{B}}_{k|k}, \ldots\; u^{\star\mathcal{B}}_{k+N-1|k}$ are generated, where the first control action $u^{\star\mathcal{B}}_{k|k}$ is applied to the low-level controller. In the next sample time instant, the optimization solves the same problem by using the solution in the previous interval as the initial guess and the updated information on current states value. More information of the structure and implementation of the PANOC controller can be found in Stella et al. (2017), while Figure 4 presents the overall proposed system architecture.

3. SIMULATION RESULTS

The proposed framework has been evaluated in the Gazebo Koenig and Howard (2004) robot simulation environment,the code is written in C++ and is developed within the ROS framework. The robot model used in the simulation environment is a simplified multirotor CAD model, equipped with a front-facing camera, a ROSFlight flight controller Jackson et al. (2016) and

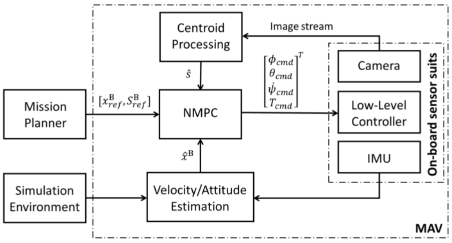

Fig. 4. Overall block diagram of the proposed system.

an IMU.Throughout the simulations, the gazebo world [1] "*tunnel_practice_2.world*", developed for the DARPA Subterranean challenge [2], has been selected for deploying the aerial platform. The NMPC has a sampling rate of 20 Hz, a prediction horizon of 40 steps, while the tuning parameters used by the NMPC were:

$$\begin{aligned} Q^x &= diag(10, 10, 10, 3, 3), & Q^x_f &= 10Q^x, \\ Q^S &= diag(10, 60), & Q^S_f &= 10Q^S, \\ R &= diag(2, 10, 10, 2) \end{aligned}$$

where *diag* denotes a square diagonal matrix. Additionally, in the control design, the following bounds have been considered as: $-0.4\,\text{rad/s} \leq [\phi_{cmd}, \theta_{cmd}, \dot{\psi}_{cmd}] \leq 0.4\,\text{rad/s}$ and $T_{cmd} \in [0, 1]$. In the simulation world, two different subterranean tunnel routes have been chosen for the MAV to explore. Figure 5 provides a graphical overview of the two tunnel morphologies.

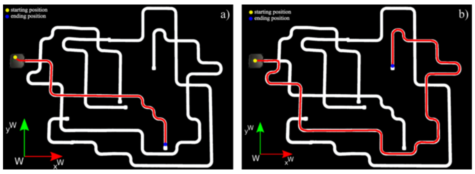

Fig. 5. Selected tunnel morphology for the evaluation of the proposed scheme from the DARPA Subterranean challenge gazebo world "*tunnel_practice_2.world*". a) depicts Scenario 1 and b) depicts Scenario 2

In both simulation scenarios, the desired reference state vectors were $x_{ref} = [2\,\text{m/s}, 0, 0, 0, 0]$ and $S_{ref} = [0, 0]$, while it should be highlighted that the MAV can perform the exploration task without any prior information from the environment, since the MAV navigates by tracking the centroid in the center of the acquired image frame, enabling in this way the proper translation along the tunnel's horizontal and vertical axes. For these simulation scenarios the following corresponding videos depict the exact responses of the aerial vehicle - Video1: `https://youtu.be/F34t3BBHu7w` and Video2: `https://youtu.be/6FEU-wxN7WM`.

First Scenario The selected tunnel focuses mainly in the s_y tracking, since the tunnel has a challenging part, where two consecutive altitude descending tunnel areas are followed by an altitude ascending tunnel area. Figure 6 depicts the xy and xz views of the ground truth path of the MAV. This information has not been used from the proposed system and is provided only for visualization purposes. A main point of the resulting translation based on the novel proposed navigation scheme is that the MAV has properly followed the center of the tunnel also on the z axis with an overall descending of 10 meters. The first subfigure of Figure 7 presents the tracking of $v^{\mathcal{B}}_x$ to the desired reference. In this case the controller was able to follow the reference with an Mean Absolute Error of 0.09 m/s. The second subfigure of Figure 7 showcases the reference tracking performance of s_x over time. For the first 20 sec, the aerial platform starts from an off-center pose outside the tunnel and therefore the centroid on the x axis is slightly oscillating in a way that guides the MAV inside the tunnel. For the time instances 38 sec, 65 sec, 140 sec, 150 sec and 165 sec the centroid reaches peak values, depicting that the tunnel is taking a turn and the MAV has to yaw along the axis. Moreover, for the time instances 85 sec and 118 sec the centroid is oscillating but the magnitude of the oscillation is with an error less than 0.05. This oscillation can also denote that the aerial platform is flying close to another obstacle in the tunnel and does a manoeuvre to avoid it. The third subfigure of Figure 7 depicts the s_y reference tracking. This plot shows that the values are close to zero, but it also includes peaks that are produced from the visual processing, since s_y sometimes jumps back and forth some pixels. Nevertheless, these peaks occur instantaneously, where the average magnitude of the error $e_{s_y} = s^{ref}_y - s_y$ is bounded $-0.05 \leq \bar{e}_{s_y} \leq 0.05$. Therefore, the controller tries to regulate the small errors and the altitude of the MAV during the navigation is not disturbed overall. For the time instances 85 sec and 110 sec the error is positive and reaches 0.3 meaning that the tunnel is vertically descending and the MAV follows it. Moreover, for the time instance 122 sec, the error reaches -0.22 and this denotes a vertical ascending of the tunnel and the fact that the MAV tries to follow the vertical axis without colliding to the walls.

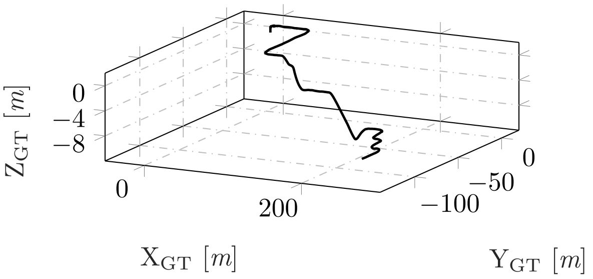

Fig. 6. Trajectory of the MAV during the navigation in the 1st tunnel scenario

The control signals (roll, pitch, and the normalized thrust references) are generated by the non linear solver and are depicted at Figure 8. An interesting note in the presented responses is that when the tunnel is taking a turn, the roll of the MAV is offsetting from zero, meaning that the roll is connected with the centroid motion on x axis, but the magnitude is negligible. This also leads to faster linear velocities in the x axis, as shown in Figure 7.

Second Scenario In this scenario, the translation along the tunnel axis has been evaluated, since it contains multiple turns in various configurations. Figure 9 depicts the ground truth 3D path that have been followed by the

[1] `https://bitbucket.org/osrf/subt/wiki/Home`
[2] `https://subtchallenge.com/`

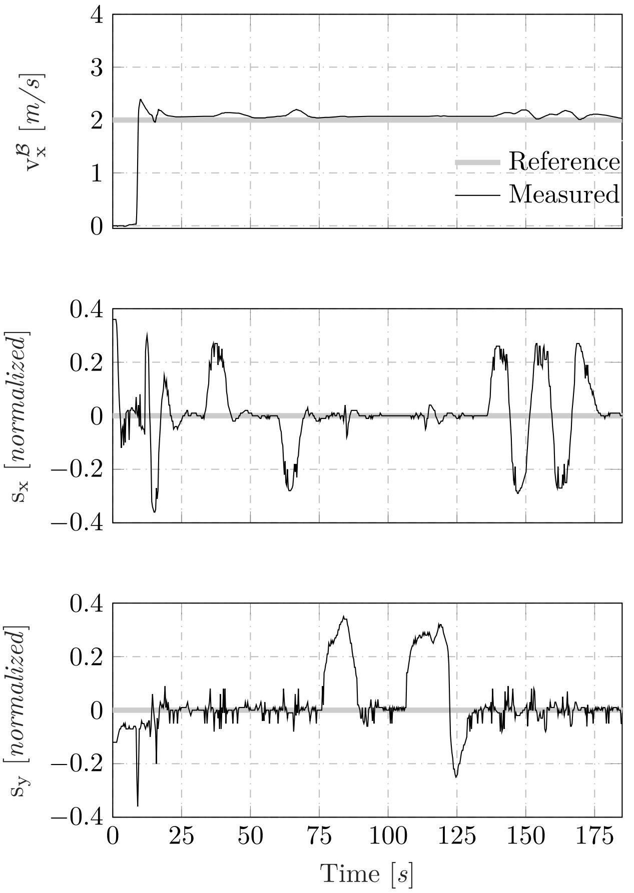

Fig. 7. Linear velocity x, centroid s_x and s_y reference tracking during the navigation in the 1st tunnel scenario

MAV. This Figure also demonstrates the altitude descend of the vehicle of an overall 8.5 m.

Similarly to the case 1, Figure 10 presents the reference tracking plots for the $v_x^{\mathcal{B}}$, s_x and s_y. The performance is similar, where the s_x has more peaks, since the tunnel contains more turns and for the s_y, tracking peaks occur similar to the case 1, without affecting majorly the navigation.

Table 1 presents a summary of the performed simulations, depicting the navigation time and the Mean Absolute Error (MAE) for s_x, s_y and $v_x^{\mathcal{B}}$.

Table 1. Comparison of the method in each tunnel for three different velocity references

	Navigation Time			**$v_x^{\mathcal{B}}$ MAE**		
Ref $v_x^{\mathcal{B}}$	1 m/s	1.5 m/s	2 m/s	1 m/s	1.5 m/s	2 m/s
Tunnel 1	358 s	240 s	195 s	0.04	0.07	0.09
Tunnel 2	873 s	592 s	467 s	0.05	0.07	0.1

	s_x MAE			**s_y MAE**		
Ref $v_x^{\mathcal{B}}$	1 m/s	1.5 m/s	2 m/s	1 m/s	1.5 m/s	2 m/s
Tunnel 1	0.03	0.045	0.07	0.036	0.05	0.06
Tunnel 2	0.04	0.06	0.08	0.02	0.025	0.03

4. CONCLUSIONS

This article approached the problem of fast autonomous navigation in dark, featureless and unknown subterranean environments like tunnels, by coupling a NMPC control architecture that plans the robot motion along the tunnel axis and regulates its altitude with a single image frame processing architecture. In the proposed architecture the robot is considered as a free flying object, while its dynamic model has been augmented with vision based perception states and is part of the objective function for the overall optimization problem. On top of that, a method based on a sequence of single frame images was used to extract a navigation centroid in the image plane in order to identify the free space of the environment both horizontally and vertically. Simulation results in a realistic simulation environment demonstrated the ability of the proposed system to navigate along dark tunnels.

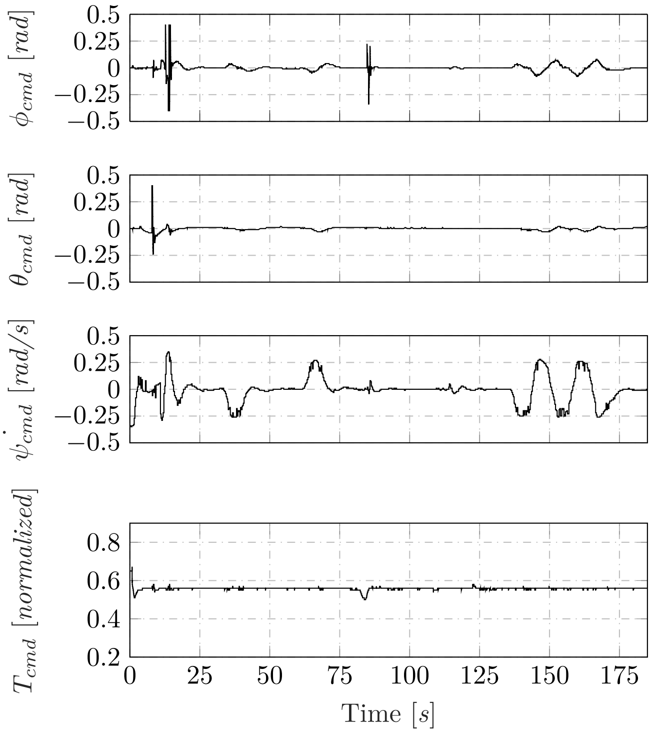

Fig. 8. Control signals generated from the PANOC solver to the low-level controller during the navigation in the 1st tunnel scenario.

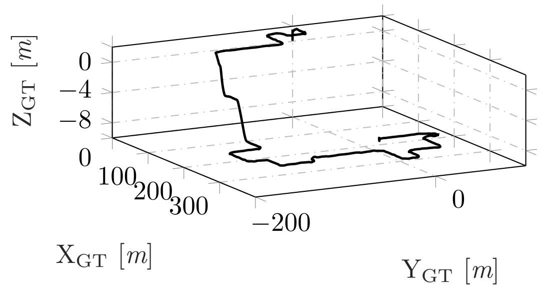

Fig. 9. Trajectory of the MAV during the navigation in the 2nd tunnel scenario.

REFERENCES

Chaumette, F. and Hutchinson, S. (2006). Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4), 82–90.

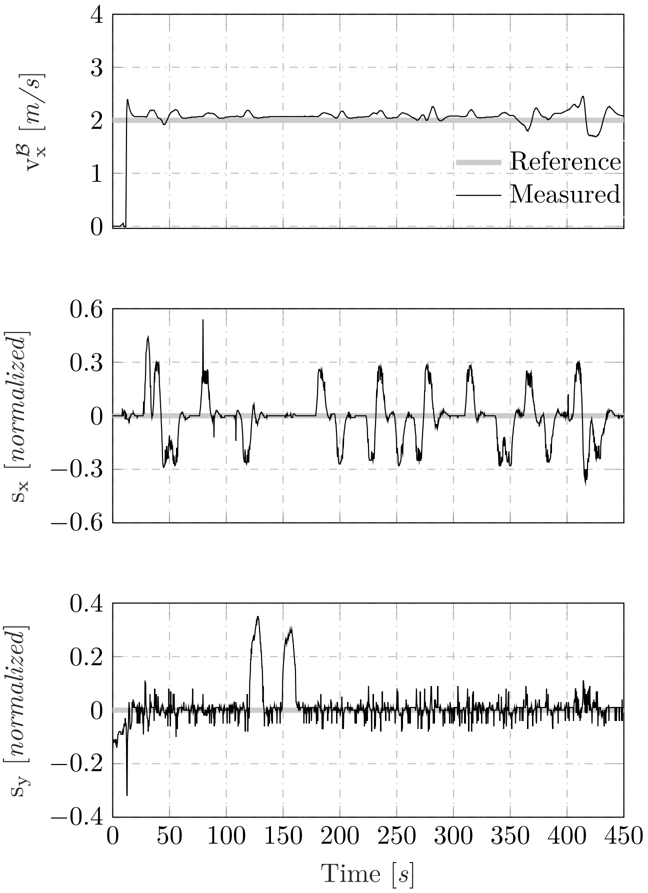

Fig. 10. Linear velocity x, centroid s_x and s_y reference tracking during the navigation in the 2nd tunnel scenario.

Cozman, F. and Krotkov, E. (1997). Depth from scattering. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 801–806. doi:10.1109/CVPR.1997.609419.

Falanga, D., Foehn, P., Lu, P., and Scaramuzza, D. (2018). Pampc: Perception-aware model predictive control for quadrotors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–8. IEEE.

Gonzalez, R.C. and Woods, R.E. (2006). *Digital Image Processing (3rd Edition).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision.* Cambridge university press.

He, K., Sun, J., and Tang, X. (2011). Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12), 2341–2353. doi:10.1109/TPAMI.2010.168. URL `http://dx.doi.org/10.1109/TPAMI.2010.168`.

Jackson, J., Ellingson, G., and McLain, T. (2016). ROSflight: A lightweight, inexpensive MAV research and development tool. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 758–762. doi:10.1109/ICUAS.2016.7502584.

Kamel, M., Stastny, T., Alexis, K., and Siegwart, R. (2017). Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system. In *Robot Operating System (ROS)*, 3–39. Springer.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, 2149–2154. IEEE.

Mascarich, F., Khattak, S., Papachristos, C., and Alexis, K. (2018). A multi-modal mapping unit for autonomous exploration and mapping of underground tunnels. In *2018 IEEE Aerospace Conference*, 1–7. IEEE.

Mcfadyen, A., Mejias, L., Corke, P., and Pradalier, C. (2013). Aircraft collision avoidance using spherical visual predictive control and single point features. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 50–56. IEEE.

Özaslan, T., Loianno, G., Keller, J., Taylor, C.J., Kumar, V., Wozencraft, J.M., and Hood, T. (2017). Autonomous navigation and mapping for inspection of penstocks and tunnels with mavs. *IEEE Robotics and Automation Letters*, 2(3), 1740–1747.

Potena, C., Nardi, D., and Pretto, A. (2017). Effective target aware visual navigation for uavs. In *2017 European Conference on Mobile Robots (ECMR)*, 1–7. IEEE.

Sheckells, M., Garimella, G., and Kobilarov, M. (2016). Optimal visual servoing for differentially flat underactuated systems. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5541–5548. IEEE.

Small, E., Sopasakis, P., Fresk, E., Patrinos, P., and Nikolakopoulos, G. (2019). Aerial navigation in obstructed environments with embedded nonlinear model predictive control. In *2019 European Control Conference (ECC).* IEEE.

Soille, P. (2003). *Morphological Image Analysis: Principles and Applications.* Springer-Verlag, Berlin, Heidelberg, 2 edition.

Stella, L., Themelis, A., Sopasakis, P., and Patrinos, P. (2017). A simple and efficient algorithm for nonlinear model predictive control. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 1939–1944. IEEE.

Stetter, H.J. (1973). *Analysis of discretization methods for ordinary differential equations*, volume 23 of *Springer Tracts in Natural Philosoph.* Springer.

Tan, C.H., Sufiyan, D., Ang, W.J., Win, S.K.H., and Foong, S. (2019). Design optimisation of sparse sensing array for extended aerial robot navigation in deep hazardous tunnels. *IEEE Robotics and Automation Letters.*

Tan, R.T. (2008). Visibility in bad weather from a single image. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. doi:10.1109/CVPR.2008.4587643.

Theodoridis, S. and Koutroumbas, K. (2008). *Pattern Recognition, Fourth Edition.* Academic Press, Inc., Orlando, FL, USA, 4th edition.