

# Non-linear MPC based Navigation for Micro Aerial Vehicles in Constrained Environments

Björn Lindqvist, Sina Sharif Mansouri and George Nikolakopoulos

**Abstract**—This article establishes a novel Non linear Model Predictive Control (MPC) scheme for the navigation of a MAV (Micro Aerial Vehicle) in constrained environments, such as narrow passages, multi-obstacle populated spaces and tight openings. The proposed NMPC optimization framework is based on the Proximal Averaged Newton type method for Optimal Control (PANOC) and has the merit to employ a penalty method for the proper consideration of the obstacles and other environmental constraints during the navigation. The proposed scheme has the ability to be a fast solution for the navigation of MAVs that can be directly applied online and thus it is creating a powerful navigation framework for demanding flights. For achieving such an agile and fast aerial navigation, the article will also present the proposed penalty creation methodology for dealing with the obstacle avoidance and the space constraint navigation. Finally, the efficacy of the proposed scheme will be demonstrated by multiple simulation results under constrained and demanding environments.

## I. INTRODUCTION

In the latest years Micro Aerial Vehicles (MAVs) have been established as a base line for multiple demanding application sectors, with the most representative ones to be the aerial inspection of infrastructure [1], the underground mine inspection [2], search-and-rescue missions [3], and the subterranean exploration [4]. Due to the agility of the MAV, these platforms can provide access to unreachable areas and can be used for monitoring, exploring and inspection. However, narrow passages, limited area for entrances, such as windows, small voids and general narrow constrained spaces and multiple obstacles that constrain the available open space are challenges for autonomous navigation, while they are introducing multiple demanding requirements to the adopted aerial navigation scheme.

To overcome these real-life induced problems in aerial navigation, this article proposes a novel Nonlinear Model Predictive Control (NMPC) for establishing a fully autonomous navigation framework in constrained environments. As it will be presented in the sequel, the NMPC optimization problem is solved using Proximal Averaged Newton-type method for Optimal Control (PANOC) [5] and employs a penalty method where the optimization problem is resolved with increasing constraint penalties, until the computed path does not enter the obstacles.

This work has been partially funded by the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 730302 SIMS.

Robotics Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå SE-97187, Sweden.

Corresponding Author's email: bjlinv-3@ltu.se

## A. Background and Motivation

As MAV's start appearing in more and more applications, the need for stable and intelligent control algorithms is increasing as well. Moving from a human controlled or human aided MAV to a fully autonomous operation, requires the controller to naturally and in real-time interact with the environment around the MAV for obstacle avoidance and fast path planning. In the related literature, there are numerous control schemes for path-planning and obstacle avoidance that have successfully been utilized for MAV's[6], [7], for example reactive navigation schemes as the widely used potential fields [8], [2] or dynamic graph search methods like the modified versions of  $A^*$  such as ADA\* [9] or MSA\* [10]. Lately, there have been a few approaches towards NMPC schemes for the combined problem of control and path planning as in [11], while the ability has been demonstrated of such advanced and demanding control schemes to operate efficiently in experimental environments [12]. In the specific case of the NMPC schemes, the main problematic issue is the solver time and the main ability to compute in real-time or close to real-time way-points that will satisfy the overall mission demands and timing issues. In the current novel proposed approach, the adopted solver is the Optimization Engine and its related algorithm PANOC (Proximal Averaged Newton-type method for Optimal Control) [13], [5] that is an optimization algorithm for non-convex and nonlinear optimization problems. As the acronym implies, it uses a Newton-type method and is designed to be used for optimal control problems, while its main advantages lie in being matrix-free, requiring only simple algebraic operations, that allows for a very fast convergence time and makes it an ideal framework for real-life implementations on MAVs. Optimization Engine also employs a Penalty Method[14] for the consideration of constraints.

## B. Contributions

Based on the aforementioned state of the art, the first major contribution of this article and the main difference between the proposed method and other control architectures, for online path planning, is the fact that the NMPC produced path is computed in relation to a non-linear dynamic of the MAV in its entirety, i.e. the path is described by a series of control inputs. With a well tuned NMPC and a good prediction model this ensures that the computed path is feasible and smooth, and that there is no disconnection between the path planner and the lower-level controllers. The second contribution is the proposed novel method for describing parametric geometric constraints for the NMPC, in particular

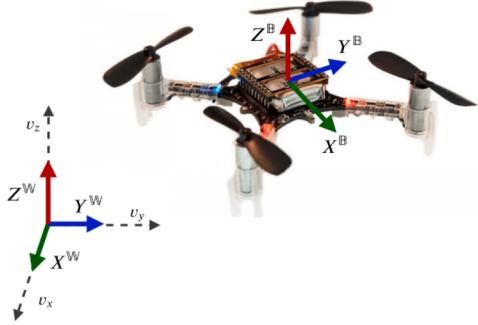


Fig. 1: Coordinate frames, where  $\mathbb{W}$  and  $\mathbb{B}$  denote the world and body coordinate frames respectively.

the parameterization of general obstacle geometries that can be extended to real-world application scenarios for obstacle avoidance and path planning. In the proposed approach, the NMPC considers the non-linear kinematics of the MAV and provides corresponding optimal paths, while avoiding obstacles with online calculated smooth control inputs. The third contribution of this article is based on the multiple simulations results and the overall demonstration of the current NMPC framework for establishing a non-linear solver that is computationally stable and fast enough to match what is required to stabilize a MAV (approximately at 20Hz way-points generation) and an approach to constrain the state-space based on the location, size and geometry of the obstacles, by including parametric constraints.

### C. Outline

The rest of the article is structured as follows. Initially, the kinematic model of the MAV is presented in Section II-A, followed by the presentation of the corresponding objective function and the formulation of the obstacle constraints in Sections II-B and II-C respectively. The proposed optimization is presented in Section II-D, while multiple simulation scenarios, with related evaluation results that prove the efficiency of the proposed framework are presented in Section III, with an additional corresponding comparison and discussion. Finally, Section IV concludes the article by summarizing the findings and offering some directions for future research.

## II. METHODOLOGY

### A. MAVs Kinematics

The considered MAV's coordinate systems are depicted in Figure 1, where the body frame  $[x^{\mathbb{B}}, y^{\mathbb{B}}, z^{\mathbb{B}}] \in \mathbb{R}^3$  is fixed to the MAV, while the world frame is denoted as  $[X^{\mathbb{W}}, Y^{\mathbb{W}}, Z^{\mathbb{W}}] \in \mathbb{R}^3$ . The six Degrees of Freedom (DoF) MAV is defined by the set of equations in (1), while the full

derivation of the adopted modeling is described in [11].

$$\dot{P}(t) = V(t) \quad (1a)$$

$$\dot{V}(t) = R(\phi, \theta) \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} - \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} V(t) \quad (1b)$$

$$\dot{\phi}(t) = 1/\tau_{\phi}(K_{\phi}\phi_{ref}(t) - \phi(t)) \quad (1c)$$

$$\dot{\theta}(t) = 1/\tau_{\theta}(K_{\theta}\theta_{ref}(t) - \theta(t)), \quad (1d)$$

where  $P(t) \in \mathbb{R}^3$  denotes the position  $[x, y, z]^T$ ,  $V(t) \in \mathbb{R}^3$  denotes the linear velocities  $[v_x, v_y, v_z]^T$  along each world frame axes as seen in Figure 1, and  $\phi(t), \theta(t) \in \mathbb{R} \cap [-\pi, \pi]$  are roll and pitch angles along the  $X^{\mathbb{W}}, Y^{\mathbb{W}}$  axis respectively. Moreover,  $R(\phi(t), \theta(t)) \in SO(3)$  is a rotation matrix that describes the attitude in Euler form, with  $\phi_{ref} \in \mathbb{R}$ ,  $\theta_{ref} \in \mathbb{R}$  and  $T(t) \in \mathbb{R}_+$  to be the references in roll, pitch and total thrust. The adopted model assumes that the acceleration is only dependant on the magnitude and angle of the thrust vector, produced by the motors, as well as the linear damping terms  $A_x, A_y, A_z \in \mathbb{R}$  and the gravity of earth denoted by  $g \in \mathbb{R}$ . The attitude-terms are modeled as a first order system of the attitude (roll/pitch) and the reference, with gains of  $K_{\phi}, K_{\theta} \in \mathbb{R}^2$  and time constants of  $\tau_{\phi}, \tau_{\theta} \in \mathbb{R}^2$ . It is also assumed that the MAV is equipped with a lower-level attitude controller that takes thrust, roll and pitch commands and provides motor commands for the MAV [15].

### B. Cost Function

The MAVs state vector is denoted as  $X = [P, V, \phi, \theta]^T$ , while the control input is  $u = [T, \phi_{ref}, \theta_{ref}]^T$  and (1) is discretized for each time instant  $(k+j|k)$  with a sampling time of  $\delta_t \in \mathbb{Z}^+$  using the forward Euler method. Additionally,  $(k+j|k)$  denotes the prediction of the time step  $k+j$ , produced at the time step  $k$ , while it is the discretized form of (1) that is used for the prediction model of the NMPC. The objective of the controller is to navigate to the reference position, while avoiding the obstacles. Thus, the objective function considers three terms, as depicted in (2).

$$J = \sum_{j=1}^N \underbrace{(x_{ref} - x_{k+j|k})^T Q_x (x_{ref} - x_{k+j|k})}_{\text{state cost}} \quad (2a)$$

$$+ \underbrace{(u_{ref} - u_{k+j|k})^T Q_u (u_{ref} - u_{k+j|k})}_{\text{input cost}} \quad (2b)$$

$$+ \underbrace{(u_{k+j|k} - u_{k+j-1|k})^T Q_{\Delta u} (u_{k+j|k} - u_{k+j-1|k})}_{\text{control actions smoothness cost}} \quad (2c)$$

where  $Q_x \in \mathbb{R}^{8 \times 8}$ ,  $Q_u, Q_{\Delta u} \in \mathbb{R}^{3 \times 3}$  are weight matrices for the state weights, input weights and input rate weights respectively. In (2) the first term of (2a) describes the *state cost*, which is the cost associated with deviating from a certain state reference  $x_{ref}$ . The second term of (2b) describes the *input cost* that penalizes a deviation from the steady-state input  $u_{ref} = [g, 0, 0]$  i.e. the inputs that describe hovering.

Finally, to guarantee that the control actions are smooth, the third term of (2c) is added that compares the input at  $(k+j-1|k)$  with the input at  $(k+j|k)$  and penalizes the changing of the input from one time step to the next one, with  $N \in N^+$  to denote the Control Horizon of the NMPC.

### C. Obstacle Definition and Constraints

The main challenges for guaranteeing collision free paths are: a) to define the obstacle constraints with a well defined gradient, and b) to design the constraints fully parametric, so that their positions and size is part of the input fed to the NMPC scheme. Moreover, the parametric constraints should have an associated cost of zero, when the MAV is outside the obstacle. Thus, in this article, the function  $\max(h, 0) = [h]_+$  is proposed for this purpose. By choosing  $h$  as an expression that is positive inside of the constrained area and negative outside of it,  $[h]_+$  will be positive inside the constraint and zero outside it. Additionally, this novel presentation of the constraints results to the ability of defining more complex geometries by taking the product of multiple such  $[h]_+$  terms, as the product is zero where any of the terms are negative. For the case of MAV navigation, there are multiple different types of obstacles that can be encountered in the surrounding environment, however for a collision free navigation, all these obstacles can be categorized to three types, such as a cylinder shape, a polytope surface, and a constrained entrance or passages, where their parametric presentation as corresponding constraints will be discussed in the sequel. Moreover, to guarantee bounds on changes in control actions, a constraint on the control input variations will be also established. Finally, it should be highlighted that all the underlying constraints are summed over the control horizon  $N$  to account for the obstacle constraints at all the predicted future time steps.

1) *Cylinder Obstacle*: This constraint can be used for any blocked area or general obstacle, where the radius of the cylinder envelops the area that is undesirable. The cylinder constraint is defined in (3) with a specified height and radius as:

$$\begin{aligned} h_{cylinder} &= r_{obs}^2 - (x_{k+j|k} - x_{obs})^2 - (y_{k+j|k} - y_{obs})^2, \\ h_{zmax} &= -z_{k+j|k} + z_{obs}, \end{aligned} \quad (3)$$

where  $x_{obs}, y_{obs}, z_{obs}$  and  $r_{obs}$  are the  $x, y, z$  coordinates and radius of the obstacle. The constraint equations (4) are obtained by multiplying these two equations as:

$$C_{cylinder} = \sum_{j=0}^N [h_{cylinder}]_+ [h_{zmax}]_+. \quad (4)$$

A visual representation of the obstacle can be seen in Figure 2, in the form of a cost map. This is a 2D slice of the 3D space where the new third axis represent the cost related to violating the obstacle constraint.

2) *Surface Obstacle*: The surface obstacle represents a wall-like obstacle with a limited width, length and height using six line/plane equations. Although using only a line could be enough to prevent collisions, the goal is a general

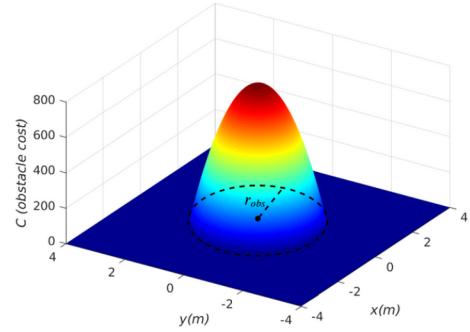


Fig. 2: Cost map of cylinder with radius 2m at  $z=1$ .

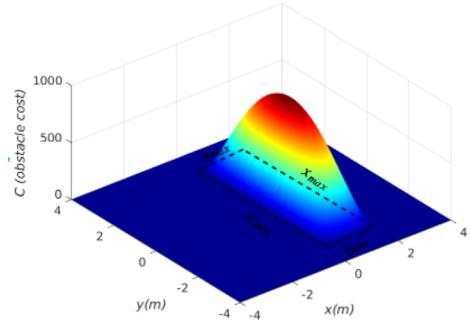


Fig. 3: Cost map of constrained surface.

parameterized presentation that can represent any such obstacle. For ease of presentation the 1D line/plane case can be denoted by a minimum/maximum extent along each axis as

$$\begin{aligned} h_{min} &= p_{k+j|k} - p_{min}, \\ h_{max} &= -(p_{k+j|k} - p_{max}), \end{aligned} \quad (5)$$

where  $p$  denotes a position coordinate of the MAV and  $p_{min}, p_{max}$  denotes the extent of the constrained area. Let every  $h$ -term for  $x, y$  and  $z$  respectively describe a line/plane equation then the full surface is:

$$\begin{aligned} C_{wall} = \sum_{j=0}^N &[h_{xmax}]_+ [h_{xmin}]_+ [h_{ymax}]_+ \dots \\ &[h_{ymin}]_+ [h_{zmax}]_+ [h_{zmin}]_+. \end{aligned} \quad (6)$$

Without any loss of generality, the framework can be directly extended for the description of the obstacle surfaces by the full plane equations that could come directly from approximation from sensor data, an attribute that can be directly linked with real life applications and real-life sensory systems for environmental perception e.g. lidars. Figure 3 shows the cost map of an example of the constrained surface type obstacle.

3) *Restricted Entrances*: In order to define the geometry of a constrained passage, three constraints are needed. First, the entrance itself is described by a cylinder geometry in the  $y - z$  plane as in (7). The minus in front of the expression inverses the constraint, so that the expression is negative

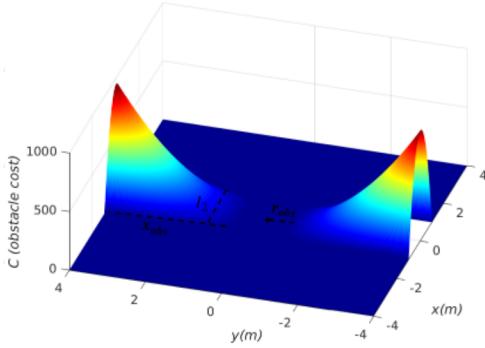


Fig. 4: Costmap of limited entrance constraint at  $z=1\text{m}$ . Centre of entrance at  $(0,0,1)$  with radius 1m and length of 2m.

inside the radius of the cylinder and positive elsewhere.

$$h_{\text{entrance}} = -(r_{\text{obs}}^2 - (y_{k+j|k} - y_{\text{obs}})^2 + (z_{k+j|k} - z_{\text{obs}})^2) \quad (7)$$

To limit the passage area size, constraints of the length of the passage in the  $x$ -direction (8) are defined. In this case,  $x_{\text{obs}}$ ,  $l_1, l_2$  are the centre of the passage, and the extent of the passage in the  $x$ -direction respectively, while  $h_{xmin}$  is negative for  $x_{k+j|k} < x_{\text{obs}} - l_1$  and  $h_{xmax}$  is negative for  $x_{k+j|k} > x_{\text{obs}} + l_2$ . Depending on the direction of the entrance, either  $l_1$  or  $l_2$  should be set to zero and this will result to that the entrance position is  $x_{\text{obs}}$ .

$$\begin{aligned} h_{xmin} &= x_{k+j|k} - x_{\text{obs}} + l_1 \\ h_{xmax} &= -(x_{k+j|k} - x_{\text{obs}}) + l_2 \end{aligned} \quad (8)$$

As in the previous cases, the constraint is summed over the prediction horizon, in 9:

$$C_{\text{passage}} = \sum_{j=0}^N [h_{\text{entrance}}] + [h_{xmax}] + [h_{xmin}] + \quad (9)$$

A visual representation of the obstacle can be found in Figure 4.

*4) Control Input Constraint::* In addition to the penalty method being applied to obstacle constraints, included is also a penalty constraint for the change in the control input. These constraints are included to ensure non-aggressive behaviour of the control inputs  $\phi_{\text{ref}}$  and  $\theta_{\text{ref}}$  even in very demanding obstacle avoidance scenarios and can be denoted as:

$$C_{\Delta\phi} = \sum_{j=0}^{N-1} [-\Delta\phi_{\max}^2 + (\phi_{\text{ref}(k+j-1|k)} - \phi_{\text{ref}(k+j|k)})^2] + \quad (10a)$$

$$C_{\Delta\theta} = \sum_{j=0}^{N-1} [-\Delta\theta_{\max}^2 + (\theta_{\text{ref}(k+j-1|k)} - \theta_{\text{ref}(k+j|k)})^2] + \quad (10b)$$

These constraints limit the change in the control inputs from one time step to the next by a maximum value, here denoted  $\Delta\phi_{\max}$  and  $\Delta\theta_{\max}$  respectively. In addition to the rate constraint, let hard bounds on the control inputs be denoted as:

$$u_{\min} \leq u_{k+j|k} \leq u_{\max}. \quad (11)$$

#### D. Solver

The utilized Optimization Engine is an open source tool for generic optimization, and its centre-piece is the non-linear non-convex solver PANOC. The algorithm of PANOC is well described in [5]. Optimization Engine can solve parametric optimization problems using a quadratic penalty method as

$$\text{Minimize}_{z \in Z} f(z, p) + c \|F(z, p)\|^2 \quad (12)$$

where  $f$  is a Lipschitz-differentiable function,  $F$  is a vector-valued mapping so that  $\|F(z, p)\|^2$  is a Lipschitz-differentiable function and  $c$  is a positive penalty parameter. The cost function in (2) and the constraints described in II-C can be fit into this framework by performing a single-shooting of the cost function by  $(u_{k+j|k})_j = z$ , and setting  $Z$  to be bound by (11). In addition let  $F$  map the obstacle and input rate equality constraints. The parameter  $p$  includes the initial condition of the state vector  $X$  as well as references and obstacle data. The idea is to re-solve (12) with an exponentially increasing penalty parameter until the constraints satisfy a specified tolerance. This can be seen as the cost-minima gradually moving out of the obstacles as  $c$  is increased until the path does not enter the obstacles. The simulations in III are performed with a maximum of four penalty method iterations.

### III. SIMULATION RESULTS

For the presented simulations, the corresponding model parameters can be described as in (1) and are chosen as  $\tau_\phi, \tau_\theta = 0.5$ ,  $K_\phi, K_\theta = 1$ , in order to approximately match the response of a low-level controller acting on a MAV. Additionally,  $g$  is set to  $9.81 \text{ m/s}^2$ , the control horizon is set to  $N = 40$  and  $\delta t = \frac{1}{20} \text{ s}$ , which implies a prediction of two seconds. The weights in (2) are chosen as:

$$\begin{aligned} Q_x &= \text{diag}(5, 5, 20, 3, 3, 3, 8, 8) \\ Q_u &= \text{diag}(5, 10, 10) \\ Q_{\Delta u} &= \text{diag}(5, 12, 12) \end{aligned} \quad (13)$$

The constraints on control inputs are chosen (in SI-units) as:

$$u_{\min} = \begin{bmatrix} 5 \\ -0.5 \\ -0.5 \end{bmatrix}, u_{\max} = \begin{bmatrix} 13.5 \\ 0.5 \\ 0.5 \end{bmatrix} \quad (14)$$

Additionally the constraints on change in input described in (10) are chosen as  $\Delta\phi_{\max} = 0.07$  and  $\Delta\theta_{\max} = 0.07$ . In the following presented simulations the task of the MAV will be to take off from a selected set-point and then follow a series of set-point position references through the constrained environment while avoiding obstacles.

#### A. Multiple Cylinder Obstacles

The first simulated scenario, depicted in Figure 5, considers the path of the MAV, while passing four cylindrical obstacles, as defined by (4). All four cylinder obstacles have a height of 1.5m and a radius of 0.8m. In this scenario the controller is provided with three separate position references at 2 seconds (take-off), 4 seconds, and 14 seconds into the simulation respectively. As it is indicated from the executed

MAV path the scheme has the ability to clearly avoid the obstacles, while passing and reaching the desired reference.

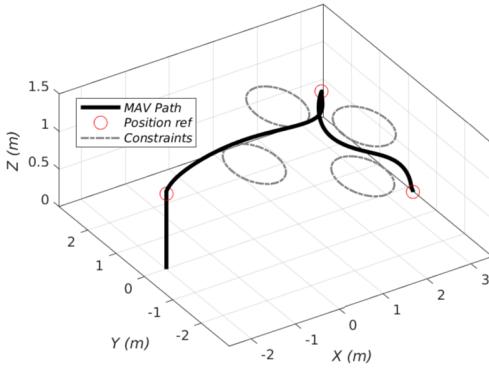


Fig. 5: Path around multiple cylinder obstacles with radius 0.8m and height 1.5m.

In Figure 6 the avoidance manoeuvres, while passing through the cylindrical obstacles can be seen more clearly, as the curved parts of the  $x$  and  $y$  position graphs.

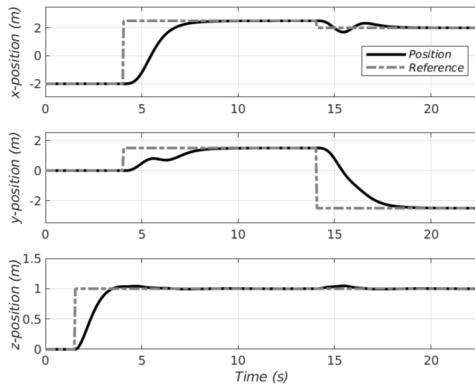


Fig. 6: Position data for path around cylinders.

In this case, the altitude is kept at the reference throughout the manoeuvre, while the computed control inputs, depicted in Figure 7 are smooth and feasible, and are kept within the input constraints with respect to the maximum and minimum values, as well as to the change in input from one time step to the next one. Finally, Figure 8 presents information towards the solver itself, including the solver iterations, the constraint violations and the general solver time. It should be noted that this is the maximum constraint violation in the full prediction and as can be seen in the simulation plots the MAV never enters the obstacle. For the simulations in sections III-B and III-C the results regarding solver data and control inputs looks very alike and the solver time continuously peaks at around 6-8ms when all four iterations of the penalty method are applied during the obstacle avoidance manoeuvres, and stays at around 0.5ms during normal flight.

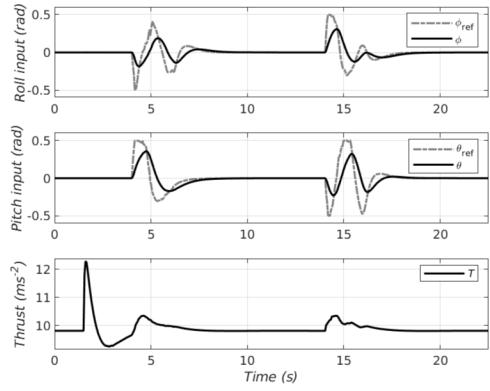


Fig. 7: Optimized Control Inputs and corresponding states for path around cylinders.

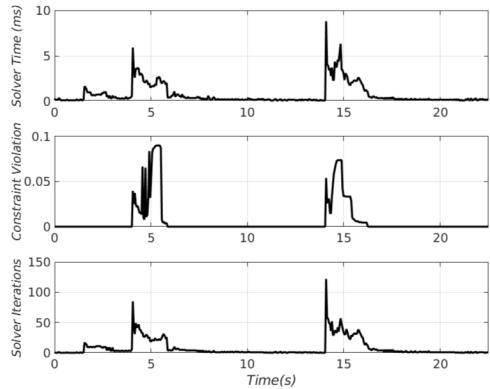


Fig. 8: Solver data for cylinder simulation.

### B. Multiple Constrained Surface Obstacles

In the second simulated case, the problem of obstacle avoidance with a constrained surface type obstacle will be considered, while Figure 9 shows the path of the MAV, when passing two surface obstacles. The first obstacle is a wall with a large extent in  $y$  ( $y_{min} = -5$  m,  $y_{max} = 5$  m) and a height of  $z_{max} = 1.5$  m that inclines the MAV to move above it. The second obstacle is defined by  $y_{max} = 1$  m and a height of  $z_{max} = 3$  m, making it more optimal for the MAV to move around the wall. Both walls have a width of 1 m, and in both cases the MAV can choose to move either above or around the obstacle but based on the choice of obstacle parameters the choice is different. The MAV cleanly moves past both obstacles without breaking the obstacle constraints.

### C. Constrained Entrance Obstacles

The final simulated scenario is depicted in Figure 10 and it considers the case of constrained entrance obstacles. More specifically, in this case the MAV is passing through two constrained entrances or narrow passages. The radius of the passage is  $r_{obs} = 0.125$  m and has an extent of 1 m, while the other parameters are  $x_{obs} = -0.5$  m,  $y_{obs} = -0.5$  m,  $z_{obs} = 1$  m for the first pass and  $x_{obs} = 0.5$  m,  $y_{obs} = -1.5$  m,  $z_{obs} = 0.3$  m for the second

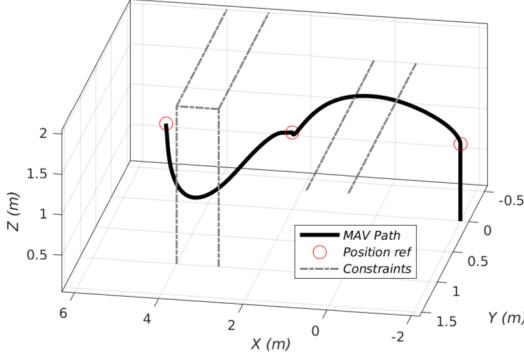


Fig. 9: Path past two wall-like obstacles with different heights and extent.

pass through the narrow passage.

As it has been demonstrated from the obtained responses, the MAV finds the entrance and passes through it without entering the obstacle. It should be noted that, as described by the obstacle geometry, the only path for reaching the final way-point from the other side is to go through the passage. Thus the difference between this scenario and the previous two is that to avoid the obstacle the MAV does not only have to move out of the way of obstacles, but also to find a specific path through the constrained environment, which is a significant merit and contribution of the overall proposed framework for the aerial navigation of MAVs.

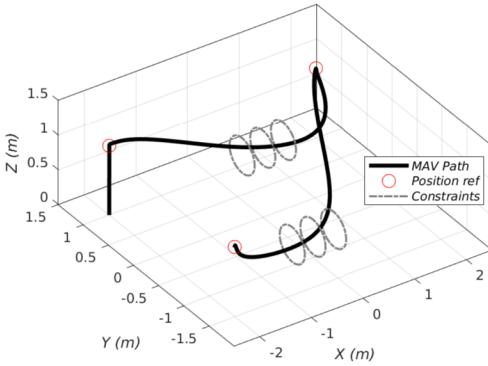


Fig. 10: Path through constrained entrances.

#### IV. CONCLUSIONS

In this article a novel obstacle avoidance method has been established that is able to represent general obstacle geometries as constraints and to result in collision free paths based on a NMPC framework that is based on PANOC. The efficiency of the proposed scheme has been demonstrated based on multiple simulation scenarios and various types of obstacles. As it has been demonstrated, the NMPC has the ability to successfully solve online the optimization problem for completing the required mission based waypoints, while avoiding obstacles or passing through them, wherever this is demanding for mission completion, without

breaking obstacle constraints nor constraints acting on the inputs. From the obtained simulation results, it has been also demonstrated that the overall computation time is kept well below the required sampling rate for the considered MAV and that only peaks to 8ms. For future work the obvious next step is integrating this method to a real MAV, starting with 2D obstacle detection via a LIDAR sensor and developing a system of tracking obstacles. Another extension could be the flight into an already mapped constrained area, where the full description of the environment could be pre-loaded as a set of multiple constraints of the proposed type that fully describe the constrained and undesirable areas of the environment to ensure collision free paths.

#### REFERENCES

- [1] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, “Cooperative coverage path planning for visual inspection,” *Control Engineering Practice*, vol. 74, pp. 118–131, 2018.
- [2] S. S. Mansouri, C. Kanellakis, D. Kominiak, and G. Nikolakopoulos, “Deploying MAVs for autonomous navigation in dark underground mine environments,” *Robotics and Autonomous Systems*, 2020.
- [3] C. Sampedro, A. Rodriguez-Ramos, H. Bavle, A. Carrio, P. de la Puente, and P. Campoy, “A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques,” *Journal of Intelligent & Robotic Systems*, pp. 1–27, 2018.
- [4] J. G. Rogers, R. E. Sherrill, A. Schang, S. L. Meadows, E. P. Cox, B. Byrne, D. G. Baran, J. W. Curtis, and K. M. Brink, “Distributed subterranean exploration and mapping with teams of uavs,” in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VIII*, vol. 10190. International Society for Optics and Photonics, 2017, p. 1019017.
- [5] A. Sathyia, P. Sopasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos, “Embedded nonlinear model predictive control for obstacle avoidance using panoc,” in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 1523–1528.
- [6] C. Goerzen, Z. Kong, and B. Mettler, “A survey of motion planning algorithms from the perspective of autonomous uav guidance,” *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, p. 65, 2010.
- [7] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [8] D. Droeßel, M. Nieuwenhuisen, M. Beul, D. Holz, J. Stückler, and S. Behnke, “Multilayered mapping and navigation for autonomous micro aerial vehicles,” *Journal of Field Robotics*, vol. 33, no. 4, pp. 451–475, 2016.
- [9] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeyns, “Autonomous obstacle avoidance and maneuvering on a vision-guided mav using on-board processing,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2472–2477.
- [10] P. P.-Y. Wu, D. Campbell, and T. Merz, “Multi-objective four-dimensional vehicle motion planning in large dynamic environments,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 3, pp. 621–634, 2010.
- [11] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, “Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system (ROS). Springer, 2017, pp. 3–39.
- [12] Elias Small, Pantelis Sopasakis, Emil Fresk, Panagiotis Patrinos, George Nikolakopoulos, “Aerial navigation in obstructed environments with embedded nonlinear model predictive control,” *2019 18th European Control Conference (ECC)*, 2018.
- [13] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, “A simple and efficient algorithm for nonlinear model predictive control,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1939–1944.
- [14] B. Hermans, P. Patrinos, and G. Pipeleers, “A penalty method based approach for autonomous navigation using nonlinear model predictive control,” *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 234 – 240, 2018.
- [15] J. Jackson, G. Ellingson, and T. McLain, “ROSflight: A lightweight, inexpensive MAV research and development tool,” in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 758–762.