**IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)**

A Unified NMPC Scheme for MAVs Navigation with 3D Collision Avoidance under Position Uncertainty

Sina Sharif Mansouri[1], Christoforos Kanellakis[1], Björn Lindqvist[1], Farhad Pourkamali-Anaraki[2], Ali-akbar Agha-mohammadi[3], Joel Burdick[4] and George Nikolakopoulos[1]

Abstract— **This article proposes a novel Nonlinear Model Predictive Control (NMPC) framework for Micro Aerial Vehicle (MAV) autonomous navigation in indoor enclosed environments. The introduced framework allows us to consider the nonlinear dynamics of MAVs, nonlinear geometric constraints, while guarantees real-time performance. Our first contribution is to reveal underlying planes within a 3D point cloud, obtained from a 3D lidar scanner, by designing an efficient subspace clustering method. The second contribution is to incorporate the extracted information into the nonlinear constraints of NMPC for avoiding collisions. Our third contribution focuses on making the controller robust by considering the uncertainty of localization in NMPC using Shannon's entropy to define the weights involved in the optimization process. This strategy enables us to track position or velocity references or none in the event of losing track of position or velocity estimations. As a result, the collision avoidance constraints are defined in the local coordinates of the MAV and it remains active and guarantees collision avoidance, despite localization uncertainties, e.g., position estimation drifts. The efficacy of the suggested framework has been evaluated using various simulations in the Gazebo environment.**

I. Introduction

The deployment of Micro Aerial Vehicles (MAVs) is gaining more attention in different applications for interior or exterior inspections. Examples include infrastructure inspection [1], underground mine tunnel inspection [2], [3], and bridge inspection [4]. The obtained information from the MAV inspections can be used for identifying various types of damages, while providing real-time map-building and navigation in unknown and complex environments.

One of the main challenges in deploying MAVs in real-world applications is the ability to provide collision-free paths, which strongly depend on an accurate and robust localization. A failure in the localization adversely impacts the overall mission quality, and in the worst-case scenario, it leads to a collision/crash. Unfortunately, localization uncertainties are inevitable in Global Positioning System (GPS)-denied environments and in areas that lack prominent visual and geometric features. Hence, there is a need to develop new methods to handle such uncertain cases in perceptually-degraded and extreme environments such as subterranean environments, to provide access to unreachable areas and increase the personnel's overall safety.

This article proposes a new Nonlinear Model Predictive Control (NMPC) architecture for path planning and collision avoidance that considers localization uncertainties as well as geometrically-induced constraints. In the proposed method, a novel and efficient subspace clustering technique [5] will be presented for the transformation of geometric constraints into equivalent plane constraints for modeling the surrounding environment. The proposed approach significantly reduces the computational complexity and memory usage of existing subspace clustering methods. Due to the real-time constraints, traditional subspace clustering methods are not applicable, as the computational time is often greater than one second, even for one thousand data points (e.g., [6]). The extracted equations are in the body frame and are thus decoupled from localization and used in the sequel as nonlinear constraints in NMPC for avoiding collisions. Additionally, to cope with uncertainties in the localization, we assume that the weights of the trajectory following in the NMPC are adaptive and vary based on Shannon's entropy [7] of the measurements. This feature permits the MAV to carry out the mission despite significant uncertainties in localization by dropping one or both of the position or velocity references from tracking objectives. Hence, even a failure in the high level path planner or localization will not lead to a collision of the platform. The proposed control architecture allows for continued mission by MAV even in the presence of localization drift.

A. Background & Motivation

In the related literature, there have been many works that addressed control, navigation and path planning of MAVs [8], [9]. Most of these works decouple the problems of designing controllers, navigation and path planning, and proposed hierarchical structures for autonomous navigation.

Towards the topic of MAV navigation and path planning, exploration algorithms like frontier exploration algorithms [10], entropy based algorithms [11], and information-gain algorithms [12] provide a global planning strategy for MAVs. These methods mainly rely on information concerning the map and localization of the platform and compute regions that reduce the map uncertainty. The obtained desired areas to visit are then fed to the controller to generate com-

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

[1] The authors are with the Robotics Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå SE-97187, Sweden.

[2] The author is with Department of Computer Science, University of Massachusetts Lowell, MA, USA.

[3] The author is with Jet Propulsion Laboratory California Institute of Technology Pasadena, CA, 91109.

[4] The author is with Division of Engineering and Applied Sciences, California Institute of Technology, Pasadena, California, USA.

Corresponding Author's email: sinsha@ltu.se

mands for navigation of MAVs [13], [14]. As localization and mapping suffer from uncertainties and drifts, an additional reactive control layer is often considered for local obstacle avoidance and to prevent collisions with the environment. The most widely used reactive control layer is the artificial potential field [15], which has seen wide use in multiple application areas, as a local path planner for both MAVs and other robotic platform applications, such as the mapping of an infrastructure [16] or in the case of the multi-robot coordination [17] for mobile robots. However, this method suffers from getting stuck in a local optimum and the result would be conservative and not optimal [18]. Few articles have addressed collision avoidance of MAVs with the NMPC framework. Much research focused on addressing formation problems with a fixed number of agents [19], considered a linear model [20], assumed global pose information of obstacles [21], uncertainties of position estimations are excluded.

Plane segmentation has been extensively investigated, and we present a brief overview. These methods can be divided into three categories. (a) Point clustering, based on similarities between the measurements such as distance and angle between surface normal [22]. (b) Region growing, where the method chooses seed points or regions, and cluster the points based on that information. The authors in [23] proposed fast segmentation of 3D point clouds for autonomous vehicles. This method extracts a set of seed points based on low height values to estimate the ground surface and then extract the points close to initial ground plane. (c) Random sample consensus (RANSAC)-based plane fitting where points from the point cloud are sampled, and planar models are fitted to them. In [24], a RANSAC method was proposed for performing normal coherence check on points and removed the data points whose normal directions were contradictory to the fitted plane. While plane detection has received significant attention over the last years, most research relies on organized point clouds, such as RGB-D images [25], where the neighbor information can be used. However, extracting planes from unorganized point clouds is more challenging because of the cloud size variations, which means that the neighbor information cannot be immediately used [26].

In machine learning, sparse subspace clustering [27] is the state-of-the art method for segmenting points drawn from a union of subspaces. These methods consist of two steps: 1) solving the sparse representation problem to find a similarity matrix, 2) employing spectral clustering to partition the data. Previous research efforts focused solely on reducing the cost associated with step 1. Unfortunately, existing methods did not attempt to reduce the cost of spectral clustering. Therefore, state-of-the-art subspace clustering methods rely on the increase in computational power to perform step 2, which is a bottleneck for real-time data processing.

B. Contributions

The first contribution is to design a scalable subspace clustering technique that finds clusters in the form of subspaces within a 3D point cloud. We develop a scalable method for partitioning the input data into clusters using three tools: (1) randomized sampling, (2) fast sparse representation solvers, and (3) efficient methods for computing the eigenvalue decomposition. Thus, the proposed method enables extracting the required information for avoiding collisions.

The second contribution stems from coupling the 3D collision avoidance and controller in the body frame. The obtained plane equations in local coordinates are considered as non-linear constraints in the optimization scheme, while the controller tracks the desired trajectory in global coordinates. The proposed NMPC is solved by Optimization Engine (OpEn) [28] that uses the Proximal Averaged Newton-type method for Optimal Control (PANOC) algorithm [29], while a penalty method is used for enforcing equality constraints that guarantees obstacle avoidance.

The third contribution enables the framework to handle localization uncertainties by defining adaptive weights for tracking the position and velocity reference way-points. The weights are calculated based on uncertainties associated with measurements, while drifts in localization estimations result in position or velocity tracking or neither. The constraints are defined based on the local point cloud and will be active in all cases, and collision avoidance is guaranteed. The proposed solution results in progressing the MAV navigation, instead of terminating the mission or collision of the platform.

The final contribution is to thoroughly examine the performance of the proposed method in corridor and confined environments using Gazebo. The proposed method successfully avoids collisions, even when the localization estimations are uncertain. These results show the capability of the proposed architecture in challenging scenarios and can be found in the following link: `https://youtu.be/76ob9HSrOAs`

C. Outline

Section II introduces notations used in the article. Then, the segmentation approach is presented in Section III. In the sequel, a presentation of the NMPC formulation, and the solver are described in Section IV. Section V presents our simulation results in the Gazebo environment. Finally, Sections VI and VII conclude the article by summarizing the findings and offering some directions for future research.

II. Notation and Preliminaries

The empty set in $\mathbb{R}^n$ is denoted by $\emptyset_n$. A vector in $\mathbb{R}^n$ is predetermined as column vector in $\mathbb{R}^{n\times 1}$. The scalar product between two vectors $\boldsymbol{r}, \boldsymbol{t} \in \mathbb{R}^n$ is denoted by $\boldsymbol{r} \cdot \boldsymbol{t}$. The identity matrix in $\mathbb{R}^{n\times n}$ is denoted by $\mathbf{I}_n$. The transpose of a matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ is denoted by $\mathbf{M}^\top$. The set $\mathcal{P} = \{(x_i, y_i, z_i), i \in \mathbb{N}\}$ presents the set of point clouds in $\mathbb{R}^3$. The set $\mathcal{S} = \{(\alpha_i, \beta_i, \gamma_i, \zeta_i), i \in \mathbb{N}\}$ denotes the set of plane equation in the form of $\alpha x+\beta y+\gamma z+\zeta = 0$, $\alpha, \beta, \gamma, \zeta \in \mathbb{R}$. $\boldsymbol{x}$, $\boldsymbol{u}$ are called *state* and *input* vectors respectively. $\boldsymbol{p} = [p_x, p_y, p_z]^\top \in \mathbb{R}^3$ is the position and $\boldsymbol{v} = [v_x, v_y, v_z]^\top \in \mathbb{R}^3$ is the vector of linear velocities, $\phi \in \mathbb{R} \cap [-\pi, \pi]$ and $\theta \in \mathbb{R} \cap [-\pi, \pi]$ are the roll and pitch angles. Figure 1 depicts the block diagram of the proposed structure with the high level NMPC controller, and the low level controller with the MAV in the loop. The set $\mathcal{S}$ is provided from the plane segmentation module (Section III) and the NMPC module (Section IV) generates control actions $\boldsymbol{u}$ for navigating to the reference way point $\boldsymbol{x}_r$, based on the odometry uncertainty $\boldsymbol{\sigma}$, the estimated states $\hat{\boldsymbol{x}}$, and the plane equations.

III. Segmentation of 3D Point Cloud

In this section, we propose an efficient subspace clustering technique to extract planes or surfaces that are presented in a point cloud. Suppose that the obtained 3D point cloud consists of n points, i.e., $\mathcal{P} = \{(x_i, y_i, z_i)\}_{i=1}^n$, then we will form a data matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{3\times n}$ by assuming each data point as a column vector. As these points are drawn from a union of subspaces in $\mathbb{R}^3$, we can express each point as a sparse linear combination of all the other data points.

Subspace clustering techniques aim to partition the data into multiple clusters and fit each cluster with a low-dimensional subspace (e.g., a 2D surface). A popular approach to subspace clustering relies on solving the following optimization problem for each point $\mathbf{b}_j$, $j = 1, \ldots, n$:

$$\min_{\mathbf{c}_j \in \mathbb{R}^n} \|\mathbf{c}_j\|_1 + \frac{\lambda}{2}\|\mathbf{b}_j - \sum_{i \neq j} c_{ij}\mathbf{b}_i\|_2^2 \;\text{ s.t. }\; \mathbf{c}_j^T \mathbf{1}_n = 1, \quad (1)$$

where $\|\cdot\|_q$ represents the ℓ_q norm for vectors, λ is a regularization parameter, $\mathbf{c}_j = [c_{1j}, \ldots, c_{nj}]^T$ is the coefficient vector associated with $\mathbf{b}_j$, and $\mathbf{1}_n$ is the vector of all ones. The constraint $i \neq j$ avoids the trivial solution of expressing $\mathbf{b}_j$ via itself, and the constraint $\mathbf{c}_j^T\mathbf{1}_n = 1$ allows us to extract the general case of affine subspaces.

The above optimization problem can be cast in a more concise form for the entire data set, i.e., $\mathbf{b}_j$, $j = 1, \ldots, n$:

$$\min_{\mathbf{C} \in \mathbb{R}^{n\times n}} \|\mathbf{C}\|_1 + \frac{\lambda}{2}\|\mathbf{B} - \mathbf{BC}\|_F^2 \quad (2a)$$

$$\text{s.t. } \operatorname{diag}(\mathbf{C}) = \mathbf{0}_n, \;\; \mathbf{C}^T\mathbf{1}_n = \mathbf{1}_n, \quad (2b)$$

where $\|\mathbf{C}\|_1 = \sum_{i,j}|c_{ij}|$, $\|\mathbf{C}\|_F^2 = \sum_{i,j} c_{ij}^2$, and diag($\mathbf{C}$) is the vector of the diagonal elements of $\mathbf{C}$. After solving the above optimization problem and finding the coefficient matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_n]$, the next step is to find the segmentation of the data into multiple subspaces. To this end, we build a weighted graph with n nodes corresponding to the n original data points, and the similarity matrix is defined as $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$. Finally, we apply spectral clustering to $\mathbf{W}$ for partitioning the data into subspaces. To be formal, we form the normalized graph Laplacian matrix $\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal degree matrix. Then, the K eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_K \in \mathbb{R}^n$ corresponding to the K smallest eigenvalues of $\mathbf{L}$ are found (K represents the number of clusters). The last step of spectral clustering is to perform *K-means* clustering on the rows of the matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_K] \in \mathbb{R}^{n\times K}$ to find the segmentation of the original data points. The optimization problem in (2) can be solved using the Alternating Direction Method of Multipliers (ADMM) [30], which scales poorly with the data size. The computational cost of existing implementations is cubic or quadratic in terms of the number of data points. Moreover, one has to form various n-dimensional square matrices (e.g., $\mathbf{C}$ and $\mathbf{W}$) that will lead to high memory usage. Hence, existing subspace clustering methods are not appropriate for our problem of interest. We address these problems by developing a scalable subspace clustering technique.

The high computational cost of constructing the coefficient matrix originates from computing a regularized representation of every single data point with respect to the whole dataset. Thus, the first step of our novel method is to form two subsets of the original data by uniform sampling without replacement. Furthermore, we eliminate the constraint $\mathbf{C}^T\mathbf{1}_n = \mathbf{1}_n$ by mapping the original data from $\mathbb{R}^3$ to $\mathbb{R}^4$. This trick is known as homogeneous embedding [31], where we add a new coordinate which is 1 (or another constant) to every column of the matrix $\mathbf{B}$, i.e., $\mathcal{P} = \{(x_i, y_i, z_i, 1)\}_{i=1}^n$.

Given two sampling parameters $0 < \kappa_1 < \kappa_2 < 0.5$, we create two sets of indices $\mathcal{I}_1$ and $\mathcal{I}_2$ with $n_1 = \lfloor \kappa_1 n \rfloor$ and $n_2 = \lfloor \kappa_2 n \rfloor$ elements from $\{1, \ldots, n\}$ selected uniformly at random, respectively. Then, we modify the sparse representation problem for each $\mathbf{b}_j$, $j \in \mathcal{I}_2$: $\min_{\mathbf{c}_j \in \mathbb{R}^{n_1}} \|\mathbf{c}_j\|_1 + \frac{\lambda}{2}\|\mathbf{b}_j - \sum_{i\in\mathcal{I}_1} c_{ij}\mathbf{b}_i\|_2^2$. This problem does not return a trivial solution because $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$, and we propose to use the SPArse Modeling Softwares (SPAMSs) package [32].

After solving the new optimization problem, we should apply spectral clustering to the obtained coefficient matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_{n_2}]$. However, the matrix $\mathbf{C}$ is not square anymore because $n_1 \neq n_2$, and we often want n_1 to be much smaller than n_2 to reduce the computational cost.

State-of-the-art techniques seek to build a square matrix of size $n_2 \times n_2$, which can be quite expensive. To tackle this problem, we *implicitly* form the similarity matrix $\mathbf{W} = \widetilde{\mathbf{C}}^T\widetilde{\mathbf{C}} \in \mathbb{R}^{n_2\times n_2}$, where $\widetilde{\mathbf{C}} = |\mathbf{C}|$. Next, we present an efficient approach to perform spectral clustering using the new similarity matrix. We compute i-th element of $\mathbf{D}$ as:

$$\sum_{j=1}^{n_2} w_{ij} = \sum_{j=1}^{n_2} \widetilde{\mathbf{c}}_i^T\widetilde{\mathbf{c}}_j = \widetilde{\mathbf{c}}_i^T\boldsymbol{\eta} = \widetilde{\mathbf{c}}_i \cdot \boldsymbol{\eta}, \quad (3)$$

where $\boldsymbol{\eta} = \sum_{j=1}^{n_2}\widetilde{\mathbf{c}}_j \in \mathbb{R}^{n_1}$. Thus, we compute the diagonal degree matrix $\mathbf{D}$ using n_2 scalar products.

The remaining task is to compute the K smallest eigenvectors of the graph Laplacian matrix. We can reduce the computational cost and memory usage of this step by computing the top K eigenvectors of $\mathbf{I}_n - \mathbf{L} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$. Let $\mathbf{U\Sigma P}^T$ be the singular value decomposition of $\widetilde{\mathbf{C}}\mathbf{D}^{-1/2} \in \mathbb{R}^{n_1\times n_2}$, where $\mathbf{U} \in \mathbb{R}^{n_1\times r}$ (left singular vectors), $\mathbf{P} \in \mathbb{R}^{n_2\times r}$ (right singular vectors), $\mathbf{\Sigma}$ contains the singular values, and r is the rank parameter. Then, the top K eigenvectors of $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ are equivalent to the top K right singular vectors of $\widetilde{\mathbf{C}}\mathbf{D}^{-1/2}$ since we have [33]:

$$\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} = (\mathbf{U\Sigma P}^T)^T(\mathbf{U\Sigma P}^T) = \mathbf{P\Sigma}^2\mathbf{P}^T. \quad (4)$$

After computing the top eigenvectors and performing *K-means* clustering, we obtain the segmentation of the data. The final step picks three points within each subspace to find the equation of each subspace for collision avoidance.

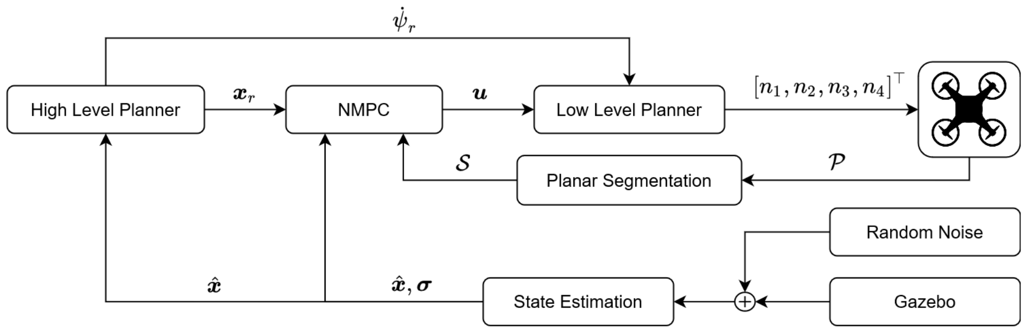

Fig. 1: Control scheme of the proposed NMPC module, where way point and yaw rate references are provided from the high level controller, NMPC generates thrust and attitude commands and the low level controller generates motor commands.

IV. Nonlinear Model Predictive Control

A. Objective Function

We develop the NMPC with 3D collision avoidance constraints, and we solve it by PANOC [29] to guarantee real-time performance. The objective of NMPC is to track the reference trajectory $\boldsymbol{x} = [p, v, \phi, \theta]^\top$ from high level planner or operator and to generate thrust T and attitude commands ϕ_d, θ_d for a low level controller, while guaranteeing safety distance from all extracted planes. Based on $\boldsymbol{u} = [T, \phi_d, \theta_d]^\top$, and reference yaw rate command $\dot{\psi}_r$ from a high level planner, the low level controller generates motor commands $[n_1, \ldots, n_4]^\top$ for the MAV.

The states of the non-linear dynamics of the MAV based on [34] can be presented as, $\boldsymbol{x} = [p_x, p_y, p_z, v_x, v_y, v_z, \phi, \theta]^\top$, $\hat{\boldsymbol{x}} = [\hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{v}_x, \hat{v}_y, \hat{v}_z, \hat{\phi}, \hat{\theta}]^\top$ is the estimated state from Extended Kalman Filter (EKF) for MAV dynamics, and the control input is: $\boldsymbol{u} = [T, \phi_d, \theta_d]^\top$. The discrete-time dynamical system is obtained by Euler method and with a sampling time of T_s as $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$.

In the NMPC approach, a finite-horizon problem with prediction horizon N is solved at every time instant k. The states and control actions $k + j$ steps ahead of the current time step k are expressed by $\boldsymbol{x}_{k+j|k}$, and $\boldsymbol{u}_{k+j|k}$ respectively. At each time step, NMPC generates an optimal sequence of control actions $\boldsymbol{u}^\star_{k|k}, \ldots, \boldsymbol{u}^\star_{k+N-1|k}$, and the first control action $\boldsymbol{u}^\star_{k|k}$ is applied to the flight controller using a zero-order hold element, that is, $\boldsymbol{u}(t) = \boldsymbol{u}^\star_{k|k}$ for $t \in [kT_s, (k+1)T_s]$. For the proposed NMPC, the following finite horizon cost function is introduced:

$$J = \sum_{j=0}^{N-1} \underbrace{\|\boldsymbol{x}_{k+j+1|k} - \boldsymbol{x}_r\|^2_{\boldsymbol{Q}_x}}_{\text{way point error}} + \underbrace{\|\boldsymbol{u}_{k+j+1|k} - \boldsymbol{u}_r\|^2_{\boldsymbol{Q}_u}}_{\text{actuation}} + \underbrace{\|\boldsymbol{u}_{k+j|k} - \boldsymbol{u}_{k+j-1|k}\|^2_{\boldsymbol{Q}_{\Delta u}}}_{\text{smoothness cost}}. \quad (5)$$

The first term of J gives the tracking of the reference way points by penalizing a deviation from $\boldsymbol{x}_r$. The second term is the hovering term, where $\boldsymbol{u}_{ref}$ is $[g,\, 0,\, 0]^\top$, which is the hover thrust with horizontal angles. The third term penalizes the aggressiveness of the obtained control actions. Additionally, $\boldsymbol{Q}_x \in \mathbb{R}^{8\times 8}$, $\boldsymbol{Q}_u \in \mathbb{R}^{3\times 3}$, $\boldsymbol{Q}_{\Delta u} \in \mathbb{R}^{3\times 3}$ are the weights for each term of the objective function, which reflects the relative importance of each term. To consider uncertainties in the localization estimation, the term $\boldsymbol{Q}_x$ is a diagonal matrix and elements are updated based on the Shannon entropy of the measurements. We consider set of measurements variances as:

$$\boldsymbol{\sigma} = \{(\sigma_i^{p_x}, \sigma_i^{p_y}, \sigma_i^{p_z}, \sigma_i^{v_x}, \sigma_i^{v_y}, \sigma_i^{v_z}), \quad i \in \{k - (n_{max} - 1), \ldots, k\}\}, \quad (6)$$

where σ is the variance of measurements and n_{max} is limited window size of stored previous variances. We compute the Shannon entropy for each set of variance at time instant k by first defining probabilities $P_i = \sigma_i / (\sum_{i=1}^{n_{max}} \sigma_i)$. Next, we compute the Shannon entropy, i.e., $H = -\sum_{i=1}^{n_{max}} P_i \log(P_i)$. Then, the obtained H will be used as diagonal terms of $\boldsymbol{Q}_x$ as $\boldsymbol{Q}_x = diag(H^{p_x}, H^{p_y}, H^{p_z}, H^{v_x}, H^{v_y}, H^{v_z})$

B. Obstacle Definition and Constraints

1) Collision Avoidance Constraint: The set of extracted planes are presented by:

$$\mathcal{S} = \{(\alpha_i, \beta_i, \gamma_i, \zeta_i), i = 1, \ldots, n_{cluster}\}, \quad (7)$$

where $n_{cluster}$ is the number of clusters and α, β, γ, ζ are the plane coefficients. To avoid the obstacles, for each extracted plane equation, the constraints are defined in (8). It should be highlighted that the constraints are defined in the body frame $(\boldsymbol{p}_{k+j|k} - \boldsymbol{p}_{k|k})$ and the MAV is considered in the center of the point cloud so that the global position is not required.

$$d_s \leq \frac{|[\alpha_i, \beta_i, \gamma_i] \cdot [\boldsymbol{p}_{k+j|k} - \boldsymbol{p}_{k|k}]^\top + \zeta_i|}{\sqrt{\alpha_i^2 + \beta_i^2 + \gamma_i^2}}, \quad (8)$$

for $i = 0, \ldots, n_{cluster}$ and $j = 0, \ldots, N-1$. The proposed constraints guarantee that the MAV has at least d_s distance to each plane.

2) Input Constraint: To prevent the aggressive behavior of control actions, the following input constraints on the successive differences of control actions are defined as:

$$|\phi_{d,k+j|k} - \phi_{d,k+j+1|k}| \leq \Delta\phi_{\max}, \quad (9a)$$

$$|\theta_{d,k+j|k} - \theta_{d,k+j+1|k}| \leq \Delta\theta_{\max}, \quad (9b)$$

for $j = 0, \dots, N-1$. Where $\Delta\phi_{\max}$ and $\Delta\theta_{\max}$ are bounds for changes.

C. Embedded Optimization

The following optimization problem can be defined:

$$\underset{\{\boldsymbol{u}_{k+j|k}\}_{j=0}^{N-1}}{\textbf{minimize}} \; J \tag{10a}$$

$$\textbf{subject to}: \boldsymbol{x}_{k+j+1|k} = f(\boldsymbol{x}_{k+j|k}, \boldsymbol{u}_{k+j|k}), \tag{10b}$$

$$\text{Constraints (8), (9),} \tag{10c}$$

$$\boldsymbol{u}_{k+j|k} \in [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}], \tag{10d}$$

for $j = 0, \dots, N-1$.

Problem (10) is a parametric non-convex optimization problem that must be solved online. We use the fast optimization solver Optimization Engine (for short `OpEn`) developed by [28]. We first eliminate the state sequence following the procedure detailed in [29]. Hence, we have a problem of the decision variable $\boldsymbol{u} = (\boldsymbol{u}_{k|k}, \boldsymbol{u}_{k+1|k}, \dots, \boldsymbol{u}_{k+N-1|k}) \in \mathbb{R}^{3N}$, which is constrained in $\boldsymbol{U} = [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]^N$. For the i^{th} plane equation, constraint (8) reduces to:

$$\max\left\{0, d_s - \frac{|[\alpha_i, \beta_i, \gamma_i] \cdot [\boldsymbol{p}_{k+j|k} - \boldsymbol{p}_{k|k}]^\top + \zeta_i|}{\sqrt{\alpha_i^2 + \beta_i^2 + \gamma_i^2}}\right\} = 0. \tag{11}$$

Likewise, constraint (9) can be written as an equality constraint for ϕ and similarly for θ as follow:

$$\max\left\{0, \phi_{d,k+j|k} - \phi_{d,k+j+1|k} - \Delta\phi_{\max}\right\} = 0, \tag{12a}$$

$$\max\left\{0, \phi_{d,k+j+1|k} - \phi_{d,k+j|k} - \Delta\phi_{\max}\right\} = 0. \tag{12b}$$

V. Results

A. Simulation Setup

The Gazebo robot simulator is used to evaluate the performance of the overall system architecture. The quad-copter in the simulations is equipped with the 3D lidar Velodyne VLP-16. During the simulations the ground truth odometry information provided from the Gazebo environment is used without considering any on-board sensor fusion. For evaluating the adaptive weights performance of the controller, random noises are added to the odometry measurements. We generate noise with the normal distribution $\mathcal{N}(\mu, \sigma^2)$ [35], where μ and σ^2 are the mean and standard deviation for each term of the states. From the practical point of view, position estimations suffer from higher uncertainties compared to velocity estimations [36], since position drift is more difficult to recover compared to velocity drifts that can recover after few time steps. Thus, we consider higher standard deviation of noise for position estimations. The normal distributions, for the position and velocity estimation, are $\mathcal{N}(0, 1.5)$ and $\mathcal{N}(0, 0.5)$, respectively.

Two environments are chosen to demonstrate the main attributes of the proposed navigation framework. The first environment is inspired from corridor areas, where the goal is to navigate from one end to the other, while avoiding collisions with the walls, the second case mainly focuses on the effect of odometry uncertainties in confined environments and demonstrates the performance when adaptively updating the uncertainty weights in the position and velocity in the controller. In all simulations, the term ψ for the MAV is set to be zero because plane segmentation and NMPC are in the body frame of MAV and independent of the yaw. The simulations can be found in the link: `https://youtu.be/76ob9HSrOAs`

The parameters of the MAV model are identical to the nonlinear model in [2]. Moreover, the tuning parameters of NMPC are $\boldsymbol{Q}_{\Delta u} = [20,\, 20,\, 20]^\top$, $\boldsymbol{Q}_u = [10,\, 10,\, 10]^\top$, $T = [0, 1]$, and ϕ_d, $\theta_d = [-0.4, 0.4]$ rad. The NMPC prediction horizon N is 40, the control sampling frequency T_s is $20\,\text{hz}$, d_s sets to $1\,\text{m}$, n_{max} and $n_{cluster}$ are 10, and $\Delta\phi_{max}$ and $\Delta\theta_{max}$ are $0.05\,\text{rad/sec}$. Moreover, the mean and maximum computation time of the proposed NMPC, in all the studied scenarios, are $1.7\,\text{msec}$ and $8.8\,\text{msec}$ respectively. Finally, for the plane segmentation κ_1 is 0.1, κ_2 is 0.2, and λ sets to 0.15, while the mean and maximum running time of the proposed segmentation method are $0.05\,\text{sec}$ and $0.4\,\text{msec}$.

B. Simulation Evaluation

1) Plane Segmentation: Figure 2 depicts the results of plane segmentation for different complex environments. The point clouds are in body coordinates, thus the MAV is located in the center. We also compared the running time of our method with Sparse Subspace Clustering Orthogonal Matching Pursuit (SSC-OMP) [6], Sparse Subspace Clustering Alternating Direction Method of Multipliers (SSC-ADMM) [27], and Elastic Net Subspace Clustering (ENSC) [37]. As depicted in Figure 3, our approach reduces the computational cost by at least an order of magnitude.

2) Navigation in Corridor Environment: Figure 4 depicts the trajectory of the MAV with the collision avoidance constraints of the NMPC in the "house_maze" world, a Gazebo environment available in VoxBlox repository[1]. The way-points are generated beforehand with different velocities, where some way-points have been intentionally selected close to the wall to demonstrate the collision avoidance capabilities of the framework in cases where the planner designs a trajectory prone to collisions with the environment. This simulation does not include uncertainty linked with localization measurements. The proposed NMPC controller avoids collisions for all velocities [0.3, 0.5, 0.8, 1.0, 1.2] m/sec. Additionally, the potential field method [16] as a baseline local collision avoidance is evaluated with way-point generated with 0.3 m/sec, while the same desired safety distance is set. The potential field trajectory is oscillating in the environment and it failed to reach the final goal. This is due to repulsive forces and large number of point clouds, which push the MAV from one side to another side.

The Mean Absolute Error (MAE) between the desired trajectory and way-points are 0.16, 0,16, 0.22, 0.27, 0.36, and 1.4 m for our proposed method with velocities of 0.3, 0.5, 0.8, 1.0, 1.2 m/sec and potential field with velocity of 0.3m/sec respectively. It should be highlighted that the MAE cannot be zero as some points are violating the constraints

[1] `https://github.com/ethz-asl/voxblox`

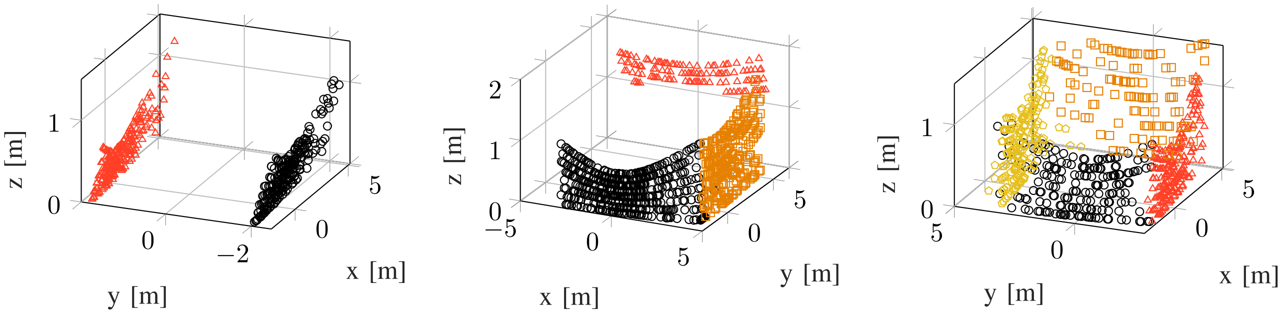

Fig. 2: The extracted surfaces from various 3D point clouds in different scenarios based on the proposed segmentation method, each cluster is indicated with a different color. Best viewed in color.

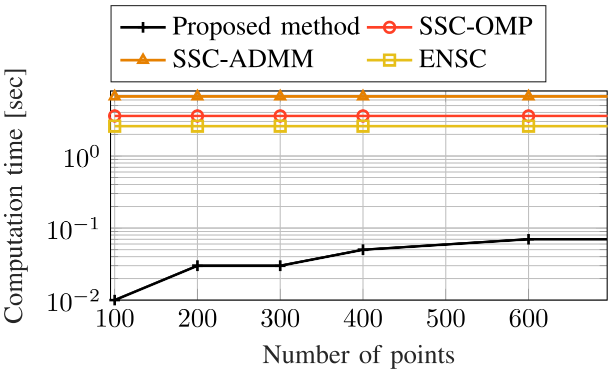

Fig. 3: Comparing running time of segmentation methods.

and the controller must avoid reaching them. It is observed that the MAE slightly increases with the increase in the generated way-point velocities. However, the increase in velocities did not result in any collision. The length of the trajectory for each velocity is 54.60, 50.60, 49.41, 49.68, 49.79 m respectively. We see that the higher velocity results in slightly lower length of the path, however the MAE increases too. Nonetheless, this article is not focused on finding optimal path, and further investigation is required. Finally, the MAE of velocity profile of the MAV are 0.04, 0.03, 0.02, 0.12, 0.25 m/sec for way-points of [0.3, 0.5, 0.8, 1.0, 1.2] m/sec velocities respectively. It is observed that the MAE is increased by increasing way-points velocities. However, it should be highlighted that in order to guarantee avoiding collision in confined environment, reaching higher speeds is not always a feasible solution.

Figure 5 shows the minimum distance to the 3D point cloud during navigation without uncertainty in localization based on the proposed method with different velocities and the potential field method. It should be highlighted that the distance between confined areas of the environment is less than 2 m, thus the constraints of $d_s = 1\,\mathrm{m}$ is slightly violated with 0.1 m. However, this does not have impact on overall performance and in all velocities we avoid collisions.

3) Navigation under Localization Uncertainties in Confined Environment: A confined environment with no entry/exit is chosen to evaluate the performance when uncertainties are induced in the position and velocity measurements in the x and y axis, while the reference way-point sets to $[0, 0, 1, 0, 0, 0]^\top$. The noise in the measurements is induced after the take-off, when the MAV reaches the desired way-point. The focus of these simulations is to highlight the performance of the framework when the localization is noisy, which adaptively changes the weights in the position and the velocity based on the noise levels. Two scenarios are considered one with and one without adaptive weights for NMPC, while in both cases the collision avoidance constraints are active. Figure 6 depicts the trajectory of the MAV in each scenario. This Figure shows that the MAV for the majority of the simulation run hovers close to the desired location when the adaptive weights are enabled, compared to the other case where it oscillates more. Figure 7 shows the value of ground truth and measurements with noise for position in x axis as an example, while the changes in the weights for tracking of x is depicted too. It is observed that NMPC with adaptive weights has less oscillation, and NMPC without adaptive weights fails and collides to the walls. Moreover, Figure 8 shows the distance of each controller during the simulation, as it can be seen the adaptive weights of NMPC improve the collision avoidance performance.

VI. Limitations and Future Improvements

The proposed efficient planar segmentation method shows satisfactory performance in simulation results. However, false connections in the similarity graph constructed from sparse representations may lead to imprecise assignments, which is problematic for extracting planes from 3D point clouds. Additionally, the proposed method is mainly used for plane segmentation, and other shapes such as cylinders are not considered. Although the proposed approach is suitable for many robotic applications such as Simultaneous Localization and Mapping (SLAM) and wall following, there is a need for more general object extraction techniques for local collision avoidance of MAV with NMPC. Another future research direction is to analyze the effect of noisy point clouds on the accuracy of our plane segmentation method.

VII. Conclusion

This article proposes a framework for autonomous navigation of MAVs in various environments with position

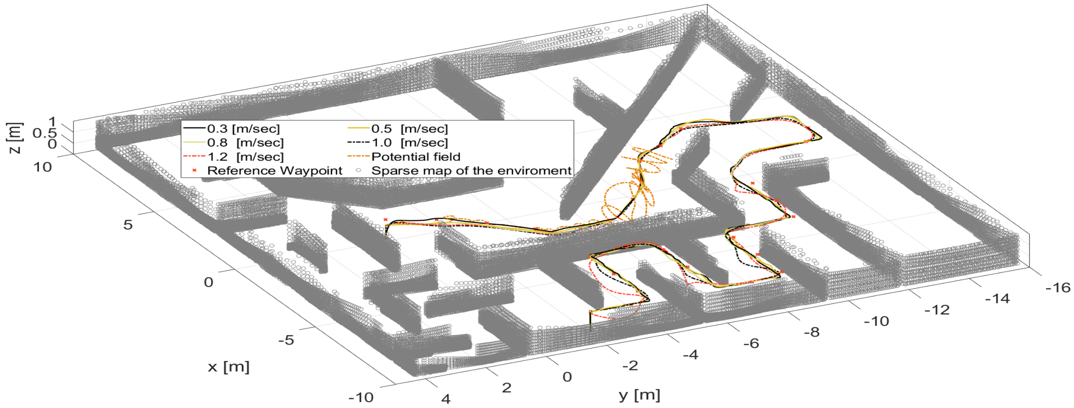

Fig. 4: The trajectory of the MAV with different velocities during navigation in the corridor environment, while the way-points are shown by cross marker.

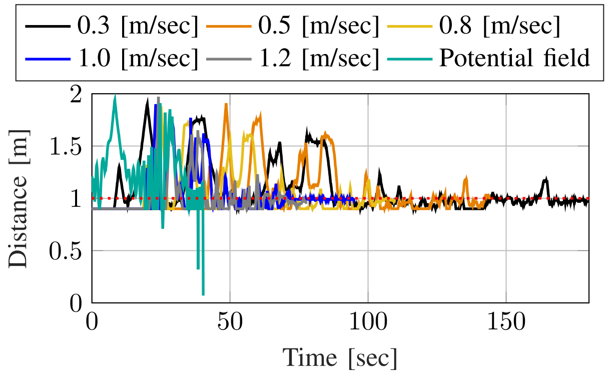

Fig. 5: The minimum distance during navigation without localization uncertainties to the points in 3D point cloud, while the reference trajectory generated close to the obstacles.

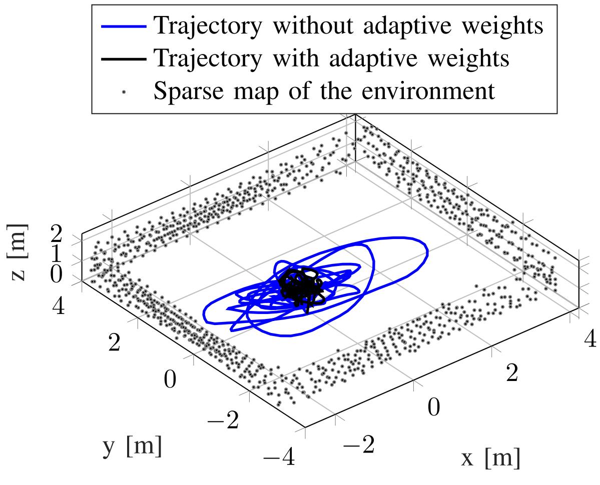

Fig. 6: The trajectory of the MAV during navigation in the confined environment, while the way-points sets to $[0, 0, 1, 0, 0, 0]^\top$ and uncertainty in the localization is added.

uncertainties. The framework consists of two main modules, where the first one is plane segmentation from a local 3D point cloud based on a highly efficient sparse subspace clustering technique. The second module is NMPC with

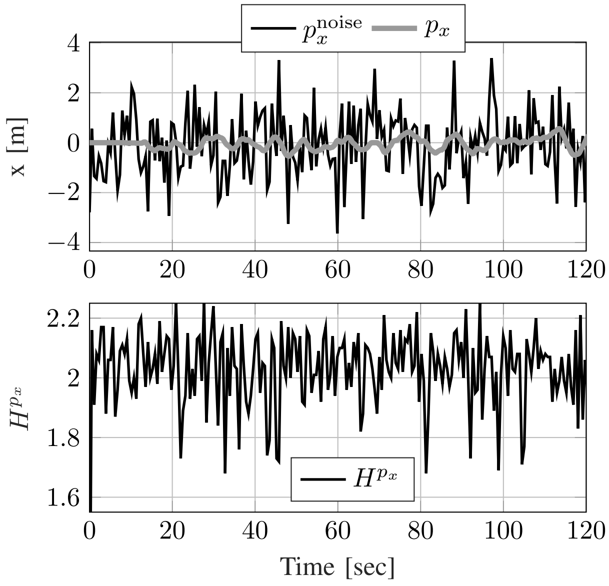

Fig. 7: The real value and noisy measurement for position in x axis and adaptive weight of x tracking in the controller, the results are down sampled for better visualization.

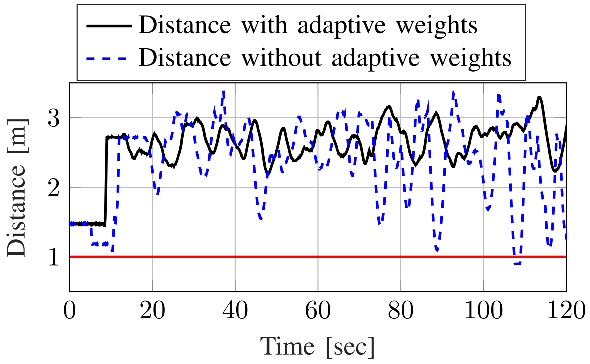

Fig. 8: The minimum distance during navigation with localization uncertainties to the points in 3D point cloud.

collision avoidance based on plane equations and adaptive weights for tracking position, velocity or none based on localization uncertainties. The overall framework is evaluated in the Gazebo environment and the obtained results show the efficiency of the proposed methods to provide a collision-free navigation. It is shown that considering the localization uncertainties in controller results in robust maneuver, while without having adaptive weights, crushing of the MAV or larger drifts from desired trajectory are unavoidable.

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*, p. 103472, 2020.

[3] S. S. Mansouri, C. Kanellakis, E. Fresk, B. Lindqvist, D. Kominiak, A. Koval, P. Sopasakis, and G. Nikolakopoulos, "Subterranean MAV navigation based on nonlinear MPC with collision avoidance constraints," *IFAC-PapersOnLine*, 2020.

[4] N. Hallermann and G. Morgenthal, "Visual inspection strategies for large bridges using unmanned aerial vehicles (uav)," in *Proc. of 7th IABMAS, International Conference on Bridge Maintenance, Safety and Management*, 2014, pp. 661–667.

[5] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *Acm Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[6] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3918–3927.

[7] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[8] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[9] H. Mo and G. Farid, "Nonlinear and adaptive intelligent control techniques for quadrotor uav–a survey," *Asian Journal of Control*, vol. 21, no. 2, pp. 989–1008, 2019.

[10] J. A. P. Fentanes, R. F. Alonso, E. Zalama, and J. G. García-Bermejo, "A new method for efficient three-dimensional reconstruction of outdoor environments using mobile robots," *Journal of Field Robotics*, vol. 28, no. 6, pp. 832–853, 2011.

[11] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[12] S. Bhattacharya, N. Michael, and V. Kumar, "Distributed coverage and exploration in unknown non-convex environments," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 61–75.

[13] H. Carrillo, P. Dames, V. Kumar, and J. A. Castellanos, "Autonomous robotic exploration using occupancy grid maps and graph slam based on shannon and rényi entropy," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 487–494.

[14] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon next-best-view planner for 3d exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.

[15] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE transactions on systems, man, and cybernetics*, vol. 22, no. 2, pp. 224–241, 1992.

[16] D. Droeschel, 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.

[17] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1217–1222.

[18] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[19] M. Ille and T. Namerikawa, "Collision avoidance between multi-uav-systems considering formation control using mpc," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2017, pp. 651–656.

[20] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6753–6760.

[21] E. Small, P. Sopasakis, E. Fresk, P. Patrinos, and G. Nikolakopoulos, "Aerial navigation in obstructed environments with embedded nonlinear model predictive control," in *2019 ECC*. IEEE, 2019.

[22] B. Sun and P. Mordohai, "Oriented point sampling for plane detection in unorganized point clouds," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2917–2923.

[23] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5067–5073.

[24] X. Qian and C. Ye, "Ncc-ransac: a fast plane extraction method for 3-d range data segmentation," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2771–2783, 2014.

[25] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6218–6225.

[26] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, "An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells," *Remote Sensing*, vol. 9, no. 5, p. 433, 2017.

[27] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[28] P. Sopasakis, E. Fresk, and P. Patrinos, "Optimization Engine," 2019. [Online]. Available: http://doc.optimization-engine.xyz/

[29] A. S. Sathya, P. Sopasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos, "Embedded nonlinear model predictive control for obstacle avoidance using panoc," in *Proceedings of the 2018 European Control Conference*, 2018.

[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[31] C. Li, C. You, and R. Vidal, "On geometric analysis of affine sparse subspace clustering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1520–1533, 2018.

[32] J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing," *Foundations and Trends in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014.

[33] F. Pourkamali-Anaraki, "Large-scale sparse subspace clustering using landmarks," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019, pp. 1–6.

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

[35] P. Z. Peebles, *Probability, random variables, and random signal principles*. McGraw-Hill New York, 2001, vol. 3.

[36] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[37] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3928–3937.