Data-Driven Predictive Planning and Control for Aerial 3D Inspection with Back-face Elimination

Savvas Papaioannou, Panayiotis Kolios, Christos G. Panayiotou and Marios M. Polycarpou

Abstract— Automated inspection with Unmanned Aerial Systems (UASs) is a transformative capability set to revolutionize various application domains. However, this task is inherently complex, as it demands the seamless integration of perception, planning, and control which existing approaches often treat separately. Moreover, it requires accurate long-horizon planning to predict action sequences, in contrast to many current techniques, which tend to be myopic. To overcome these limitations, we propose a 3D inspection approach that unifies perception, planning, and control within a single data-driven predictive control framework. Unlike traditional methods that rely on known UAS dynamic models, our approach requires only input-output data, making it easily applicable to off-the-shelf black-box UASs. Our method incorporates back-face elimination, a visibility determination technique from 3D computer graphics, directly into the control loop, thereby enabling the online generation of accurate, long-horizon 3D inspection trajectories.

I. INTRODUCTION

Unmanned Aerial Systems (UAS) are increasingly employed across a range of application domains, such as reconnaissance tasks [1], [2], search-and-rescue operations [3]–[5], and inspection missions [6], [7]. Despite progress in automation and aviation, fully autonomous UASs with integrated planning and control remain in their infancy. Automated robotic inspection tasks require precise navigation, efficient trajectory planning, and accurate path execution. Long-horizon planning is critical for predicting future actions and optimizing cumulative rewards. However, many existing approaches rely on short-horizon planning in simplified 2D environments, neglecting the complexity of 3D settings [8], [9]. Conventional methods assume known UAS dynamics, either through explicit modeling [10], [11] or learning-based approaches [12], [13]. Model-based techniques often lead to intractable non-linear optimization problems, while learning-based methods demand large datasets and retraining when applied to new platforms. Recent efforts favor data-driven approaches [14], [15], which leverage past input-output data without requiring explicit models.

Moreover, automated inspection is inherently a challenging task because it requires tight integration of perception,

planning, control, whereas in many other planning problems these tasks are often handled separately [16]–[20]. For instance, [16], [17] address power-line inspection and trajectory following, respectively, assuming pre-defined reference trajectories, effectively treating the planning problem as solved. In [18], planning is decoupled into path searching and motion control, while the works in [19], [20] treat perception, planning, and control as separate modules for UAV navigation and coverage. In contrast, automated inspection demands the joint solution of planning and control under perception-aware constraints. The UAS must dynamically determine the inspection sequence, timing, and reference trajectory to cover multiple surface points, significantly increasing problem complexity.

To address these challenges, this work formulates the 3D inspection task as an optimization problem that maximizes cumulative reward over a receding planning horizon, integrating perception, planning, and control within a data-driven predictive control framework. The proposed method jointly optimizes the UAS motion control inputs and camera viewpoints, incorporating back-face elimination, a technique from 3D computer graphics used for visibility determination, directly into the control loop. This enables the generation of accurate, long-horizon 3D inspection trajectories.

The rest of the paper is organized as follows: Section II reviews related work. Section III presents the problem formulation. Section IV describes the proposed approach, while Section V provides the evaluation. Finally, Section VI concludes the paper.

II. RELATED WORK

Most existing inspection planning approaches primarily address 2D environments, such as terrain coverage [21]. Visibility-aware methods with fixed sensors, as presented in [22], rely on myopic strategies, limiting their capacity for long-horizon planning. In 3D environments, [23] formulates the inspection task as a sampling-based path planning problem, enabling online receding horizon planning through geometric random trees. The method in [24] decomposes the problem into view and motion planning, solved heuristically via genetic algorithms. A finite element method (FEM)-based approach is introduced in [25], where offline inspection paths are generated using potential and distance fields. Particle swarm optimization (PSO) is employed in [26] and [27], with the latter framing the problem as discrete graph optimization. Recent works [28], [29] formulate the planning problem as a traveling salesman problem (TSP), solved using meta-heuristic techniques. Receding horizon control has also been

applied to 3D inspection planning in [30], [31], however, these methods are constrained to cuboid-like structures and do not incorporate visibility considerations. Finally, the approach in [6] is limited to the generation of myopic plans.

III. PROBLEM FORMULATION

A. UAS Dynamical Model

Without loss of generality, we assume that the dynamical behavior $\mathscr{B}$ of a UAS agent can be described by a linear time-invariant (LTI) system of the following form [32]:

$$\mathscr{B}(A,B,C,D) := \begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t), \end{cases} \quad (1)$$

where $\mathscr{B}(A,B,C,D)$ is a minimal input/output/state representation of the system, with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$ unknown. The state, control input, and output of the system at time-step $t \in \mathbb{N}$ are given respectively by $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$.

Definition 1: The signal $u \in \mathbb{R}^{m \times T}$ of length $T \in \mathbb{N}^+$ with $u(t) \in \mathbb{R}^m, t \in \{1, \ldots, T\}$ is persistently exciting of order $N \in \mathbb{N}^+ \leq T$ if the Hankel matrix:

$$\mathscr{H}_N(u) := \begin{pmatrix} u(1) & u(2) & \ldots & u(T-N+1) \\ u(2) & u(3) & \ldots & u(T-N+2) \\ \vdots & \vdots & \ddots & \vdots \\ u(N) & u(N+1) & \ldots & u(T) \end{pmatrix}, \quad (2)$$

has full rank, i.e., $\text{rank}(\mathscr{H}_N(u)) = mN$.

Lemma 1 (Fundamental Lemma [14]): Consider a controllable LTI system $\mathscr{B}(A,B,C,D)$ described by Eq. (1) and assume that the input sequence (i.e., control input samples) $u^d \in \mathbb{R}^{m \times T}$ is persistently exciting of order $N + n$, and $y^d \in \mathbb{R}^{p \times T}$ is the corresponding output (i.e., collected output data samples). Then, the input-output pair of sequences u and y of length N is a valid input-output trajectory of $\mathscr{B}(A,B,C,D)$ if and only if there exists $g \in \mathbb{R}^{T-N+1}$ such that:

$$\begin{bmatrix} \mathscr{H}_N(u^d) \\ \mathscr{H}_N(y^d) \end{bmatrix} g = \begin{bmatrix} u \\ y \end{bmatrix}. \quad (3)$$

Subsequently, a valid input-output trajectory (u,y) of the system of length N can be constructed by collecting a sufficiently rich and long (i.e., persistently exciting) input-output sequence (u^d, y^d) of length T such that $T \geq (m+1)(N+n)-1$. In this case, the subspace consisting of all valid input and output trajectories of length N of the system is the same as the range space of the Hankel matrices $\mathscr{H}_N(u^d)$ and $\mathscr{H}_N(y^d)$ respectively. For extensions to nonlinear and stochastic systems, we refer readers to [33].

B. UAS Camera Model

The UAS carries a gimballed camera module with a finite field of view (FOV), which is utilized for performing automated inspection missions. The camera FOV is represented as a convex polyhedron i.e., a regular right pyramid with square base, characterized by its vertices $V_{\text{FOV}} \in \mathbb{R}^{3 \times 5}$,

where $V_{\text{FOV}}(i) \in \mathbb{R}^3, i \in \{1, \ldots, 5\}$. V_{FOV} is adjusted in three-dimensional space by instructing the gimbal controller to perform two sequential basic rotations: first, a rotation by an angle $\theta_y \in \Theta_y$ around the y-axis, and then a rotation by an angle $\theta_z \in \Theta_z$ around the z-axis as:

$$V_{\theta_z,\theta_y}(i) = R_z(\theta_z)R_y(\theta_y)V_{\text{FOV}}(i), i \in \{1, \ldots, 5\}, \quad (4)$$

where, with slight abuse of notation, V_{θ_z,θ_y} is the matrix containing the rotated vertices of the FOV corresponding to the angles (θ_z, θ_y), which are used here as indices. $R_y(\theta)$ and $R_z(\theta)$ are the basic 3×3 rotation matrices that rotate vectors by an angle θ around the y-axis and z-axis, respectively.

The convex hull of V_{θ_z,θ_y} is defined as $C_{\theta_z,\theta_y} = \{\mathrm{x} \in \mathbb{R}^3 | \Gamma_{\theta_z,\theta_y}\mathrm{x} \leq \Delta_{\theta_z,\theta_y}\}$, where $\Gamma_{\theta_z,\theta_y}$ is a 5×3 matrix whose rows are the outward normal vectors $\Gamma_{\theta_z,\theta_y}^\top(i) \in \mathbb{R}^3, i \in \{1, \ldots, 5\}$ on the pyramid FOV facets formed by the vertices in V_{θ_z,θ_y} and $\Delta_{\theta_z,\theta_y}$ is a 5×1 constant vector containing the offset $\Delta_{\theta_z,\theta_y}(i)$ of each facet i from the origin. Finally, the set of points $C_{\theta_z,\theta_y}(t)$ that can be observed through the FOV V_{θ_z,θ_y} along the agent's trajectory described by Eq. (1) at time-step t is given by:

$$C_{\theta_z,\theta_y}(t) = \{\mathrm{x} \in \mathbb{R}^3 | \Gamma_{\theta_z,\theta_y}\mathrm{x} \leq \Delta_{\theta_z,\theta_y} + \Gamma_{\theta_z,\theta_y}hy(t)\}, \quad (5)$$

where the output of the system $hy(t) \in \mathbb{R}^3$ denotes the UAS position (3D Cartesian coordinates) at time-step t, i.e., the matrix h extracts the spatial coordinates from the measurement vector $y(t)$.

C. Problem Statement

Consider a known structure of interest, denoted by $\mathcal{S} \subset \mathbb{R}^3$, whose surface is represented by a mesh $\Delta\mathcal{S}$ composed of triangular facets. Let $\Delta\hat{\mathcal{S}} = \{s_1, \ldots, s_{|\Delta\hat{\mathcal{S}}|}\} \subseteq \Delta\mathcal{S}$ be a subset of facets. Each facet $s_i \in \Delta\hat{\mathcal{S}}$ is defined by three vertices, νs_1^i, νs_2^i, and $\nu s_3^i \in \mathbb{R}^3$. The centroid of each facet is given by $\bar{s}_i = \text{mean}(\nu s_1^i, \nu s_2^i, \nu s_3^i)$, and each facet is associated with an outward normal vector, denoted as $\vec{s}_i$, and a reward value, r_i which may vary across facets.

The objective of the UAS agent is to design and execute inspection trajectories over a rolling planning horizon of N time-steps, aiming to maximize the cumulative reward obtained by inspecting the facets $s_i \in \Delta\hat{\mathcal{S}}$ with its camera, and to conclude the mission once all facets in $\Delta\hat{\mathcal{S}}$ have been inspected.

Let γ_i denote a decision variable indicating whether facet s_i has been inspected by the end of a planning horizon of length N time-steps. The agent's objective can be formulated as: $\max \sum_{i=1}^{|\Delta\hat{\mathcal{S}}|} r_i \gamma_i(N)$. In the scenario where $r_i = r, \forall i \in \{1, \ldots, |\Delta\hat{\mathcal{S}}|\}$ the objective simplifies to maximizing the number of facets inspected within the planning horizon. On the other hand, by assigning different reward values to the facets, we can design trajectories that prioritize different parts of the structure to be inspected. However, the objective above does not capture the requirement to inspect all facets in $\Delta\hat{\mathcal{S}}$. To accomplish this, the agent must track which facets have been inspected, ensuring mission progress and avoiding redundant work. Additionally, the agent must determine in

an on-line fashion the visibility of each facet based on its predicted state, enabling the planning of trajectories that ensure the inspection of visible areas on the structure's surface.

In summary, the agent is required to inspect all facets, collecting the reward from each facet only once and only if the facet is visible. Next, we discuss how inspection planning dynamics and perception-aware constraints via back-face elimination have been integrated within a data-driven predictive control framework to tackle this problem.

IV. PLANNING AND CONTROL FOR 3D INSPECTION

A. Inspection Planning Dynamics

As previously mentioned, the task is to inspect $|\Delta\hat{\mathcal{S}}|$ facets on the surface of $\mathcal{S}$. The inspection value of each facet s_i over the planning horizon is represented by the binary decision variable $\gamma_i(t+\tau|t) \in \{0,1\}$, for $\tau \in \{0,\ldots,N-1\}$. Here, $\gamma_i(t+\tau|t) = 1$ indicates that facet s_i is predicted to be inspected at time-step $t + \tau$ of the planning horizon, based on the plan computed at time-step t, while $\gamma_i(t + \tau|t) = 0$ implies that the facet has not yet been scheduled for inspection as of time-step $t + \tau$.

To enable the agent to design long-horizon inspection trajectories, anticipate which facets can be inspected at future time-steps while integrating past information (i.e., facets already inspected), we design the following inspection planning dynamics:

$$\xi_i(t + \tau + 1|t) - \xi_i(t + \tau|t) = \varpi_i(\tau) \quad (6a)$$

$$\varpi_i(\tau) \leq \gamma_i(t + \tau|t) + \Xi_i(t) \quad (6b)$$

$$\xi_i(t|t) = \xi_i(t|t-1) \quad (6c)$$

where the variable $\xi_i(t + \tau + 1|t) \in [0,1]$ captures the discrete-time dynamics of the inspection value for each facet via the difference equation (6a), initialized from the previous plan, and $\varpi_i(\tau) \in [0,1]$ is an auxiliary decision variable which integrates the inspection value $\gamma_i(t + \tau|t)$ during the current planning horizon with a memory component $\Xi_i(t) \in [0,1]$. Specifically, $\Xi_i(t) = 1$ only if there exists a time-step $t' < t$ at which the facet s_i has been inspected. Subsequently, $\xi_i(t + \tau + 1|t)$ captures the evolution of inspection value for each facet based on current and past information up to time-step t. Based on this, the agent's objective can be re-formulated as: $\max_{\gamma} \sum_{i=1}^{|\Delta\hat{\mathcal{S}}|} r_i \xi_i(t+N|t)$ where $\gamma \in \{0,1\}^{|\Delta\hat{\mathcal{S}}| \times N}$ is the collection of binary decision variables $\gamma_i(t+\tau|t) \, \forall i, \forall \tau$. Observe that the inspection value of some facet i can be activated either through $\Xi_i(t)$ or via $\gamma_i(t + \tau|t)$. Consequently, if a facet i has been inspected at some time-step $t' < t$ in the past (i.e., $\Xi_i(t) = 1$), then turning on $\gamma_i(t + \tau \mid t)$ (i.e., planning to inspect it again at some future time step $t + \tau \mid t$) does not improve the objective. Therefore, in such cases, the agent is encouraged to design a plan for inspecting some other facet $j \neq i$ that has not yet been inspected, which will improve the objective. Next, we discuss the design of perception-aware constraints based on back-face elimination, enabling the agent to identify visible facets along its path and plan accurate long-horizon inspection strategies. Additionally, we explain how γ is integrated with perception-aware constraints within a data-driven predictive control framework to guide the agent in maximizing the cumulative reward.

B. Back-face Elimination

Back-face elimination [34] is an optimization technique in 3D computer graphics that enhances rendering performance by removing polygons (typically triangles) oriented away from the camera. Since these back-facing polygons are not visible to the viewer, discarding them reduces the number of polygons that need to be processed and rendered, thereby conserving computational resources. This method determines a polygon's visibility by assessing the orientation of its normal vector relative to the viewing direction. In this work, this methodology is utilized to integrate perception-aware constraints to 3D inspection planning aimed at determining facet visibility relative to the agent's predicted state, thereby enabling the design of accurate inspection plans. More specifically, a facet is considered visible if its normal vector is oriented opposite to the camera's viewing direction (i.e., the camera is facing the facet). Additionally, the facet's centroid must lie within the convex hull of the camera's field of view, a criterion that can be extended to include all vertices of the facet. Let the agent's camera viewing direction at a given orientation and position be denoted by $\vec{c}_{\theta_z,\theta_y}(t+\tau|t) = c^o_{\theta_z,\theta_y} - hy(t+\tau|t)$, for $\tau \in \{0,\ldots,N-1\}$, where $c^o_{\theta_z,\theta_y} = \text{mean}(V_{\theta_z,\theta_y}(1),\ldots,V_{\theta_z,\theta_y}(4))$ is the centroid of the FOV's base. For a given facet $s_i, i \in \{1,\ldots,|\Delta\hat{\mathcal{S}}|\}$ with an outward normal vector $\vec{s}_i$ and centroid $\bar{s}_i$, we design the following perception-aware constraints to determine visibility:

$$\text{dot}\left(\vec{c}_{\theta_z,\theta_y}(t+\tau|t), \vec{s}_i\right) \leq M\left(1 - b_{i,\theta_z,\theta_y}(t+\tau|t)\right) \quad (7a)$$

$$\text{dot}\left(\vec{c}_{\theta_z,\theta_y}(t+\tau|t), \vec{s}_i\right) > -Mb_{i,\theta_z,\theta_y}(t+\tau|t) \quad (7b)$$

$$\psi_{i,\theta_z,\theta_y}(t+\tau|t) = 1 \implies \bar{s}_i \in C_{\theta_z,\theta_y}(t+\tau|t) \quad (7c)$$

where $\text{dot}(a,b)$ is the dot-product between vectors a and b, M is a large positive constant, and $b_{i,\theta_z,\theta_y}(t + \tau|t) \in \{0,1\}$, $\psi_{i,\theta_z,\theta_y}(t + \tau|t) \in \{0,1\}$ are binary decision variables indicating whether the camera faces the facet s_i and whether its centroid $\bar{s}_i$ belongs to the convex hull of the camera's FOV, respectively. More specifically, if $b_{i,\theta_z,\theta_y}(t + \tau|t) = 1$, then from Eq. (7a)-(7b), the dot product between $\vec{c}_{\theta_z,\theta_y}(t + \tau|t)$ and $\vec{s}_i$ is negative, and restricted to the range $(-M,0]$, indicating visibility, where the camera viewing direction faces the facet's normal vector. On the other hand, if the dot product between these two vectors is positive, it indicates that they are not pointing toward each other, therefore, the camera does not face the facet. Consequently, if $\text{dot}\left(\vec{c}_{\theta_z,\theta_y}(t+\tau|t), \vec{s}_i\right) > 0$, the decision variable $b_{i,\theta_z,\theta_y}(t + \tau|t)$ is forced to become zero to satisfy the constraints within the interval $(0, M]$. Finally, the binary decision variable $\psi_{i,\theta_z,\theta_y}(t + \tau|t)$ is activated only if the centroid $\bar{s}_i$ resides within the convex hull of the camera's FOV at time-step $t + \tau|t$ of the planning horizon as shown in (7c). We note here that the first two constraints above can be simplified since the camera direction vector $\vec{c}_{\theta_z,\theta_y}(t +$

$\tau|t)$ remains unchanged under translation. Therefore, the constraints in Eq. (7a)-(7b) are time-invariant, allowing the time index to be dropped from all variables (i.e., the viewing direction vector can be computed by rotating the camera at the origin).

C. Data-Driven Predictive Inspection Control

Given our UAS agent represented by $\mathscr{B}(A, B, C, D)$ as described in Sec. III-A, and based on Lemma 1, we collect a sequence of input $u^d \in \mathbb{R}^{m \times T}$ and output $y^d \in \mathbb{R}^{p \times T}$ data of length $T \geq (m+1)(L+n) - 1$, which are persistently exciting of order $L+n$, with $L \in \mathbb{N}^+ = K + N$. Here, $K \geq \ell$ denotes the length of the initialization horizon (i.e., a window of historical input-output data) used to capture the initial state of the system at the start of the predictive control horizon, ensuring that predictions for future actions are consistent with the system's past behavior, i.e., the executed trajectory. Subsequently, $\ell \in \mathbb{N}^+$ denotes the smallest integer such that the rank of the observability matrix $\mathcal{O}_\ell(A, C) = \left[C^\top, (CA)^\top, \cdots, (CA^{\ell-1})^\top \right]^\top \in \mathbb{R}^{p\ell \times n}$ is equal to n, i.e., $\mathrm{rank}(\mathcal{O}_\ell(A, C)) = n$ [14]. Finally, as already discussed, N is the planning horizon, defining the length of time-steps into the future to predict the system. Subsequently, the system's dynamical behavior is captured through the two Hankel matrices: $U_H = \mathcal{H}_L(u^d)$, and $Y_H = \mathcal{H}_L(y^d)$, where U_H is divided into two parts (i.e., U_p and U_f), used to capture the past and future behavior of the system, respectively. Here, $U_p \in \mathbb{R}^{mK \times T-L+1}$ and $U_f \in \mathbb{R}^{mN \times T-L+1}$. The same procedure applies to the Hankel matrix Y_H for the output data. The proposed controller is formulated as a quadratic mixed-integer optimization problem (MIQP) shown in Problem (P1) via Eq. (8a)-(8u). This controller, is initialized based on the past input-output sequence (u_o, y_o) of length K, which effectively sets the underlying initial states from which the predicted trajectory (u, y) will evolve. This initialization procedure is shown in Eq. (8b)-(8e), where $u_o \in \mathbb{R}^{m \times K}$ and $y_o \in \mathbb{R}^{p \times K}$, so that $u_o(t) \in \mathbb{R}^m$ and $y_o(t) \in \mathbb{R}^p$. At each time-step t, the sequence pair (u_o, y_o) tracks the executed trajectory, updating its values by shifting those from the previous time-step one position to the left (i.e., Eq. (8b), (8d)), and then incorporating the applied control input $u(t-1|t-1)$ and observed output $y(t-1|t-1)$ from the previous time-step $t-1$ (i.e., Eq. (8c), (8e)).

The constraints shown in Eq. (8f)-(8i), construct the future trajectory of the agent by identifying a consistent future input-output sequence (u, y) of length N aligned with the past data (u_o, y_o) of length K, via the decision variable $g \in \mathbb{R}^{T-L+1}$. The notation $U_p(\kappa)$ in Eq. (8f) represents the κ_{th} block row of $U_p \in \mathbb{R}^{mK \times T-L+1}$, so that $U_p(\kappa) \in \mathbb{R}^{m \times T-L+1}$. Subsequently, the dot product operation $\mathrm{dot}(U_p(\kappa), g) = u_o(t-\kappa|t)$ is performed between all m rows of $U_p(\kappa)$ and g, resulting in an m-dimensional signal, which we equate to $u_o(t-\kappa|t)$. The same notation is applied to the remaining constraints in Eq. (8g)-(8i). The predicted control inputs and system outputs over the rolling planning horizon $\tau \in \{0, \ldots, N-1\}$ are represented by the decision variables $u(t+\tau|t)$ and $y(t+\tau|t)$.

(P1) Predictive Planning and Control

$$\min_{\gamma, g, \boldsymbol{u}} \mathcal{J} \tag{8a}$$

subject to:

– Initialize past behavior:

$$u_o(t-\kappa|t) = u_o(t-\kappa+1|t-1), \kappa \in \{1, .., K-1\} \tag{8b}$$

$$u_o(t-K|t) = u(t-1|t-1) \tag{8c}$$

$$y_o(t-\kappa|t) = y_o(t-\kappa+1|t-1), \kappa \in \{1, .., K-1\} \tag{8d}$$

$$y_o(t-K|t) = y(t-1|t-1) \tag{8e}$$

– Construct future behavior:

$$\mathrm{dot}(U_p(\kappa), g) = u_o(t-\kappa|t), \qquad \forall \kappa \tag{8f}$$

$$\mathrm{dot}(Y_p(\kappa), g) = y_o(t-\kappa|t), \qquad \forall \kappa \tag{8g}$$

$$u(t+\tau|t) = \mathrm{dot}(U_f(\tau+1), g), \qquad \forall \tau \tag{8h}$$

$$y(t+\tau|t) = \mathrm{dot}(Y_f(\tau+1), g), \qquad \forall \tau \tag{8i}$$

– Inspection planning dynamics:

$$\xi_i(t+\tau+1|t) = \xi_i(t+\tau|t) + \varpi_i(\tau), \qquad \forall \tau, i \tag{8j}$$

$$\xi_i(t|t) = \xi_i(t|t-1), \qquad \forall i \tag{8k}$$

$$\varpi_i(\tau) \leq \gamma_i(t+\tau|t) + \Xi_i(t), \qquad \forall i \tag{8l}$$

– Perception-aware constraints:

$$\mathrm{dot}(\vec{c}_\phi, \vec{s}_i) \leq M(1 - b_{i,\phi}), \qquad \forall i, \phi \tag{8m}$$

$$\mathrm{dot}(\vec{c}_\phi, \vec{s}_i) > -Mb_{i,\phi}, \qquad \forall i, \phi \tag{8n}$$

$$\psi_{i,\phi}(t+\tau|t) = 1 \implies$$
$$\bar{s}_i \in C_\phi(t+\tau|t), \qquad \forall i, \phi \tag{8o}$$

– Constraint Integration:

$$\gamma_i(t+\tau|t) \leq [b_{i,\phi} \times \psi_{i,\phi}(t+\tau|t)] \times$$
$$\omega_\phi(t+\tau|t), \qquad \forall \tau, i, \phi \tag{8p}$$

– Collision avoidance:

$$y(t+\tau|t) \notin C_{\Delta S} \qquad \forall \tau \tag{8q}$$

– Variable definitions:

$$\tau \in \{0, .., N-1\}, \kappa \in \{1, .., K\}, i \in \{1, .., |\Delta \hat{S}|\} \tag{8r}$$

$$\phi \in \{1, .., |\{\Theta_z \times \Theta_y\}|\}, \omega_\phi(t+\tau|t) \in \{0, 1\} \tag{8s}$$

$$\gamma_i(t+\tau|t), \psi_{i,\phi}(t+\tau|t), b_{i,\phi} \in \{0, 1\}, \tag{8t}$$

$$\xi_i(t+\tau+1|t), \Xi_i(t), \varpi_i(\tau) \in [0, 1], g \in \mathbb{R}^{T-L+1} \tag{8u}$$

The constraints in Eq. (8j)-(8l) capture the inspection planning dynamics discussed in Sec. IV-A. These constraints are designed to minimize redundant work by (a) tracking already inspected facets through the memory component $\Xi_i(t)$ and (b) preventing the generation of plans that inspect the same facet more than once within the planning horizon.

Next, the constraints in Eq. (8m)-(8n) identify the visible facets through back-face elimination as discussed in Sec. IV-B, where with slight change of notation $\phi \in \{1, \ldots, |\{\Theta_z \times \Theta_y\}|\}$ serves as index pointing to a specific FOV orientation $\{\theta_z, \theta_y\} \in \{\Theta_z \times \Theta_y\}$. The viewing direction $\vec{c}_\phi, \forall \phi$, is precomputed for all camera FOV rotations since it is invariant under translation. Therefore, the binary decision variable $b_{i,\phi}$ indicates whether facet i can be observed through the

camera orientation ϕ, based on the facet's normal vector and the camera's viewing direction. To determine visibility, this result must be combined with the binary decision variable $\psi_{i,\phi}(t+\tau|t)$, which indicates whether the centroid of facet s_i resides within the convex hull of the FOV with orientation ϕ at time step $t+\tau|t$. This is achieved with the constraint shown in Eq. (8o) as follows: The convex hull of the agent's ϕ_{th} camera FOV at time step $t+\tau|t$ is given, as discussed in Sec. III-B, by $C_\phi(t+\tau|t) = \{x \in \mathbb{R}^3 \mid \Gamma_\phi x \le \Delta'_\phi(t+\tau|t)\}$, where $\Delta'_\phi(t+\tau|t) = \Delta_\phi + \Gamma_\phi hy(t+\tau|t)$ and $hy(t+\tau|t)$ indicates the predicted output of the system, i.e., the agent's position inside the planning horizon. The binary decision variable $\psi_{i,\phi}(t+\tau|t)$ can then be derived from the auxiliary binary variable $\psi'_{j,i,\phi}(\tau)$ via the constraint:

$$\text{dot}\left(\Gamma_\phi(j), \bar{s}_i\right) + M\psi'_{j,i,\phi}(\tau) - \Delta_\phi(j)\psi'_{j,i,\phi}(\tau) - \\ \text{dot}\left(\Gamma_\phi(j), hy(\tau)\right)\psi'_{j,i,\phi}(\tau) \le M, \quad (9)$$

where for brevity $t+\tau|t$ is abbreviated as τ, $j \in \{1, \dots, 5\}$ denotes the FOV faces, M is a large positive constant, $\Gamma_\phi(j)$ denotes the j_{th} row of the matrix Γ_ϕ, $\Delta_\phi(j)$ is the j_{th} element of the column vector Δ_ϕ, and $\psi'_{j,i,\phi}(\tau) \in \{0,1\}$. $\psi'_{j,i,\phi}(\tau)$ will take the value of 1 if the inequality $\text{dot}(\Gamma_\phi(j), \bar{s}_i) \le \Delta_\phi(j) + \text{dot}(\Gamma_\phi(j), hy(\tau))$ holds for some j, ϕ, i, and τ; otherwise, it will take the value of 0. Subsequently, $\bar{s}_i \in C_\phi(t+\tau|t)$ implies that $\psi'_{j,i,\phi}(\tau) = 1, \forall j$. This can be implemented with the following constraint: $5\psi_{i,\phi}(\tau) \le \sum_{j=1}^{5} \psi'_{j,i,\phi}(\tau), \forall \phi, i, \tau$, where the controllable binary decision variable $\psi_{i,\phi}(\tau)$, when driven to 1 forces $\psi'_{j,i,\phi}(\tau) = 1, \forall j$ in order to satisfy the constraint. In essence, by setting $\psi_{i,\phi}(t+\tau|t)$ to 1, we are effectively guiding the agent's ϕ_{th} camera FOV through the output $hy(t+\tau|t)$ at time step $t+\tau|t$ to cover the centroid of facet s_i.

Next, the perception-aware constraints are given in Eq. (8p) via the decision variable $\gamma_i(t+\tau|t)$, which indicates that there exists a camera FOV orientation such that facet s_i is visible and can be inspected via the agent's output at time-step $t+\tau|t$. This is accomplished by multiplying the binary decision variables $b_{i,\phi}$ (back-face elimination), $\psi_{i,\phi}(t+\tau|t)$ (centroid inside the FOV), and $\omega_\phi(t+\tau|t) \in \{0,1\}$. In particular $\omega_\phi(t+\tau|t)$ ensures that, at each time-step within the planning horizon, the camera orientation is restricted to one active configuration, i.e., $\sum_\phi \omega_\phi(t+\tau|t) = 1, \forall \tau$. This multiplication, can easily be implemented in modern mixed-integer solvers using the logical AND operator. To ensure the UAS agent avoids collision with the structure $\mathcal{S}$, we enforce the constraint in Eq. (8q). The convex hull of $\mathcal{S}$ is defined by the intersection of $|\Delta\mathcal{S}|$ half-spaces. Each half-space j, for $j = \{1, \dots, |\Delta\mathcal{S}|\}$, corresponds to a plane described by the equation $\text{dot}(\alpha(j), x) = \beta(j)$ with $x \in \mathbb{R}^3$, effectively partitioning the 3D space into two regions. To prevent a collision with $\mathcal{S}$, the following conditions must be satisfied for all time-steps τ and all planes j:

$$\text{dot}\left(\alpha(j), hy(t+\tau|t)\right) > \beta(j) - Mo_j(t+\tau|t), \ \forall \tau, j, \ (10)$$

where M is a sufficiently large positive constant, and $o_j(t+\tau|t) \in \{0,1\}$ are binary variables. When $o_j(t+\tau|t) = 1$, it

indicates that the inequality $\text{dot}(\alpha(j), hy(t+\tau|t)) \le \beta(j)$ holds at time-step $t+\tau|t$, suggesting potential collision along plane j. The agent is considered to be in collision with $\mathcal{S}$ at time-step $t+\tau|t$ - meaning it resides within the convex hull defined by $\Delta\mathcal{S}$ if $\text{dot}(\alpha(j), hy(t+\tau|t)) \le \beta(j)$ holds for all j. Subsequently, the constraint $\sum_{j=1}^{|\Delta\mathcal{S}|} o_j(t+\tau|t) + 1 \le |\Delta\mathcal{S}|, \ \forall \tau$ ensures that the total number of activated $o_j(t+\tau|t)$ variables at any given time-step is less than $|\Delta\mathcal{S}|$, thereby preventing the agent from simultaneously satisfying all the inequalities that define the convex hull of $\mathcal{S}$ and thus avoiding collision. Finally, the cost function $\mathcal{J}$ to be minimized over γ, g and u i.e., $\min_{\gamma, g, u} \mathcal{J}$, is given by:

$$\mathcal{J} = w_1 \sum_{\tau=1}^{N-1} ||\Delta u(t+\tau|t)||_2^2 - w_2 \sum_{i=1}^{|\Delta\hat{\mathcal{S}}|} r_i \xi_i(t+N|t) \ (11)$$

where $g \in \mathbb{R}^{T-L+1}$, $\gamma := \gamma_i(t+\tau|t), \forall i, \tau$, $u := u(t+\tau|t), \forall \tau$, and $\Delta u(t+\tau|t) = u(t+\tau|t) - u(t+\tau-1|t), \tau \in \{1, \dots, N-1\}$. The parameters w_1 and w_2 are tuning weights used to balance control effort and inspection performance, respectively.

V. EVALUATION

A. Simulation Setup

To evaluate our approach, we simulated a quadrotor UAS agent with a 12-dimensional state vector $x(t) = [x_1, \dot{x}_1, x_2, \dot{x}_2, x_3, \dot{x}_3, x_4, \dot{x}_4, x_5, \dot{x}_5, x_6, \dot{x}_6]^\top$, and full state measurement $y(t)$. The state vector includes both 3D Cartesian position and velocity components, $(x_1, \dot{x}_1)$, $(x_2, \dot{x}_2)$, $(x_3, \dot{x}_3)$, corresponding to the x-, y-, and z-axes, as well as angular positions and velocities, $(x_4, \dot{x}_4)$, $(x_5, \dot{x}_5)$, and $(x_6, \dot{x}_6)$, representing roll, pitch, and yaw, respectively.

Quad-rotor control $u(t) = [u_1, u_2, u_3, u_4]^\top$ is achieved by managing four independent inputs, where the lift force $u_1 \in [-3, 3]$N along the z-axis controls vertical motion. Additionally, torques u_2, u_3, and u_4 are applied around the x-, y-, and z-axes, respectively, to control roll, pitch, and yaw, with $u_2, u_3, u_4 \in [-2, 2]$Nm.

The dynamical behavior of the UAS is linearized around the hover state and is described by $\mathcal{B}(A, B, C, D)$ with $A \in \mathbb{R}^{12 \times 12}$, $B \in \mathbb{R}^{12 \times 4}$, $C = I_{12}$ (identity matrix), and $D = 0$. The non-zero elements of A and B are as follows: $A(1,2) = T_s$, $A(2,2) = 1$, $A(2,9) = g_r T_s$, $A(3,4) = T_s$, $A(4,4) = 1$, $A(4,7) = -g_r T_s$, $A(5,6) = T_s$, $A(6,6) = 1$, $A(7,8) = T_s$, $A(8,8) = 1$, $A(9,10) = T_s$, $A(10,10) = 1$, $A(11,12) = T_s$, $A(12,12) = 1$, and $B(6,1) = \frac{T_s}{ma}$, $B(8,2) = \frac{lT_s}{I_x}$, $B(10,3) = \frac{lT_s}{I_y}$, $B(12,4) = \frac{T_s}{I_z}$.

The parameters $T_s = 0.7$ s, $g_r = 9.81$ m/s^2, $ma = 1.2$ kg, $l = 0.21$ m, and $I_x = I_y = I_z = 0.004$ N $\cdot$ s^2/rad denote the sampling time, gravity, UAV mass, quadrotor arm length, and moments of inertia about the $x-$, $y-$, and $z-$axes, respectively. The camera FOV at the origin is given by $V_{\text{FOV}} = \begin{bmatrix} -W/2 & W/2 & W/2 & -W/2 & 0 \\ W/2 & W/2 & -W/2 & -W/2 & 0 \\ H & H & H & H & 0 \end{bmatrix}$ with $W = 5.5$ m and $H = 6.5$ m. The camera can perform $|\Theta_z| = |\Theta_y| = 8$ equally spaced rotations around the y- and z-axes. The
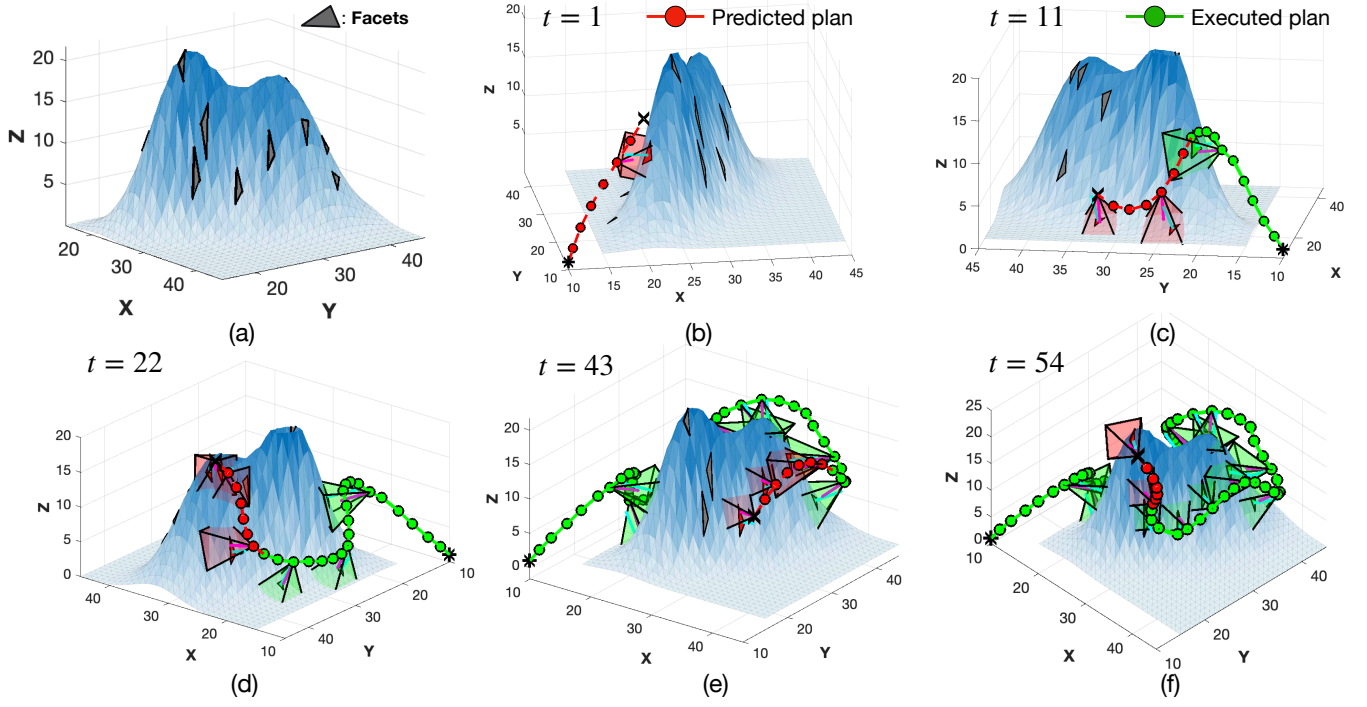
Fig. 1. The figure provides an example to illustrate the proposed approach of data-driven predictive planning and control for 3D inspection.

parameters w_1 and w_2 in Eq. (11) are set to $5E-3$ and 15 respectively, and the controller in Problem (P1) is implemented utilizing the Gurobi MIQP solver.

B. Results

Figure 1 illustrates a 3D inspection scenario executed using the proposed controller from Problem (P1), configured with parameters $K = 1$, $N = 8$, and a collected input-output data sequence of length $T = (m+1)(L+n)-1 = 104$, where $L = K+N$. The structure $\mathcal{S}$ to be inspected is represented as a mesh $\Delta\mathcal{S}$ ($|\Delta\mathcal{S}| = 1008$) shown in Fig. 1(a), with the facets to be inspected $\Delta\hat{\mathcal{S}}$ shown in gray color, randomly sampled from $\Delta\mathcal{S}$, with cardinality $|\Delta\hat{\mathcal{S}}| = 15$, and assigned rewards $r_i, i \in \{1, \dots, 15\}$ randomly sampled within the interval $[1, 20]$. The agent is initialized at the positional state of $(x, y, z) = (10, 10, 1)$, marked with $*$ as shown in Fig. 1(b). The evolution of the inspection mission is shown at different time-steps in Fig. 1(b)-1(f), showcasing the predicted and executed plan (i.e., trajectory and camera FOV) in red and green, respectively. As shown in the figure, the approach simultaneously solves planning and control under perception-aware constraints. The agent determines in an online manner the sequence of facets to inspect next, by optimizing the cumulative reward while considering visibility, within the planning horizon, and generates the optimal trajectory and camera states accordingly. The mission concludes after 60 time-steps, once all facets have been inspected.

Finally, the next experiment evaluates the significance of back-face elimination (BFE) in identifying visible facets for the inspection task. A Monte Carlo simulation with 50 trials was conducted, randomly initializing a UAS agent in the simulated environment shown in Fig. 1, and running
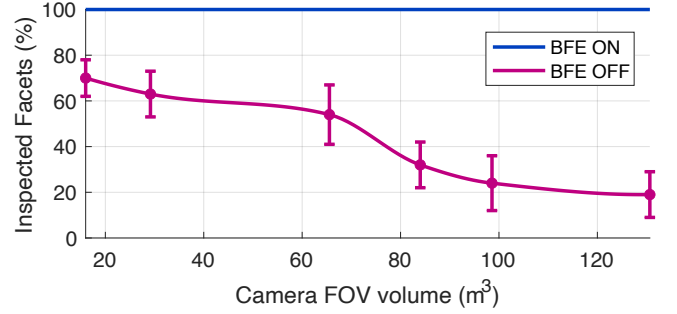

Fig. 2. The effect of back-face elimination (BFE) on the performance of 3D inspection planning. Average percentage of visible inspected facets at the end of the mission as a function of the FOV size.

the proposed controller with and without BFE constraints (i.e., by setting $b_{i,\phi} = 1, \forall i, \phi$ in Eq. (8p)). The facets for inspection were randomly selected from $\Delta\mathcal{S}$, with their quantity randomly sampled from the range [10, 30]. Ground truth was obtained through ray-tracing [35]. The findings, illustrated in Fig. 2, reveal that with BFE enabled, the controller successfully inspects all facets regardless of FOV size, without performance loss. However, disabling BFE results in a performance drop as FOV size increases. This is due to larger FOVs including facets that are within the convex hull of the FOV but not truly visible. Small FOVs require the agent to move closer to the structure, naturally aligning with visible facets. In contrast, larger FOVs may cover more facets in the structure, even if some are not actually visible. Hence, the incorporation of back-face elimination (BFE) enables perception-aware guidance, thereby enhancing the effectiveness of the inspection process.

VI. Conclusion

This paper presents a novel UAS-based, data-driven control approach to 3D inspection planning. By integrating perception, planning, and control within a unified predictive control framework, the method dynamically generates optimal inspection trajectories guided by a perception-aware objective aimed at maximizing inspection performance. The inspection planning task is formulated as an optimization problem that maximizes cumulative rewards over a receding planning horizon, incorporating back-face elimination into the control loop for online visibility determination. Furthermore, this work explores the application of data-driven control to complex optimization problems beyond traditional regulation and tracking tasks. Future work will focus on deploying the proposed approach in real-world scenarios using off-the-shelf UAS platforms.

References

[1] P. Valianti, S. Papaioannou, P. Kolios, and G. Ellinas, "Multi-agent coordinated interception of multiple rogue drones," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[2] S. Papaioannou, C. Laoudias, P. Kolios, T. Theocharides, and C. G. Panayiotou, "Joint estimation and control for multi-target passive monitoring with an autonomous UAV agent," in *2023 31st Mediterranean Conference on Control and Automation (MED)*. IEEE, 2023, pp. 176–181.

[3] J. Moon, S. Papaioannou, C. Laoudias, P. Kolios, and S. Kim, "Deep reinforcement learning multi-uav trajectory control for target tracking," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 441–15 455, 2021.

[4] S. Papaioannou, S. Kim, C. Laoudias, P. Kolios, S. Kim, T. Theocharides, C. Panayiotou, and M. Polycarpou, "Coordinated CRLB-based control for tracking multiple first responders in 3D environments," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1475–1484.

[5] S. Papaioannou, P. Kolios, C. G. Panayiotou, and M. M. Polycarpou, "Synergising human-like responses and machine intelligence for planning in disaster response," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–8.

[6] A. Zacharia, S. Papaioannou, P. Kolios, and C. Panayiotou, "Distributed Control for 3D Inspection using Multi-UAV Systems," in *2023 31st Mediterranean Conference on Control and Automation (MED)*, 2023, pp. 164–169.

[7] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "Rolling horizon coverage control with collaborative autonomous agents," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 383, no. 2289, 2025.

[8] S. Huang, R. S. H. Teo, and W. L. Leong, "Review of coverage control of multi unmanned aerial vehicles," in *2017 11th Asian Control Conference (ASCC)*. IEEE, 2017, pp. 228–232.

[9] X. Kan, H. Teng, and K. Karydis, "Online exploration and coverage planning in unknown obstacle-cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5969–5976, 2020.

[10] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for Collision Avoidance and Control of UAVs With Dynamic Obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6001–6008, 2020.

[11] B. S. Guevara, L. F. Recalde, V. Moya, J. Varela-Aldas, D. C. Gandolfo, and J. M. Toibero, "Estimation of Unmodeled Dynamics: Nonlinear MPC and Adaptive Control Law With Momentum Observer Dynamic," *IEEE Access*, vol. 12, pp. 77 121–77 132, 2024.

[12] A. Saviolo, G. Li, and G. Loianno, "Physics-Inspired Temporal Learning of Quadrotor Dynamics for Accurate Model Predictive Trajectory Tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 256–10 263, 2022.

[13] L. Bauersfeld, E. Kaufmann, P. Foehn, S. Sun, and D. Scaramuzza, "NeuroBEM: Hybrid Aerodynamic Quadrotor Model," in *2021 Robotics: Science and Systems*, 2021.

[14] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

[15] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 307–312.

[16] J. Xing, G. Cioffi, J. Hidalgo-Carrio, and D. Scaramuzza, "Autonomous Power Line Inspection with Drones via Perception-Aware MPC," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1086–1093.

[17] J. Goricanec, A. Milas, L. Markovic, and S. Bogdan, "Collision-Free Trajectory Following With Augmented Artificial Potential Field Using UAVs," *IEEE Access*, vol. 11, pp. 83 492–83 506, 2023.

[18] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "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.

[19] S. Papaioannou, C. Vitale, P. Kolios, C. G. Panayiotou, and M. M. Polycarpou, "Hierarchical Fault-Tolerant Coverage Control for an Autonomous Aerial Agent," *IFAC-PapersOnLine*, vol. 58, no. 4, pp. 532–537, 2024.

[20] 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.

[21] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, vol. 9, pp. 119 310–119 342, 2021.

[22] Y. Kantaros, M. Thanou, and A. Tzes, "Distributed coverage control for concave areas by a heterogeneous robot–swarm with visibility sensing constraints," *Automatica*, vol. 53, pp. 195–207, 2015.

[23] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3D exploration and surface inspection," *Autonomous Robots*, vol. 42, pp. 291–306, 2018.

[24] W. Jing, J. Polden, P. Y. Tao, C. F. Goh, W. Lin, and K. Shimada, "Model-based coverage motion planning for industrial 3D shape inspection applications," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, 2017, pp. 1293–1300.

[25] S. Ivic, B. Crnkovic, L. Grbcic, and L. Matlekovic, "Multi-UAV trajectory planning for 3D visual inspection of complex structures," *Automation in Construction*, vol. 147, p. 104709, 2023.

[26] Z. Shang, J. Bradley, and Z. Shen, "A co-optimal coverage path planning method for aerial scanning of complex structures," *Expert Systems with Applications*, vol. 158, p. 113535, 2020.

[27] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

[28] L. Shi, G. Mehrooz, and R. H. Jacobsen, "Inspection path planning for aerial vehicles via sampling-based sequential optimization," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2021, pp. 679–687.

[29] A. Anastasiou, A. Zacharia, S. Papaioannou, P. Kolios, C. G. Panayiotou, and M. M. Polycarpou, "Automated Real-Time Inspection in Indoor and Outdoor 3D Environments with Cooperative Aerial Robots," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2024.

[30] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "UAV-based receding horizon control for 3D inspection planning," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2022, pp. 1121–1130.

[31] ——, "Distributed search planning in 3-D environments with a dynamically varying number of agents," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 7, pp. 4117–4130, 2023.

[32] J. P. How, E. Frazzoli, and G. V. Chowdhary, "Linear flight control techniques for unmanned aerial vehicles," in *Handbook of unmanned aerial vehicles*. Springer, 2015, pp. 529–576.

[33] L. Huang, J. Lygeros, and F. Dörfler, "Robust and Kernelized Data-Enabled Predictive Control for Nonlinear Systems," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 2, pp. 611–624, 2024.

[34] I. Pantazopoulos and S. Tzafestas, "Occlusion culling algorithms: A comprehensive survey," *Journal of Intelligent and Robotic Systems*, vol. 35, pp. 123–156, 2002.

[35] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "Jointly-optimized Trajectory Generation and Camera Control for 3D Coverage Planning," *IEEE Transactions on Mobile Computing*, 2025, doi:10.1109/TMC.2025.3551362.