

Visual Tracking Nonlinear Model Predictive Control Method for Autonomous Wind Turbine Inspection

Abdelhakim Amer¹, Mohit Mehndiratta¹, Jonas le Fevre Sejersen¹, Huy Xuan Pham¹, Erdal Kayacan¹

Abstract—Automated visual inspection of on-and offshore wind turbines using aerial robots provides several benefits, namely, a safe working environment by circumventing the need for workers to be suspended high above the ground, reduced inspection time, preventive maintenance, and access to hard-to-reach areas. A novel nonlinear model predictive control (NMPC) framework alongside a global wind turbine path planner is proposed to achieve distance-optimal coverage for wind turbine inspection. Unlike traditional MPC formulations, visual tracking NMPC (VT-NMPC) is designed to track an inspection surface, instead of a position and heading trajectory, thereby circumventing the need to provide an accurate predefined trajectory for the drone. An additional capability of the proposed VT-NMPC method is that by incorporating inspection requirements as visual tracking costs to minimize, it naturally achieves the inspection task successfully while respecting the physical constraints of the drone. Multiple simulation runs and real-world tests demonstrate the efficiency and efficacy of the proposed automated inspection framework, which outperforms the traditional MPC designs, by providing full coverage of the target wind turbine blades as well as its robustness to changing wind conditions. The implementation codes¹ are open-sourced.

I. INTRODUCTION

Wind turbine blades operate in harsh working conditions, underlying exposure to high centrifugal loads, erosion, lightning strikes, and bird hits. Thus, their condition needs consistent monitoring via regular inspections. Traditionally, human operators conduct arduous visual inspections, which are expensive and underlie significant risks, as the inspectors are suspended high above the ground. Consequently, piloted drones are deployed due to lower operating costs and reduced downtime of turbines by performing faster inspection [1]. However, manual inspections heavily rely on pilots' experience and skills. Hence, the need for an automated inspection framework that substitutes a skilled human pilot is evident.

There are several challenges in the automated drone-based inspection of wind turbine blades. One of them is maintaining a specific distance from the blades' surface. A large distance negatively affects image resolution, while being too close involves the risk of crashing into the blade, especially when exposed to strong winds. Another challenge is that the drone should be perpendicular to the blades' surface

for an optimal view angle of the region of interest. This challenge underlies two aspects: (i) obtaining the relative viewing angle to the surface gets tricky for a trajectory with desired position and heading attributes, and (ii) maintaining the desired relative view requires consistent rejection of operational disturbances from the controller.

The core of the proposed methodology is the underlying surface trajectory – comprising of an optimal sequence of inspection surfaces – that replaces an explicit position and heading trajectory for the drone. Such a novel approach essentially circumvents closely tracking a predefined trajectory that is no longer appropriate to achieve full inspection coverage due to operational disturbances. To start with, a global distance-optimal sequence of inspection surfaces is obtained via a greedy search-based optimization strategy over each wind turbine blade surface. Next, a novel visual tracking nonlinear model predictive control (VT-NMPC) method is proposed which manipulates the drone's pose to precisely cover each inspection surface. Owing to the well-crafted objective function, the VT-NMPC method allows the drone to consistently correct its relative pose to the blade's surface, thus maintaining the desired distance and optimal viewing angle to the inspected turbine blade. The contributions of this work are as follows:

- An automated wind turbine inspection framework that renders end-to-end inspection by taking turbines' global position and dimensions as inputs.
- A novel nonlinear model predictive control (NMPC) method with visual tracking objectives, rendering an optimal relative drone's pose to the inspection surface at all times.
- Implementation and analysis of the proposed method on a customized drone to demonstrate its real-world applicability.

The rest of this paper is organized as follows: Section II introduces the state-of-the-art inspection methods. Section III presents the problem formulation and illustrates the proposed automated inspection framework. The mathematical model of the inspection drone is presented in Section IV followed by the VT-NMPC problem formulation in Section V. The simulation results are presented in Section VI, followed by real-world implementation experiments in Section VII. Finally, some conclusions are drawn from this work in Section VIII.

II. RELATED WORK

Several approaches tried to address the problems of state estimation, planning, and control for aerial vehicles in chal-

A. Amer, J. le Fevre and H. X. Pham are with Artificial Intelligence in Robotics Laboratory (AiR Lab), the Department of Electrical and Computer Engineering, Aarhus University, 8000 Aarhus C, Denmark, {abdelhakim, jonas.le.fevre, huy.pham} @ece.au.dk. Mohit Mehndiratta is with GIM Robotics, Espoo, Finland, mohit.mehndiratta@gimrobotics.fi. E. Kayacan is with the Automatic Control Group, Department of Electrical Engineering and Information Technology, Paderborn University, Paderborn, Germany, {erdal.kayacan} at uni-paderborn.de.

¹<https://www.github.com/open-airlab/VTNMPC-Autonomous-Wind-Turbine-Inspection>

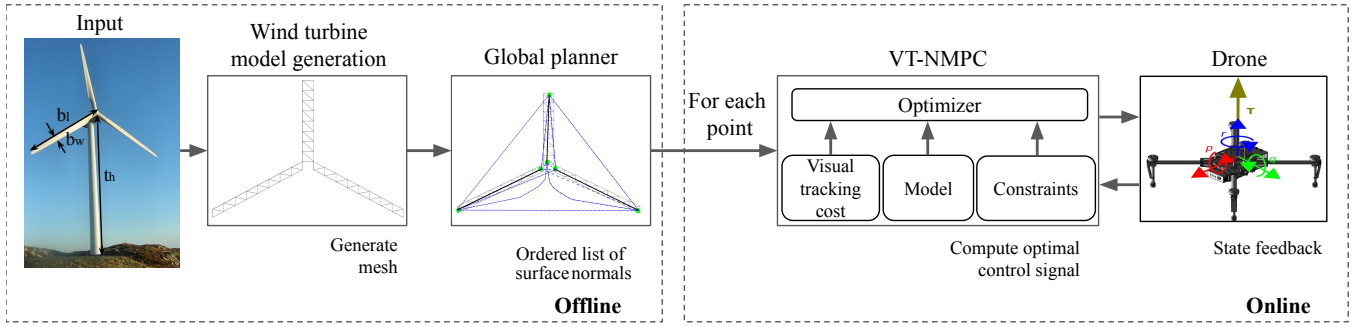


Fig. 1: Overview of the automated inspection framework modules. The user specifies the dimensions and location of the turbines and the output is the motor commands for the drone. The model generation and global planning are calculated pre-flight (offline), while the optimization and control are performed onboard (online).

lenging outdoor scenarios [2]–[4]. For autonomous wind turbine inspection, trajectory-tracking-based approaches are proposed in the literature wherein an optimal waypoint sequence comprising of 3D position and heading, is generated beforehand [5]–[7]. However, disturbances such as wind render the predefined trajectory insufficient to achieve full coverage of the inspection area. Alternatively, other solutions, [8], [9] generate the inspection trajectory on the fly by utilizing relative distance measurements. The main limitation of these methods is the global suboptimality of the generated trajectory. Additionally, distance-optimality is a desired feature for an automated wind turbine inspection framework because of the limited battery life of the drones.

Other approaches, such as Visual servoing (VS) methods provide a trajectory that allows the drone to maintain a collision-free sight to a given point of interest. For instance, in [10], an optimal trajectory for a quadrotor is obtained by solving a nonlinear optimization problem, followed by the trajectory tracking via NMPC. Another method that provides an optimal trajectory by developing a path parameterization algorithm for quadrotors, considering their limited field of view (FOV), is proposed in [11]. As such, all these VS methods treat the trajectory generation and tracking as separate tasks. Conversely, a more efficient way is to generate the trajectory as part of the control problem. This approach is illustrated in [12], wherein they design a linear MPC that renders trajectory tracking while generating VS-based velocity profiles to navigate the quadrotor. Another work based on a similar principle is illustrated in [13], where the perception-aware model predictive control (PAMPC) method is proposed. This method adds a cost to the problem formulation of NMPC to maintain a target within the FOV-center while minimizing image blur.

The above methods plan a local path that fulfills a predefined visual goal rather than resulting in a global inspection trajectory. Therefore, they are insufficient on their own to realize all inspection-related goals, such as finding a global distance-optimal path and achieving total coverage of the inspection surface. In the next section we present our hierarchical inspection framework, wherein an offline distance-optimal planner results in a global trajectory, followed by the

VT-NMPC method rendering local planning.

III. AUTOMATED INSPECTION FRAMEWORK

The turbine blades need to be inspected in two phases. In the first phase, the wind turbine rests in an inverted Y configuration, wherein the top blade is oriented vertically upward, while the other two are at a 120 degree angle from the vertical. In order to inspect the surfaces pointing towards the ground, the wind turbine is rotated 120 degrees, and a re-planning is performed on the last two surfaces. The inspection framework is demonstrated for a Vestas V100, having tower height (t_h) of 120m, blade length (b_l) of 50m, and blade width (b_w) of 3m (See Fig. 1). An overview of the automated inspection framework is presented in Fig. 1.

A. Triangular Mesh Generation

First, a simplified triangular mesh representation of the turbine blades is created. For this purpose, wind turbine dimensions, b_l , b_w , and t_h , are given as inputs. Three cuboids of dimensions $b_l \times b_w \times \frac{b_w}{3}$ are then created with a 120° relative angle between them. The cuboids represent the turbine blades in a simplified form, wherein each cuboid comprises right-angle triangular elements on each surface. This simplified model overestimates the blade's width as it does not account for the tapering towards the tip. However, introducing this simplification facilitates generating a model of any commercial wind turbine, irrespective of design or size, thus rendering the proposed method generic. It should be noted that the mesh generation process can be customized to support any desired initial wind turbine configuration, and is not limited to the Y configuration.

B. Wind Turbine Inspection Path Planner

In essence, the function of this planner is to obtain a distance-optimal sequence of inspection points, which are the centers of each triangular mesh surface, utilizing the generated simplified mesh model. For this purpose, the path-planning algorithm underlies three steps: Clustering, ordering, and interpolation. Within the clustering step, each triangular element from the wind turbine model is grouped into different surfaces based on its centroid location and value of the surface normal. The wind turbine model then

consists of twelve clusters (surfaces) in total, four for each blade. Then, in the second step, two nodes are defined for each cluster or blade surface, whereby the nodes are located at each end of the surface (root and tip). Thus, a graph structure is created by connecting each node, as shown in Fig. 2, where the edge length connecting the nodes is the cost to be minimized. Subsequently, the ordered list for the node is obtained via solving a traveling salesman problem (TSP), with a constraint that renders the entry and exit node to always belong to the same cluster. This constraint facilitates sequential surface inspection while ensuring that the drone does not jump between surfaces. Since the generated graph comprises a relatively small number of nodes (24), we adopt a brute force algorithm to solve the TSP, thus guaranteeing distance optimality. Finally, the ordered list of intermediate points and corresponding normals between the nodes are found via linear line fitting and interpolation techniques. A very simplified version of the wind turbine-specific path planner algorithm is provided as pseudocode in Algorithm 1. Similar to [7], the proposed path planner uses a mesh generated model of the turbine for computing the distance-optimal path. However, this work exploits the wind turbine structure, wherein the graph is created based on surfaces rather than individual triangles. This simplification reduces the number of nodes by several orders of magnitude, thus significantly lowering the computation time. Yet, the simplification compromises the generality as a universal planner, restricting it to only wind turbine inspection. While [7] is a general planner for any structure, the output of the planner is not compatible with the proposed controller. The output from [7] is the reference point of the inspection drone, while the proposed planner and controller are working with reference of the inspection point on the mesh surface. One could project the drone's reference point onto the inspection mesh, but as [7] computes optimal position by utilizing the FOV of the camera, there is no direct correlation between the projected drone references points and the inspection point. In conclusion, a new planning method was needed in order to fully utilize the potential of the proposed controller.

C. Visual Tracking Nonlinear Model Predictive Control

An NMPC is an optimization-based control method that is commonly preferred for robotic applications [14]–[24]. It calculates control actions utilizing constrained optimization and a model-based prediction. In this work, based on the given inspection point trajectory and the state measurements, the designed VT-NMPC method calculates the desired attitude rates and thrust, which are given as inputs to the PID controller performing the actuator control. Moreover, the VT-NMPC method facilitates visual coverage of the inspected structure by satisfying some visual tracking costs that are specified later in Section V. Unlike other MPC controllers, the designed VT-NMPC method does not require the desired position trajectory for the drone. Note that the proposed VT-NMPC is a generic inspection controller and is not limited to wind turbine inspection. The proposed method has two advantages. Firstly, the waypoints are not

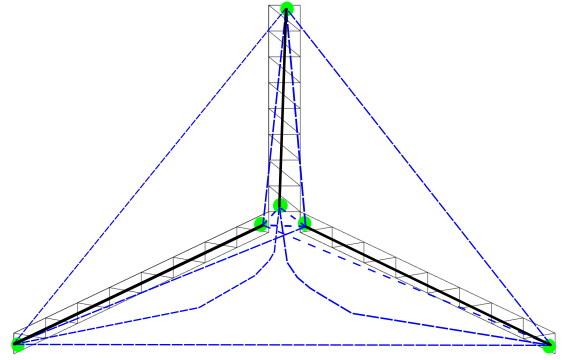


Fig. 2: A topological representation of our graph-based search algorithm for distance-optimal path planner. A node (green) is placed at the end of each blade's surface, which are all connected by (blue) edges to form a graph. The goal is to find the sequence of nodes to visit that minimizes the total distance traveled while satisfying the imposed constraints.

Algorithm 1 Wind turbine specific path planner

A) Create the graph (clustering)

$groups \leftarrow$ group elements by normals & centroid location
 $groups \leftarrow$ sort elements in each group by z-value

B) Solve the TSP via brute force (ordering)

while there are more permutations of the tour, T **do**

 T = []

 shuffle groups

for group in groups **do**

 T append one of end nodes in group

 T append other end node in group

end for

if T is not new **then**

 continue

end if

if distance(T) < $shortest_distance$ **then**

$best_tour \leftarrow T$

$shortest_distance \leftarrow distance(T)$

end if

end while

C) Find intermediate points (interpolation)

$intermediate_points \leftarrow$ line fitting and interpolation

always available for an inspection problem, whereas it is straightforward to provide the inspection surface's position as input. Secondly, safety is ensured throughout the flight, as the VT-NMPC method consistently optimizes the objective of keeping the drone at a specific distance from the inspection surface.

IV. INSPECTION DRONE

We present the dynamic model of the utilized quadrotor in Newton-Euler format [25], while replacing Euler angles with quaternion. The translational kinematics is obtained using the transformation from body frame (\mathcal{F}_B) to Earth-fixed frame

(\mathcal{F}_E) as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_{EB} \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad (1)$$

where x, y, z represent the translational position that is defined in frame \mathcal{F}_E , while u, v, w are the translational velocities and are defined in the frame \mathcal{F}_B and, finally, R_{EB} represents the rotation matrix between frames \mathcal{F}_E and \mathcal{F}_B , associated with the unit quaternion vector $\mathbf{q}_{EB} = [q_x, q_y, q_z, q_w]$. The rigid-body dynamic equations in the body-fixed coordinate are given as follows, assuming the quadrotor to be a point mass such that all the forces act at the center of gravity.

$$\dot{u} = rv - wq + 2g(q_x q_z - q_w q_y), \quad (2a)$$

$$\dot{v} = pw - ru - 2g(q_y q_z + q_w q_x), \quad (2b)$$

$$\dot{w} = qu - pv - g(1 - 2q_x q_x - 2q_y q_y) + \frac{1}{m}T, \quad (2c)$$

where p, q, r are the angular body rates defined in the body frame, T is the total thrust force generated by the drone's propellers in the body frame, and m is the quadrotor's total mass. Additionally, the following represents the relation between the global angular rates in \mathcal{F}_E and the body rates in \mathcal{F}_B :

$$\dot{q}_x = 0.5(pq_w + rq - qq_z), \quad (3a)$$

$$\dot{q}_y = 0.5(qq_w - rq_x + pq_z), \quad (3b)$$

$$\dot{q}_z = 0.5(rq_w + qq_x - pq_y), \quad (3c)$$

$$\dot{q}_w = 0.5(-pq_w - qq_y - rq_z). \quad (3d)$$

Finally, we can rewrite the lumped nonlinear dynamic model of the quadrotor at a high level in its discretized form as follows:

$$\mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k), \quad (4)$$

where $\mathbf{x} \in \mathbb{R}^{10}$ and $\mathbf{u} \in \mathbb{R}^4$, are the state and control vectors and they comprise of:

$$\mathbf{x} = [x, y, z, u, v, w, q_x, q_y, q_z, q_w]^T, \quad (5)$$

$$\mathbf{u} = [p, q, r, T]^T. \quad (6)$$

The state function is denoted by $\mathbf{f}_d(\cdot, \cdot): \mathbb{R}^{10} \times \mathbb{R}^4 \rightarrow \mathbb{R}^{10}$.

V. PROPOSED VT-NMPC FORMULATION

The least-square objective of the proposed VT-NMPC with nonlinear costs and constraints can be formulated in discrete time as follows:

$$\min_{\mathbf{x}_k, \mathbf{u}_k} \frac{1}{2} \left\{ \sum_{k=0}^{N_c-1} (C_h + C_d + C_r + C_o + C_{\mathbf{x}^*} + C_{\mathbf{u}}) + C_{N_c} \right\} \quad (7a)$$

$$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k), \quad k \in [0, N_c - 1], \quad (7b)$$

$$\mathbf{u}_{k,\min} \leq \mathbf{u}_k \leq \mathbf{u}_{k,\max}, \quad k \in [0, N_c - 1], \quad (7c)$$

where the terms $\mathbf{u}_{k,\min} \leq \mathbf{u}_{k,\max} \in \mathbb{R}^4$ in (7c), specify the lower and upper bounds on the controls, respectively. The

terms inside the summation in (7a) together account for the stage cost. Amongst them, the first four costs, namely C_h, C_d, C_r, C_o represent the cost associated with visual tracking objective and are expressed in the following form:

$$C_h = (h_k - h^{\text{ref}})^T \mathbf{w}_h (h_k - h^{\text{ref}}), \quad (8)$$

$$C_d = (d_k - d_k^{\text{ref}})^T \mathbf{w}_d (d_k - d_k^{\text{ref}}), \quad (9)$$

$$C_r = (r_k - r_k^{\text{ref}})^T \mathbf{w}_r (r_k - r_k^{\text{ref}}), \quad (10)$$

$$C_o = (o_k - o_k^{\text{ref}})^T \mathbf{w}_o (o_k - o_k^{\text{ref}}), \quad (11)$$

The first visual tracking cost, C_h essentially controls the heading of the drone, which penalizes the deviation of the inspection area's centroid from the drone's FOV-center. Accordingly, a heading function h is formulated as a dot product between a vector representing the forward direction of the drone in the Earth-fixed frame and a vector \mathbf{a} that connects the drone to the center of the inspection area in the X-Y plane, as illustrated in Fig. 3. Consequently, the heading function is expressed as follows:

$$h = (R_{EB} \mathbf{x}_B) \cdot \mathbf{a}_{xy}, \quad (12)$$

where the vector $\mathbf{x}_B = [1, 0, 0]^T$, and $\mathbf{a}_{xy} = [a_x, a_y, 0]^T$ reflects a vector comprising the $x - y$ components of the original vector \mathbf{a} , which is given as:

$$\mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} p_x - x \\ p_y - y \\ p_z - z \end{bmatrix}. \quad (13)$$

where $\mathbf{p} = [p_x, p_y, p_z]^T$ refers to the center of the inspection area. Moreover, the reference value for the heading function h^{ref} is set to 1 so that the two vectors are aligned.

The second visual tracking cost, C_d controls the position of the drone such that the drone stays within a specified radius from the center point \mathbf{p} . This cost penalizes the deviation of the current Euclidean distance (d) from its specified reference value (d^{ref}). Also, the Euclidean distance is calculated in the X-Y plane as follows:

$$d = \|\mathbf{a}_{xy}\|. \quad (14)$$

The reference Euclidean distance d^{ref} is the distance that the user desires to maintain from the wind turbine for safety. In this work, d^{ref} is set to 7m, which has been the de-facto reference distance in the literature for wind turbine inspection [26]–[28]. The third visual tracking cost, C_r ensures that the drone stays within the desired region of interest, which is maintaining a specific distance from the inspection surface along its surface-normal in the X-Y plane. The region of interest function r is calculated by taking the dot product of the vector \mathbf{a}_{xy} and the normal vector to the surface being inspected:

$$r = \mathbf{a}_{xy} \cdot \mathbf{n}, \quad (15)$$

where the normal vector is given by $\mathbf{n} = [n_x, n_y, 0]^T$. The reference distance to be maintained (r^{ref}) is set equal to d^{ref} , i.e., $r^{\text{ref}} = d^{\text{ref}} = 7\text{m}$. It is to be noted that, the region of interest function r together with the distance function d guide

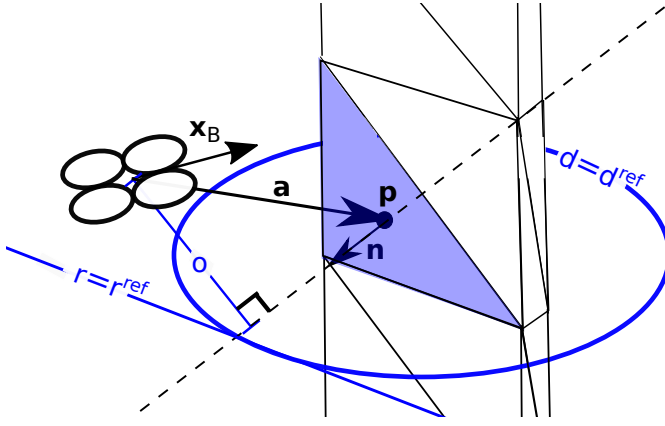


Fig. 3: Graphical depiction of the visual costs for the VT-NMPC for inspecting a surface of normal \mathbf{n} . The intersection between the line $r = r^{\text{ref}}$, the circle $d = d^{\text{ref}}$ and the normal vector \mathbf{n} represent the optimal position that the drone should attain during inspection relative to the surface.

the drone to the desired position – at a specified distance from the inspection surface centroid –, while aligning with the positive direction of the surface-normal.

The final visual tracking cost, \mathcal{C}_o comprises of an orthogonality function o (visually depicted in Fig. 3), which is defined as the magnitude of the orthogonal projection of the vector \mathbf{a} on the surface normal \mathbf{n} , and is expressed as follows:

$$o = \|(\mathbf{a} - (\mathbf{a} \cdot \mathbf{n})\mathbf{n})\|, \quad (16)$$

in essence, the role of this cost is to align the drone with the surface normal, hence, o^{ref} is set to 0. The last two stage costs in (7a) are the costs associated with states and control, penalizing the deviations of the predicted states and control trajectories from their references. They are given as follows:

$$\mathcal{C}_{\mathbf{x}^*} = (\mathbf{x}_k^* - \mathbf{x}_k^{\text{ref}})^T \mathbf{W}_{\mathbf{x}^*} (\mathbf{x}_k^* - \mathbf{x}_k^{\text{ref}}), \quad (17)$$

$$\mathcal{C}_{\mathbf{u}} = (\mathbf{u}_k - \mathbf{u}_k^{\text{ref}})^T \mathbf{W}_{\mathbf{u}} (\mathbf{u}_k - \mathbf{u}_k^{\text{ref}}), \quad (18)$$

where $\mathbf{x}^* = [u, v, w, q_x, q_y]^T \in \mathbf{x}$. The following trajectories are selected for a smooth response from the drone:

$$\mathbf{x}^{*\text{ref}} = \mathbf{x}_{N_c}^{*\text{ref}} = [u_r, v_r, w_r, 0, 0]^T, \quad (19)$$

$$\mathbf{u}^{\text{ref}} = [0, 0, 0, mg]^T. \quad (20)$$

The weights defined in the stage cost are given by $w_h \in \mathbb{R}^{1 \times 1}$, $w_d \in \mathbb{R}^{1 \times 1}$, and $w_r \in \mathbb{R}^{1 \times 1}$, $\mathbf{W}_{\mathbf{x}^*} \in \mathbb{R}^{5 \times 5}$, $\mathbf{W}_{\mathbf{u}} \in \mathbb{R}^{4 \times 4}$. The weight matrices associated with state and control costs are selected to be diagonal and positive-(semi) definite matrices. Finally, the second component in (7a) refers to the terminal cost which is given by:

$$\mathcal{C}_{N_c} = (\mathbf{x}_{N_c}^* - \mathbf{x}_{N_c}^{\text{ref}})^T \mathbf{W}_{N_c} (\mathbf{x}_{N_c}^* - \mathbf{x}_{N_c}^{\text{ref}}). \quad (21)$$

The above cost penalizes the finite nature of the prediction horizon, which caters to the stability of the overall optimization problem. Herein, $\mathbf{W}_{N_c} \in \mathbb{R}^{5 \times 5}$ is the corresponding positive-(semi) definite weight matrix which is also selected as diagonal. In terms of the implementation of the VT-NMPC

method, the weight matrices are selected based on the method proposed in [25]. The incorporated active exploration approach renders automated tuning of the controller weights based on the drone's performance during tuning trials. The method essentially utilizes an intelligent trial-and-error-based strategy that utilizes retrospective knowledge to improve the next set of controller weight attempts, balancing exploration and exploitation in an automated fashion. Accordingly, the following weights are achieved:

$$w_h = 80, w_d = 30, w_r = 25, w_o = 60$$

$$\mathbf{W}_{\mathbf{x}^*} = \text{diag}(0.3, 0.3, 1, 80, 80)$$

$$\mathbf{W}_{\mathbf{u}} = \text{diag}(1, 1, 0.25, 0.03), \mathbf{W}_{N_c} = 1.5\mathbf{W}_{\mathbf{x}^*}.$$

Additionally, the following constraints are imposed on the control input to obtain a smooth response,

$$-100 \text{ (}^\circ/\text{s)} \leq p, q, r \leq 100 \text{ (}^\circ/\text{s)}, \quad (22)$$

$$0.3mg \text{ (N)} \leq T \leq 2mg \text{ (N)}. \quad (23)$$

A. Implementation Details

The optimization problem underlying the VT-NMPC is solved by utilizing the direct multiple shooting method with a shooting grid size of 0.01s, to enable real-time feasibility. Direct solution methodology is adopted over the indirect methods, as the former can account for both inequality and equality constraints. Subsequently, the resulting discretized optimal control problem (OCP) reduces to a sequential quadratic program which is further solved via the generalized Gauss-Newton method with the help of a special real-time iteration (RTI) scheme proposed in [29]. Moreover, ACADO toolkit [30], along with qpOASES solver is utilized as the solution platform for solving the OCP in (7) as it incorporates the direct multiple shooting method and the RTI approach altogether. Additionally, the prediction horizon $N_c = 30$ and a sampling time of 0.01s for computational tractability.

VI. SIMULATIONS

This section demonstrates the results of a wind turbine inspection scenario conducted in a Gazebo simulation environment. A sinusoidal wind disturbance with a time period of 10s and a standard deviation of 0.5m/s is applied in the y-direction to further model a wind farm environment. The results are obtained for three different mean wind speed levels, namely, 0m/s, 4m/s, and 7m/s. Note that only the plots for 4m/s mean wind speed are presented for brevity. The efficacy of the proposed VT-NMPC method is compared with the PAMPC [13] and the conventional NMPC. Both the PAMPC and NMPC follow a reference trajectory consisting of 3-D positions. Besides, the NMPC requires the reference yaw angle, while the PAMPC uses the inspection point as a reference. The reference position trajectory is obtained as an offset from the inspection point, at a distance equal to d^{ref} in the direction of the surface normal, while the reference yaw angle is precomputed based on the surfaces' normal \mathbf{n} . The performance of all three methods is evaluated in terms of three metrics, i.e., safety, coverage, and inspection quality. The obtained results are summarized in Table I.

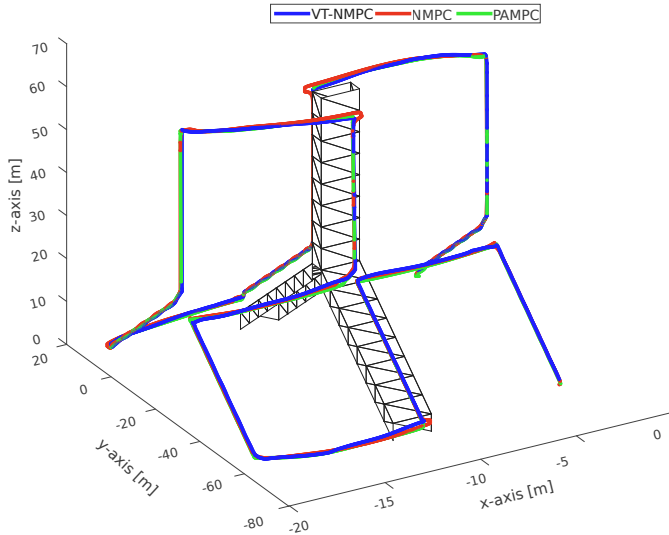


Fig. 4: Obtained optimal 3-D trajectory of the drone during inspection using the proposed novel VT-NMPC, traditional NMPC and PAMPC.

Safety implies the distance from the inspection surface that the drone maintains. Coverage illustrates the ability to view all the triangular mesh elements that underly the turbine model. Inspection quality refers to the quality of the obtained images. The obtained 3-D trajectory of the drone using different methods is shown in Fig. 4. Note that the drone's average speed during the inspection is around 0.7m/s.

A. Safety Metric

First, a safety region requirement is defined as a boundary from the desired reference, d^{ref} , in the positive and negative direction. A 1-meter range (safety margin) is selected, beyond which the drone is considered to be violating this requirement. Accordingly, the safety metric is calculated as follows:

$$\text{safety metric (SM)} = \begin{cases} 100\%, & |d - d^{ref}| < \text{safety margin} \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

which is then averaged over the entire trajectory. The VT-NMPC, PAMPC, and NMPC manage to maintain comparable distances on average with 7.03m, 7.07m, and 7.06m respectively. Nevertheless, the amplitude of fluctuations for the VT-NMPC is significantly lower than the NMPC and PAMPC, obtaining a minimum distance of 6.15m compared to 5.84m for the PAMPC and 5.63m for the NMPC.

The ability of the VT-NMPC to maintain the desired distance compared to the other two methods can be attributed to the inclusion of distance cost in (9). In essence, it explicitly penalizes the distance from the inspection surface compared to the other two which are tracking-error driven, i.e., separately minimizes the error in x , y , and z components.

	NMPC	PAMPC	VT-NMPC
Coverage	91.6%	99.8%	100%
Safety (SM)	75.0%	84.7%	100%
Quality (CM)	98.5%	99.9%	99.9%

TABLE I: Simulation results for the defined performance indices. Tests were conducted at a wind speed of 4 m/s.

B. Coverage

During the inspection, it is required that all surface areas of the parts being inspected are to be captured by the drone's camera. We use the triangular mesh of the generated wind turbine model to measure coverage by checking whether each triangle that comprises the model has been viewed. For each time instance along the trajectory, we check the triangles that are within FOV of the drone by projecting its vertices onto the drone's image plane using a pinhole camera projection. Subsequently, we calculate the coverage as the ratio between the total number of viewed triangles to the total number of triangles. A 100% coverage is achieved by all the three methods with 0m/s wind condition. When introducing a wind speed of 4m/s, both the VT-NMPC and PAMPC maintain a 99.8% coverage value, while the NMPC's coverage reduces to 91.6%. The presence of the visual feedback in the vision-based control methods (VT-NMPC and PAMPC) allows the drone to maintain high coverage for all the three wind levels.

C. Inspection Quality

An additional requirement is the quality of the images collected throughout the inspection operation. The wind turbine blade should be always centered within the image frame. This criterion is evaluated by the centering metric (CM), which is calculated as $CM = h$, where h is the heading function defined in (12). As can be seen in Table I, high average values for CM are achieved for vision-based control methods, i.e., 99.9% for the VT-NMPC and the PAMPC.

VII. REAL-WORLD EXPERIMENTS

A. Experimental Setup

The real-world experiments are conducted in a motion capture system environment having 16 cameras. It is important to note that the use of this system is consistent across all methods and should not have affected the presented comparative study. Additionally, a blowing tunnel is incorporated to provide disturbances with a wind speed of 4 (m/s), reflecting the operational conditions during wind turbine inspections. As such, we develop in-house a small-scale quadrotor, having a total mass of 1090g and a rotor-to-rotor diagonal dimension of 250 mm. The drone is equipped with an Nvidia Jetson TX2 onboard computer that runs the trajectory generation and VT-NMPC codes online. Also, a Pixhawk flight controller running the PX4 firmware renders the controller mapping from thrust and body rates to rotor speeds. In terms of the experimental setup, the drone is

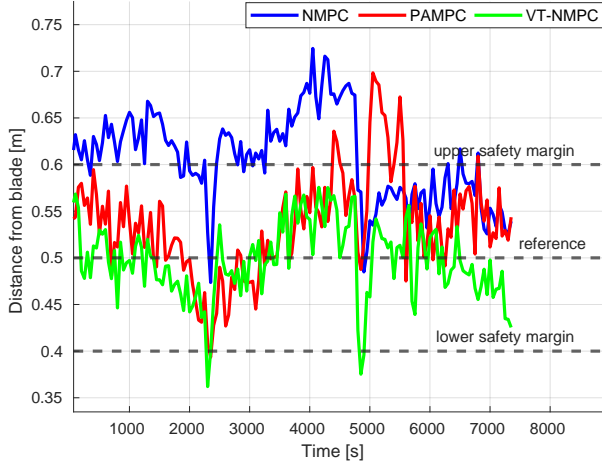


Fig. 5: A plot of the distance from the blade throughout the trajectory, showing the upper and lower safety margins and the reference (d^{ref}).

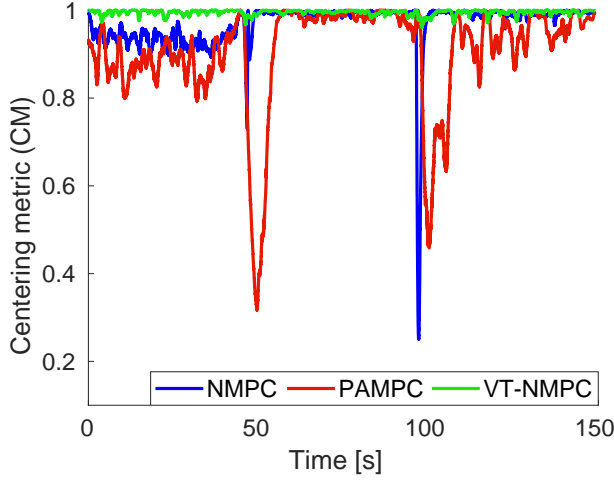


Fig. 6: A plot of the centering metric (CM) vs. time. VT-NMPC consistently tracks the center of the inspected surface area, even under wind disturbance, as opposed to PAMPC and NMPC.

tasked with inspecting a single blade of the turbine inspected in Section VI. Besides, the blade is scaled by a factor of $1/10$, $1/10$, and $1/15$ in x , y , and z dimensions, respectively, and $d^{ref} = 0.5\text{m}$ is set, to cater to the limited testing space.

B. Experimental Results

Next, we present the experimental results, wherein the proposed VT-NMPC method and the baselines are evaluated in a similar setting to the simulation study. As such, the tracked trajectories from the three methods are utilized to compute the defined metrics, i.e., safety, coverage, and inspection quality.

For the specific experimental run, the obtained coverage values from the VT-NMPC, PAMPC, and conventional NMPC are 100%, 90%, and 96%, respectively. This validates the superiority of the added visual tracking costs within the VT-NMPC method. Then, to illustrate the safety metric, the

	NMPC	PAMPC	VT-NMPC
Coverage	98.0%	91.0%	99.8%
Safety (SM)	43.0%	89.9%	98.0%
Quality (CM)	97.0%	90.8%	99.5%

TABLE II: Statistical analysis of the experimental results. For each metric, the average values in percentage are calculated based on three experimental runs. d^{ref} is specified as 0.5m and the safety margin is 0.1 m.

resulting absolute distance error throughout the trajectory is shown in Fig. 5. Note that a safety margin of 10cm is selected in real-world experiments due to the limited testing area. As can be seen, the proposed VT-NMPC method on average maintains the lowest error to the reference in comparison to the baselines. Accordingly, it enables the drone to stay within the safety limits for 98% of the run time, even for such a small safety margin. It may be worth noting that for a marginal increase of 5cm in the safety margin, an SM = 100% is realized for the VT-NMPC method, while the other competitors still violate the limits for substantial run time. The centering metric performance can be observed in Fig. 6. As visualized, the real-world results manifest a dominating performance of VT-NMPC compared to the PAMPC and conventional NMPC. One may note a performance drop for the PAMPC method in the real-world results. This is attributed to the conservative tuning of the perception cost, which otherwise resulted in instability. The main reason for this behavior is the complexity of the perception cost, which also has the possibility of a division by zero during the projection step. Furthermore, the computational effort is measured for the three controllers. The execution time is lowest for the conventional NMPC (5 milliseconds), followed by VT-NMPC (10 milliseconds), then PAMPC (13 milliseconds), implying that the benefits of vision-based MPC methods come at an additional computational burden. Nevertheless, the proposed VT-NMPC method does show a 30% performance improvement over the baseline PAMPC, which can be attributed to the simplicity of its visual costs.

VIII. CONCLUSIONS

In this work, an automated wind turbine inspection framework incorporating a novel NMPC method with visual tracking objectives is presented. Multiple simulations and real-world tests validated that the proposed VT-NMPC method outperforms traditional MPC methods in several aspects. When compared to the PAMPC, the heading-control term is more intuitive, facilitating easier implementation, and shows better performance in achieving full coverage and high inspection quality. Moreover, the VT-NMPC method shows promising improvements in obtaining better image quality by maintaining an optimal relative pose to the inspection surface. Besides, the proposed method infers a potential benefit of directly incorporating the reference surface's pose in the MPC formulation rather than utilizing a position trajectory for the drone during inspection. It is worth noting

that even though the proposed method is exclusive to wind turbine inspection, it can be extended to a broader inspection spectrum by replacing the wind turbine-specific path planner. As part of future work, we intend to collaborate with a wind turbine manufacturer to conduct outdoor testing of the proposed framework.

IX. ACKNOWLEDGEMENT

This work is supported by EIVA a/s and Innovation Fund Denmark under grants 2040-00032B and 2035-00052B. The authors would like to thank Peter Harling Lykke for his support during the real-world experiments. Furthermore, the authors would further like to acknowledge Upteko Aps for bringing the use-case challenge.

REFERENCES

- [1] J. Hwang, D. Platenkamp, and R. Beukema, "A literature survey on remote inspection of offshore wind turbine blades: Automated inspection and repair of turbine blades (airtub-wp1)," 2021.
- [2] J. G. Hansen, M. Heiß, D. Li, M. Kozłowski, and E. Kayacan, "Vessel inspection in-the-wild: Practical planning in large-scale industrial environments," in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 812–817.
- [3] M. Heiß, J. G. Hansen, D. Li, M. Kozłowski, and E. Kayacan, "Predictiveslam-robust visual slam through trajectory-aware object masking," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2023, pp. 1256–1261.
- [4] J. G. Hansen, M. Heiß, M. Kozłowski, and E. Kayacan, "Uav trajectory evaluation in large industrial environments: A cost-effective solution," in *2022 European Control Conference (ECC)*. IEEE, 2022, pp. 1336–1341.
- [5] 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.
- [6] S. Song, D. Kim, and S. Jo, "Online coverage and inspection planning for 3d modeling," *Autonomous Robots*, vol. 44, no. 8, pp. 1431–1450, 2020.
- [7] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 6423–6430.
- [8] M. Mehndiratta and E. Kayacan, "Gaussian process-based learning control of aerial robots for precise visualization of geological outcrops," in *2020 European Control Conference (ECC)*. IEEE, 2020, pp. 10–16.
- [9] M. Car, L. Markovic, A. Ivanovic, M. Orsag, and S. Bogdan, "Autonomous wind-turbine blade inspection using lidar-equipped unmanned aerial vehicle," *IEEE Access*, vol. 8, pp. 131 380–131 387, 2020.
- [10] C. Potena, D. Nardi, and A. Pretto, "Effective target aware visual navigation for uavs," in *2017 European Conference on Mobile Robots (ECMR)*. IEEE, 2017, pp. 1–7.
- [11] I. Spasojevic, V. Murali, and S. Karaman, "Perception-aware time optimal path parameterization for quadrotors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3213–3219.
- [12] P. Roque, E. Bin, P. Miraldo, and D. V. Dimarogonas, "Fast model predictive image-based visual servoing for quadrotors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7566–7572.
- [13] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.
- [14] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe, "Model predictive control in aerospace systems: Current state and opportunities," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541–1566, 2017.
- [15] M. Mehndiratta and E. Kayacan, "Receding horizon control of a 3 DOF helicopter using online estimation of aerodynamic parameters," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2017.
- [16] —, "Online learning-based receding horizon control of tilt-rotor tricopter: A cascade implementation," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 6378–6383.
- [17] —, "A constrained instantaneous learning approach for aerial package delivery robots: onboard implementation and experimental results," *Autonomous Robots*, vol. 43, no. 8, pp. 2209–2228, Dec 2019.
- [18] M. Mehndiratta, K. Singh, E. Kayacan, and M. Feroskhan, "Receding horizon-based fault-tolerant control of quadplus: an over-actuated quadrotor," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 853–859.
- [19] A. Prach and E. Kayacan, "An mpc-based position controller for a tilt-rotor tricopter vtol uav," *Optimal control applications and methods*, vol. 39, no. 1, pp. 343–356, 2018.
- [20] M. Mehndiratta and E. Kayacan, "Reconfigurable fault-tolerant nmpe for y6 coaxial tricopter with complete loss of one rotor," in *2018 IEEE conference on control technology and applications (CCTA)*. IEEE, 2018, pp. 774–780.
- [21] M. Mehndiratta, E. Kayacan, S. Patel, E. Kayacan, and G. Chowdhary, "Learning-based fast nonlinear model predictive control for custom-made 3d printed ground and aerial robots," in *Handbook of Model Predictive Control*. Springer, 2019, pp. 581–605.
- [22] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, "Learning in centralized nonlinear model predictive control: Application to an autonomous tractor-trailer system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 197–205, 2014.
- [23] T. Kraus, H. J. Ferreau, E. Kayacan, H. Ramon, J. De Baerdemaeker, M. Diehl, and W. Saeys, "Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles," *Computers and electronics in agriculture*, vol. 98, pp. 25–33, 2013.
- [24] N. Imanberdiyev and E. Kayacan, "Redundancy resolution based trajectory generation for dual-arm aerial manipulators via online model predictive control," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2020, pp. 674–681.
- [25] M. Mehndiratta, E. Camci, and E. Kayacan, "Can deep models help a robot to tune its controller? a step closer to self-tuning model predictive controllers," *Electronics*, vol. 10, no. 18, p. 2187, 2021.
- [26] P. Durdevic, D. Ortiz-Arroyo, and Z. Yang, "Lidar assisted camera inspection of wind turbines: experimental study," in *2019 ISt International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*. IEEE, 2019, pp. 1–7.
- [27] M. Stokkeland, K. Klausen, and T. A. Johansen, "Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 998–1007.
- [28] W. Gu, D. Hu, L. Cheng, Y. Cao, A. Rizzo, and K. P. Valavanis, "Autonomous wind turbine inspection using a quadrotor," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 709–715.
- [29] M. Diehl, H. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577 – 585, 2002.
- [30] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit—an open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.