IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Model-Based Trajectory Planning of a Hybrid Robot for Powerline Inspection

Zhishuo Li[1], Yunong Tian[2], Guodong Yang[2], Yanfeng Zhang[1], En li[2], Zize Liang[1], and Min Tan[1]

***Abstract*—This letter presents the first trajectory planning method for hybrid robot to perform powerline inspection involving obstacle navigation and landing. We develop a geometric model that incorporates constraints for landing the hybrid robot on a powerline, obstacle avoidance, and objectives that maximize the visibility of the powerline during flight. The trajectory generation is achieved via solving a multiple shooting nonlinear programming problem with respect to system dynamics and geometric constraints. The formulation of the problem accommodates both powerline-to-powerline and air-to-powerline trajectory planning scenarios. It runs onboard and is capable of generating trajectories within 50 ms, regardless of whether the hybrid robot's initial state is positioned on the powerline or hovering above it. Through simulation experiments, we illustrate the impact of our proposed geometric model on trajectory planning. Furthermore, real-world experimental results validate the efficacy of the proposed planning method. Compared with the existing feedback-control-based work, the landing and obstacle navigation time are significantly reduced.**

***Index Terms*—Constrained Motion Planning, Motion and Path Planning, Aerial Systems: Applications.**

I. Introduction

REGULAR powerline inspections are crucial for ensuring the safe and reliable transmission of electricity, preventing hazards and optimizing the efficiency of the electrical grid. In recent years, there have been many efforts focused on inspection robots to improve the efficiency and quality of powerline inspections. The climbing robot [1]–[3] can roll along a powerline to achieve high-precision and long-distance inspections, but it struggles with navigating obstacles. UAVs [4]–[6] can conduct inspections by flying above powerlines, providing high flexibility but with limited endurance. Hybrid robots [7]–[12] are designed to combine the strengths of climbing robots and UAVs, aiming to achieve both high maneuverability and extended endurance.

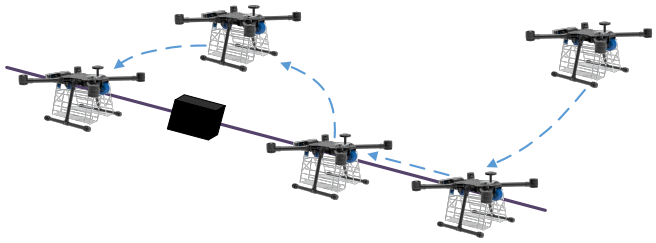

Fig. 1. Illustration of powerline inspection by a hybrid robot. It lands on the powerline and conducts inspection through climbing, flies over the obstacle and lands back on the powerline.

As depicted in Fig. 1, the hybrid robot is expected to autonomously land on the overhead ground wire (OGW), advance for inspections, fly over obstacles like insulators and clamps, and subsequently return to the OGW to continue the inspection. Nevertheless, previous work on hybrid robots has primarily focused on the platform design [7], [8] and autonomous landing [9]–[12], but has given little attention to trajectory planning. These aspects are crucial for enabling hybrid robots to perform autonomous inspections. Some researchers have investigated trajectory planning techniques for multirotors perching on variant surfaces [13]–[16] and cables [17], [18]. Although these works successfully direct the multirotors towards the targets while respecting specific constraints, they often represent multirotors as inflated point masses, an oversimplification that fails to capture the complex geometry of the hybrid robot. Besides, most of them encounter difficulties in balancing complex constraints with high computational efficiency, restricting their applicability to other tasks.

Drawing inspiration from trajectory planning methods used in multirotors perching, we have pioneered a novel approach to address the gap in trajectory planning for hybrid robots. We model the trajectory planning problem as a nonlinear programming (NLP) problem, with the dynamical and geometric constraints as hard constraints, and the quality of inspection as cost function to optimize. The main contributions are as follows: 1) A versatile trajectory generation method for hybrid robot powerline inspection. This method facilitates powerline-to-powerline and air-to-powerline trajectory planning, resulting in a substantial reduction in both landing and obstacle navigation time cost. 2) A general geometric model for hybrid robot powerline inspection, including analytical formulations for both precise landing and powerline visibility maximization.

Manuscript received: September 19, 2023; Revised December 22, 2023; Accepted January 23, 2024.

This paper was recommended for publication by Editor Giuseppe Loianno upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 61973300, in part by the National Natural Science Foundation of China under Grant 62206275, in part by the National Natural Science Foundation of China under Grant 62273344 and in part by the National Natural Science Foundation of China under Grant 61973300.

[1]Zhishuo Li, Yanfeng Zhang, Zize Liang and Min Tan are with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: lizhishuo2019@ia.ac.cn; zhangyanfeng2020@ia.ac.cn; zize.liang@ia.ac.cn; min.tan@ia.ac.cn).

[2]Yunong Tian, Guodong Yang and En Li are with the Engineering Laboratory of Industrial Vision and Intelligent Equipment Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yunong.tian@ia.ac.cn; guodong.yang@ia.ac.cn; en.li@ia.ac.cn).

Digital Object Identifier (DOI): see top of this page.

3) Sufficient simulations and real-world experiments demonstrate the effectiveness of our implementation.

The remainder of this paper is organized as follows. We review the related literature in Section II. The dynamic and geometric models are developed in Section III, and the trajectory generation method is presented in Section IV. Section V provides the experimental results and analysis. Finally, Section VI concludes the paper.

II. Related works

The integrated hybrid robots that combine flying and climbing capabilities have been developed in recent years. Authors in [7] provided a detailed overview of the the mechanical structure, basic flight and climbing functionalities of hybrid robots. Mirallès *et al.* [10] utilized a monocular camera and LIDAR to determine the pose of the vehicle relative to the powerline, and developed a position-based casaded P/PI controller to align and keep the robot centered along the powerline. At last, the robot was manually landed on the powerline by the pilot. Chang *et al.* [9] use a swingable 2D Laser Range Finder to detect powerline and obstacles. The hybrid robot is commanded to approach the powerline guided by a feedback control strategy and achieves fully automated landing by reducing thrust upon alignment. Additionally, it demonstrates obstacle navigation capabilities through vertical takeoff and navigation over obstacles. Later, Li *et al.* [11] expanded the perception capabilities by deploying powerline detection and tracking methods in multi-powerline scenarios. Additionally, the landing time was slightly reduced through a multi-stage feedback control approach. These studies have notably advanced the automation of hybrid robots. However, feedback control-based maneuvering methods struggle with various constraints, including dynamic and inspection-related limitations. Additionally, the considerable time required for multi-stage autonomous landing and obstacle navigation significantly limits the endurance capacity of hybrid robots. Consequently, it is essential to explore trajectory planning methods to improve the efficiency and endurance of hybrid robots in autonomous inspections.

While the literature directly addressing hybrid robots remains limited, we expand our investigation to include the pertinent domain of multirotors perching due to its intrinsic connection. Recently, many works [13]–[15] leveraged the differential flatness property [19] to carry out trajectory planning in flat space. Thomas *et al.* [13] formulated a Quadratic Program (QP) for perching on the inclined surfaces. The trajectory is parameterized as piece-wise polynomials, and the trajectory planning is complished by solving the QP using a series of linear approximations. Mao *et al.* [14] also formulated a QP with physical constraints and boundary conditions to perform trajectory planning. They increased the trajectories' time iteratively and recursively solved the QP to deal with nonlinear actuators constraints. However, these QP-based trajectory planning approaches have limitations in handling general nonlinear optimization tasks due to their restrictions on quadratic objective functions and linear constraints.

With advancements in onboard computer performance, direct trajectory optimization using nonlinear programming (NLP) and collocation [20] has been applied to aerial robots [16]–[18], [21]. Similar to QP-based methods, this approach formulates an optimization problem, but the primary idea is to directly optimize control inputs or states over a discretized time horizon, considering system dynamics and nonlinear constraints. Vlantis *et al.* [16] developed a discrete-time nonlinear model predictive controller (MPC) with the objective of landing a multirotor on a moving inclined platform. The MPC is designed to minimize errors in position, attitude, and velocity while simultaneously avoiding collisions. However, due to the problem's complexity and high computing demands, it must be solved using ground stations. Later, Popek *et al.* [17] proposed a direct collocation approach for multirotor vehicle perching on cylinders. This method uses splines to parameterize the system's state and input, allowing for onboard processing. Yet, their method primarily addresses point-to-point planning without integrating obstacle avoidance or comprehensive perching constraints. The perception-aware model predictive control (PAMPC) framework [21] further extends NLP's utility in multirotors, optimizing action and perception objectives. This framework was demonstrated in a simple experiment where it maintained the visibility of a target point during a circular flight. Building on PAMPC [21], Paneque *et al.* [18] formulated an NLP for powerline perching. It computes a dynamically feasible, collision-free, and perception-aware solution to guide the multirotor to the desired final state. This study treats the multirotor as an ellipsoid and optimizes the final state by minimizing the terminal position errors. However, it does not restrict the multirotor's path to the final state, which could pose collision risks for hybrid robots. Besides, the complexity of multirotor models and constraint formulation leads to a significant computational load, with solving times ranging from 400 to 1500 milliseconds.

These NLP-based approaches, while highly task-specific and context-sensitive, have computation and convergence times that depend on the problem's formulation and initial conditions, making them challenging to adapt to other tasks. We take inspiration in these works and introduce a novel NLP framework for hybrid robots trajectory planning. This framework effectively addresses the identified gaps in computational efficiency and trajectory safety by utilizing simplified dynamic models and appropriate constraints and objectives.

III. Mathematical modeling

This section presents the dynamic model of the hybrid robot and the geometric model for powerline inspection task. These models are integrated as constraints or objectives in the NLP problem discussed in Section IV to guide the trajectory generation process.

A. Nomenclature

In this work, we follow common mathematical conventions and denote scalars with lowercase letters, vectors with boldface lowercase letters, and matrices with boldface uppercase letters. We make use of a world frame defined by axes $\{\mathbf{x}_w, \mathbf{y}_w, \mathbf{z}_w\}$, with $\mathbf{z}_w$ pointing upward. The body frame is represented

by $\{\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b\}$, with $\mathbf{x}_b$ coinciding with the preferred forward direction and $\mathbf{z}_b$ pointing vertically up during perfect hover. We denote the translations and orientations between two coordinate frames as vector $\mathbf{p}_{F_1F_2}$ and unit quaternion $\mathbf{q}_{F_1F_2}$, respectively. Consequently, a vector $\mathbf{v}_{F_2}$ in frame F_2 is expressed in frame F_1 as: $\mathbf{v}_{F_1} = \mathbf{q}_{F_1F_2} \odot \mathbf{v}_{F_2} + \mathbf{p}_{F_1F_2}$. The operator $\odot$ denotes the quaternion product, which combines the rotational information of quaternion with the vector to produce the rotated vector. Moreover, we use subscripts "$,x$", "$,y$", and "$,z$" to indicate the x, y, and z coordinates of a vector, respectively (e.g., $\mathbf{p}_{F_1F_2,x}$ denotes the x-coordinate of the vector $\mathbf{p}_{F_1F_2}$).

B. Dynamic Model

The dynamic model is based on a simplified model proposed in [22], which takes thrust and body rate as input. However, when generating trajectories, excessive instantaneous changes in thrust can result in infeasible trajectories due to the physical limitations of the robot. Therefore, we replace the thrust with its derivative as the system input, as suggested in [23]. This allows the system to generate smoother and more stable trajectories that can be effectively tracked by the robot. The 11-dimensional robot state space is defined as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}}_{wb} \\ \dot{\mathbf{q}}_{wb} \\ \dot{\mathbf{v}}_{wb} \\ \dot{f} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{wb} \\ \frac{1}{2}\Lambda(\boldsymbol{\omega}_b)\mathbf{q}_{wb} \\ \mathbf{q}_{wb} \odot f\mathbf{z}_w + g\mathbf{z}_w \\ \tau \end{bmatrix}, \tag{1}$$

where $\mathbf{x}$ represents the system state variables, $\mathbf{p}_{wb}$ and $\mathbf{q}_{wb}$ are the position and orientation of the body frame with respect to the world frame, $\mathbf{v}_{wb}$ and $\boldsymbol{\omega}_b$ are the linear and angular velocity of the robot expressed in world and body frame respectively, f and τ denote the mass-normalized net thrust and its derivative, $\Lambda(\boldsymbol{\omega}_b)$ is a skew-symmetric matrix derived from $\boldsymbol{\omega}_b$

$$\Lambda(\boldsymbol{\omega}_b) = \begin{bmatrix} 0 & -\boldsymbol{\omega}_{b,x} & -\boldsymbol{\omega}_{b,y} & -\boldsymbol{\omega}_{b,z} \\ \boldsymbol{\omega}_{b,x} & 0 & \boldsymbol{\omega}_{b,z} & -\boldsymbol{\omega}_{b,y} \\ \boldsymbol{\omega}_{b,y} & -\boldsymbol{\omega}_{b,z} & 0 & \boldsymbol{\omega}_{b,x} \\ \boldsymbol{\omega}_{b,z} & \boldsymbol{\omega}_{b,y} & -\boldsymbol{\omega}_{b,x} & 0 \end{bmatrix}. \tag{2}$$

C. Landing on Powerline

Powerline can be represented by line segment since trajectory planning only involves a short segment of powerline. The vector form equation of the powerline is given by $\mathbf{l}_s + \sigma\mathbf{l}_v$, where $\mathbf{l}_s$ is the position vector of a known point on the powerline, $\mathbf{l}_v$ is the direction vector of the powerline, and σ is a parameter.

Let the guiding mechanism of the hybrid robot be modeled as a tri-prism with two parallel triangular faces (P_1 and P_2) and length l, see Fig. 2. The base of the triangle is $\|\mathbf{ef}\|$, the height is h, and the vertex angle is θ. The coordinates of the vertices of the triangle in the body frame is given by $\mathbf{p'}_b = (-\frac{l}{2}, 0, 0)$, $\mathbf{e}_b = (-\frac{l}{2}, h\tan\frac{\theta}{2}, -h)$, $\mathbf{f}_b = (-\frac{l}{2}, -h\tan\frac{\theta}{2}, -h)$. We can transform them to the world frame:

$$\mathbf{p'}_w = \mathbf{q}_{wb} \odot \mathbf{p'}_b + \mathbf{p}_{wb}, \tag{3a}$$

$$\mathbf{e}_w = \mathbf{q}_{wb} \odot \mathbf{e}_b + \mathbf{p}_{wb}, \tag{3b}$$

$$\mathbf{f}_w = \mathbf{q}_{wb} \odot \mathbf{f}_b + \mathbf{p}_{wb}. \tag{3c}$$

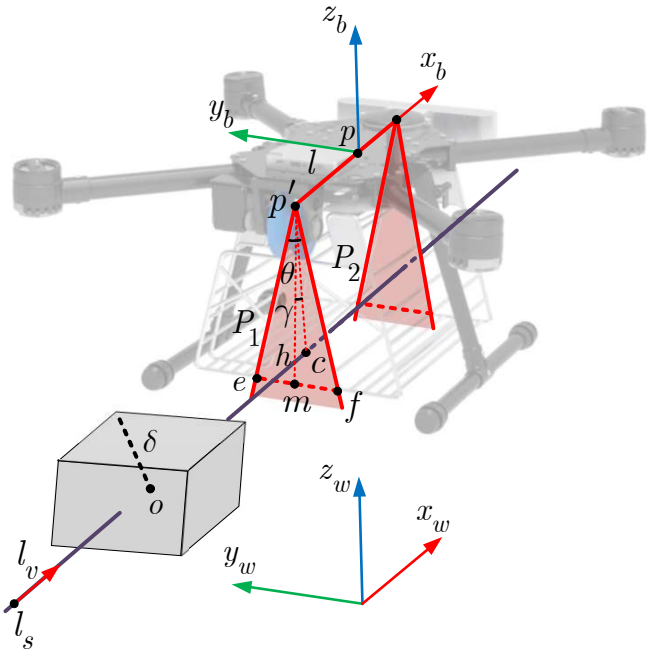

Fig. 2. The geometry of the hybrid robot and its workspace.

Additionally, m is the midpoint of $\mathbf{ef}$, $\mathbf{n}_w = \mathbf{p'e}_w \times \mathbf{p'f}_w$ is the normal vector of plane P_1, and $\mathbf{c}_w = \mathbf{l}_s + \lambda\mathbf{l}_v$ is the intersection point of the powerline and the plane P_1, where λ can be derived from $\mathbf{p'c}_w \perp \mathbf{n}_w$.

Having established the geometric models for the hybrid robot and the powerline, ensuring the robot's successful landing on the powerline can be reduced to verifying that the powerline's intersection points with the triangular planes are both situated within the triangles. For plane P_1, we can achieve this by restricting the cosine value of the angle γ_{P_1} between $\mathbf{p'm}_w$ and $\mathbf{p'c}_w$ to be greater than $\cos\frac{\theta}{2}$:

$$\cos\gamma_{P_1} - \cos\frac{\theta}{2} = \frac{\mathbf{p'm}_w \cdot \mathbf{p'c}_w}{\|\mathbf{p'm}_w \cdot \mathbf{p'c}_w\|} - \cos\frac{\theta}{2} > 0. \tag{4}$$

The constraint activation is required only as the hybrid robot nears the target point. On the one hand, the initial position may potentially conflict with the constraint, consequently rendering the NLP problem infeasible. On the other hand, the robot should complete most of its rotation before landing on the powerline, instead of attempting to pivot at the target. This aspect is addressed by incorporating an additional position-related term into the constraint. The minimum value of (4) is $(-1-\cos\frac{\theta}{2})$ when $\mathbf{p'm}_w$ and $\mathbf{p'c}_w$ are opposite in direction. Consequently, we can force the constraint (4) to always hold true by summing $(1+\cos\frac{\theta}{2})$ to it. We define the following function to add that value whenever the hybrid robot is outside of the target's surroundings

$$g(\mathbf{x}) = sigm(\|\mathbf{p}_{wb} - \mathbf{p}_{\mathbf{r}w}\|)(1+\cos\frac{\theta}{2}), \tag{5}$$

where $\mathbf{p}_{\mathbf{r}w}$ is the target point in the world frame, $\|\mathbf{p}_{wb} - \mathbf{p}_{\mathbf{r}w}\|$ is the distance from the hybrid robot to the target point, and $sigm(\cdot)$ is an activation function that defined as

$$sigm(\cdot) = \frac{1}{2} + \frac{1}{2}tanh(k(\cdot - d)). \tag{6}$$

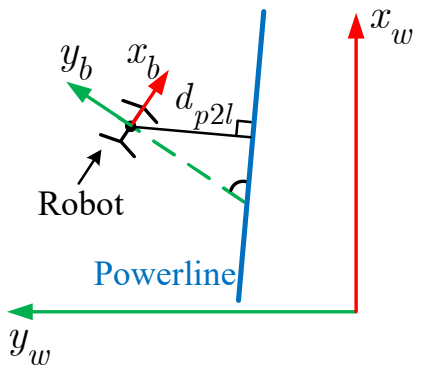

Fig. 3. Visualization of proposed geometric modeling for (12) and (13).

which is approximately 1 when the distance is greater than a threshold d, and close to 0 otherwise. Finally, the constraint that enforces the hybrid robot landing on the powerline comes to be

$$h_{lpP_1}(\mathbf{x}) = \frac{\mathbf{p}'\mathbf{m}_w \cdot \mathbf{p}'\mathbf{c}_w}{\|\mathbf{p}'\mathbf{m}_w \cdot \mathbf{p}'\mathbf{c}_w\|} - \cos\frac{\theta}{2} + g(\mathbf{x}) > 0. \quad (7)$$

Similarly, we can derive another constraint based on the relationship between the powerline and plane P_2

$$h_{lp_{P_2}}(\mathbf{x}) > 0. \quad (8)$$

Regardless of whether the robot's heading aligns with or opposes the powerline's direction, the landing constraints can be satisfied. Consequently, the robot's heading should be constrained to align with the powerline's direction. We can achieve this by constraining the cosine value of the x-axis (representing the heading of the hybrid robot) and the direction vector of the powerline to be greater than 0

$$h_{dir}(\mathbf{x}) = \frac{\mathbf{q}_{wb} \odot \mathbf{x}_b \cdot \mathbf{l}_v}{\|\mathbf{q}_{wb} \odot \mathbf{x}_b \cdot \mathbf{l}_v\|} + k(\mathbf{x}) > 0. \quad (9)$$

where $\mathbf{q}_{wb} \odot \mathbf{x}_b$ represents the x-axis of the body frame in the world frame, $k(\mathbf{x})$ is an activation function defined for the same purpose as in (5)

$$k(\mathbf{x}) = 1sigm(\|\mathbf{p}_{wb} - \mathbf{p}_{\mathbf{r}_w}\|). \quad (10)$$

D. Powerline inspection

To ensure the inspection quality during the maneuver, the hybrid robot should be as close as possible to the powerline in the horizontal direction and align its head with the direction of the powerline. To solve this, we first minimize the distance of their projection onto the $\mathbf{x}_w$-$\mathbf{y}_w$ plane. As shown in Fig. 3, the projection of the powerline can be approximated as a line segment, and its general equation is given by

$$Ax + By + C = 0, \quad (11)$$

where $A = -\mathbf{l}_{v,y}$, $B = \mathbf{l}_{v,x}$, $C = \mathbf{l}_{v,y}\mathbf{l}_{s,x} - \mathbf{l}_{v,x}\mathbf{l}_{s,y}$. And the projection of the hybrid robot in the $\mathbf{x}_w$-$\mathbf{y}_w$ plane can be considered as a point represented by $(\mathbf{p}_{wb,x}, \mathbf{p}_{wb,y})$. Therefore, the distance between them can be calculated as

$$d_{p2l} = \frac{|A\mathbf{p}_{wb,x} + B\mathbf{p}_{wb,y} + C|}{\sqrt{A^2 + B^2}}. \quad (12)$$

It is worth noting that aligning the hybrid robot with the powerline is equivalent to making its $\mathbf{y_b}$ axis perpendicular to the powerline. Hence, we further minimize the absolute value of the dot product between the hybrid robot's $\mathbf{y_b}$ axis and the direction vector of the powerline

$$a_{p2l} = |\mathbf{q}_{wb} \odot \mathbf{y}_b \cdot \mathbf{l}_v|. \quad (13)$$

The minimum values of (12) and (13) occur when the hybrid robot is positioned directly above and aligned with the powerline. This configuration ensures optimal visibility of the powerline and facilitates the hybrid robot's landing on it.

E. Obstacle Navigation

When encountering obstacles on the powerline, the hybrid robot sets the target point to a location on the powerline in front of the obstacle and carries out powerline-to-powerline trajectory planning. If the obstacle blocks the view of the powerline ahead, the robot initially sets the target point above the obstacle to gain a forward view. Subsequently, the hybrid robot sets the target point to a location on the powerline in front of the obstacle and carries out air-to-powerline trajectory planning. To avoid collisions with obstacles, we model the hybrid robot as a sphere with radius r, and simplify the obstacle as its minimum bounding box centered at $\mathbf{o}$, with a diagonal radius of δ. The obstacle navigation formula is derived as follows

$$h_{oa}(\mathbf{x}) = \|\mathbf{p}_{wb} - \mathbf{o}_w\| - \delta - r - \epsilon > 0, \quad (14)$$

where ϵ represents a safety margin.

As for collision avoidance with powerlines, we cannot treat the hybrid robot as a sphere due to its need to perch on the powerline. Referring back to the aforementioned perching constraints (7) and (8), they not only ensure that the hybrid robot perches on the powerline but also enforce the continuous alignment of the powerline within the extension range of the guiding mechinism $\mathbf{p}'\mathbf{e}$ and $\mathbf{p}'\mathbf{f}$, which effectively prevents lateral collisions between the robot and the powerline. Furthermore, considering the inspection task, we impose a motion constraint to confine the hybrid robot's movement exclusively to the space above the powerline.

IV. Trajectory Generation

The trajectory planning problem is formulated as a discrete time multiple-shooting NLP problem, where the trajectory is discretized into N shooting nodes over a variable time horizon T. Each shooting node is associated with a set of variables that represent the state and control input of the system. And the objective is to find the optimal values for the state and control variables at the discrete time points that minimize the

cost function while satisfying the constraints. The complete formulation of the optimization problem is provided below:

$$\min \quad \sum_{k=0}^{N-1} L_k^T \mathbf{Q}_k L_k + M_N^T \mathbf{Q}_N M_N \tag{15a}$$

$$\text{s.t.} \quad \mathbf{x}_0 = \mathbf{x}_{init} \tag{15b}$$

$$T_{min} < T < T_{max} \tag{15c}$$

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \quad \forall k \in [0, N-1] \tag{15d}$$

$$0 < f_k < f_{max} \quad \forall k \in [0, N] \tag{15e}$$

$$\tau_{min} < \tau_k < \tau_{max} \quad \forall k \in [0, N] \tag{15f}$$

$$\mathbf{v}_{min} < \mathbf{v}_k < \mathbf{v}_{max} \quad \forall k \in [0, N] \tag{15g}$$

$$\boldsymbol{\omega}_{min} < \boldsymbol{\omega}_k < \boldsymbol{\omega}_{max} \quad \forall k \in [0, N] \tag{15h}$$

$$z_{min} < \mathbf{p}_{wb,zk} \quad \forall k \in [0, N] \tag{15i}$$

$$0 < h_{lpP_1}(\mathbf{x}_k) \quad \forall k \in [0, N] \tag{15j}$$

$$0 < h_{lpP_2}(\mathbf{x}_k) \quad \forall k \in [0, N] \tag{15k}$$

$$0 < h_{dir}(\mathbf{x}_k) \quad \forall k \in [0, N] \tag{15l}$$

$$0 < h_{oa}(\mathbf{x}_k) \quad \forall k \in [0, N]. \tag{15m}$$

The construction of problem (15) is described as follows: (15a) is the cost function to minimize (detailed in the next paragraph); (15b) is the initial values of the state variables; (15c) impose a constraint on the total maneuvering time T; (15d) represents the discrete-time dynamic model. It is discretized by the 4th order Runge-Kutta (RK4) method, which approximates the continuous-time dynamic model at discrete time intervals with a fixed time step $T/(N-1)$; (15e) and (15f) are the limits of thrust and its derivative; (15g) and (15h) are the constraints for the linear and angular velocities; (15i) represents the height restriction for the hybri robot; (15j) and (15k) are the constraints for the hybrid robot to land on the powerline that defined by (7) and (8); (15l) is the landing direction constraint that defined by (9); (15m) is the obstacle navigation constraint defined by (14).

The cost function consist of a combination of terms that capture various aspects of the trajectory. At each shooting node, these terms are assigned weight factors through a diagonal matrix Q. The first component of the cost function represents the running cost, which is given below

$$L_k = [\tau_k \quad \boldsymbol{\omega}_k^T \quad \tilde{\mathbf{p}}_k^T \quad d_{p2lk} \quad a_{p2lk}]^T. \tag{16}$$

It minimizes the derivative of the thrust and the body rate to encourage smooth control inputs, as well as d_{p2lk} and a_{p2lk} to align the hybrid robot with the powerline. Furthermore, it minimizes the position error $\tilde{\mathbf{p}}_k^T$ defined as the difference between the actual and reference positions in world frame coordinates. Consequently, the hybrid robot can reach the destination as soon as possible while respecting the actuator constraints. The second component corresponds to the terminal cost, which is defined as

$$M_N = [\tilde{\mathbf{p}}_N^T \quad \tilde{\mathbf{q}}_N^T \quad \tilde{\mathbf{v}}_N^T \quad \tilde{\boldsymbol{\omega}}_N^T]^T. \tag{17}$$

It minimizes the cost associated with reaching the desired final state $\mathbf{x}_r$ at the endpoint, considering the position and orientation errors $\tilde{\mathbf{p}}_N^T$, $\tilde{\mathbf{q}}_N^T$, as well as the linear and angular velocity errors $\tilde{\mathbf{v}}_N^T$, $\tilde{\boldsymbol{\omega}}_N^T$.

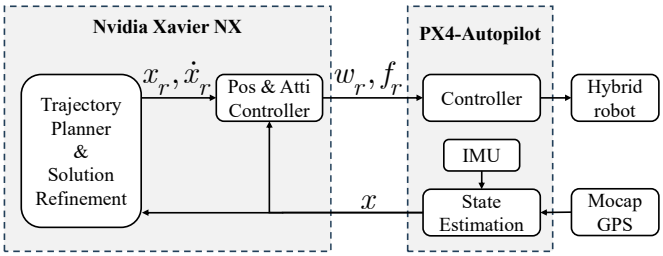

Fig. 4. The architecture for estimation, planning, control.

We leverage the ForcesPRO framework [24], [25] to solve our NLP formulation, which offers a powerful toolset to tackle optimization problems efficiently and effectively. For discretization, we select $N = 51$ with a time horizon of T that falls within the range of 3 to 15 seconds. Convergence is typically achieved within 20 to 70 iterations, and the computation time remains below 50 ms. After the optimization process completed, we employ RK4 intergration to enhance the trajectory resolution, which involves subdividing each time interval into smaller sub-intervals and iteratively calculating the trajectory using RK4 within each sub-interval. Then we perform collision detection between the generated high-resolution trajectory and the obstacle. If a collision is detected, we proceed to replan the trajectory by increasing N and repeat collision detection process until a collision-free trajectory is obtained.

V. Experiments

We evaluated the proposed method onboard a hybrid robot developed in our previous work [11]. It weighs 3.5kg and has a flight time over 20 minutes. The trapezoidal-shaped guiding mechinism allows for self-alignment with the powerline, thereby enhancing the fault tolerance during the robot's descent. Fig. 4 presents the architecture and a detailed description of the system. The state estimates of the hybrid robot are derived from an Extended Kalman Filter (EKF) that fuses data from external positioning systems like the NOKOV Motion Capture (Mocap) System or GPS, with IMU data from the PX4 Autopilot. Additionally, the positions of the powerlines and obstacles are established within the coordinate framework of these external positioning systems. An Nvidia Jetson Xavier NX is used to perform trajectory planning and acts as a high level flight controller. The control actuations are computed by a position and attitude controller described at [26], which takes the reference state vector and produces the desired net thrust and body rate. The PX4-Autopilot is used as a low-level controller that translates the aforementioned commands into individual motor commands.

A. Simulation Experiments

In order to evaluate the impact of landing constraint and inspection cost on our proposed trajectory generation method, we simulated a powerline in Matlab and placed an obstacle above it, and then conducted a series of ablation experiments. First, the hybrid robot was initially set to hover above the powerline with its head oriented 60 degrees relative to the

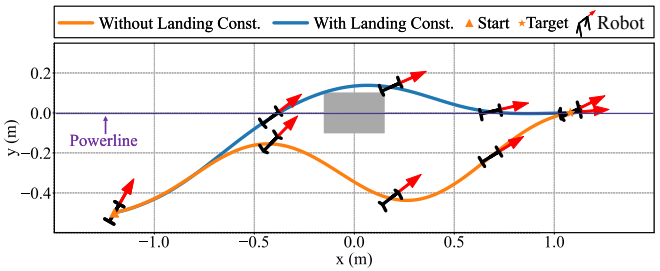

Fig. 5. Comparison of two trajectories with and without landing constraints.

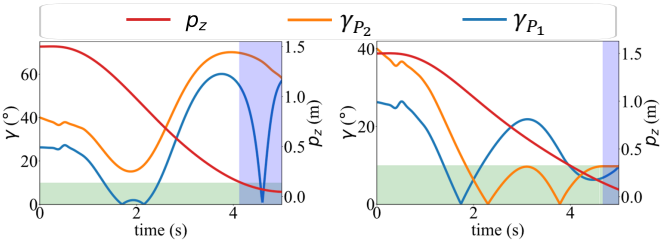

Fig. 6. The relative altitude between the hybrid robot and the powerline and angle γ during the experiment in Fig. 5. The left subfigure corresponds to trajectory without landing constraint, while the right subfigure corresponds to trajectory with landing constraint.

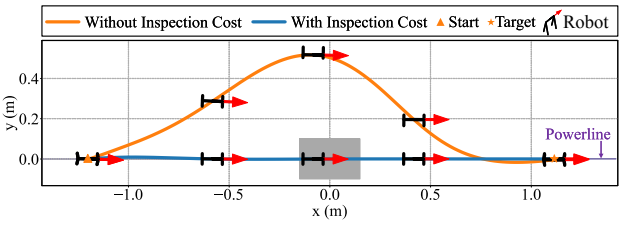

Fig. 7. Comparison of two trajectories with and without inspection cost.

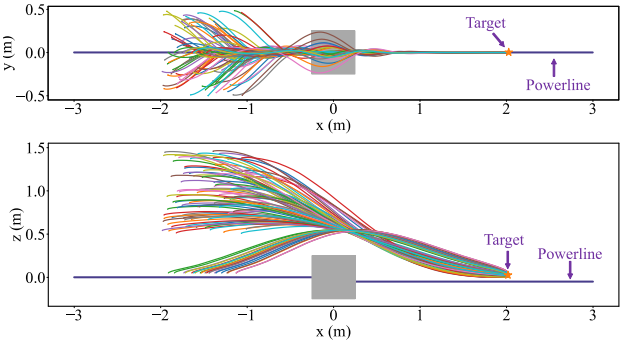

Fig. 8. Visualization of 100 simulated trajectories.

powerline, and the target point was set as a point on the powerline. We removed the inspection cost from problem (15), and then generated trajectories with and without including the landing constraints (15j), (15k) and (15l) respectively, shown in Fig. 5. Fig. 6 shows the angle γ_{P_1} and γ_{P_2} that defined in Section III.C, as well as the relative altitude between the hybrid robot and the powerline. The green region represents angles smaller than $\theta/2$, while the purple region represents relative altitudes smaller than the height of the guiding mechanism. In the case of the trajectory with the landing constraint, when the relative altitude is below the height of the guiding mechanism, the angle is smaller than $\theta/2$. However, for the unconstrained trajectory, when the relative altitude is below the height of the guiding mechanism, the angle is not within $\theta/2$, indicating that the hybrid robot cannot catch the powerline. Experimental results demonstrate the effective nature of our landing constraint in ensuring the hybrid robot's successful landing on the powerline.

Next, the robot was positioned on the powerline, with the target point set to a location on the powerline behind the obstacle. We maintained the landing constraints and generated two trajectories with and without including the inspection cost, as shown in Fig. 7. Both trajectories allow the robot to navigate the obstacle and land back on the powerline. However, in the trajectory without inspection cost, the hybrid robot navigates the obstacle from the side, while in the trajectory with inspection cost, it navigates the obstacle from above, maintaining constant visibility of the powerline. This contrast illustrates the effectiveness of the proposed inspection cost.

Finally, we randomly selected 20 initial points where the robots were perched on the powerline, and 80 initial points within a $1\times1\times1$ cubic region centered 1 meter above the powerline. The initial x-axis of the body frame was set at an angle ranging from -30 to 30 degrees relative to the direction of the powerline. The target point was positioned to the right of the powerline. We then generated trajectories using the complete formula (15) and presented the results in Fig. 8. Our method achieves a 100% successful landing rate, which is defined as the hybrid robot effectively navigates the obstacle and lands on the powerline. Additionally, the hybrid robot quickly aligned itself with the powerline to maintain visibility during both trajectories.

B. Obstacle Navigation and Landing Experiments

To validate the effectiveness of our proposed methods in real-world, we setup a powerline and a obstacle in the room and conducted several experiments for obstacle navigation and landing on the powerline, shown in Fig. 9. First, we placed the hybrid robot on a powerline, and leveraged the trajectory planning in Section IV to generate trajectory to navigate the obstacle. Then the hybrid robot was commanded to perform the maneuver following the scheme in Fig. 4. We present the trajectory planning and tracking results in Fig. 10, and also plot the simplified hybrid robot with the x-axis of the body frame to further show the orientation. As depicted in the top view, the planned trajectory keeps the hybrid robot positioned directly above the powerline, ensuring maximum visibility of the powerline. Additionally, it can be observed that the hybrid robot is capable of navigating the obstacle when viewed from the side. Based on the final states of the hybrid robot depicted in both aforementioned figures, we can see that the hybrid robot ultimately lands back on the powerline.

Next, the hybrid robot was brought above the powerline with a specific orientation relative to it, and then carried out air-to-powerline trajectory planning and tracking. As illustrated in Fig. 11, our inspection cost imposed through (15a) enforce

TABLE I
RMSE AND SUCCESSFUL LANDING RATE

Component	Powerline-to-Powerline	Air-to-Powerline
RMSE-x (m)	0.020	0.011
RMSE-y (m)	0.025	0.028
RMSE-z (m)	0.007	0.018
RMSE-yaw (rad)	0.023	0.016
Successful Landing Rate	4/4	4/4

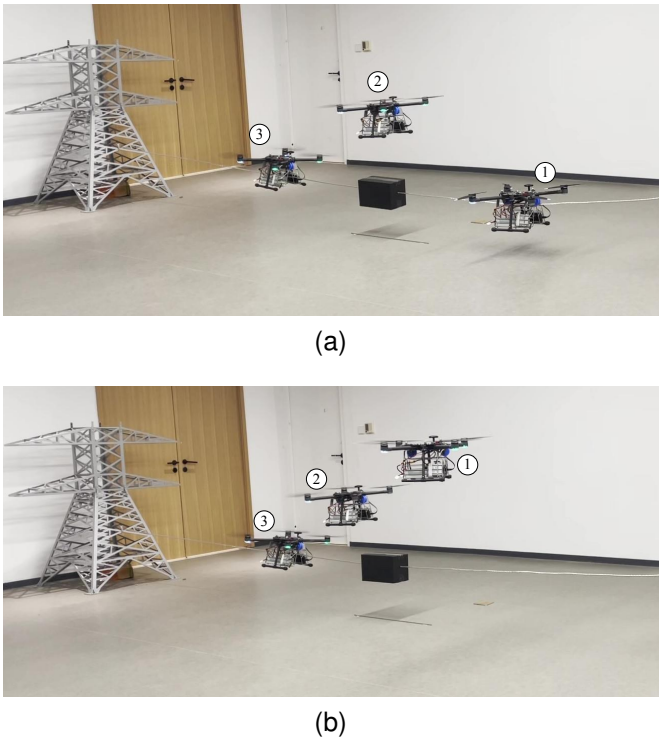

Fig. 9. Obstacle navigation and landing maneuver for (a) powerline-to-powerline and (b) air-to-powerline. The numbers denote the evolution of the trajectory.

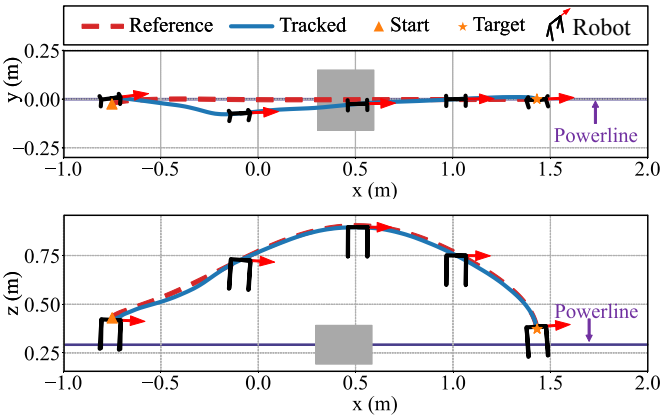

Fig. 10. Top and side view of trajectory planning and tracking for powerline-to-powerline.

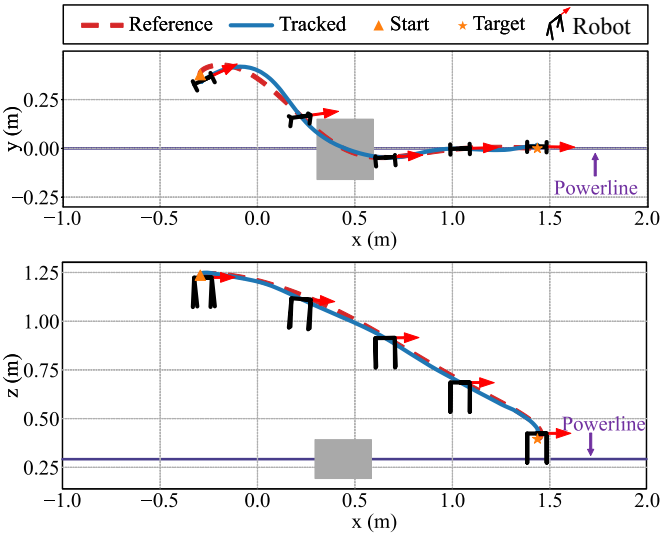

Fig. 11. Top and side view of trajectory planning and tracking for air-to-powerline.

the hybrid robot aligns itself with the powerline as quickly as possible. Moreover, the hybrid robot achieves a successful landing onto the powerline without any collisions.

Furthermore, we conducted a total of 8 iterations for trajectory planning and tracking procedures involving both powerline-to-powerline and air-to-powerline scenarios, and recorded the successful landing rate and the mean Root Mean Square Error (RMSE) as seen in Table I. The position tracking errors remained below 0.03 m, while the yaw tracking error was less than 2 degrees. Notably, these errors are mostly at the beginning of trajectories, gradually decreasing towards the end, as depicted in Fig. 10 and Fig. 11. Considering the margin brought by the guiding mechanism, the accuracy can meet the requirements of the landing task. Finally, a video of the experimental results is provided in the supplementary material.

C. Comparison

Lastly, we conducted outdoor landing and obstacle navigation experiments (see Fig. 12) following the architecture depicted in Fig. 4, and comprehensively compared the performance of our hybrid robot with the existing hybrid robots [9]–[11]. The system performance statistics are shown in Table II. In contrast to previous research, we are the first to propose a trajectory planning method for hybrid robots. The trajectories we planned take into consideration dynamics constraints, energy consumption, and constraints associated with inspection tasks, making them smoother and more practical. As a result, the landing time (when landing from 1 m above and 0.2 m to the side of the powerline) and obstacle navigation time are significantly reduced. Furthermore, the planned trajectories maximize the visibility of the powerline during the maneuver by incorporating inspection cost, which is an aspect that cannot be taken into account by feedback-control-based methods. However, since our work primarily focuses on planning, it lacks perception capabilities.

VI. CONCLUSION

In this work, we present an NLP-based trajectory generation framework for hybrid robot powerline inspection. Our method is capable of planning a dynamically feasible trajectory that

Fig. 12. Obstacle navigation and landing maneuver. The numbers denote the evolution of the trajectory.

TABLE II
SYSTEM PERFORMANCE COMPARISON BETWEEN HYBRID UAV [9], LINEDRONE [10], HYBRID ROBOT [11] AND OURS.

Model	Hybrid UAV	LineDrone	Hybrid robot	Ours
Landing	**Yes**	**Yes**	**Yes**	**Yes**
Landing time (s)	20	60	14	**<5**
Obstatcle Navigation	**Yes**	No	No	**Yes**
Navigation time (s)	14	/	/	**<8**
Inspection	No	No	No	**Yes**
Perception	**Yes**	**Yes**	**Yes**	No

enables collision-free landing on the powerline while maintaining maximum visibility of the powerline during the flight. A series of simulations and real-world experiments demonstrated the robustness of our method and notably enhanced inspection performance. We demonstrate that the hybrid robot can generate and execute both the powerline-to-powerline and air-to-powerline trajectories within 50 ms, leveraging onboard computational resources and an external positioning system. Additionally, we show that our method significantly reduce the landing and obstacle navigation time, potentially improving inspection efficiency. Due to the lack of onboard sensing in our work, our method is currently not applicable in scenarios without pre-existing maps. In the future, we will extend our method to autonomous powerline inspection where the estimation, powerline and obstacle detection are obtained totally from on-board sensors.

REFERENCES

[1] R. S. Goncalves and J. C. M. Carvalho, "A mobile robot to be applied in high-voltage power lines," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 37, pp. 349–359, 2015.

[2] N. Pouliot and S. Montambault, "Linescout technology: From inspection to robotic maintenance on live transmission power lines," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 1034–1040.

[3] P.-L. Richard, N. Pouliot, F. Morin, M. Lepage, P. Hamelin, M. Lagac, A. Sartor, G. Lambert, and S. Montambault, "Lineranger: Analysis and field testing of an innovative robot for efficient assessment of bundled high-voltage powerlines," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9130–9136.

[4] J. Bian, X. Hui, X. Zhao, and M. Tan, "A novel monocular-based navigation approach for uav autonomous transmission-line inspection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–7.

[5] A. Dietsche, G. Cioffi, J. Hidalgo-Carrió, and D. Scaramuzza, "Powerline tracking with event cameras," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6990–6997.

[6] C. Xu, Q. Li, Q. Zhou, S. Zhang, D. Yu, and Y. Ma, "Power line-guided automatic electric transmission line inspection system," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–18, 2022.

[7] H. Wang, E. Li, G. Yang, and R. Guo, "Design of an inspection robot system with hybrid operation modes for power transmission lines," in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2019, pp. 2571–2576.

[8] X. Qin, B. Jia, J. Lei, J. Zhang, H. Li, B. Li, and Z. Li, "A novel flying–walking power line inspection robot and stability analysis hanging on the line under wind loads," *Mechanical Sciences*, vol. 13, no. 1, pp. 257–273, 2022.

[9] W. Chang, G. Yang, J. Yu, Z. Liang, L. Cheng, and C. Zhou, "Development of a power line inspection robot with hybrid operation modes," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 973–978.

[10] F. Mirallès, P. Hamelin, G. Lambert, S. Lavoie, N. Pouliot, M. Montfrond, and S. Montambault, "Linedrone technology: Landing an unmanned aerial vehicle on a power line," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6545–6552.

[11] Z. Li, Y. Tian, G. Yang, E. Li, Y. Zhang, M. Chen, Z. Liang, and M. Tan, "Vision-based autonomous landing of a hybrid robot on a powerline," *IEEE Transactions on Instrumentation and Measurement*, 2022.

[12] P. Hamelin, F. Mirallès, G. Lambert, S. Lavoie, N. Pouliot, M. Montfrond, and S. Montambault, "Discrete-time control of linedrone: An assisted tracking and landing uav for live power line inspection and maintenance," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 292–298.

[13] J. Thomas, M. Pope, G. Loianno, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, and V. Kumar, "Aggressive flight with quadrotors for perching on inclined surfaces," *Journal of Mechanisms and Robotics*, vol. 8, no. 5, p. 051007, 2016.

[14] J. Mao, G. Li, S. Nogar, C. Kroninger, and G. Loianno, "Aggressive visual perching with quadrotors on inclined surfaces," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5242–5248.

[15] J. Ji, T. Yang, C. Xu, and F. Gao, "Real-time trajectory planning for aerial perching," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 516–10 522.

[16] P. Vlantis, P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Quadrotor landing on an inclined platform of a moving ground vehicle," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2202–2207.

[17] K. M. Popek, M. S. Johannes, K. C. Wolfe, R. A. Hegeman, J. M. Hatch, J. L. Moore, K. D. Katyal, B. Y. Yeh, and R. J. Bamberger, "Autonomous grasping robotic aerial system for perching (agrasp)," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[18] J. L. Paneque, J. R. Martínez-de Dios, A. Ollero, D. Hanover, S. Sun, A. Romero, and D. Scaramuzza, "Perception-aware perching on powerlines with multirotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3077–3084, 2022.

[19] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 2520–2525.

[20] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of guidance, control, and dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[21] 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.

[22] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE transactions on robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[23] M. Jacquet, G. Corsini, D. Bicego, and A. Franchi, "Perception-constrained and motor-level nonlinear mpc for both underactuated and tilted-propeller uavs," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4301–4306.

[24] A. Domahidi and J. Jerez, "Forces professional," Embotech AG, url=https://embotech.com/FORCES-Pro, 2014–2019.

[25] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "Forces nlp: an efficient implementation of interior-point... methods for multistage nonlinear nonconvex programs," *International Journal of Control*, pp. 1–17, 2017.

[26] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2017.