2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Obstacle-Aware IBVS Target Tracking via Feature-Space Projection and Virtual Imaging Guidance with ADP-Shaped Terminal Cost

Mingcong Li, Zhen Chen*, and Xiangdong Liu

***Abstract*— We propose a visual-servoing and obstacle-avoidance controller for a wheeled mobile robot (WMR) with a two-axis gimbal camera that operates without mapping, using only vision and lightweight forward sensing. A task-allocation MPC with online terminal-cost iteration is introduced. Specifically, task projection in the image-feature space mitigates underactuation and coupling–induced local optima; Virtual Imaging Constraint Guidance (VICG) yields a visibility-preserving heading reference that steers the trajectory around obstacles; and an Approximate Dynamic Programming (ADP) module learns a context-aware terminal cost online, providing long-horizon guidance for mid-horizon prediction. Relying solely on image feedback plus lightweight ranging, the method coordinates the WMR and gimbal to accomplish obstacle avoidance and visual-servo tracking jointly. Hardware experiments validate the feasibility and effectiveness of the proposed approach.**

I. INTRODUCTION

Vision-guided mobile robotics is a cornerstone technology for applications such as infrastructure inspection, autonomous surveillance, and search-and-rescue operations, where perception and control must be tightly integrated [1], [2]. Among vision-based control paradigms, Image-Based Visual Servoing (IBVS) is particularly advantageous as it directly regulates features in the image plane, granting it inherent robustness to calibration and sensing noise compared to position-based schemes [3], [4]. Mounting the camera on a two-axis gimbal substantially amplifies these capabilities by decoupling the camera's orientation from the vehicle's chassis, enabling the system to maintain target visibility even while the base executes complex maneuvers [5], [6].

However, in outdoor environments, this enhanced agility introduces a formidable control challenge. The system must simultaneously track a target and avoid obstacles, often without a reliable global map and relying only on lightweight forward sensing. This creates a tension between the local, reactive nature of IBVS and the global, predictive requirements of safe navigation (see Fig. 1). The system is both underactuated—with four control inputs for an eight-dimensional feature space—and kinematically coupled, making instantaneous error annihilation generically infeasible. A structural tension thus exists: aggressive servoing actions may compromise safety, while prioritizing safety can degrade tracking performance or lead to navigation failure [2]. A particularly common failure mode is the "gimbal-only" local minimum, where the gimbal perfectly tracks the target, but the robot base stalls, unable to achieve a more favorable position. Prior work has addressed this from two distinct perspectives. One line of research focuses on the image domain, developing visibility-aware IBVS controllers using tools like barrier functions [2], [7], [8], [9]. While successful, these methods often consider simpler single-body systems and do not explicitly resolve the control authority allocation between a nonholonomic base and an active gimbal. A complementary body of work has focused on the world-space problem, employing corridor-based abstractions to generate smooth, safe paths [10], [11], [12], but often lacks a deep integration with the underlying image-feature dynamics.

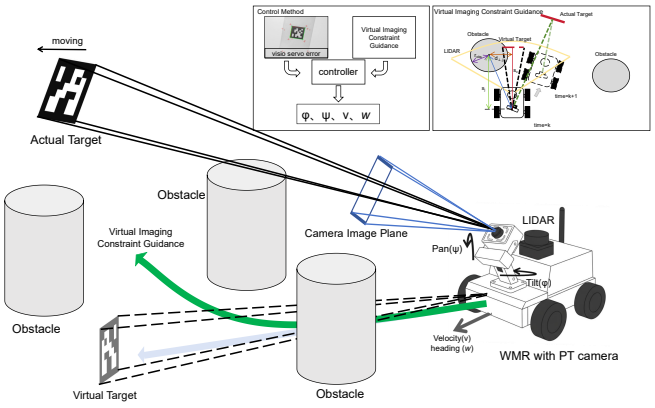

Fig. 1: Conceptual diagram of the CoVG-MPC framework. The system coordinates a nonholonomic base and a gimbal camera to track a target while avoiding obstacles using a hierarchical control scheme.

Model Predictive Control (MPC) has emerged as a principled framework for arbitrating such multi-objective problems [13], [6], [14]. Yet, classical MPC is often hampered by a reliance on manually tuned, static terminal cost functions [15]. While recent advances in learning-based MPC seek to alleviate this burden [16], [17], [18], their application in visual servoing has predominantly focused on aerial platforms, without addressing the specific, tightly-coupled dynamics of a WMR-gimbal system navigating among ground-level obstacles [1], [19]. A unified framework that coordinates the base and gimbal while adaptively resolving the servoing-safety conflict online thus remains an open challenge.

We adopt MPC for its ability to impose multi-objective trade-offs and constraints, but we reshape it to address the above failure modes. First, we introduce a task projection in the image-feature space that allocates fast error suppres-

All authors are with the Institute of Electrical Engineering and Control, School of Automation, Beijing Institute of Technology, Beijing 100081, China.
*Corresponding author: Zhen Chen, chenzhen76@bit.edu.cn

sion to the gimbal and delegates the orthogonal complement to the base, preventing gimbal-only traps. Second, we propose Virtual Imaging Constraint Guidance (VICG)—a virtual camera/target construction that produces a visibility-preserving heading reference so that trajectories route around potential occlusions without relying on a global map. Third, these components are embedded in a linearized QP-MPC for real-time operation and augmented by an adaptive MPC component that learns a context-aware terminal cost online, providing long-horizon guidance for mid-horizon prediction.

The work is organized as follows: Section II formulates the problem. Section III presents the complete CoVG-MPC methodology. Section IV reports simulation and hardware experiments. Section V concludes.

II. Problem Statement

A. System Kinematics and Visual Servoing Model

We consider a wheeled mobile robot (WMR) equipped with a two-axis gimbal camera. Let $\mathscr{W}$ be the world frame, $\mathscr{B}$ the base frame at the robot center, and $\mathscr{C}$ the camera frame at the optical center. The robot state is

$$\mathbf{x} = \begin{bmatrix} x & y & \theta & \psi & \phi \end{bmatrix}^\top \tag{1}$$

where (x, y, θ) is the base planar pose (position and yaw), and (ψ, ϕ) are gimbal pan and tilt. The control input is

$$\mathbf{u} = \begin{bmatrix} v & \omega & \dot{\psi} & \dot{\phi} \end{bmatrix}^\top \tag{2}$$

with base forward speed v, base yaw rate ω, and gimbal rates $\dot{\psi}, \dot{\phi}$. A calibrated pinhole camera observes a rigid target providing four 3D points $\{P_k^{\mathscr{T}} \in \mathbb{R}^3\}_{k=1}^4$ in the target frame $\mathscr{T}$. Adopting a conventional extrinsic chain with pan-then-tilt parameterization, the target points are transformed into the camera frame. The pixel coordinates follow

$$\tilde{q}_k = \lambda_k \begin{bmatrix} u_k \\ v_k \\ 1 \end{bmatrix} = \mathbf{K} \frac{1}{Z_k} \begin{bmatrix} X_k \\ Y_k \\ Z_k \end{bmatrix}, \quad u_k = \frac{f_x X_k}{Z_k} + c_x, \; v_k = \frac{f_y Y_k}{Z_k} + c_y \tag{3}$$

Stacking the four points,

$$\mathbf{s} = \mathrm{vec}(\,[u_1, v_1, u_2, v_2, u_3, v_3, u_4, v_4]\,) \in \mathbb{R}^8, \quad \mathbf{e} = \mathbf{s} - \mathbf{s}^\star \tag{4}$$

The feature velocity relates to the camera's spatial twist $\mathbf{V}_{\mathscr{C}}$ via the standard interaction matrix $\mathscr{L}(\mathbf{s}, \mathbf{Z})$:

$$\dot{\mathbf{s}} = \mathscr{L}(\mathbf{s}, \mathbf{Z})\, \mathbf{V}_{\mathscr{C}}, \qquad \mathbf{V}_{\mathscr{C}} = \begin{bmatrix} V_x & V_y & V_z & \Omega_x & \Omega_y & \Omega_z \end{bmatrix}^\top \tag{5}$$

where $\mathbf{V}_{\mathscr{C}}$ is the spatial twist of the camera expressed in $\mathscr{C}$. Since the feature depth vector $\mathbf{Z}$ is required for the interaction matrix but is not directly measured, it is estimated online using an iterative Broyden method. The camera twist relates to the inputs via a kinematic Jacobian $J_{\mathscr{C}}(\mathbf{x}) \in \mathbb{R}^{6\times 4}$:

$$\mathbf{V}_{\mathscr{C}} = J_{\mathscr{C}}(\mathbf{x})\, \mathbf{u} \tag{6}$$

Combining these yields the overall system dynamics:

$$\dot{\mathbf{s}} = \underbrace{\mathscr{L}(\mathbf{s}, \mathbf{Z}) J_{\mathscr{C}}(\mathbf{x})}_{\mathbf{A}(\mathbf{x}, \mathbf{s}, \mathbf{Z})} \mathbf{u}, \qquad \mathbf{A} \in \mathbb{R}^{8\times 4} \tag{7}$$

Due to underactuation (four inputs for eight features) and other constraints, instantaneous error annihilation is generally infeasible; the system typically converges to a best-achievable error $\mathbf{e}^\star \neq \mathbf{0}$. The base follows unicycle kinematics

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega \tag{8}$$

and gimbal angles evolve as $\dot{\psi}, \dot{\phi}$. State and input box and rate constraints are enforced.

B. Obstacle Representation and Safety Constraints

A 2D forward-facing LiDAR returns points $\mathscr{O} = \{o_j\}$ expressed in $\mathscr{B}$. We model the robot footprint by a disc of radius r_{rob} plus a safety buffer ρ_s. The total safety radius is $r_f = r_{\text{rob}} + \rho_s$.

For each point $o_j = [o_{j,x}, o_{j,y}]^\top$, the distance $r_j = \|o_j - c_{\text{rob}}\|_2$ yields the clearance d_j:

$$d_j = r_j - r_f \tag{9}$$

where c_{rob} is the geometric center of the robot base in the local frame. Collision avoidance requires $d_{\min} := \min_j d_j \geq 0$.

To ensure safe path feasibility, the system must avoid obstacles in its forward sector. We quantify the clearance δ_j and longitudinal position t_j relative to the base heading. We define a safety corridor of width $2\rho_v$ aligned with the robot's heading. The minimal forward corridor margin m_c represents the minimum longitudinal distance to any obstacle that falls within this lateral corridor ($t_j \geq 0$):

$$m_c = \min_{j:\, t_j \geq 0} d_j^{\text{ray}} \geq 0 \tag{10}$$

The calculation of d_j^{ray} (corridor-thickened distance) implicitly uses the lateral distance δ_j to the heading ray. This metric allows us to evaluate the safety of the current heading choice relative to the local obstacle map.

C. Overall Control Objective

We consider a mapless outdoor problem requiring multi-objective coordination. The controller must continuously reduce the visual tracking error $\mathbf{e} = \mathbf{s} - \mathbf{s}^\star$ using only image feedback, while simultaneously ensuring collision avoidance ($d_{\min} \geq 0$) and path feasibility ($m_c \geq 0$). This task presents two key practical challenges: (1) the system is underactuated (four inputs for eight features), making instantaneous, perfect error correction infeasible under constraints; and (2) aggressive tracking minimization often leads to the "gimbal-only" local minimum, where the robot base stalls and navigation fails. Our objective is to synthesize a real-time policy $\pi : (\mathbf{s}_t, \mathscr{O}_t, \mathbf{x}_t) \mapsto \mathbf{u}_t$ that robustly balances tracking, safety, and dynamic constraints.

III. Coordinated Vehicle-Gimbal Model Predictive Control Method

This section details the proposed CoVG-MPC framework, which systematically addresses the control challenges outlined in Section II. The framework is built upon three synergistic principles. First, we introduce a control allocation via geometric decomposition to resolve the system's

underactuation and kinematic coupling. Second, a novel guidance law, Virtual Imaging Constraint Guidance (VICG), translates non-convex obstacle avoidance into a tractable heading reference. Third, at a guidance layer, an online adaptive MPC component learns a context-aware terminal cost to provide long-horizon foresight. These components are then integrated into a real-time, convex Quadratic Program (QP) MPC, ensuring robust and efficient performance. The overall control architecture, illustrating the interplay between these components, is shown in Fig. 2.

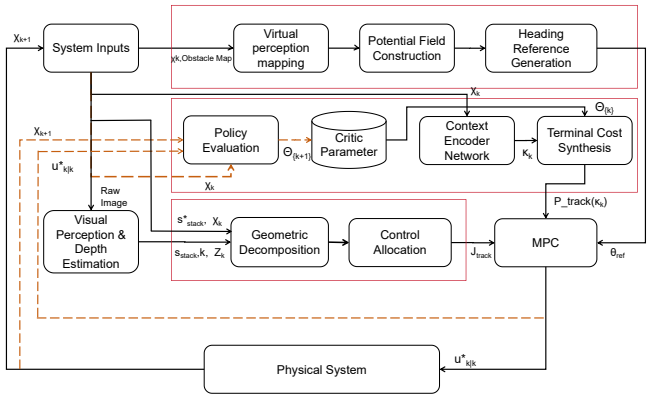

Fig. 2: The proposed CoVG-MPC control framework.

A. Control Allocation via Geometric Decomposition

The nominal control objective is defined in the image feature space. Given the stacked feature vector $\mathbf{s} \in \mathbb{R}^8$ and its error $\mathbf{e} = \mathbf{s} - \mathbf{s}^\star$ from Section II, a standard Image-Based Visual Servoing (IBVS) command drives the feature velocity towards a reference:

$$\dot{\mathbf{s}}^{\text{ref}} = -\mathbf{K}\mathbf{e}, \qquad \mathbf{K} \in \mathbb{R}^{8\times 8},\ \mathbf{K} \succeq 0 \tag{11}$$

where $\mathbf{K}$ is a positive semi-definite gain matrix. To account for varying feature importance, field-of-view (FOV) margins, and depth uncertainty, we equip the feature space $\mathbb{R}^8$ with a safety-aware, state-dependent inner product defined by a weighted metric $\mathbf{W}$:

$$\|\mathbf{a}\|_{\mathbf{W}}^2 = \mathbf{a}^\top \mathbf{W}\,\mathbf{a}, \quad \mathbf{W} = \text{blkdiag}(\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4) \in \mathbb{S}^8 \tag{12}$$

Each block $\mathbf{W}_k \in \mathbb{S}^2$ is a symmetric positive-definite matrix that can encode (i) pixel anisotropy from camera intrinsics (f_x, f_y), (ii) penalties for proximity to FOV boundaries, and (iii) feature reliability weights. This metric acts to normalize the feature space and prioritize reliable features.

Next, the control inputs are mapped to the feature space. The total influence matrix, or analytic image Jacobian $\mathbf{A}$, is given by:

$$\mathbf{A}(\mathbf{x}, \mathbf{s}, \mathbf{Z}) = \mathscr{L}(\mathbf{s}, \mathbf{Z})\, J_{\mathscr{C}}(\mathbf{x}) \in \mathbb{R}^{8\times 4} \tag{13}$$

where $\dot{\mathbf{s}} = \mathbf{A}\,\mathbf{u}$. We partition the control input $\mathbf{u}$ and the Jacobian $\mathbf{A}$ into parts corresponding to the base (subscript b) and the gimbal (subscript g):

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_g \end{bmatrix} = \begin{bmatrix} v \\ \omega \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_b & \mathbf{A}_g \end{bmatrix}, \quad \mathbf{A}_b \in \mathbb{R}^{8\times 2},\ \mathbf{A}_g \in \mathbb{R}^{8\times 2} \tag{14}$$

The feature dynamics are thus $\dot{\mathbf{s}} = \mathbf{A}_b\,\mathbf{u}_b + \mathbf{A}_g\,\mathbf{u}_g$. The set of all feature velocities that can be instantaneously generated by the gimbal forms the gimbal-achievable subspace $\mathscr{S}_g = \text{Range}(\mathbf{A}_g) \subset \mathbb{R}^8$. Its $\mathbf{W}$-orthogonal complement is defined as:

$$\mathscr{S}_g^{\perp_{\mathbf{W}}} = \{\mathbf{y} \in \mathbb{R}^8 \mid \langle \mathbf{y}, \mathbf{z} \rangle_{\mathbf{W}} = 0,\ \forall \mathbf{z} \in \mathscr{S}_g\} \tag{15}$$

Any vector in $(\mathbb{R}^8, \langle\cdot,\cdot\rangle_{\mathbf{W}})$ admits a unique decomposition into components in $\mathscr{S}_g$ and its complement. Applying this to $\dot{\mathbf{s}}^{\text{ref}}$:

$$\dot{\mathbf{s}}^{\text{ref}} = \underbrace{P_g\,\dot{\mathbf{s}}^{\text{ref}}}_{\text{gimbal component}} + \underbrace{P_b\,\dot{\mathbf{s}}^{\text{ref}}}_{\text{base component}}, \quad P_b := I - P_g \tag{16}$$

where $P_g : \mathbb{R}^8 \to \mathscr{S}_g$ is the $\mathbf{W}$-orthogonal projector onto $\mathscr{S}_g$. Using the weighted Moore-Penrose pseudoinverse, $\mathbf{A}_g^{\#_{\mathbf{W}}} := (\mathbf{A}_g^\top \mathbf{W} \mathbf{A}_g)^\dagger \mathbf{A}_g^\top \mathbf{W}$, the projector is explicitly constructed as:

$$P_g = \mathbf{A}_g\,\mathbf{A}_g^{\#_{\mathbf{W}}} = \mathbf{A}_g\,(\mathbf{A}_g^\top \mathbf{W} \mathbf{A}_g)^\dagger \mathbf{A}_g^\top \mathbf{W}, \qquad P_b = I - P_g \tag{17}$$

This $\mathbf{W}$-orthogonal projector satisfies idempotency ($P_g^2 = P_g$) and symmetry ($P_g^\top \mathbf{W} = \mathbf{W} P_g$), guaranteeing a unique, energy-minimal decomposition of the task into two orthogonal sub-tasks:

$$\dot{\mathbf{s}}_g := P_g\,\dot{\mathbf{s}}^{\text{ref}}, \qquad \dot{\mathbf{s}}_b := P_b\,\dot{\mathbf{s}}^{\text{ref}} \tag{18}$$

The gimbal tracks $\dot{\mathbf{s}}_g$, the closest achievable rate, while the $\mathbf{W}$-orthogonal residual $\dot{\mathbf{s}}_b$ is reserved for the base, structurally preventing gimbal-only local minima.

The projection can be realized procedurally through a two-stage, posture-priority control (PPC) scheme. First, the optimal gimbal control is found by solving a weighted least-squares problem:

$$\mathbf{u}_g^\star = \arg\min_{\mathbf{u}_g} \|\mathbf{A}_g \mathbf{u}_g - \dot{\mathbf{s}}^{\text{ref}}\|_{\mathbf{W}}^2 = (\mathbf{A}_g^\top \mathbf{W} \mathbf{A}_g + \lambda_g^2 I)^{-1} \mathbf{A}_g^\top \mathbf{W}\,\dot{\mathbf{s}}^{\text{ref}} \tag{19}$$

with Tikhonov damping $\lambda_g \geq 0$ for conditioning. The residual feature rate left for the base is precisely the projected base component:

$$\mathbf{r} = \dot{\mathbf{s}}^{\text{ref}} - \mathbf{A}_g \mathbf{u}_g^\star = P_b\,\dot{\mathbf{s}}^{\text{ref}} \equiv \dot{\mathbf{s}}_b \tag{20}$$

The base control is then computed to track this residual:

$$\mathbf{u}_b^\star = \arg\min_{\mathbf{u}_b} \|\mathbf{A}_b \mathbf{u}_b - \mathbf{r}\|_{\mathbf{W}}^2 = (\mathbf{A}_b^\top \mathbf{W} \mathbf{A}_b + \lambda_b^2 I)^{-1} \mathbf{A}_b^\top \mathbf{W}\,\mathbf{r} \tag{21}$$

This geometric decomposition provides the crucial inputs for our MPC formulation. The reference feature-rates for each actuation channel are:

$$\dot{\mathbf{s}}_g = P_g(-\mathbf{K}\mathbf{e}), \qquad \dot{\mathbf{s}}_b = P_b(-\mathbf{K}\mathbf{e}) \tag{22}$$

These targets are used to construct a quadratic cost term, J_{proj}, which will be part of the MPC's objective function. Over the prediction horizon H, this cost is:

$$J_{\text{proj}} = \sum_{k=0}^{H-1} \left\| \mathbf{A}_g^{(k)} \mathbf{u}_g^{(k)} - \dot{\mathbf{s}}_g^{(k)} \right\|_{\mathbf{W}^{(k)}}^2 + \sum_{k=0}^{H-1} \left\| \mathbf{A}_b^{(k)} \mathbf{u}_b^{(k)} - \dot{\mathbf{s}}_b^{(k)} \right\|_{\mathbf{W}^{(k)}}^2 + \mu \sum_{k=0}^{H-1} \left\langle \mathbf{A}_g^{(k)} \mathbf{u}_g^{(k)}, \mathbf{A}_b^{(k)} \mathbf{u}_b^{(k)} \right\rangle_{\mathbf{W}^{(k)}} \tag{23}$$

where $\mu \geq 0$ is a weak coupling coefficient, typically set to zero for strict decoupling. This cost term enforces the hierarchical control allocation within the MPC framework, forming the basis for the linearized QP detailed in Section III-C.

Alternatively, the hierarchical allocation objective for each time step can be consolidated into a single quadratic penalty using an effective state-dependent weighting matrix. The single-stage tracking cost is then written as:

$$\ell_{\text{track}}(\mathbf{x}_k, \mathbf{u}_k) = \|\mathbf{A}_k \mathbf{u}_k - \dot{\mathbf{s}}_k^{\text{ref}}\|_{\mathbf{W}_{\text{eff},k}}^2 \tag{24}$$

where the effective weighting matrix, $\mathbf{W}_{\text{eff},k}$, is explicitly constructed from the feature-space metric $\mathbf{W}_k$ and the gimbal projector $P_{g,k}$:

$$\mathbf{W}_{\text{eff},k} = \mathbf{W}_k \big(\alpha I + (1-\alpha) P_{g,k}\big) \tag{25}$$

Here, $\alpha > 1$ yields an effective weight of α in the base subspace and 1 in the gimbal subspace (since $\alpha I + (1-\alpha)P_g = I + (\alpha - 1)P_b$), prioritizing base motion to resolve tracking errors.

B. Guidance via Virtual Imaging Constraint Guidance

To address the non-convex obstacle avoidance constraints without compromising the real-time feasibility of the MPC, we introduce a high-level guidance module. Termed Virtual Imaging Constraint Guidance (VICG), this module reframes the geometric navigation problem into a tractable motion planning task that generates a stable heading reference for the MPC.

We attach a virtual sensory (imaging) plane perpendicular to the current base heading at a fixed look-ahead distance $s_v > 0$. World/robot-frame obstacles $\{o_j\}$ are mapped onto this plane via a projection $\mathscr{P}_v$, producing for each obstacle j a signed lateral displacement $d_{\perp,j}$ and a longitudinal coordinate s_j (nonnegative for points in front).

To ensure smooth activation as obstacles enter or leave the forward sector, we employ a C^∞ smooth transition function family:

$$g(x) \triangleq \begin{cases} e^{-1/x}, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{26}$$

$$f(x; a, b) \triangleq \frac{g(x-a)}{g(x-a) + g(b-x+a)} \tag{27}$$

and instantiate left/right activation functions f_1, f_2 using Eq. (27) by selecting suitable intervals (a,b) that correspond to the forward sector limits.

We quantify the degree of obstacle encroachment into the desired path with the Path Intrusion Metric:

$$\rho_j \triangleq (r_f + r_{\text{obs},j}) - |d_{\perp,j}| \tag{28}$$

where r_f is the robot's safety radius (body + buffer) and $r_{\text{obs},j}$ is the obstacle's effective radius on the virtual plane. A positive value, $\rho_j > 0$, indicates a potential collision risk.

Combining the activation functions, intrusion metric, and lateral geometry, we define a Repulsive Influence Function for each obstacle:

$$\Psi_j = f_j \cdot \exp\left(k_p \frac{\rho_j}{d_{\perp,j}^2 + \varepsilon} \right), \qquad k_p > 0,\ \varepsilon > 0 \tag{29}$$

This function quantifies the threat from each obstacle. These individual influences are then aggregated into two bounded, dimensionless scalars, Ψ_1 and Ψ_2, representing the total repulsive influence from the left and right sides of the robot, respectively.

On the lateral axis d of the virtual plane, we define a scalar potential field that combines a center-seeking objective with obstacle avoidance:

$$U(d; \{o_j\}) = U_{\text{att}}(d) + U_{\text{rep}}(d) \tag{30}$$

The attractive term, U_{att}, creates a quadratic well that pulls the desired trajectory towards the center of the lane:

$$U_{\text{att}}(d) = \frac{1}{2} k_a d^2, \qquad k_a > 0 \tag{31}$$

The repulsive term, U_{rep}, introduces an asymmetric potential based on the imbalance between left and right threats:

$$U_{\text{rep}}(d) = -k_{\text{vicg}} (\Psi_1 - \Psi_2)\, d, \qquad k_{\text{vicg}} > 0 \tag{32}$$

By evaluating the negative gradient of the total potential at the robot's current lateral position ($d = 0$), we obtain a net guidance "force":

$$F_{\text{net}} = -\left. \frac{\partial U}{\partial d} \right|_{d=0} = k_{\text{vicg}} (\Psi_1 - \Psi_2) \tag{33}$$

This bounded, signed scalar value steers the robot away from the more threatened side, while maintaining a centered path when threats are balanced ($\Psi_1 \approx \Psi_2$).

The guidance force F_{net} is first converted into an instantaneous corrective heading, $\delta_{\text{oa},k}$, using the corridor depth s_v:

$$\delta_{\text{oa},k} = \arctan\left(\frac{F_{\text{net},k}}{s_v} \right) \tag{34}$$

To ensure smooth behavior and encode the higher-level objective of tracking the visual target, we filter $\delta_{\text{oa},k}$ toward a nominal, task-optimal heading bias, $\delta_{\text{nom},k}$:

$$\begin{aligned} \delta_{\text{filtered},k+1} = (1 - \kappa_m \Delta t)\, \delta_{\text{filtered},k} + \kappa_f \Delta t \cdot \delta_{\text{oa},k} \\ + \kappa_m \Delta t \cdot \delta_{\text{nom},k} \end{aligned} \tag{35}$$

where $\kappa_m, \kappa_f > 0$ are filter gains. The bias $\delta_{\text{nom},k} = \text{wrap}(\theta_{t,k} - \theta_k)$ represents the direction to the actual target, where $\theta_{t,k}$ is the base-plane azimuth of the camera ray pointing to the target's image centroid, and $\text{wrap}(\cdot)$ maps angles

to $(-\pi, \pi]$. This term provides an alignment pull, ensuring that when it is safe to do so, the guidance system steers the robot in a direction favorable for the visual servoing task.

Finally, the complete safe heading reference is formed by applying the filtered correction to the current heading:

$$\theta_{\text{ref},k} = \text{wrap}(\theta_k + \delta_{\text{filtered},k}) \tag{36}$$

This dynamically generated $\theta_{\text{ref},k}$ serves as the feasible and stable heading target for the MPC controller.

C. Convex MPC Formulation

This section integrates the hierarchical task projection from Section III-A and the VICG guidance from Section III-B into a single, convex, finite-horizon Model Predictive Control (MPC) problem. The key insight is to encode the non-convex obstacle avoidance constraints as a soft quadratic penalty on the VICG heading reference, thereby preserving the convexity and real-time solvability of the final optimization problem. Instead of hard constraints, obstacle avoidance is enforced implicitly via the tracking cost of the VICG reference heading θ_{ref}.

Let $\mathbf{U}_k = \text{col}(\mathbf{u}_{k|k}, \mathbf{u}_{k+1|k}, \ldots, \mathbf{u}_{k+H-1|k}) \in \mathbb{R}^{4H}$ be the sequence of control inputs over the prediction horizon H. By linearizing the image-feature dynamics along the receding trajectory with sampling time Δt, we obtain the terminal error prediction:

$$\mathbf{e}_{k+H|k} = \mathbf{e}_{k|k} + \underbrace{\begin{bmatrix}\Delta t\mathbf{A}_k, & \Delta t\mathbf{A}_{k+1}, & \ldots, & \Delta t\mathbf{A}_{k+H-1}\end{bmatrix}}_{T_s \in \mathbb{R}^{8\times 4H}} \mathbf{U}_k \tag{37}$$

which induces a quadratic terminal cost:

$$\ell_f(\mathbf{U}_k) = (\mathbf{e}_{k|k} + T_s\mathbf{U}_k)^\top P_{\text{track}}(\kappa_k)\,(\mathbf{e}_{k|k} + T_s\mathbf{U}_k) \tag{38}$$

Here, $P_{\text{track}}(\kappa_k)$ is the context-aware terminal cost matrix, which is learned online by the adaptive MPC layer detailed in Section III-D.

The total cost function for the MPC is a sum of stage costs over the horizon and the final terminal cost. The stage cost at each step, $\ell(\mathbf{x}_k, \mathbf{u}_k)$, includes: (i) the task-projected tracking penalty derived from J_{proj} in (23), (ii) a quadratic penalty for deviations from the VICG heading reference θ_{ref}, (iii) control effort regularization (weighted by $R \succ 0$), and (iv) a penalty on the rate of change of control inputs (weighted by $S \succ 0$).

All per-step actuator bounds, rate limits, and Field-of-View (FOV) box constraints are linearized and stacked into a single horizon-level linear inequality:

$$C_k\mathbf{U}_k \le d_k, \quad \text{where} \quad C_k = \begin{bmatrix} C_{\text{limits}} \\ C_{\text{rate}} \\ C_{\text{fov}} \end{bmatrix}, \quad d_k = \begin{bmatrix} d_{\text{limits}} \\ d_{\text{rate}} \\ d_{\text{fov}} \end{bmatrix} \tag{39}$$

For example, the FOV constraints are constructed by enforcing box constraints $s_{\min} \le s_{k+i|k} \le s_{\max}$ at each prediction step $i \in [1, H]$. The predicted feature vector at step $k+i$ is linearized as $s_{k+i|k} \approx s_{k|k} + T_{s,i}\mathbf{U}_k$, where $T_{s,i}$ is the matrix mapping the control sequence $\mathbf{U}_k$ to the feature change at step i. Stacking these constraints for all steps across the horizon yields the compact matrix form:

$$\begin{bmatrix} \mathscr{T}_s \\ -\mathscr{T}_s \end{bmatrix} \mathbf{U}_k \le \begin{bmatrix} \mathbf{1}_H \otimes (s_{\max} - s_{k|k}) \\ -\mathbf{1}_H \otimes (s_{\min} - s_{k|k}) \end{bmatrix} \tag{40}$$

where $\mathscr{T}_s = \text{col}(T_{s,1}, \ldots, T_{s,H})$ is the stacked prediction matrix for the entire horizon. These inequalities form the rows of the submatrices C_{fov} and d_{fov}.

Crucially, direct non-convex obstacle-avoidance constraints are not imposed here; their influence is captured entirely within the convex VICG heading objective.

By collecting all stage and terminal cost terms, we formulate the standard convex Quadratic Program (QP):

$$\mathbf{U}_k^\star = \arg\min_{\mathbf{U}_k} \ \frac{1}{2}\mathbf{U}_k^\top H_k \mathbf{U}_k + f_k^\top \mathbf{U}_k \quad \text{s.t.} \quad C_k\mathbf{U}_k \le d_k \tag{41}$$

The Hessian H_k and gradient f_k are assembled from the linearized dynamics and cost components as follows:

$$\begin{aligned} H_k &= 2\Big(\sum_{i=0}^{H-1} \mathbf{A}_{k+i}^\top \mathbf{W}_{k+i}\mathbf{A}_{k+i} + I_H \otimes R + D^\top (I_H \otimes S) D \\ &\qquad + T_\theta^\top (\beta I) T_\theta + T_s^\top P_{\text{track}}(\kappa_k) T_s \Big) \\ f_k &= 2\Big(\sum_{i=0}^{H-1} \big(-\mathbf{A}_{k+i}^\top \mathbf{W}_{k+i}\, \dot{\mathbf{e}}_{k+i|k}^\star \big) - D^\top (I_H \otimes S)\, d_0 \\ &\qquad - T_\theta^\top (\beta I)\, b_\theta + T_s^\top P_{\text{track}}(\kappa_k)\, \mathbf{e}_{k|k} \Big) \end{aligned} \tag{42}$$

where I_H is the horizon identity, $\otimes$ denotes the Kronecker product, D is the block first-difference operator, d_0 contains the last applied control $\mathbf{u}_{k-1}$ for the initial rate penalty, and T_θ and b_θ represent the linearized yaw dynamics and reference residual for the VICG objective.

Since the weighting matrices $\mathbf{W}_{k+i}$, R, S, and the terminal cost matrix $P_{\text{track}}(\kappa_k)$ are all positive definite, the Hessian H_k is also positive definite ($H_k \succ 0$). This guarantees that the QP in (41) is strictly convex, ensuring a unique optimal solution $\mathbf{U}_k^\star$ that can be found efficiently by standard solvers.

D. ADP-Based Synthesis of the Terminal Cost

The final component of our framework is the guidance layer, which addresses the challenge of providing long-horizon foresight to the mid-horizon MPC. Instead of relying on a fixed, manually-tuned terminal cost, we introduce an Approximate Dynamic Programming (ADP) layer that learns a context-aware terminal cost matrix $P_{\text{track}}(\kappa_k)$ online. This approach allows the MPC to adapt its behavior to the perceived risk of the environment, without sacrificing the convexity or real-time solvability of the QP problem defined in (41).

First, we construct a low-dimensional context vector $\mathbf{m}_k \in [0,1]^4$ by normalizing state-dependent operational margins:

$$\mathbf{m}_k = [m_{\text{fov},k}, m_{\text{cond},k}, m_{\text{control},k}, m_{\text{obs},k}]^\top \tag{43}$$

These components represent margins for FOV limits, numerical conditioning, control effort, and obstacle proximity, respectively. This vector is then mapped to a scalar risk factor

$\kappa_k \in [0,1]$ using a shallow Radial Basis Function (RBF) network with a sigmoid activation:

$$\kappa_k = \sigma\left(\sum_{i=1}^{N_c} w_i \exp\left(-\frac{\|\mathbf{m}_k - \mathbf{c}_i\|^2}{2\beta_i^2}\right)\right), \quad \sigma(s) \triangleq \frac{1}{1+e^{-s}} \tag{44}$$

where $\mathbf{c}_i$ and β_i are the centers and widths of the RBF kernels. The scalar κ_k acts as a continuous risk-modulating factor: $\kappa_k \to 0$ in high-risk scenarios (e.g., near obstacles) and $\kappa_k \to 1$ in safe, benign situations.

We employ a quadratic critic to approximate the value function, parameterized by the risk factor κ_k:

$$V(\mathbf{e}, \kappa_k; \Theta) = \mathbf{e}^\top P(\kappa_k; \Theta)\,\mathbf{e} \tag{45}$$

where $\Theta = \{P_{\text{risk}}, P_{\text{safe}}\}$ are the learnable parameters. The terminal cost matrix $P_{\text{track}}(\kappa_k)$ is structured as a parsimonious blend of two SPD matrices:

$$P_{\text{track}}(\kappa_k) = (1-\kappa_k)\,P_{\text{risk}} + \kappa_k\,P_{\text{safe}}, \qquad P_{\text{risk}}, P_{\text{safe}} \in \mathbb{S}_{++} \tag{46}$$

Here, P_{risk} learns a cost function that emphasizes safety and visibility conservatism (used in high-risk states), while P_{safe} learns a cost that prioritizes target-tracking performance (used in low-risk states).

The critic's parameters, P_{risk} and P_{safe}, are updated online using a temporal difference (TD) learning rule. The one-step TD error serves as the learning signal:

$$\delta_{\text{TD},k} = c_k + \gamma V(\mathbf{e}_{k+1}, \kappa_{k+1}; \Theta_k) - V(\mathbf{e}_k, \kappa_k; \Theta_k) \tag{47}$$

where $\gamma \in (0,1)$ is a discount factor and the stage cost $c_k = \ell(\mathbf{x}_k, \mathbf{u}_k)$ is consistent with the MPC stage cost.

To ensure that P_{risk} and P_{safe} remain strictly positive definite throughout the learning process, we perform the parameter updates on the manifold of SPD matrices using a Riemannian gradient step under the log-Euclidean metric [20]. This update involves mapping the current matrices to the tangent space via the matrix logarithm, performing a gradient descent step scaled by the TD error $\delta_{\text{TD},k}$ and the context factor κ_k, and mapping the result back to the SPD manifold via the matrix exponential. This procedure guarantees that the learned cost matrices remain well-conditioned and SPD. The resulting $P_{\text{track}}(\kappa_k)$ is then fed into the Hessian and gradient assembly in (42), closing the loop between the high-level ADP layer and the real-time MPC layer.

The entire CoVG-MPC control cycle integrates guidance, cost synthesis, policy improvement, and online learning within a continuous loop. At each time step, the framework executes a complete sense-plan-act-learn sequence: First, the VICG module generates a safe heading reference from the current obstacles, while the ADP layer synthesizes a context-aware terminal cost matrix. These are then used to formulate and solve a convex QP for the optimal control plan. The first action from the plan is executed on the robot. Finally, the critic's parameters are updated online using the resulting state transition via a TD-learning rule, ensuring continuous adaptation and robust performance.

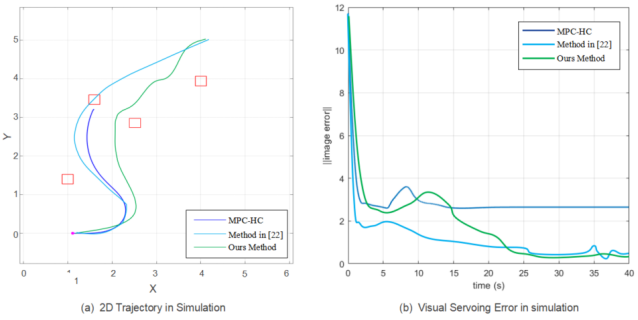

Fig. 3: Comparative simulation results: (a) 2D trajectories and (b) visual servoing error over time for the proposed method (Ours), MPC with Hard Constraints (MPC-HC), and a Vision-Centric IBVS Controller.

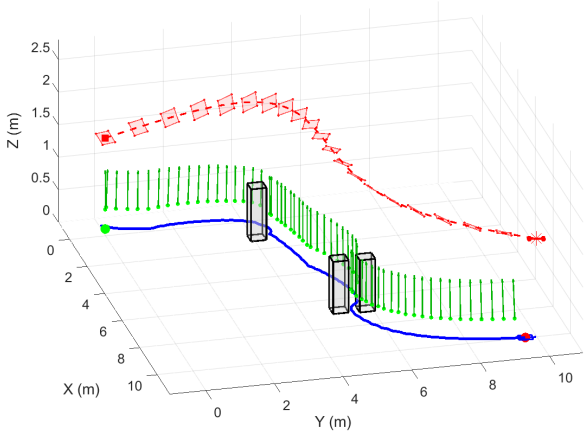

Fig. 4: Hardware experiment results: 3D trajectory of the robot (blue), dynamic target path (red), and camera's pointing vector (green). The robot avoids obstacles while maintaining target visibility.

IV. EXPERIMENTAL VALIDATION

We validate the CoVG-MPC framework through comparative simulations and hardware experiments.

A. Simulation-Based Comparative Analysis

We began by conducting a comparative study in MATLAB to highlight the fundamental advantages of our approach. The simulation environment contained several static, non-convex obstacles. The shared objective was to navigate the robot to an optimal pose for observing a target defined by four feature points in the world frame $\mathscr{W}$:

$$P_{\mathscr{W}} = \begin{bmatrix} 6.0 & 6.0 & 4.0 & 4.0 \\ 6.0 & 4.0 & 6.0 & 4.0 \\ 2.0 & 2.0 & 1.0 & 1.0 \end{bmatrix} \tag{48}$$

This configuration implies an optimal robot location in the vicinity of $(x,y) \approx (5,5)$. We compared our method against two baselines with carefully tuned parameters.

1) Our CoVG-MPC: The controller was configured with a prediction horizon $H = 10$. Key weights in the cost function were set to $Q_{\text{track}} = 1.2 I_8$, $Q_\theta = 0.8$, and

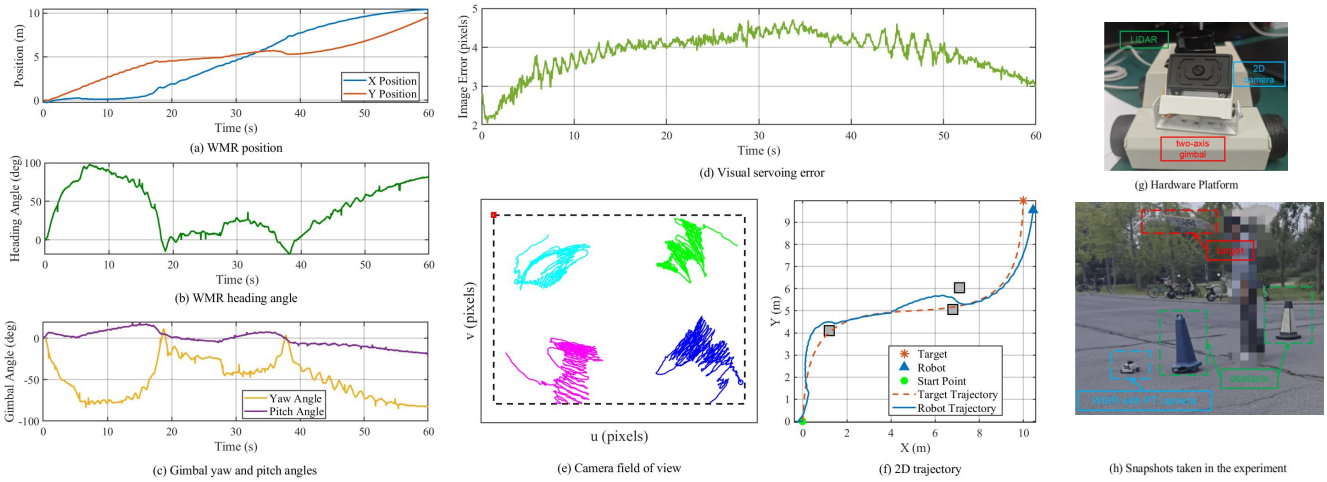

Fig. 5: Detailed hardware experiment results: (a) Robot position over time. (b) Robot heading angle. (c) Gimbal pan and tilt angles. (d) Visual servoing error. (e) Feature points trajectory in the camera's field of view. (f) Top-down 2D trajectory. (g) The hardware platform. (h) Snapshots from the experiment.

$R = \mathrm{diag}(0.1, 0.1, 0.05, 0.05)$ for tracking, heading, and control effort, respectively.

2) A Vision-Centric IBVS Controller [21]: This method handles FOV constraints and unknown feature depth via prescribed performance control, using an error transformation to confine features within visibility bounds. Its primary focus is visibility maintenance, not obstacle avoidance.
3) MPC with Hard Constraints (MPC-HC): This standard MPC uses the same horizon ($H = 10$) and cost weights as our method but encodes obstacles as direct non-convex inequality constraints, solved using a generic nonlinear solver.

As shown in Fig. 3, the results clearly demonstrate the limitations of the baseline methods in this multi-objective scenario. The MPC-HC controller failed to find a feasible solution and stalled before the obstacles, a common failure mode when dealing with non-convex constraints. The vision-centric IBVS controller, while driving the visual error down, failed to adequately account for the upcoming obstacle, resulting in a collision. Only our CoVG-MPC controller successfully identified and followed a safe, efficient path to the target region.

The error plot (Fig. 3b) confirms these findings: the MPC-HC error plateaus at $\sim$2.0, the IBVS method oscillates around 0.3 due to the underactuated system continuously attempting to null an infeasible error near obstacles, while our method converges smoothly to the lowest value of $\sim$0.34, demonstrating a stable balance between navigation and tracking.

B. Hardware Implementation and Results

We further deployed the framework on a physical platform to validate its robustness in a dynamic, real-world scenario.

1) Experimental Setup: Our platform is a custom four-wheeled differential-drive robot with a 2-axis gimbal, a ZED 2i stereo camera, and a 2D LiDAR. Computations were performed onboard via ROS 2, with the control loop running at 50 Hz. The experiment required the robot to track a manually moved ArUco marker through a cluttered outdoor passage.

The CoVG-MPC controller was configured with parameters mirroring the simulation but tuned for the real-world dynamics. The MPC prediction horizon was set to $H = 8$. The cost function weights were $Q_s = 1.5 \cdot \mathbf{I}$ for the projected tracking cost, $Q_\theta = 1.0$ for the VICG heading reference, $R = \mathrm{diag}(0.1, 0.15, 0.05, 0.05)$ for control inputs, and $S = 0.05 \cdot \mathbf{I}$ for control rates. The VICG module used a look-ahead distance of $s_v = 1.5$ m with attraction and repulsion gains of $k_a = 0.5$ and $k_{\text{vicg}} = 1.2$. For the ADP layer, the discount factor was $\gamma = 0.98$ and the Riemannian gradient step size was $\alpha_{\text{TD}} = 0.01$.

2) Results and Discussion: The hardware experiments validate that our framework robustly achieves its core objectives.

As shown in Figs. 4-5, the robot base executes large maneuvers while the gimbal provides precise, anti-correlated compensation to maintain target lock. This effective decoupling, a result of our geometric decomposition, ensures a consistently low visual servoing error (Fig. 5d). The VICG module's guidance is highlighted in Fig. 6, where the robot's heading intelligently deviates from the direct line-of-sight to ensure safety.

Averaged over multiple runs, the system maintained a minimum obstacle distance of 0.51 m (well above the 0.3 m safety threshold) and achieved a low visual servoing Root Mean Squared Error (RMSE) of 2.17 pixels. The framework demonstrated high reliability, with a 0% failure rate for both FOV violations and QP solver convergence.

Both simulation and hardware experiments confirm that the proposed framework enables robust, coordinated visual servoing in complex environments, systematically overcom-

ing the failure modes of traditional approaches.

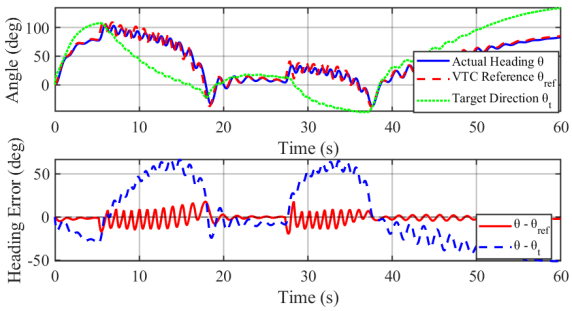

Fig. 6: Analysis of the VICG guidance law. (Top) The robot's actual heading precisely follows the safe VICG reference, intelligently deviating from the direct target direction when needed. (Bottom) The heading error plots confirm that safety guidance correctly overrides the raw tracking objective.

V. CONCLUSION

In this paper, we presented CoVG-MPC, a hierarchical control framework that resolves the fundamental conflict between local, reactive tracking and global, predictive safety for a wheeled mobile robot with an active gimbal. The framework's success is built upon three synergistic contributions: a task-space decomposition, grounded in geometric control theory, that optimally allocates control authority between the base and gimbal; a novel Virtual Imaging Constraint Guidance law that transforms non-convex obstacle constraints into a tractable heading objective; and an online adaptive critic, based on Approximate Dynamic Programming, that learns a context-dependent terminal cost to provide long-horizon foresight.

Hardware experiments validated the framework's effectiveness, demonstrating the ability to navigate smoothly through cluttered spaces while maintaining an uninterrupted visual lock on a dynamic 3D target. Crucially, this was accomplished without building a global map, relying solely on instantaneous forward-facing sensor data to perceive the environment and avoid obstacles. The principled control allocation successfully coordinates the base and gimbal, eliminating failure modes like gimbal-stalling and demonstrating superior performance over traditional approaches.

REFERENCES

[1] S. N. Aspragkathos, P. Rousseas, G. C. Karras, and K. J. Kyriakopoulos, "Multirotor target tracking through policy iteration for visual servoing," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 3240–3246.

[2] S.-L. Dai, J. Liang, K. Lu, and X. Jin, "Adaptive image-based moving-target tracking control of wheeled mobile robots with visibility maintenance and obstacle avoidance," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 2, pp. 488–501, 2023.

[3] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 2002.

[4] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE robotics & automation magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[5] R.-J. Wai and Y.-W. Lin, "Adaptive moving-target tracking control of a vision-based mobile robot via a dynamic petri recurrent fuzzy neural network," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 4, pp. 688–701, 2012.

[6] G. Olivas-Martínez, A. Miranda-Moya, C. Katt, and H. Castañeda, "Non-singular fast terminal adaptive visual tracking control with reduced tuning parameters for an aerial vehicle under perturbations," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 406–10 412.

[7] X. Jin, S.-L. Dai, J. Liang, and D. Guo, "Multirobot system formation control with multiple performance and feasibility constraints," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 4, pp. 1766–1773, 2021.

[8] C. P. Bechlioulis, S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, "Robust image-based visual servoing with prescribed performance under field of view constraints," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 1063–1070, 2019.

[9] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.

[10] M.-G. Kim, M. Jung, J. Hong, and K.-K. K. Kim, "Mppi-ipddp: A hybrid method of collision-free smooth trajectory generation for autonomous robots," *IEEE Transactions on Industrial Informatics*, 2025.

[11] L. Campos-Macías, D. Gómez-Gutiérrez, R. Aldana-López, R. de la Guardia, and J. I. Parra-Vilchis, "A hybrid method for online trajectory planning of mobile robots in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 935–942, 2017.

[12] L. Schäfer, S. Manzinger, and M. Althoff, "Computation of solution spaces for optimization-based trajectory planning," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 216–231, 2021.

[13] E. Sabouni, H. S. Ahmad, V. Giammarino, C. G. Cassandras, I. C. Paschalidis, and W. Li, "Reinforcement learning-based receding horizon control using adaptive control barrier functions for safety-critical systems," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 401–406.

[14] P. Roque, "Coordination of multi-agent systems: Predictive and vision-based control for aerial and space robotics," Ph.D. dissertation, KTH Royal Institute of Technology, 2022.

[15] M. Lin, Z. Sun, Y. Xia, and J. Zhang, "Reinforcement learning-based model predictive control for discrete-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 3312–3324, 2023.

[16] V.-A. Le and A. A. Malikopoulos, "Controller adaptation via learning solutions of contextual bayesian optimization," *IEEE Robotics and Automation Letters*, 2025.

[17] B. Amos, I. Jimenez, J. Sacks, B. Boots, and J. Z. Kolter, "Differentiable mpc for end-to-end planning and control," *Advances in neural information processing systems*, vol. 31, 2018.

[18] M. Zanon and S. Gros, "Safe reinforcement learning using robust mpc," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2020.

[19] A. Romero, Y. Song, and D. Scaramuzza, "Actor-critic model predictive control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 777–14 784.

[20] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM journal on matrix analysis and applications*, vol. 29, no. 1, pp. 328–347, 2007.

[21] Z. Miao, H. Zhong, J. Lin, Y. Wang, Y. Chen, and R. Fierro, "Vision-based formation control of mobile robots with fov constraints and unknown feature depth," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 2231–2238, 2020.