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

Reactive Whole-Body Control of Mobile Manipulators for Dynamic Target Tracking via Adaptive-Predictive Visual Servoing

Andrea Monguzzi[1], Giuseppe Alfonso[1], Navvab Kashiri[1]

***Abstract*— This paper addresses the challenging problem of enabling a mobile manipulator with an eye-in-hand camera to track dynamic targets with time-varying positions and orientations in an unbounded workspace. Specifically, we propose an optimization-based whole-body control framework for dynamic target tracking. The framework enables the mobile manipulator to maintain the target within the camera's field of view while reaching the desired pose, by dynamically regulating the priorities of the optimization constraints and objectives according to the task execution state. Moreover, we present an adaptive-predictive position-based visual servoing strategy to generate the Cartesian references sent to the controller. To enhance the tracking performance, we introduce (1) adaptive gains to avoid abrupt motions and the resulting vibrations while preserving final precision; (2) dynamic addition of a feedforward term incorporating a velocity estimate of the target using a Kalman Filter. The proposed approach is validated on a real robotic setup, as compared to a state-of-the-art approach, demonstrating superior performance in dynamic target tracking.**

I. Introduction

Tracking dynamic targets remains a challenging problem for mobile manipulators (MMs), as it requires the simultaneous coordination of the motion of the mobile base and the robotic manipulator, while also dealing with uncertainty in perception and estimation. There are two main approaches for controlling an MM: considering the base and the arm as independent subsystems, as in [1], or considering the MM as an integrated whole-body system, as in [2], where the authors propose a Quadratic Program (QP)-based controller. As pointed out in [3], even if designing separate control strategies for the mobile base and the manipulator may offer implementation convenience, it typically leads to sub-optimal performance. In contrast, coordinating the two subsystems in a whole-body fashion allows the system to exploit the complementary capabilities of the base and the arm, thereby improving the overall effectiveness. Focusing on whole-body strategies, it is important to distinguish between planning-based approaches and reactive ones. Planning can compute an optimal trajectory from the current state to the desired state, but its execution is often open-loop and highly sensitive to estimation errors. Although modern planners can handle high-dimensional constrained problems, they require significant computation time, as analyzed in [3]. On the other hand, reactive control continuously closes the loop using sensor measurements, enabling fast and robust adaptation to environmental changes without explicit trajectory generation.

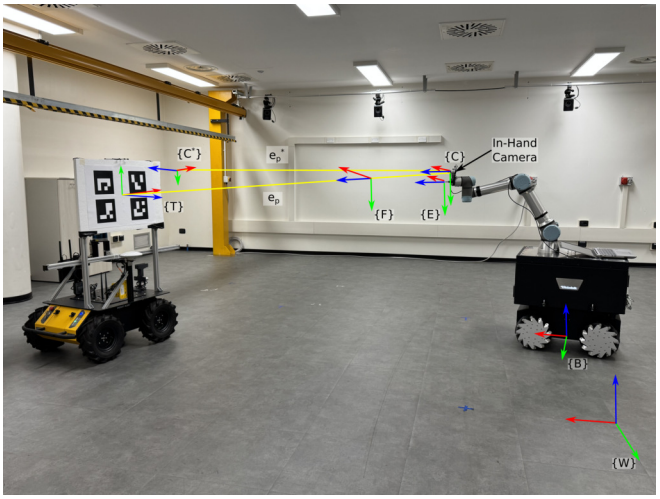

Fig. 1: Holonomic MM with an in-hand camera (on the right) tracking a moving target (on the left) that changes its position and orientation in the unbounded workspace. Relevant frames are highlighted: $\{\mathbf{W}\}$ world frame, $\{\mathbf{B}\}$ base frame, $\{\mathbf{E}\}$ end-effector frame, $\{\mathbf{C}\}$ camera frame, $\{\mathbf{T}\}$ target frame, $\{\mathbf{C}^*\}$ desired camera frame, $\{\mathbf{F}\}$ FoV frame. The x, y, z axes are depicted in red, green and blue, respectively. Error vectors $\mathbf{e}_p$ and $\mathbf{e}_p^*$ are highlighted.

This paper considers the problem of tracking a dynamic target moving in an unbounded workspace while changing its position and orientation. Fig. 1 shows the considered scenario: a moving target is tracked by an MM characterized by a holonomic base and equipped with an in-hand camera. Since the target trajectory is unknown to the MM, we propose a QP-based whole-body reactive Dynamic Target Tracking Control (DT^2C), coupled with an enhanced Position-Based Visual Servoing (PBVS) scheme that generates Cartesian references to track the dynamic target.
The main contributions of this paper are the following:

1) Building on the QP-based Holistic Motion Controller (HMC) of [2], we extend the formulation with additional cost terms and an adaptive weighting strategy. The proposed terms drive the MM to keep the target centered in the camera's Field of View (FoV) to avoid loss during dynamic tracking. Simultaneously, adaptive weights shift the objective from FoV centering when the target is distant to accurate pose alignment when it is near. This makes the controller robust against dynamic target motion and guarantees reaching the final goal.
2) We propose an Adaptive-Predictive PBVS scheme to

[1] The Authors are with Robotics Laboratory, Innovation Hub, Leonardo S.p.A., Via R. Pieragostini 80, 16151, Genoa, Italy, email{name.surname}@leonardo.com. This research work has been funded by the Key Digital Technologies Joint Undertaking (KDT JU) through the project MATISSE, grant agreement No. 101140216.

generate Cartesian references for dynamic tracking, integrated with our DT^{2}C. This scheme employs adaptive gains to prevent sudden accelerations of the mobile base that could lead to vibrations, and dynamically enables or disables a feedforward term based on the Kalman Filter estimate of the target's linear velocity, thereby enhancing tracking performance.

The rest of the paper is organized as follows. Sec. II reviews the related works and highlights the differences compared to our approach. Sec. III formalizes the problem under consideration. In Sec. IV, we present our DT^{2}C, and in Sec. V, we describe the Adaptive-Predictive PBVS scheme. Sec. VI details the experimental validation, highlighting the superior performance of DT^{2}C compared to HMC in dynamic target tracking. Finally, Sec. VII concludes the paper.

II. Related Works

Strategies for whole-body control of MMs have been extensively studied in the literature. As mentioned in Sec. I, a first distinction can be made between reactive control and planning-based approaches. A further classification concerns how references are expressed: either in configuration space, as in [4], or in Cartesian space. In the latter case, two main strategies are widely discussed in the literature: the stack-of-tasks approach, which computes an analytic solution by projecting secondary tasks into the null space of a primary task [5], and optimization-based methods, which relax the task hierarchy by formulating a single optimization problem where tasks are combined and weighted according to their priority. Other strategies also exist: for example, [6] considers the teleoperation of a holonomic humanoid platform to reach static targets. They propose an approach that exploits manipulator dexterity to coordinate the mobile base and arm motions, in contrast to the stack-of-tasks framework, where dexterity may be included as a task but does not directly influence the generation of mobile base motions. The work in [7] also considers a holonomic humanoid platform performing a vision-based pick of a static object. However, in this case, Cartesian trajectories are generated by a planner and commands are computed using the stack-of-tasks approach.

As mentioned, several works in the literature exploit optimization-based methods. For example, [8] proposes a constraint-based approach for dual-arm mobile robots using a virtual kinematic chain to decouple whole-body and dual-arm constraints. The method requires solving two cascaded QPs, which limits the control command frequency to 200 Hz, and is tested with Cartesian references generated from either a planner or an admittance control law. The authors of [9] propose a reference governor based on model predictive control to enhance an image-based visual servoing (IBVS) scheme. In particular, they add predictive behaviour while maintaining the reactivity of the QP-based low-level control loop. However, for MM, [9] validates the strategy only in simulation with a fixed target. The authors of [2] consider a non-holonomic base with an arm equipped with an in-hand camera and propose the reactive HMC formulated as a constrained QP, solved at 200 Hz. HMC aims to minimize joint velocities while maximizing translational manipulability and respecting joint position and velocity limits. In the experimental validation, the strategy is tested by controlling the MM to reach predefined Cartesian poses and follow planned trajectories, and subsequently applying standard PBVS to reach a static target close to the considered Cartesian poses. The same type of MM considered in [2] is also addressed in [10], which builds on the HMC proposed in [2] and defines an architecture for reactive manipulation on the move. In this framework, Cartesian velocity references are generated by a trajectory planner to reach a predefined area of interest. Once the robot enters this area, standard PBVS is employed to reach a target that moves within a limited portion of the workspace while maintaining a fixed orientation. A robotic arm on a non-holonomic base equipped with an actuated eye-to-hand camera mounted on the base is studied in [11]. They extend the reactive controller of [2] to control the camera link so that it remains oriented toward the target while preventing the arm from entering the camera's FoV. In the experiments, Cartesian velocity references are provided by standard PBVS, aiming to reach a static target. Table I summarizes the main optimization-based approaches for reactive whole-body control and highlights their differences compared to our proposed approach.

III. Problem Formalization

We address the problem of tracking a dynamic target that changes its position and orientation in an unbounded workspace using an MM with a holonomic base and a robotic arm equipped with an in-hand RGB camera. Fig. 1 highlights the relevant coordinate frames. In the following, we denote by ${}^{A}\mathbf{h}$ the vector $\mathbf{h}$ expressed in the reference frame $\{\mathbf{A}\}$. When no superscript is indicated, $\mathbf{h}$ is expressed in the world frame $\{\mathbf{W}\}$. The position of the origin of frame $\{\mathbf{A}\}$, $\mathbf{o}_A$, is represented by $\mathbf{p}_A \in \mathbb{R}^3$ and its orientation is denoted by the unit quaternion $\mathcal{Q}_A$. The rotation matrix from frame $\{\mathbf{A}\}$ to $\{\mathbf{B}\}$ is indicated by ${}^{A}\mathbf{R}_B$, while ${}^{A}\mathbf{t}_B$ denotes the translation vector between the two frames. The corresponding homogeneous transformation is given by ${}^{A}\mathbf{T}_B$.

The target trajectory is assumed to be unknown to the MM. The tracking objective is to control the camera frame $\{\mathbf{C}\}$ so that it reaches a desired pose, expressed by frame $\{\mathbf{C}^*\}$, defined relative to the target frame $\{\mathbf{T}\}$ by a fixed, user-defined homogeneous transformation ${}^{T}\mathbf{T}_{C^*}$. We define the Euclidean position error between frame $\{\mathbf{T}\}$ and $\{\mathbf{C}\}$ as $\mathbf{e}_p$, and the error between frame $\{\mathbf{C}^*\}$ and $\{\mathbf{C}\}$ as $\mathbf{e}_p^*$.

As shown in Fig. 1, the target consists of four ArUco markers arranged in a rectangle. We estimate the pose of the target frame $\{\mathbf{T}\}$, defined at the rectangle's center, using the following vision algorithm. For each RGB image acquired by the in-hand camera, the individual ArUco poses are first computed and smoothed using a moving average filter. Given the positioning of the markers, the filtered poses are averaged and further processed with a distance-dependent low-pass filter, yielding the pose estimate $({}^{C}\mathbf{p}_T, {}^{C}\mathcal{Q}_T)$, where ${}^{C}\mathbf{p}_T$ is the position vector and ${}^{C}\mathcal{Q}_T$ the unit quaternion. The

TABLE I: Summary of the comparison between our approach and the most relevant existing approaches. In the "Target Motion" field, *P* denotes dynamic target position and *O* denotes dynamic target orientation. In the "Base" field, *H* denotes a holonomic mobile base and *NH* denotes a non-holonomic one.

Paper	Method	Cartesian References	Target Motion	Camera	Base
[8]	Cascaded QPs	Planner / Admittance law	-	-	*NH*
[9]	QP	IBVS with Reference Governor	Static (Only in simulation)	Eye-to-hand	*H*
HMC ([2])	QP	Predefined poses or trajectories + Standard PBVS (locally)	Static	Eye-in-hand	*NH*
[10]	QP	Planner + Standard PBVS (locally)	Dynamic (*P*) moving in a limited workspace	Eye-in-hand	*NH*
[11]	QP	Standard PBVS	Static	Eye-to-hand	*NH*
DT^{2}C (Our)	QP	Adaptive-Predictive PBVS	Dynamic (*P* and *O*) in unbounded workpsace	Eye-in-hand	*H*

filter gain decreases linearly with the target distance, and no filtering is applied below 2.5 m. The filter parameters are tuned to balance noise reduction with the reactivity of the proposed approach.

The estimated target pose is used in the Adaptive-Predictive PBVS to generate reference Cartesian linear and angular velocities of the end-effector in the base frame, collected in ${}^B\boldsymbol{\nu}_E = [{}^B\mathbf{v}_E \;\; {}^B\boldsymbol{\omega}_E]^\top \in \mathbb{R}^6$ (as noted in Sec. V), which are then provided to the QP-based whole-body DT^{2}C (detailed in Sec. IV) that computes the joint velocities for the kinematic model of the MM (described in Sec. IV).

Since the camera is mounted in an in-hand configuration, the end-effector frame $\{\mathbf{E}\}$ and the camera frame $\{\mathbf{C}\}$ are assumed to be rigidly attached to the same body. Moreover, we assume that the target lies within the camera's field of view at the initial moment. A recovery routine for target search, such as one based on behaviour trees, can be integrated at startup or upon target loss; this aspect, however, lies beyond the scope of this work.

IV. Whole-Body Controller for Dynamic Target Tracking

The proposed QP-based whole-body controller relies on a kinematic model of the MM. We consider velocity-controlled robots, where $\mathbf{q}_b$ and $\mathbf{q}_a$ denote the joint positions of the kinematic models of the base and of the arm, respectively, and $\dot{\mathbf{q}}_b$ and $\dot{\mathbf{q}}_a$ indicate the corresponding velocities. The arm is modeled using its full kinematic structure with n_a Degrees of Freedom (DoFs), while the base is represented by a simplified virtual kinematic model characterized by $n_b = 3$ DoFs. Since the holonomic base can translate and rotate in the XY plane, we model it as a sequence of two prismatic joints that represent linear displacements and a revolute joint that represents rotation. Accordingly, the base configuration is defined as $\mathbf{q}_b = [x_b \;\; y_b \;\; \psi_b]^\top$, where x_b, y_b correspond to the forward and lateral displacements of the mobile base, respectively, and ψ_b denotes its yaw rotation. The overall configuration of the MM, characterized by $n = n_b + n_a$ DoFs, is given by $\mathbf{q} = [\mathbf{q}_b \;\; \mathbf{q}_a]^\top \in \mathbb{R}^n$, and the joint velocities are given by $\dot{\mathbf{q}} = [\dot{\mathbf{q}}_b \;\; \dot{\mathbf{q}}_a]^\top \in \mathbb{R}^n$. From the considered kinematic model, ${}^B\boldsymbol{\nu}_E = {}^B\mathbf{J}_E \;\; \dot{\mathbf{q}}$ holds, where ${}^B\mathbf{J}_E \in \mathbb{R}^{6\times n}$ denotes the Jacobian of the MM kinematic chain. ${}^B\mathbf{J}_E$ maps joint velocities to the end-effector velocity from the world frame $\{\mathbf{W}\}$ to the end-effector frame $\{\mathbf{E}\}$ and is expressed in the base frame $\{\mathbf{B}\}$.

We build upon the HMC formulation of [2], extending it as follows. The objective of the controller is to compute $\dot{\mathbf{q}}$ that minimizes the cost function f, achieving the desired Cartesian reference ${}^B\boldsymbol{\nu}_E$ minus a slack vector $\boldsymbol{\delta} \in \mathbb{R}^6$, which is included as an optimization variable and introduced to satisfy constraints and ensure feasibility. The controller is formulated as the following QP:

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^\top \mathbf{W}\mathbf{x} + \mathbf{c}^\top \mathbf{x}, \tag{1a}$$

$$\text{subject to} \quad \mathbf{G}\mathbf{x} = {}^B\boldsymbol{\nu}_E, \tag{1b}$$

$$\mathbf{A}\mathbf{x} \leq \mathbf{b}, \tag{1c}$$

$$\boldsymbol{\kappa}^- \leq \mathbf{x} \leq \boldsymbol{\kappa}^+. \tag{1d}$$

where $\mathbf{x} = [\dot{\mathbf{q}} \;\; \boldsymbol{\delta}]^\top$ is the vector of optimization variables. To extend the HMC proposed by [2] to dynamic target tracking within an unbounded workspace, we introduce significant modifications to the cost function, as described below.

A. Constraints

As in [2], we use the constraint (1b) to achieve the desired Cartesian reference velocity, subject to the flexibility introduced by the slack variables $\boldsymbol{\delta}$. Specifically:

$$\mathbf{G} = [{}^B\mathbf{J}_E(\mathbf{q}) \quad \mathbf{I}_6] \in \mathbb{R}^{6\times(n+6)}, \tag{2}$$

where $\mathbf{I}_6$ denotes the 6×6 identity matrix. Note that, since in the cost function we also maximize arm manipulability (see next subsection), ${}^B\mathbf{J}_E(\mathbf{q})$ remains well defined.

Constraint (1c) enforces the joint position limits. Since the base is represented by a virtual kinematic model, theoretically no limits should be imposed on its first three joints. Hence, the first three elements of $\mathbf{b}$ are set to a large constant M, in contrast to [2]. For the arm joints, we adopt the joint position-limit avoidance damper formalism proposed in [12], [2]. In particular, $\mathbf{A} = \mathbf{I}_{n\times(n+6)} \in \mathbb{R}^{n\times(n+6)}$ and

$$\mathbf{b} = \begin{bmatrix} M & M & M & \eta\frac{\rho_1-\rho_s}{\rho_i-\rho_s} & \dots & \eta\frac{\rho_{n_a}-\rho_s}{\rho_i-\rho_s} \end{bmatrix}^\top \in \mathbb{R}^n, \tag{3}$$

where M is a large constant and η is a positive weighting factor. For any joint $j = 1, \dots, n_a$, ρ_j is the distance to the nearest joint limit, ρ_i denotes the influence distance within which the damper is active, and ρ_s represents the minimum admissible distance of the joint from its limit.

Finally, constraint (1d) imposes lower and upper limits on the joint velocities and slack variables, with $\boldsymbol{\kappa}^-, \boldsymbol{\kappa}^+ \in \mathbb{R}^{n+6}$.

B. Cost Function for Dynamic Target Tracking Control

This Subsection describes the core of the proposed DT^{2}C. In our strategy, the weights in (1a) are expressed as:

$$\mathbf{W} = \begin{bmatrix} \Lambda_q + 2\,{}^B\mathbf{J}_E{}^\top\,\mathbf{W}_{FOV}\,{}^B\mathbf{J}_E & \mathbf{0}_{n\times 6} \\ \mathbf{0}_{6\times n} & \Lambda_\delta \end{bmatrix}, \quad (4)$$

$$\mathbf{c}^\top = \begin{bmatrix} (\hat{\mathbf{J}}_m + \varepsilon)^\top - 2\,{}^B\nu_{E,FOV}{}^\top\,\mathbf{W}_{FOV}\,{}^B\mathbf{J}_E & \mathbf{0}_{6\times 1} \end{bmatrix}, \quad (5)$$

and the cost function (1a) is expressed as follows:

$$f(\mathbf{x}) = \tfrac{1}{2}\dot{\mathbf{q}}^\top\Lambda_q\dot{\mathbf{q}} + \underbrace{\dot{\mathbf{q}}^\top\,{}^B\mathbf{J}_E{}^\top\,\mathbf{W}_{FOV}\,{}^B\mathbf{J}_E\,\dot{\mathbf{q}}}_{\mathscr{A}} + \tfrac{1}{2}\delta^\top\Lambda_\delta\delta + (\hat{\mathbf{J}}_m + \varepsilon)^\top\dot{\mathbf{q}} \underbrace{- 2\,{}^B\nu_{E,FOV}{}^\top\,\mathbf{W}_{FOV}\,{}^B\mathbf{J}_E\,\dot{\mathbf{q}}}_{\mathscr{B}}, \quad (6)$$

where $\hat{\mathbf{J}}_m$ denotes the manipulability Jacobian [12], and ${}^B\nu_{E,FOV}$ represents the end-effector velocity required to keep the target centered in the camera's FoV. The vector $\varepsilon = [0\ 0\ -w_\varepsilon\theta_\varepsilon\ 0\ldots 0]^\top \in \mathbb{R}^n$ contains the positive weight w_ε and the base-to-end-effector angle θ_ε, defined about the z-axis of $\{\mathbf{B}\}$ from ${}^B\mathbf{p}_E$, minimized to align the base and the end-effector. $\Lambda_q \in \mathbb{R}^{n\times n}$, $\Lambda_\delta \in \mathbb{R}^{6\times 6}$, $\mathbf{W}_{FOV} \in \mathbb{R}^{6\times 6}$ are weighting matrices, which we propose to define adaptively:

$$\Lambda_q = \mathrm{diag}\Big(\tfrac{w_b}{\|{}^B\mathbf{e}_p{}^*\|+\mu}, \tfrac{w_b}{\|{}^B\mathbf{e}_p{}^*\|+\mu}, \tfrac{w_b}{\|{}^B\mathbf{e}_p{}^*\|+\mu}, w_a, \ldots, w_a\Big), \quad (7a)$$

$$\Lambda_\delta = \mathrm{diag}\big(M, M, M, g({}^B\mathbf{e}_p), g({}^B\mathbf{e}_p), g({}^B\mathbf{e}_p)\big), \quad (7b)$$

$$\mathbf{W}_{FOV} = \mathrm{diag}\big(0, 0, 0, h({}^B\mathbf{e}_p), h({}^B\mathbf{e}_p), h({}^B\mathbf{e}_p)\big), \quad (7c)$$

in which $g(\cdot) = M\,\sigma(-\|\cdot\|)$, $h(\cdot) = w_{FOV}\,\sigma(\|\cdot\|)$, and $\sigma(x) = (1 + e^{-k_\sigma(x-d_0)})^{-1}$ denotes a sigmoid function with the steepness parameter k_σ. The positive constants w_b, w_a, w_{FOV} are user-defined weights, M is a large positive constant, μ is a small positive constant to avoid control gain singularity, and d_0 is a distance threshold.

The manipulability Jacobian $\hat{\mathbf{J}}_m$ is defined as in [2], $\hat{\mathbf{J}}_m = [\mathbf{0}_{3\times 1}\ -\mathbf{J}_m(\mathbf{q}_a)]^\top \in \mathbb{R}^n$, where $\mathbf{J}_m(\mathbf{q}_a) \in \mathbb{R}^n_a$ represents the translational manipulability Jacobian of the arm. As detailed in [13], the translational manipulability measure is defined as $m_t = (\det({}^B\mathbf{J}_{E,T}\,{}^B\mathbf{J}_{E,T}{}^\top))^{-0.5}$, where ${}^B\mathbf{J}_{E,T}$ is the translational component of the Jacobian; and therefore $\dot{m}_t = \mathbf{J}_m(\mathbf{q}_a)^\top\dot{\mathbf{q}}_a$ holds. We define $\mathbf{J}_m(\mathbf{q}_a)$ as in [12] and compute the manipulator Hessian, which appears in the definition of $\mathbf{J}_m$, following the approach proposed in [14]. The introduction of $\hat{\mathbf{J}}_m$ in the cost functions allows for maximizing the translational manipulability of the arm, while as discussed in [2], the inclusion of ε in the cost function causes the solver to select joint velocities that align the mobile base with the end-effector.

Compared to [2], we extend the cost function by introducing two additional terms, $\mathscr{A}$ and $\mathscr{B}$, together with a new adaptive definition of the weighting matrices (7). The introduction of $\mathscr{A}$ and $\mathscr{B}$ allows us to minimize $\|{}^B\nu_{E,FOV} - {}^B\mathbf{J}_E\,\dot{\mathbf{q}}\|^2_{\mathbf{W}_{FOV}}$, which represents the weighted error between the current end-effector velocity and the reference velocity ${}^B\nu_{E,FOV}$ required to keep the target centered in the camera's FoV. In particular, this reference involves only an angular velocity component, computed as detailed in the next subsection: ${}^B\nu_{E,FOV} = [\mathbf{0}_3\ {}^B\omega_{E,FOV}]$. Hence, $\mathscr{A}$ and $\mathscr{B}$ cause the optimizer to make the in-hand camera continuously look at the target, keeping it centered in the FoV and reducing the risk of losing it from view during dynamic motion in the unbounded workspace.

Constraint (1b) enforces the desired velocity ${}^B\nu_E$, which drives the camera frame $\{\mathbf{C}\}$ to the desired pose $\{\mathbf{C}^*\}$. However, as we show in Sec. VI, the HMC proposed in [2] continuously drives $\{\mathbf{C}\}$ to the pose of $\{\mathbf{C}^*\}$, which can shift the FoV away from a dynamically moving target, leading to task failure. To address this problem, we introduced $\mathscr{A}$ and $\mathscr{B}$ in (6) and the adaptive weighting strategy (7). When the target is far from the MM, i.e. $\|{}^B\mathbf{e}_p\| > d_0$, the function $g({}^B\mathbf{e}_p)$ in Λ_δ decreases the weight on the slack variables associated with ${}^B\omega_E$ (the angular component of ${}^B\nu_E$), thus allowing deviations from the commanded ${}^B\omega_E$. Simultaneously, the weights in $\mathbf{W}_{FOV}$, depending on $h({}^B\mathbf{e}_p)$, augment the contribution of $\mathscr{A}$ and $\mathscr{B}$, prioritizing the task of keeping the target centered in the FoV and thereby attempting to realize ${}^B\omega_{E,FOV}$.

Note that the slack weights on ${}^B\mathbf{v}_E$ (the translational components of ${}^B\nu_E$) always remain high, ensuring reaching the target position. This results in the MM approaching the moving target while regulating its end-effector orientation to center the target in the FoV. As the MM arrives closer to the target, i.e. $\|{}^B\mathbf{e}_p\| < d_0$, the value of $h({}^B\mathbf{e}_p)$ decreases, reducing the effect of the FoV centering terms, while $g({}^B\mathbf{e}_p)$ increases the weight on the commanded angular velocity of ${}^B\omega_E$. This smoothly shifts the control objective toward matching the final orientation of $\{\mathbf{C}^*\}$, which is the final goal of the strategy.

Finally, Λ_q contains a variable gain applied to the mobile base joints (inversely proportional to the pose error $\|{}^B\mathbf{e}_p^*\|$), and a constant weight applied to the arm joints, so that the optimizer favors base motion when the error is large and progressively increases the arm joint velocities as the error reduces. It is important to highlight that the proposed QP is strictly convex as $\mathbf{W}$ in (4) is positive definite.

C. Angular Velocity for Target Centering in the FoV

This subsection introduces and defines the ${}^B\omega_{E,FOV}$, the angular velocity component of ${}^B\nu_{E,FOV}$. To define ${}^B\omega_{E,FOV}$, a time-varying FoV frame $\{\mathbf{F}\}$ (Fig. 1) is defined. The orientation difference between $\{\mathbf{F}\}$ and $\{\mathbf{C}\}$ is used to compute ${}^B\omega_{E,FOV}$. The z axis of $\{\mathbf{F}\}$ is chosen as ${}^B\mathbf{z}_{FOV} = ({}^B\mathbf{p}_T - {}^B\mathbf{p}_C)/\|{}^B\mathbf{p}_T - {}^B\mathbf{p}_C\|$, thereby ensuring alignment with ${}^C\mathbf{e}_p$. To avoid unnecessary camera motion, the y axis of $\{\mathbf{F}\}$, ${}^B\mathbf{y}_{FOV}$, is designed to be aligned as closely as possible with the downward direction of the base frame $-{}^B\mathbf{z} = [0\ \ 0\ \ -1]^\top$. Accordingly, we define the x axis of $\{\mathbf{F}\}$ as ${}^B\mathbf{x}_{FOV} = (-{}^B\mathbf{z}) \times {}^B\mathbf{z}_{FOV}$ and ${}^B\mathbf{y}_{FOV} = {}^B\mathbf{z}_{FOV} \times {}^B\mathbf{x}_{FOV}$. This construction makes the FoV frame always point toward the target frame $\{\mathbf{T}\}$ while keeping its vertical axis stable.

Once the frame $\{\mathbf{F}\}$ is defined, we compute the rotation matrix ${}^{FOV}\mathbf{R}_C$ and extract its corresponding axis-angle representation, characterized by the angle θ_{FOV} and the unit axis ${}^{FOV}\mathbf{u}_C$. We define ${}^C\omega_{C,FOV} = -k_{FOV}\,\theta_{FOV}\,{}^{FOV}\mathbf{u}_C$, where

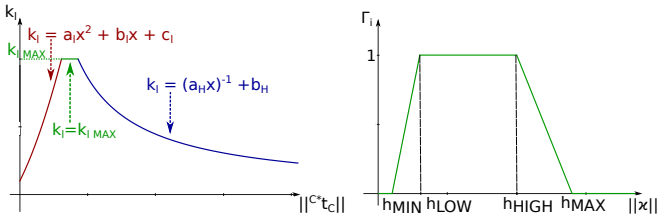

Fig. 2: Definition of the adaptive gains used in the PBVS scheme. The plots illustrate the gain k_l and the function Γ_i, $i=1,2$ that compose k_{fl}. Specifically, Γ_1 depends on $\|{}^{C^*}\mathbf{t}_C\|$ and is characterized by $t_{MIN}, t_{LOW}, t_{HIGH}, t_{MAX}$, while Γ_2 is a function of $\|{}^{C^*}v_{C^*,FF}\|$ and is defined by $v_{MIN}, v_{LOW}, v_{HIGH}, v_{MAX}$.

k_{FOV} is a positive gain. Since frames $\{\mathbf{C}\}$ and $\{\mathbf{E}\}$ are rigidly attached to the same body, ${}^C\omega_{C,FOV} = {}^C\omega_{E,FOV}$ holds. Finally, the angular velocity in the base frame is obtained as ${}^B\omega_{E,FOV} = {}^B\mathbf{R}_C\,{}^C\omega_{E,FOV}$.

V. Adaptive-Predictive Position-Based Visual Servoing for Dynamic Target Tracking

The end-effector velocity ${}^B\nu_E = [{}^B\mathbf{v}_E \;\; {}^B\omega_E]^\top$, used in constraint (1b), is computed by exploiting the Adaptive-Predictive PBVS scheme described in this section. We adopt the formalization of PBVS presented in [15], [16], which, in the case of a moving target, leads to the following control law which expresses ${}^C\nu_C = [{}^C\mathbf{v}_C \;\; {}^C\omega_C]^\top$ and drives the frame $\{\mathbf{C}\}$ to $\{\mathbf{C}^*\}$:

$$ {}^C\mathbf{v}_C = -k_l\,{}^C\mathbf{R}_{C^*}\,{}^{C^*}\mathbf{t}_C + k_{fl}(-{}^C\mathbf{R}_{C^*}\,\hat{\mathbf{e}}_l + {}^C\hat{\mathbf{v}}_C), \tag{8a} $$

$$ {}^C\omega_C = -k_o\,\theta\,{}^{C^*}\mathbf{u}_C + k_{fo}(-\hat{\mathbf{e}}_o + {}^C\hat{\omega}_C), \tag{8b} $$

where $\mathbf{R}$ and $\mathbf{t}$ are a rotation matrix and a translation vector defined according to the formalism of Sec. III, $(\theta, {}^{C^*}\mathbf{u}_C)$ indicates the axis-angle representation of ${}^{C^*}\mathbf{R}_C$, and $\hat{\mathbf{e}} = [\hat{\mathbf{e}}_l \;\; \hat{\mathbf{e}}_o]^\top$ denoted the error between the visual feature $s = ({}^{C^*}\mathbf{t}_C,\, \theta\,{}^{C^*}\mathbf{u}_C)$ and the constant desired value of the feature vector $s^* = \mathbf{0}$ (refer to [15], [16] for additional details). ${}^C\hat{\mathbf{v}}_C$ and ${}^C\hat{\omega}_C$ are the estimates of the current camera linear and angular velocity, that we compute using differential kinematics. The gains k_l, k_{fl}, k_o and k_{fo} are positive adaptive parameters, whose tuning is detailed in the next subsection.

Inspired by [17], we deploy a Kalman filter to predict the state of the moving target and hence estimate the value of the linear error $\hat{\mathbf{e}}_l$. As discussed in [17], for tracking a target moving along an unknown and highly dynamic trajectory, a motion model based on constant acceleration combined with adaptive filter parameters is an appropriate choice. We use the same dynamic model and adaptive parameter update laws for the Kalman filter proposed in [17]. Specifically, the state vector is

$$ \xi = [{}^Cx_{C^*}\;{}^C\dot{x}_{C^*}\;{}^C\ddot{x}_{C^*}\;{}^Cy_{C^*}\;{}^C\dot{y}_{C^*}\;{}^C\ddot{y}_{C^*}\;{}^Cz_{C^*}\;{}^C\dot{z}_{C^*}\;{}^C\ddot{z}_{C^*}]^\top. \tag{9} $$

At each time instant k, using $({}^C\mathbf{p}_T, {}^C\mathscr{Q}_T)$ from the vision algorithm described in Sec. III, we define ${}^C\mathbf{T}_T$ and subsequently compute ${}^C\mathbf{T}_{C^*} = {}^C\mathbf{T}_T\;{}^T\mathbf{T}_{C^*}$, using the known constant transformation ${}^T\mathbf{T}_{C^*}$. By extracting the translation vector from ${}^C\mathbf{T}_{C^*}$, we set the output of the model as $\mathbf{m} = {}^C\mathbf{t}_{C^*}$. Hence, the discrete-time linear model of the observed motion of the frame $\{\mathbf{C}^*\}$ is:

$$ \xi(k+1) = \mathbf{A}\xi(k) + \mathbf{n}_v(k), \tag{10a} $$

$$ \mathbf{m}(k) = \mathbf{H}\xi(k) + \mathbf{n}_w(k), \tag{10b} $$

where $\mathbf{A} = \mathrm{diag}(\mathbf{A}_i)$, $\mathbf{H} = \mathrm{diag}(\mathbf{H_i})$, with:

$$ \mathbf{A}_i = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{H}_i = [1\;0\;0], \quad i = x,y,z, \tag{11} $$

in which T is the time interval between two consecutive inputs from the vision algorithm detailed in Sec. III. $\mathbf{n}_v$ and $\mathbf{n}_w$ are white, zero-mean Gaussian noise characterized by block covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively. Specifically, $\mathbf{R} = r\mathbf{I}_3$ with $r = 0.0075$, and $\mathbf{Q} = \mathrm{diag}(Q_i)$ and

$$ \mathbf{Q}_i = \varsigma_{\mathbf{Q}_i}^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix}, \quad i = x,y,z, \tag{12} $$

where $\varsigma_{\mathbf{Q}_i}$ is the standard deviation adaptively computed for each i-th component of the estimated ${}^C\mathbf{t}_{C^*}$:

$$ \varsigma_{\mathbf{Q}_i} = a_s|({}^C\mathbf{t}_{C^*})_i| + b_s, \quad i = x,y,z, \tag{13} $$

with $a_s = 49$ and $b_s = 0.01$. At each update of the Kalman Filter, we obtain the position ${}^C\mathbf{t}_{C^*,KF}$ and the linear velocity ${}^C\mathbf{v}_{C^*,KF}$ (the subscript KF denotes that the quantity is predicted using the Kalman Filter).

Note that we do not estimate the orientation or the angular velocity of $\{\mathbf{C}^*\}$. Indeed, during the tracking phase, our QP-based strategy directly imposes the angular velocity to enforce ${}^C\omega_{C,FOV}$. Then, once the MM is close the target, the contribution of the angular velocity term ${}^C\omega_C = -k_o\,\theta\,{}^{C^*}\mathbf{u}_C$ is sufficient to ensure convergence to the desired pose $\{\mathbf{C}^*\}$.

The control laws of the proposed PBVS with Kalman filter are given by (recall that ${}^C\mathbf{t}_{C^*} = -{}^C\mathbf{R}_{C^*}\,{}^{C^*}\mathbf{t}_C$):

$$ {}^C\mathbf{v}_C = k_l\,{}^C\mathbf{t}_{C^*,KF} + k_{fl}({}^C\mathbf{v}_{C^*,KF} + {}^C\hat{\mathbf{v}}_C), \tag{14a} $$

$$ {}^C\omega_C = -k_o\,\theta\,{}^{C^*}\mathbf{u}_C. \tag{14b} $$

Since the frames $\{\mathbf{E}\}$ and $\{\mathbf{C}\}$ are rigidly attached to the same rigid body, ${}^C\omega_E = {}^C\omega_C$ holds. We then compute ${}^B\omega_E = {}^B\mathbf{R}_C\,{}^C\omega_E$. To compute ${}^B\mathbf{v}_E$ we apply the rigid-body velocity composition formula: ${}^B\mathbf{v}_E = {}^B\mathbf{R}_C\,{}^C\mathbf{v}_C + {}^B\omega_E\,{}^C\mathbf{t}_E$, where ${}^C\mathbf{T}_E$ is constant and known.

A. Adaptive gains for PBVS

Three adaptive positive gains are employed in the PBVS scheme: (I) k_l is specified as a piecewise function combining a quadratic, a constant, and a hyperbolic component, each expressed as a function of ${}^{C^*}\mathbf{t}_C$. (II) k_{fl} is defined as the product of two trapezoidal flat-top functions $\Gamma_1\Gamma_2$ and saturated at a maximum value $k_{fl,MAX}$: Γ_1 depends on ${}^{C^*}\mathbf{t}_C$, while Γ_2 depends on the feedforward velocity ${}^C\mathbf{v}_{C^*,FF} = {}^C\mathbf{v}_{C^*,KF} + {}^C\hat{\mathbf{v}}_C$. (III) k_o is tuned as a quadratic function of θ, saturated at a maximum value $k_{o,MAX}$: $k_o = a_o\theta^2 + b_o|\theta| + c_o$. The profiles of the functions k_l, Γ_1 and Γ_2 are illustrated in Fig. 2, while the tuning of the parameters is reported

TABLE II: Parameters tuning for the adaptive PBVS scheme.

Gain	Values
k_l	a_l = 3.0 b_l = 2.0 c_l = 0.1 $k_{l,MAX}$ = 0.7 a_h= 2.4 b_h= 0.03
k_{fl}	t_{MIN} = 0.08 t_{LOW}= 0.1 t_{HIGH} = 1.0 t_{MAX} = 1.1 v_{MIN} = 0.01 v_{LOW} = 0.02 v_{HIGH} = 1.0 v_{MAX} = 1.1 $k_{fl,MAX}$ = 0.5
k_o	a_o = 2.0 b_o = 0.5 c_o = 0.1 $k_{o,MAX}$= 1.0

TABLE III: Tuned parameters of the whole-body controller.

Parameters
$M = 10^3$ $\mu = 10^{-6}$ η = 0.1 ρ_s = 0.1 ρ_i = 0.9 w_b = 0.075 w_a = 0.01
κ^+ = [1.0 1.0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 1.0 1.0 0.5 0.5 0.5]$^\top$
κ^- = [-1.0 -1.0 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -1.0 -1.0 -1.0 -0.5 -0.5 -0.5]$^\top$
k_σ = 50 d_0= 0.75 w_{FOV} = 0.5 w_ε = 0.05 k_{FOV} = 0.8

in Table II. The hyperbolic component of k_l is included to account for the inertia of the mobile base, ensuring a gradual increase in the velocity command at the onset of the MM's motion, thus mitigating sudden accelerations that could induce vibrations.

Finally, the definition of k_{fl} allows disabling the feedforward term when (a) either the position error or the feedforward velocity are small (since in this case the first part of the control law is sufficient for tracking a slowly moving or proximal target), or (b) when the position error is large (as the estimate may be unreliable at greater distances) or the feedforward velocity is excessively high (likely arising from estimation errors). The value $\mathbf{t}_{MAX}$ is tuned with respect to the camera resolution.

VI. Experimental Validation

To validate the proposed approach and benchmark it against the HMC method presented in [2], we consider a UR5e manipulator mounted on the holonomic RB-KAIROS as MM. An Intel RealSense D455 is used as an in-hand camera mounted on the final link of the UR5e. The target composed of four ArUco markers is rigidly mounted on a Husky A300, which allows creating a dynamic target that follows a user-defined, repeatable trajectory (unknown to the MM). The described setup is shown in Fig. 1: we define $\{\mathbf{C}^*\}$ by setting ${}^T\mathbf{R}_{C^*} = \text{diag}(1.0, -1.0, -1.0)$ and ${}^T\mathbf{t}_{C^*} = [0.0\ 0.1\ 0.7]^\top$. The proposed approach is implemented using C++ and ROS2 and executed on an HP-ZBook-Fury-16-G9-Mobile-Workstation-PC with a 12th Gen Intel Core i7-12850HX × 24 processor. We solved the QP using PROX-QP [18] available in the ProxSuite GitHub repository[1]. The QP is solved at a frequency slightly greater than 500 Hz. The values of the parameters described in Sec. IV used in the experiments are reported in Table III.

We performed two types of experiments, each repeated across five different runs: (I) tracking of a static target, and (II) tracking of a dynamic target. In each experiment, we compared two different control strategies: the HMC of [2] and our proposed DT^{2}C. Additionally, we considered two alternative strategies for generating Cartesian references: the Adaptive-Predictive PBVS (APVS), described in Sec. V, and the Adaptive PBVS (AVS), which corresponds to the same scheme without the integration of the Kalman Filter. Note that in [2], Cartesian velocity references are generated considering predefined Cartesian poses to guide the robot toward a predefined area of interest; thereafter, standard PBVS is employed to reach a static target. In contrast, within our framework, the target exhibits an unknown dynamic motion in an unbounded workspace, which precludes a priori trajectory planning. Consequently, all Cartesian references must be generated exclusively through visual servoing.

To evaluate the experiments, we use the Vicon Vantage System and retro-reflective markers placed on the camera and on the target to obtain the poses of the camera frame $\{\mathbf{C}\}$ and the desired camera frame $\{\mathbf{C}^*\}$ in the Vicon frame $\{\mathbf{V}\}$, shown in Fig. 3A. It is important to note that the data collected from the Vicon System are not used in the control strategies; they are used solely to provide a ground truth for assessing and fairly benchmarking the different strategies in the experiments.

The results of the experiments are detailed in the following subsections, where the Root Mean Square Errors (RMSEs) are computed starting from a defined settling time. Specifically, for each error metric, the settling time is defined as the largest settling time value observed across all runs: a tolerance range of 0.005 m is used for the Euclidean position error, while a tolerance of range of 2° is used for the orientation errors in roll, pitch, and yaw. Since the tracking goal is to control $\{\mathbf{C}\}$ to match the pose of $\{\mathbf{C}^*\}$, we primarily analyze the Euclidean error ${}^Ve_p^*$ between the origins of $\{\mathbf{C}\}$ and $\{\mathbf{C}^*\}$, expressed in the $\{\mathbf{V}\}$ frame, and the yaw angle error ${}^Ve_\psi^*$ between their orientations relative to $\{\mathbf{V}\}$. This choice is motivated by the experimental setup, as orientation changes of $\{\mathbf{C}\}$ occur predominantly around the z-axis of $\{\mathbf{V}\}$. Moreover, in the dynamic target case, the motion of the Husky A300 (and consequently the target) primarily leads to changes in the orientation of $\{\mathbf{C}^*\}$ around the same axis.

The accompanying video shows several of the tests.

A. Static Target

The first experiments involve a static target (i.e. the Husky A300 stands still during the tests). The initial configuration of the experiment is shown in Fig. 3A and the Euclidean Distance between the origin of frame $\{\mathbf{C}\}$ and that of frame $\{\mathbf{C}^*\}$ is 4.1 m. In the considered configuration, we compare two strategies: (I) AVS + HMC and (II) APVS + DT^{2}C. For each strategy, Fig. 3B shows the Euclidean position error ${}^Ve_p^*$, while Fig. 3C shows the yaw angle error ${}^Ve_\psi^*$. Both strategies successfully achieve the tracking of the static target. As shown by Fig. 3B, the behaviors are very similar in terms of Euclidean position error: the RMSE values of the considered run are 1.328 m for AVS + HMC and 1.333 m for APVS + DT^{2}C. This similarity arises partly because the target is static and the predicted velocity ${}^C\mathbf{v}_{C,FF} = {}^C\mathbf{v}_{C^*,KF} + {}^C\hat{\mathbf{v}}_C$ correctly drops to zero, and partly because the DT^{2}C scheme acts on the position in the same way as HMC, differing only in how it modifies the orientation of $\{\mathbf{C}\}$ to maintain the target within the camera FoV when the target is distant. In contrast, Fig. 3C highlights differences in the yaw angle

[1] https://github.com/Simple-Robotics/proxsuite/tree/main

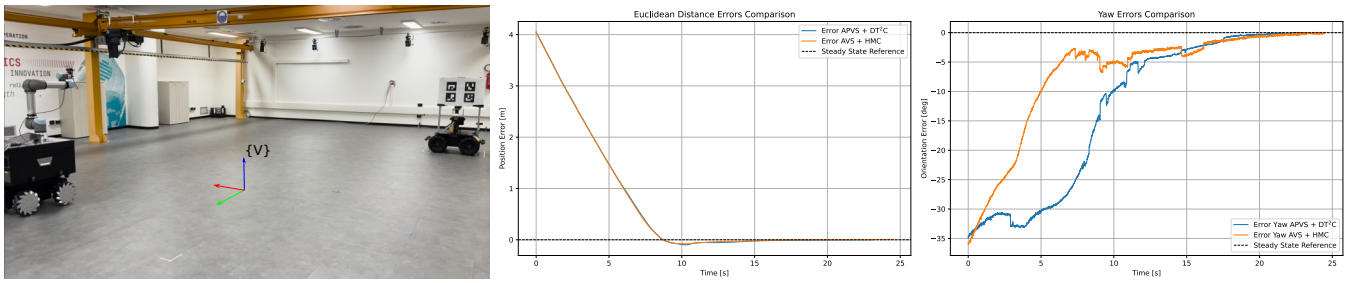

Fig. 3: Representative results from the static target tracking experiment. (A) Experimental setup for the static target evaluation. The Vicon frame $\{\mathbf{V}\}$ is highlighted. (B) Euclidean position error ${}^{V}e_{p}^{*}$ in representative runs for the two strategies. (C) Yaw angle error ${}^{V}e_{\psi}^{*}$ in representative runs for the two strategies. All quantities are expressed with respect to the Vicon Frame $\{\mathbf{V}\}$.

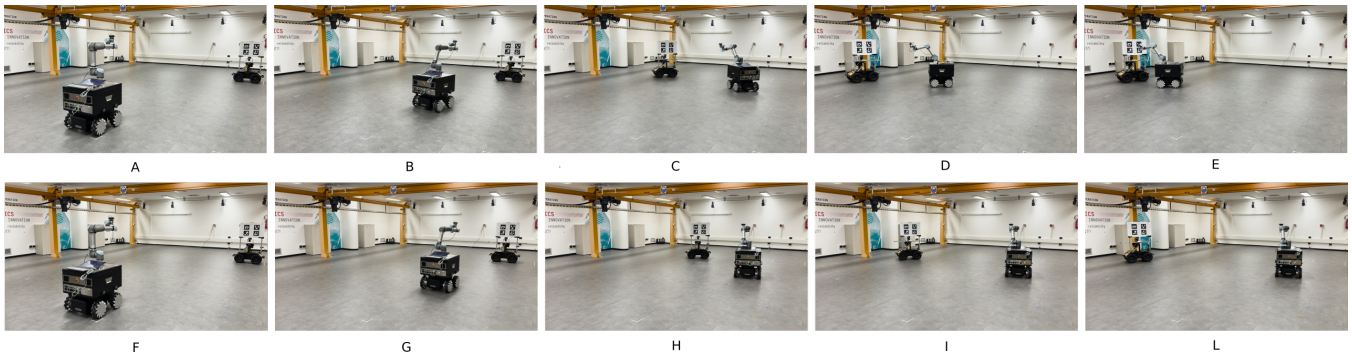

Fig. 4: Snapshots of a run with APVS + DT^{2}C (top row, A-E) and of a run with APVS + HMC (bottom row, F-L) in the dynamic target tracking experiment. Note that, with APVS + HMC, the target exits the camera's FoV, causing the MM to stop, which leads to task failure.

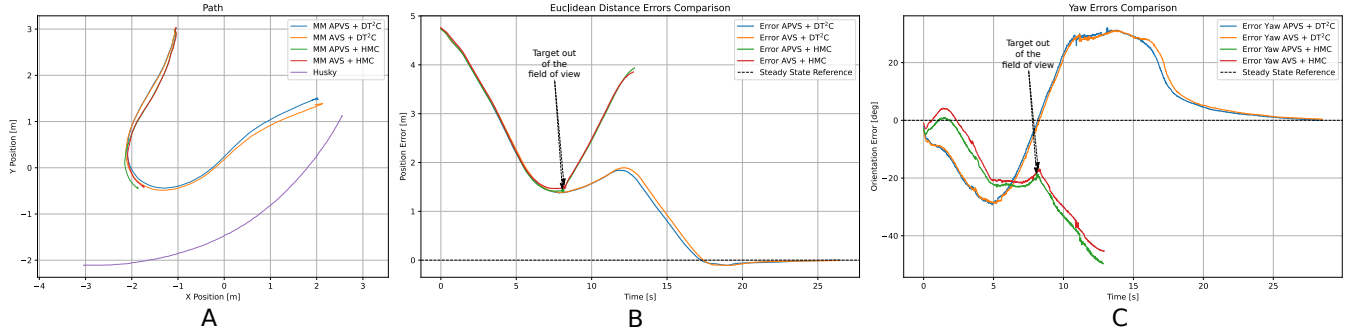

Fig. 5: Representative results from the dynamic target tracking experiment. (A) Paths executed by the MM and the Husky A300 in representative runs of each of the four benchmarked strategies. (B) Euclidean position error ${}^{V}e_{p}^{*}$ in representative runs for the four strategies. (C) Yaw angle error ${}^{V}e_{\psi}^{*}$ in representative runs for the four strategies. All quantities are expressed with respect to the Vicon frame $\{\mathbf{V}\}$.

TABLE IV: Success rates and average RMSEs together with their standard deviations (SDs) for the four evaluated strategies in the dynamic target tracking experiment. Average RMSEs are reported only for successful runs. Each strategy was tested across five runs.

	Success Rate	Mean RMSE (Euclidean Distance Error) [m]	RMSE SD (Euclidean Distance Error) [m]	Mean RMSE (Roll Error) [deg]	RMSE SD (Roll Error) [deg]	Mean RMSE (Pitch Error) [deg]	RMSE SD (Pitch Error) [deg]	Mean RMSE (Yaw Error) [deg]	RMSE SD (Yaw Error) [deg]
APVS + DT^{2}C	5/5	1.880	0.005	4.647	0.281	1.071	0.071	19.630	0.263
AVS + DT^{2}C	5/5	1.905	0.028	4.798	0.132	1.109	0.0467	20.090	0.314
APVS + HMC	0/5	-	-	-	-	-	-	-	-
AVS + HMC	0/5	-	-	-	-	-	-	-	-

error ${}^{V}e_{\psi}^{*}$. In this case, the error is larger with the DT^{2}C strategy: the RMSE values of the considered run are 14.649° for AVS + HMC and 22.869° for APVS + DT^{2}C, even though both strategies reach the settling time around the same moment. This behavior is explained by the QP formulation of DT^{2}C, which prioritizes keeping the target centered in the FoV when the MM is far from it, and only later shifts the control objective toward aligning with the final orientation of

$\{\mathbf{C}^*\}$ once $\|^B\mathbf{e}_p\| < d_0$. In contrast, HMC always aims to align with the final orientation of $\{\mathbf{C}^*\}$. In summary, HMC and DT^{2}C exhibit comparable performance in static target tracking. However, as will be demonstrated in the following subsection, HMC may fail in scenarios involving dynamic target tracking.

B. Dynamic Target in Unbounded Workspace

The second type of experiment involves the tracking of a dynamic target. The initial configuration of the experiment is shown in Fig. 4A, and the Euclidean distance between the origin of frame $\{\mathbf{C}\}$ and that of frame $\{\mathbf{C}^*\}$ is 4.75 m. The Husky A300 is controlled to stand still for 6 seconds from the experiment start and then perform a trajectory defined by the path shown in Fig. 5A and by a cubic timing law with a total duration of 10 s, reaching a maximum velocity of 1.5 m/s. This trajectory involves changes in both the position and orientation of $\{\mathbf{C}^*\}$, as shown in Fig. 4. In the considered experiment, we compare four different strategies: (I) AVS + HMC, (II) APVS + HMC, (III) AVS + DT^{2}C, and (IV) APVS + DT^{2}C. For each strategy, we conducted five independent runs of the experiment. Fig. 4A-E shows snapshots of a run of APVS + DT^{2}C, while Fig. 4F-L shows snapshots of a run of APVS + HMC. Table IV summarizes the obtained results and highlights that strategies AVS + HMC and APVS + HMC always fail to successfully perform the tracking task. Indeed, in these cases, the MM is unable to track the dynamic target that exits from the camera's FoV, since HMC persistently drives frame $\{\mathbf{C}\}$ to match the orientation of $\{\mathbf{C}^*\}$. Once the target leaves the FoV, the visual servoing scheme outputs zero velocities, causing the MM to stop. In contrast, both AVS + DT^{2}C and APVS + DT^{2}C succeed in tracking the dynamic target. This is because DT^{2}C prioritizes maintaining the target within the FoV when the MM is distant, and only shifts its objective to matching the orientation of $\{\mathbf{C}^*\}$ once the target is close to the MM. Fig. 5A shows the paths performed by the MM and the Husky A300 in representative runs of each of the four benchmarked strategies. Fig. 5B reports the Euclidean distance error ${}^V e_p^*$, while Fig. 5C shows yaw angle error ${}^V e_\psi^*$, both for a representative run of each strategy. Considering Fig. 5B and the average RMSEs of the Euclidean distance error detailed in Table IV, it can be observed that APVS + DT^{2}C outperforms AVS + DT^{2}C. This improvement is attributed to the Kalman filter integrated into the PBVS scheme: in APVS + DT^{2}C, the filter enables the estimation of the target's linear velocity, which is incorporated as a feedforward term in the command velocity. As a result, the MM reduces its distance to the target more effectively, resulting in a lower average RMSE. In summary, APVS + DT^{2}C achieves the best performance for dynamic target tracking in the considered scenario.

VII. Conclusions

This work presented DT^{2}C, a reactive whole-body controller, integrated with an Adaptive-Predictive PBVS scheme for dynamic target tracking with a mobile manipulator in an unbounded workspace. Experimental results demonstrated that, in contrast to the state-of-the-art HMC, the proposed controller successfully tracks dynamic targets in the considered scenario. Furthermore, the results showed that the Adaptive-Predictive PBVS effectively improves tracking performance, thanks to the Kalman Filter-based feedforward term. Future work will focus on extending the QP-based controller with constraints to address collision and self-collision avoidance, as well as developing more sophisticated target motion models for use in the Kalman Filter to further enhance dynamic tracking capabilities.

References

[1] S. Chitta, E. G. Jones, M. Ciocarlie, and K. Hsiao, "Mobile manipulation in unstructured environments: Perception, planning, and execution," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 58–71, 2012.

[2] J. Haviland, N. Sünderhauf, and P. Corke, "A holistic approach to reactive mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3122–3129, 2022.

[3] T. Sandakalum and M. H. Ang Jr, "Motion planning for mobile manipulators—a systematic review," *Machines*, vol. 10, no. 2, p. 97, 2022.

[4] G. B. Avanzini, A. M. Zanchettin, and P. Rocco, "Constrained model predictive control for mobile robotic manipulators," *Robotica*, vol. 36, no. 1, pp. 19–38, 2018.

[5] F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 637–654, 2015.

[6] D. Torielli, L. Muratore, and N. Tsagarakis, "Manipulability-aware shared locomanipulation motion generation for teleoperation of mobile manipulators," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 6205–6212.

[7] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 20–33, 2012.

[8] Y. Wang, C. Smith, Y. Karayiannidis, and P. Ögren, "Whole body control of a dual-arm mobile robot using a virtual kinematic chain," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1550047, 2016.

[9] A. Paolillo, M. Forgione, D. Piga, and E. M. Hoffman, "Fast predictive visual servoing: A reference governor-based approach," *Control Engineering Practice*, vol. 136, p. 105521, 2023.

[10] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke, "An architecture for reactive mobile manipulation on-the-move," in *IEEE International Conference on Robotics and Automation 2023*. IEEE, Institute of Electrical and Electronics Engineers, 2023, pp. 1623–1629.

[11] K. He, R. Newbury, T. Tran, J. Haviland, B. Burgess-Limerick, D. Kulić, P. Corke, and A. Cosgun, "Visibility maximization controller for robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8479–8486, 2022.

[12] J. Haviland and P. Corke, "Neo: A novel expeditious optimisation algorithm for reactive motion control of manipulators," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1043–1050, 2021.

[13] ——, "A purely-reactive manipulability-maximising motion controller," *arXiv preprint arXiv:2002.11901*, 2020.

[14] ——, "Manipulator differential kinematics: Part 2: Acceleration and advanced applications," *IEEE Robotics & Automation Magazine*, vol. 31, no. 4, pp. 159–169, 2023.

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

[16] ——, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[17] G. Milighetti, L. Vallone, and A. De Luca, "Adaptive predictive gaze control of a redundant humanoid robot head," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 3192–3198.

[18] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "Prox-qp: Yet another quadratic programming solver for robotics and beyond," in *RSS 2022-Robotics: Science and Systems*, 2022.