Visual Predictive Control for Mobile Manipulator: Visibility, Manipulability, and Stability

H Bildstein, Viviane Cadenat, Adrien Durand-Petiteville

Visual Predictive Control for Mobile Manipulator: Visibility, Manipulability, and Stability

H. Bildstein[†], V. Cadenat[†] and A. Durand-Petiteville[‡]

***Abstract*—This paper proposes a visual predictive control solution adapted to mobile manipulators and able to cope with several issues related to visibility, manipulability, and stability. To address these problems, the proposed strategy relies on (i) the use of two complementary cameras, (ii) the definition of a cost function depending on both the vision-based task and the manipulability, (iii) the integration of time-varying constraints allowing to prioritize the former against the latter. The strategy has been analyzed through simulation using ROS and Gazebo and implemented on our TIAGo robot. The obtained results fully validate the proposed approach.**

I. Introduction

The need for autonomous mobile manipulators is at the core of several applications in diverse scenarios, such as precision agriculture [1], industrial installation [2], Search and Rescue [3], or human assistance [4]. Generally speaking, a mobile manipulator must simultaneously carry out a navigation task for the mobile base and manipulation one for the robotic arm. Several challenges must be taken into account to carry out these two tasks. From a perception point of view, the robotic system must be equipped with sensors that can detect different landmarks and analyze the surrounding environment. Moreover, it is necessary to guarantee that the landmarks used to perform the tasks remain in the sensors' field of view. From a control point of view, the control scheme must simultaneously deal with the mobile base and the robotic arm to enable collaboration between the two subsystems and avoid movements that penalize completing the other task. Finally, it is necessary to harmonize the control of the robotic arm with the displacement of the mobile base to avoid cases where the robotic system is navigating with an extended arm, leading to significant vibrations at the end-effector level and increasing the risk of singular configurations and collisions with external elements.

As with any robotic system, there are many ways to control a mobile manipulator. A widely used solution consists of expressing the tasks in Euclidean space. In this case, the robot uses the onboard sensors to estimate the system configuration. Lidar-type sensors provide geometric data, allowing accurate estimation, but do not provide an advanced perception of the environment. Vision-based sensors offer rich environmental information, but the pose estimation is highly sensitive to errors. When using cameras, another widely used solution consists of expressing the task in the image space, which is the natural space for such a sensor. In this paper, we propose to explore the use of image-based servoing to control a mobile manipulator. The aim is to develop an approach that does not require estimating the pose of the robot, as it is traditionally required when expressing the problem relying on the pose or the generalized coordinates. Thus, it was decided to use an advanced visual-based controller, the Visual Predictive Control (VPC) [5] scheme. VPC is the combination of Image-Based Visual Servoing (IBVS) [6] with Nonlinear Model Predictive Control (NMPC) [7]. Thus, it is possible to express in a single constrained optimization problem: the end-effector positioning task in the image space, the manipulability, the visibility, and structure joint limits. Numerous VPC schemes were designed to control robotic arms [8] [9] [10], quadrotor UAVs [11], mobile robots [12] [13], autonomous underwater vehicles [14] or a tendon-driven continuum robot [15]. However, concerning mobile manipulators, NMPC schemes usually express the task using the end-effector pose [16] or the generalized coordinates [17] [18] [19]. Cameras are sometimes used as the main sensor to control mobile manipulators; however, the task is not defined in the image space [20] [21]. In such cases, the end-effector pose estimation accuracy has a significant impact on the control performances [6].

When designing the optimization process at the core of the VPC scheme used to control the mobile manipulator, we must take into account several aspects. First, the whole system contains many degrees of freedom, leading to a large search space for the NMPC optimization problem. We must then state the problem in the most suitable form to ease the optimization and rely on an efficient solver to compute an optimal solution in a very short time. Next, the system is redundant, and the end-effector pose can be obtained with infinite configurations. However, these configurations are not equally suitable for the task to perform, and it is necessary to include a term in the optimization process dealing with manipulability to prioritize the most relevant ones. Then, the VPC scheme must deal with the visibility of the landmarks of interest. Thus, if the mobile manipulator is equipped with a single camera to perform both navigation and manipulation tasks, the arm's movement might be too restricted to perform an efficient trajectory while keeping the landmark in the field. Thus, it might be interesting to consider a second camera to guarantee landmark visibility. Moreover, forcing at least one camera to conserve the landmark of interest in its field of view seems interesting. Finally, unlike fixed robotic arms, a camera attached to a mobile manipulator has to perform a large displacement to reach the desired pose. This impacts the stability of the

[†]H. Bildstein and V. Cadenat are with CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France and Univ. de Toulouse, UPS, LAAS, F-31400, Toulouse, France {cadenat,hugo.bildstein}[at]laas.fr

[‡]A. Durand-Petiteville is with Universidade Federal de Pernambuco UFPE, Departamento de Engenharia Mecânica, Av. da Arquitetura, 50740-550, Recife - PE, Brazil adrien.durandpetiteville[at]ufpe.br

closed-loop system, and it might be necessary to use large prediction horizons. It is also impacted by the numerous terms dealing with visibility and manipulability, making it necessary to include mechanisms guaranteeing closed-loop stability. To our knowledge, the works presented in [22], [23], and [24] are among the few ones tackling some of the aforementioned challenges. In [22], the nominal VPC scheme was introduced, while [23] was a first attempt to navigate with a tucked arm. It relied on a two-step control scheme, which, despite promising results, suffered from a slow convergence rate and needed to be carefully set up.

In this paper, based on [24][1], we present a VPC scheme considering the aforementioned challenges. First, the robot has two cameras, one on the end-effector and one on the head. Thus, when the end-effector camera cannot perceive the landmark, the head camera computes the visual features, which are then projected on the end-effector image sphere to manage the classical perspective projection issue, *i.e.,* without projection singularities. Next, the constrained optimization problem is defined as follows. The cost function to minimize is the sum of two terms. The first one allows the definition of the positioning task using image moments [25], which facilitates the mapping between the task and the pose spaces and then the computation of the optimal solution. The second term is based on a measure of manipulability, which deals with the robotic system's redundancy and promotes configurations far from singularities. This approach was preferred to the use of a constraint because it allows to carry out tasks requiring going through configurations with a small manipulability value. For a constraint-based approach, it would be necessary to tune the constraint threshold, making the task realization possible while impacting the manipulability. The cost function being defined, we now propose to extend the problem by adding a set of constraints, such as the classical visibility and joint limits constraints, similarly to [22]. Finally, we present the time-varying positioning constraint set guaranteeing the end-effector positioning despite using a local solver and the manipulability measure in the optimization cost function. This method first includes the prediction-reference equality constraint, a modified version of the terminal constraint [7]. Next, the velocity constraint on the last predicted step is relaxed to ensure the problem's feasibility. Finally, we include a novel logarithm-based [26] constraint prioritizing the visual task over the manipulability maximization. Last but not least, the optimization problem is implemented using a symbolic representation to reduce the processing time while computing a solution sufficiently relevant to achieve the task successfully.

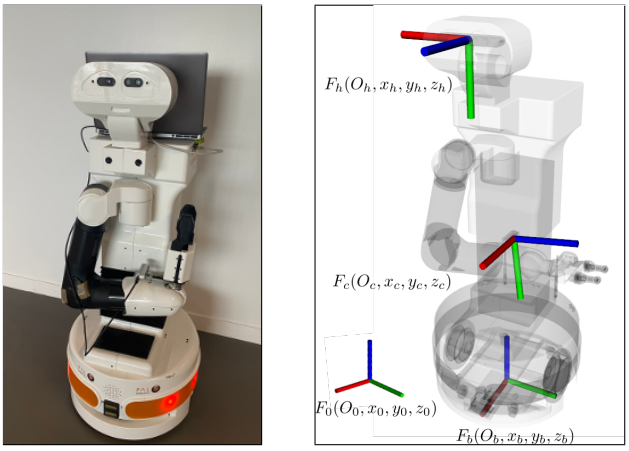
The rest of the paper is organized as follows. First, the different models are introduced before detailing the proposed VPC strategy. Next, this latter is evaluated both in simulation and experimentally on our TIAGo robot. The obtained results are then presented and thoroughly discussed. A conclusion and some prospects end the paper.

[1]This paper includes an extended presentation of the contribution, additional simulation results, and experimental results.

II. Preliminaries

A. Robotic system description and modeling

In this work, we consider the TIAGo robot designed by PAL Robotics. It is made of an upper body fixed to a differential mobile base (cf. Fig. 1a). The upper body has a 2-degree-of-freedom (DoF) head and a 7-DoF arm. Two RGB-D cameras are attached to the head and the wrist, respectively. The first is controlled using only the yaw joint ($n_h = 1$), while the second is moved thanks to only 5 DoF ($n_a = 5$). The to-be-achieved task consists of positioning the wrist camera with respect to a given landmark.



(a) The TIAGo robot (b) The robot model

Fig. 1: The robotic system

First, four frames are introduced: $F_0(0_0, x_0, y_0, z_0)$, $F_b(0_b, x_b, y_b, z_b)$, $F_{c_h}(O_{c_h}, x_{c_h}, y_{c_h}, z_{c_h})$, and $F_{c_{ee}}(O_{c_{ee}}, x_{c_{ee}}, y_{c_{ee}}, z_{c_{ee}})$, which respectively represent the world, the mobile base, head camera, and end-effector camera frames (cf. Fig. 1b). The generic subscript c will be used to highlight the relations for both cameras. It will be complemented by h or ee to specify the considered camera whenever needed.

The mobile robot is a differential-drive base. Its configuration and control vectors are then classically expressed by:

$$\chi_b = \left[X, Y, \theta\right]^T, u_b = \left[v, \omega\right]^T \tag{1}$$

where X, Y and θ are respectively its 2D-base coordinates in F_0 and its orientation around the vertical axis. This configuration vector χ_b is defined in the local sense of the term: the reference frame for a specific optimal control problem is the current mobile base frame. The base is controlled by the linear and rotational velocities along x_b and around z_b denoted by v and ω.

The arm configuration and control vectors are defined as follows:

$$\chi_a = \left[q_1, q_2, q_3, q_4, q_5\right]^T, u_a = \left[\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5\right]^T \tag{2}$$

where q_i is the i^{th} joint angle and $\dot{q}_i$ is the i^{th} joint velocity.

Lastly, the head configuration and control vectors are introduced similarly:

$$\chi_h = h_1, u_h = \dot{h}_1 \tag{3}$$

From this, it follows that the mobile manipulator state and control vectors can be defined by:

$$\chi_{mm} = \left[\chi_b^T, \chi_a^T, \chi_h\right]^T, u_{mm} = \left[u_b^T, u_a^T, u_h\right]^T \tag{4}$$

From the previous definitions, it is possible to formulate the end-effector camera kinematic model. These models are required to consider the manipulability within the control problem. To do so, we first introduce J_a as the Jacobian matrix mapping the end-effector camera kinematic screw ν_{F_c} to the arm control vector u_a. Next, we define J_{b+a} as the Jacobian matrix mapping ν_{F_c} to the arm and mobile base control vector, *i.e.,* $[u_b^T, u_a^T]^T$ such as:

$$\nu_{F_c} = J_{b+a}\begin{bmatrix} u_b^T & u_a^T \end{bmatrix}^T \tag{5}$$

where $J_{b+a} = \bar{J}_a + \bar{J}_b$ with:

$$\bar{J}_a = \begin{bmatrix} 0_{6\times 2} & J_a \end{bmatrix} \tag{6}$$

$$\bar{J}_b = {}^{c_{ee}}X_b \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0_{4\times 7} & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

where $0_{n\times m}$ corresponds to a n by m block of zeros, and ${}^{c_{ee}}X_b$ is the spatial motion transform matrix corresponding to the homogeneous transformation matrix ${}^{c_{ee}}H_b$ with:

$${}^{c_{ee}}H_b = \begin{pmatrix} {}^{c_{ee}}R_b & {}^{c_{ee}}t_b \\ 0 & 1 \end{pmatrix}, {}^{c_{ee}}X_b = \begin{pmatrix} {}^{c_{ee}}R_b & \hat{t}\,{}^{c_{ee}}R_b \\ 0 & {}^{c_{ee}}R_b \end{pmatrix} \tag{8}$$

where ${}^{c_{ee}}R_b$, ${}^{c_{ee}}t_b$ and $\hat{t}$ are respectively the rotation matrix between both frames, the position vector $O_{c_{ee}}O_b$, the skew-symmetric matrix deduced from ${}^{c_{ee}}t_b$.

B. Landmark description and selection of the visual features

It has been chosen to use landmarks made of AprilTags [27] (figure 2). It is worth mentioning the choice of using a planar target is justified by the robustness of visual servoing schemes [25]. This assumption of a planar object or having a planar limb surface is indeed used in the theoretical interaction matrix analysis.

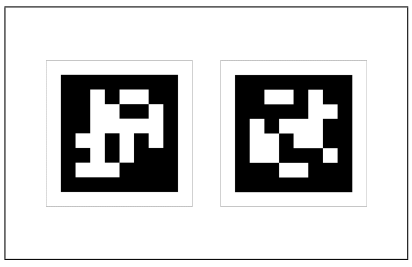


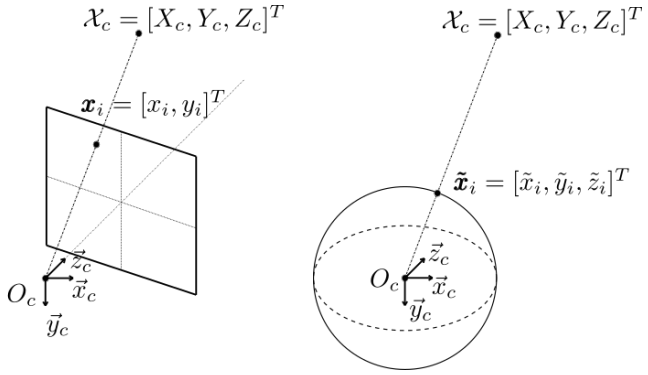
Fig. 2: Landmark

It is now necessary to characterize the landmark of interest by choosing suitable visual features. To control a camera with 6 DoFs, a classical choice considers four interest points (see [6] for more details). This leads to the following visual feature vector S_{ip}:

$$S_{ip} = \left[x_1, y_1, \ldots x_i, y_i, \ldots x_4, y_4\right]^T \tag{9}$$

where (x_i, y_i) are the 2D coordinates of the target interest points.

However, at this step, two main issues can be raised. First, the interest point coordinates present a strong coupling regarding the 6 DoF of the task, which could increase the control challenges in complex systems such as mobile manipulators. A solution highlighted in the literature is considering image moments, as shown in [25]. Indeed, using such features allows good decoupling properties, which presents the advantage of avoiding many non-linearities and more easily influencing the Cartesian trajectory. Second, this vector is intrinsically based on the perspective projection, and this projection method presents a singularity and discontinuity around $Z_c = 0$, if we denote by $\mathcal{X}_c = [X_c, Y_c, Z_c]^T$ the 3D point position in the camera frame. The arm mobility might thus be significantly restricted, reducing the realizable tasks for the mobile manipulator. These two problems can be overcome by considering the image moments based on the spherical projection and an additional camera [22], [25]. The two projection methods are presented in figure 3.



(a) Perspective (b) Spherical

Fig. 3: Projection methods

To define these moments, it is first necessary to determine the 3D point position $\mathcal{X}_c = [X_c, Y_c, Z_c]^T$ in the camera frame from the extracted 2D points coordinates. As the robot is equipped with RGB-D cameras, Z_c is directly available. Then, considering a normalized focal distance, X_c and Y_c can be classically deduced using the perspective projection model:

$$\begin{bmatrix} X_c \\ Y_c \end{bmatrix} = \begin{bmatrix} Z_c & 0 \\ 0 & Z_c \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{10}$$

The spherical projection consists in the projection of the 3D points $\mathcal{X}_c = \left[X_c, Y_c, Z_c\right]^T$ on the unit sphere centered in O_c:

$$\left[\tilde{x}_i, \tilde{y}_i, \tilde{z}_i\right]^T = \mathcal{X}_c / \|\mathcal{X}_c\| \tag{11}$$

If O is the observed object and O_{sp} its spherical projection, 3D discrete moments are defined by [25]:

$$m_{l,j,k} = \sum_{O_{sp}} \tilde{x}_i^l \tilde{y}_i^j \tilde{z}_i^k \tag{12}$$

From them, it is possible to design an adequate visual features vector [22]:

$$S = \left[x_g, y_g, I_1, C_{xy}, z_g, \alpha_{sp}\right]^T \tag{13}$$

where the triple (x_g, y_g, z_g) is the landmark gravity center coordinates, $C_{xy} = \|N_v \times z_c\|$ with N_v the normal vector to the target plane and α_{sp} the orientation of the object projection around z_c. Finally, I_1 is a suitable combination of 3D moments:

$$I_1 = m_{200}m_{020} - m_{200}m_{002} - m_{020}m_{002} + m_{110}^2 + m_{101}^2 + m_{011}^2$$

More details can be found in [22] and [25]. Thanks to this projection method and the designed visual features vector, it is possible to avoid the inherent singularity around $Z_c = 0$ and obtain nice decoupling properties for the camera DoF [25].

C. The re-projection model: the multi-camera solution

This work aims to position the end-effector camera relative to the target using a visual servoing scheme. The camera must remain oriented towards the landmark to compute the visual features. Therefore, the arm must initially be at least partially extended and then navigate with its arm outstretched. This increases the risk of collisions with the environment and might induce vibrations and perturbations. To avoid these problems and allow motions with a tucked arm, we propose to rely on the head camera when the end-effector camera cannot perceive the landmark. The idea is to compute the visual features in the head camera image and project them in the end-effector camera frame. To do so, the visual features are first projected from the head image to the head frame using (10). The head camera is controlled to face the landmark, and thus, the depth Z_c remains strictly positive, preventing any projection singularity. Next, the visual features are expressed in the camera frame using the homogeneous transformation matrix ${}^{c_{ee}}H_{c_h}$, which depends on χ_a and χ_h. Finally, they are projected on the end-effector image. Unlike the head camera, the end-effector one is not always facing the landmark, and inverting projection (10) might lead to a singularity when $Z_c = 0$, *i.e.,* when the camera is perpendicular to the landmark. For this reason, the visual features are projected on the unitary image sphere centered on the end-effector camera using projection (11), eliminating projection singularities. The control law is therefore supplied with the visual features, either directly obtained from the wrist camera information or recalculated from data provided by the head vision system using the re-projection model. Thanks to this approach, visibility problems can be overcome, and a wider range of motions can be considered.

Note that an estimate of the visual feature's depth is necessary to perform this projection. It can be provided by an RGB-D camera or using an observer. For both solutions, the provided values do not have to be highly accurate. Indeed, the head camera is only used to orientate the end-effector camera towards the landmark. The actual positioning of the end-effector camera is achieved using the visual features measured by this camera.

III. The multi-camera VPC strategy

In this section, we present the proposed VPC strategy. First, we state the optimal control problem and then detail the different elements and constraints used in this approach.

A. The VPC control problem

VPC consists of coupling NMPC with IBVS and can be defined as the following optimal control problem[2]:

$$U^*(\cdot) = \min_{U(\cdot)} \left(J_{N_p}(S(k), \chi_a(k), U(\cdot))\right) \tag{14}$$

subject to

$$\hat{S}(k) = S(k) \tag{15a}$$
$$\hat{\chi}_a(k) = \chi_a(k) \tag{15b}$$
$$\hat{S}(p+1) = f(\hat{S}(p), U(p)) \tag{15c}$$
$$\hat{\chi}_a(p+1) = g(\hat{\chi}_a(p), U(p)) \tag{15d}$$
$$C(U^*(\cdot)) \leq 0 \tag{15e}$$

where $U^*(\cdot)$ is an optimal control sequence calculated using Eq. 14. It minimizes the cost function J_{N_p} detailed in the sequel (see Eq. 17) over a N_p steps prediction horizon under a set of user-defined constraints $C(U(\cdot))$ presented in Eq. 15.

The computed optimal control sequence is denoted as $U^*(\cdot) = [u^*_{mm}(p), \ldots, u^*_{mm}(p+N_c-1)]$, where the $p = k$ notation is used to indicate that the command is obtained using predicted visual features knowing their measures at instant k. Moreover, $U^*(\cdot)$ is a N_c-dimensional vector where N_c is called the control horizon. In other words, the N_c first predictions of the N_p long prediction horizon are computed using independent control inputs, while all the remaining ones are obtained using a unique control input equal to the N_c^{th} element of $U(\cdot)$ (see [7] for more details). Once the problem is solved, only $u^*_{mm}(p)$ is applied to the robot, and the process is repeated using the previous optimization results to warm-start the solver.

The prediction process is performed through two dedicated models denoted by f and g (see constraints presented in Eq. 15). They compute the predicted visual features $\hat{S}$ and arm configuration $\hat{\chi}_a$, respectively. They use the current measures as the initial values at each iteration. If g is straightforward and corresponds to integrating the robotic arm kinematic model, f is computed using the global and exact method detailed in [22]. To summarize, two steps are required: (i) the computation of the homogeneous transformation matrix ${}^bH_c$ between the base frame F_b and the camera frame F_c thanks to the forward kinematic model; (ii) the determination of matrix ${}^{b_p}H_{b_{p+1}}$ which connects two successive predicted mobile robot poses by exactly integrating the mobile base kinematic model. The

[2] k denotes the discrete instant $t_k = kT_s$, with T_s being the sampling period.

prediction model for points in the camera frames can then be expressed as follows [22]:

$$\overline{\boldsymbol{X}}_i(p+1) = {}^{c_{p+1}}H_{b_{p+1}}\,{}^{b_{p+1}}H_{b_p}\,{}^{b_p}H_{c_p}\overline{\boldsymbol{X}}_i(p) = H(p)\overline{\boldsymbol{X}}_i(p) \quad (16)$$

where the bar indicates the homogeneous coordinates. Moreover, ${}^{b_p}H_{c_p}$ is the homogeneous transformation matrix between the camera and mobile base frames at the current prediction p, ${}^{b_{p+1}}H_{b_p}$ the one between the mobile base frames at prediction p and $p+1$, and finally ${}^{c_{p+1}}H_{b_{p+1}}$ the one between the mobile base and camera frame at predicted instant $p+1$. Using this result, the chosen visual features vector S can be deduced as explained in section II-B.

In the sequel, we focus on the definition of a cost function allowing to perform the task and of the set of inequality constraints $C(U^*(\cdot))$ aiming at dealing with the visual features visibility, the arm joints boundaries and the closed-loop stability.

B. The cost function

As explained before, the desired task consists of positioning the wrist camera with respect to a given landmark using vision. To perform this task, accounting for the robotic system's complex structure, the cost function J_{N_p} considers two objectives: the vision-based positioning task itself and the manipulability improvement. It is the sum over the prediction horizon of the cost $F(p)$. This latter is made of two terms F_{vs} and F_w, weighted using a dedicated gain denoted by $K_w > 0$, as shown in Eq. 18:

$$J_{N_p}(S(k), \chi_a(k), U(\cdot)) = \sum_{p=k+1}^{k+N_p} F(p) \quad (17)$$

with

$$F(p) = F_{vs}(p) + K_w F_w(p) \quad (18)$$

1) Modelling the visual task: F_{vs} aims at controlling the pose of the wrist camera and represents an error expressed in the image space. More precisely, similarly to IBVS, it consists of the quadratic error between the current visual features S and the desired ones S^*. Moreover, the error is weighted using a diagonal positive definite matrix Q_S, which allows specific DoF to be prioritized against others. This matrix can be easily tuned thanks to the decoupling properties of the chosen visual features S. Thus, the positing task F_{vs} is expressed as:

$$F_{vs}(p) = \left[\hat{S}(p) - S^*\right]^T Q_S \left[\hat{S}(p) - S^*\right] \quad (19)$$

2) Modelling the manipulability: The second term, F_w, aims at maximizing the manipulability of both the arm and the entire mobile manipulator. To do so, we first rely on a measure of the manipulability given by [28]:

$$w_a = \det\left(J_a^{red}(\chi_a)\, J_a^{red}(\chi_a)^T\right) \quad (20)$$

$$w_{b+a} = \det\left(J_{b+a}^{red}(\chi_a)\, J_{b+a}^{red}(\chi_a)^T\right) \quad (21)$$

where w_a and w_{b+a} are a measure of the manipulability of the sole arm, and of the arm and mobile base, respectively. Note that the number of DoFs of the considered robotic system has to be greater or equal to the number of DoFs included in the manipulability measure [28]. The wrist camera and the robotic arm have 6 and 5 DoFs, respectively. It is then not possible to consider the 6 DoFs of the camera, and it was decided only to consider the 3 translations. For this reason, J_a^{red} and J_{a+b}^{red} represent the reduced version of J_a and J_{a+b} only taking into account the translations. Figure 4 illustrates the main principle of this metric for a unique DoF. The manipulability measure is close to zero when the angle is equal to zero, *i.e.,* the two links are aligned, and the measure increases as the system moves away from the singularity.

A second metric is used to force (not to prevent) the robot to stay away from the boundaries of the DoF. It is done using the following measure [28]:

$$P = 1 - \exp\left(-k \prod_{i=0}^{5} \frac{(q_i - q_{imax})(q_{imin} - q_i)}{q_{imax} - q_{imin}}\right) \quad (22)$$

where q_{imax} and q_{imin} denote the minimal and maximal joint limits, while k is a positive constant. In Fig. 4, it can be seen that P has a low value when the angle is close its maximal or minimal values. These two measures are combined as follows:

$$w'_a = P w_a^2 \qquad w'_{b+a} = P w_{b+a}^2 \quad (23)$$

Finally, these two measures are used to define the manipulability cost function to minimize such as:

$$F_w(p) = \alpha_w / \hat{w}'_a(p) + (1 - \alpha_w) / \hat{w}'_{b+a}(p) \quad (24)$$

where $\alpha_w \in [0, 1]$. Note that in (24), the hat symbol denotes predicted measures.

C. The constraints

Now, we focus on the set of inequality constraints $C(U^*(\cdot))$ dealing with the visual features' visibility, the arm joint boundaries, and the closed-loop stability.

1) The visibility constraint: For the positioning task to be realized, preserving the target visibility during the whole motion is mandatory. A dedicated constraint has thus been introduced. However, it is worth mentioning that, in the proposed approach, the visibility constraint is set on the head image only, which allows the arm to move freely. The visibility will be ensured if the visual cues do not exceed the image boundaries. This leads to the following constraint:

$$\begin{bmatrix} \hat{S}_{ip}^{c_h}(p) - S_u \\ S_l - \hat{S}_{ip}^{c_h}(p) \end{bmatrix} \leq 0, \forall p \in [\![k+1, k+N_p]\!] \quad (25)$$

where S_l and S_u are, respectively, the lower and upper image boundaries of the head camera.

2) The joint limits constraints: It is also necessary to avoid the arm joints exceeding their lower and upper bounds χ_{al} and χ_{au} defined by the elements q_{imax} and q_{imin}. It yields to the following constraints:

$$\begin{bmatrix} \hat{\chi}_a(p) - \chi_{au} \\ \chi_{al} - \hat{\chi}_a(p) \end{bmatrix} \leq 0, \forall p \in [\![k+1, k+N_p]\!] \quad (26)$$

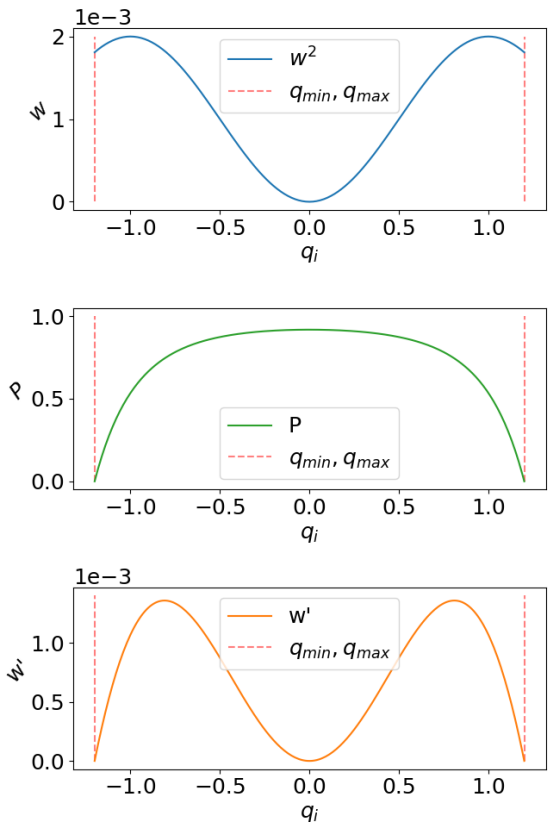


Fig. 4: Example of manipulability measure for a unique degree of freedom

3) The time-varying positioning constraint set: This part presents a set of time-varying constraints necessary to successfully perform the positioning task. It allows dealing with three challenges induced by the proposed VPC strategy. First, as this latter relies on a finite prediction horizon, the closed-loop stability must be guaranteed. Second, the manipulability index is included in the cost function to minimize because it cannot be expressed as a constraint. Thus, it cannot vanish, and the minimization of the cost function does not guarantee the positioning of the end-effector camera. Lastly, the positioning task is expressed in the image space, and a local, and thus sub-optimal, solver is used to obtain a solution at a frequency suitable for the robot's control. The computed trajectories are, therefore, subject to sub-optimality in the Euclidean space, and the system might be stuck in a local minima. To deal with these inter-correlated issues, we propose a set of three time-varying constraints defined hereafter.

a) The prediction-reference equality constraint: One of the classical ways to guarantee closed-loop stability consists of relying on a terminal constraint. This latter imposes that the last predicted state is equal to the desired one [7], forcing the computed trajectory to reach the goal. In this work, we propose a modified version of this latter, namely the prediction-reference equality constraint. It imposes that given predicted visual features are equal to the reference ones. This condition is expressed as follows:

$$\hat{S}(p+I_{TC}) = S^* \tag{27}$$

where the $I_{TC} \in [1,\ N_p]$ is the constrained prediction index whose precise role and evolution will be explained in the sequel.

b) The velocity constraints: When relying on a terminal constraint or the above prediction-reference equality constraint, it is mandatory to guarantee the feasibility of the problem, i.e., the computed solution satisfies the set of constraints. Regarding the terminal or a prediction-reference equality constraint, it is necessary to provide a sufficiently large prediction horizon. To do so, the velocity constraints of the last inputs can be relaxed, as shown in [13]. This approach leads to the following set of constraints for the mobile manipulator velocities:

$$\begin{aligned} \begin{bmatrix} u_{mm}(p) - u_{u|t} \\ u_{l|t} - u_{mm}(p) \end{bmatrix} &\leq 0, \forall p \in [\![k, k+N_c-N_r-1]\!] \\ \begin{bmatrix} u_{mm}(p) - u_{u|r} \\ u_{l|r} - u_{mm}(p) \end{bmatrix} &\leq 0, \forall p \in [\![k+N_c-N_r, k+N_c-1]\!] \end{aligned} \tag{28}$$

N_r is the number of prediction steps with relaxed boundaries, $u_{l|t}$ and $u_{u|t}$ are, respectively, the lower and upper tight boundaries corresponding to the 'true' limits of the actuator, and $u_{l|r}$ and $u_{u|r}$ are respectively the lower and upper relaxed boundaries.

c) The prediction-prediction decrease constraint: A terminal constraint and a sufficiently large prediction horizon guarantee closed-loop stability when computing the optimal solution with a global solver. However, this is insufficient when relying on local solvers providing sub-optimal trajectories. Indeed, a trajectory leading to the desired pose exists, but the first piece of this trajectory might be null. In such a case, the robot remains stuck in a local minimum. In [29], it is shown that adding a constraint imposing the decrease of the cost function allows dealing with this issue. In the present work, such an approach cannot be directly used and must be adapted. Indeed, the cost function contains the manipulability term and does not solely represent the vision-based task. Thus, optimizing the cost function does not guarantee the correct positioning of the camera but offers a trade-off between positioning and manipulability.

In this work, to ensure the correct positioning of the camera, we propose to force the decrease of the transformation between two given successive predicted camera poses. To do so, we first define $H_{I_{TC}}$ as the transformation matrix between the pose at the predicted instant $p+I_{TC}-1$ and the one at $p+I_{TC}$. Next, we rely on the logarithmic map log_6 that allows transferring an element H of the Lie group $SE(3)$ to the corresponding element ν of its Lie algebra $se(3)$ [26]:

$$\nu = \log_6(H) \tag{29}$$

In this work, $H = H_{I_{TC}}$ is used in its homogeneous transformation matrix form and ν in its 6-dimensional motion

vector form. In fact, ν corresponds to the velocity, linear and rotational, that should be applied during 1 second to obtain the transformation described by H. Thus, the constraint can be written as:

$$\|log_6(H_{I_{TC}})\| < min_{log} - \delta_{min} \tag{30}$$

where $H_{I_{TC}} = H(k+I_{TC}-1)$, and min_{log} represents the smallest $\|\log_6(H_{I_{TC}})\|$ value observed up to the current instant. Similarly to the approach presented in [29], a term δ_{min} is introduced to force a minimum decrease. It must be large enough to speed up the convergence but small enough to let the solver focus on the tasks.

d) Evolution of the time-varying constraint set: The three presented constraints aim to guarantee that, at each new iteration, a shorter trajectory reaching the goal is computed. In other words, by contracting the trajectory, we seek avoidance of null pieces of the solution, preventing the robot from being stuck in local minima and guaranteeing the successful realization of the positioning task. To do so, we rely on the I_{TC} index as follows. First, the prediction-reference constraint is set on the last prediction, and the prediction-prediction one is on the last piece of the trajectory, *i.e.,* $I_{TC} = N_p$. Thus, the predicted trajectory reaches the goal, and the length of the last piece is forced to decrease at each iteration. When this last piece of trajectory is no longer necessary, *i.e.,* null, the constraints are shifted to the previous predictions, *i.e.,* I_{TC} is decremented by 1. The process is repeated until $I_{TC} = 1$, meaning that the predicted trajectory is made of the sole $u^*(p)$ command, the one applied to the robot.

This process is illustrated in Fig. 5 where $N_p = N_c = 5$. Let us define the initial iteration as $k = k_0 = 0$, where I_{TC} is initialized to N_p. Thus, the prediction-reference constraint (27) between $\hat{S}(p+5)$ and S^* is respected, and the trajectory reaches the desired state (see Fig. 5a). This constraint can be satisfied from the initial state thanks to the relaxed velocity constraint (28). During the next iterations, *e.g.,* for iteration $k_1 > k_0$ in Fig. 5b, the piece of trajectory between $\hat{S}(p+4)$ and $\hat{S}(p+5)$ is forced to decrease thanks to the prediction-prediction decrease constraint (30). Once the logarithm of the transformation between $\hat{S}(p+4)$ and $\hat{S}(p+5)$ becomes smaller than the threshold δ_{log}, *i.e.,* iteration $k_2 > k_1$ in Fig. 5c, $\hat{S}(p+4)$ and $\hat{S}(p+5)$ are close enough to be merged. From now on, the current constraint set does not have an impact on the optimization anymore, and the constraint configuration must then be updated by applying $I_{TC} = I_{TC} - 1 = 4$, *i.e.,* iteration $k_3 = k_2 + 1$ in Fig. 5d. From now on, the prediction-reference constraint forces $\hat{S}(p+4) = S^*$, and the prediction-prediction constraint acts on $\hat{S}(p+3)$ and $\hat{S}(p+4)$ and forces this piece of trajectory to decrease, *i.e.,* iteration $k_4 > k_3$ in Fig. 5e This process is repeated until $I_{TC} = 1$ so that the command applied to the robot actually makes it reach the desired pose, *i.e.,* iteration $k_5 > k_4$ in Fig. 5f.

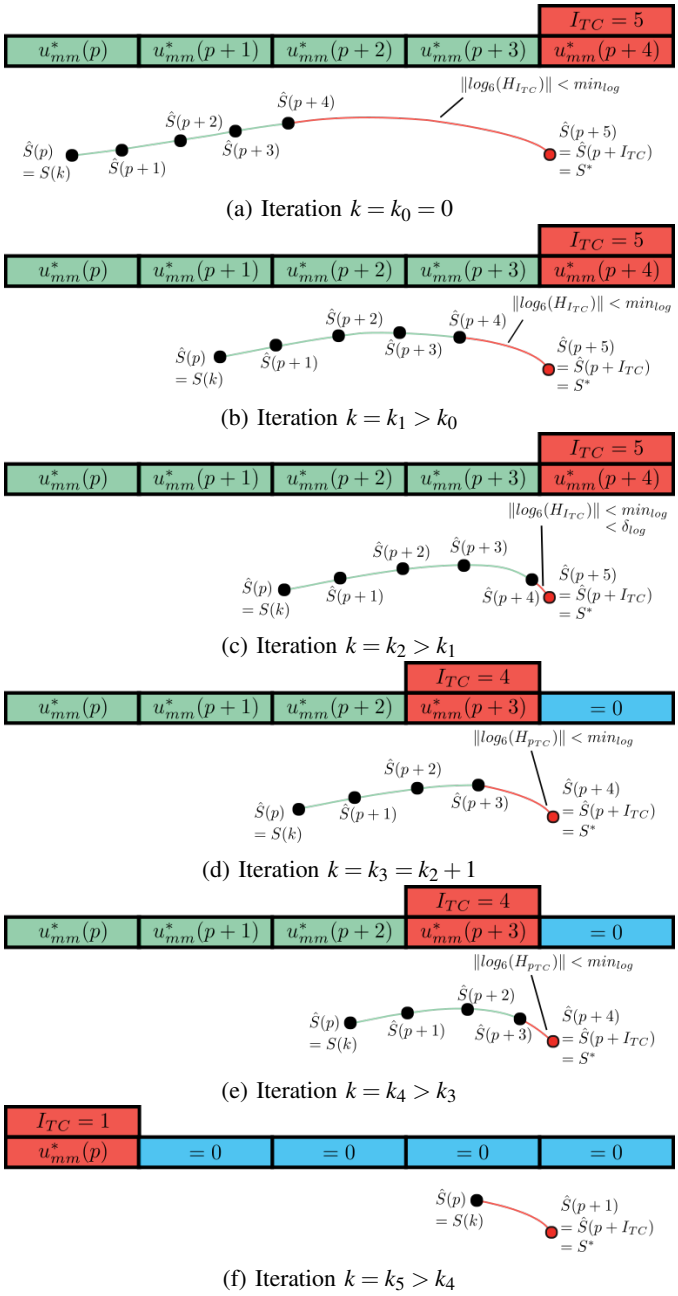


Fig. 5: Evolution of the time-varying constraint set

IV. Simulation and Experimental Results

This section presents simulation and experimental results to evaluate the performance of the presented approach. All algorithms are coded using the c++ language, and the optimal control problem is tackled using a direct single shooting method employing the SLSQP solver from the NLopt package. Gradients are computed symbolically offline using CasADi software [30] and are only evaluated online. Matrices ${}^bH_c$ and ${}^{b_k}H_{b_{k+1}}$ are derived using Pinocchio [31], a rigid body dynamics library. The tests are conducted on an Intel Core i7-10850H, and the control loop operates at 5 Hz. The solver timeout is set to $0.15s$, N_p and N_c are set up to 10 steps with a sampling time $T_s = 0.4s$. The target is a rectangle centered at $(3, 0, 1.08625)$.

For all the conducted tests, the camera and the mobile

base have to travel about 2m to reach the target, and the robotic system starts with the arm initially tucked. The bounds on the mobile base linear and angular velocities equal ± 0.1 m/s and ± 0.3 rad/s, respectively. The minimal and maximal joint limits are given by: $\chi_{au} = [2.68, 1.02, 1.50, 2.29, 2.07]$, $\chi_{al} = [0.07, -1.50, -3.46, -0.32, -2.07]$, $\chi_{hu} = 1.24$ and $\chi_{hl} = -1.24$. Finally, matrix $Q_S(p)$ is the identity matrix, while $N_r = 1$. The time units of the plots are the control loop iterations.

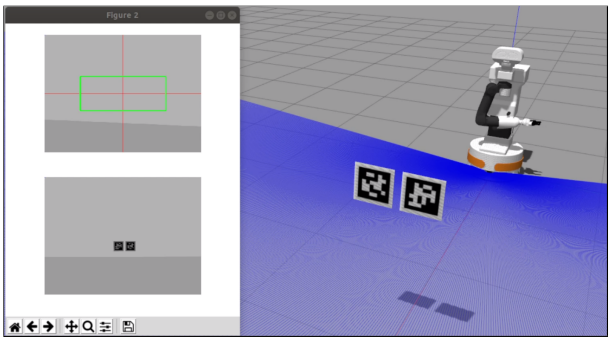
The different simulation tests have been conducted to highlight the strengths of the proposed approach and select the best options for our VPC scheme to test on our robot. The section is divided into four parts. The first is intended to analyze and validate the proposed approach while showing the impact of evaluating the gradients using CasADI. The second one is focused on the decrease constraint choice, comparing the logarithmic solution proposed here to the one based on the command norm proposed in our previous works [23]. The third one analyzes the impact of the manipulability measure, trying to highlight the best combination of the manipulability indices. Finally, the last part presents the experimental results of the most adequate selected setups.

A. Simulation results – Evaluation of the approach and impact of CasADI

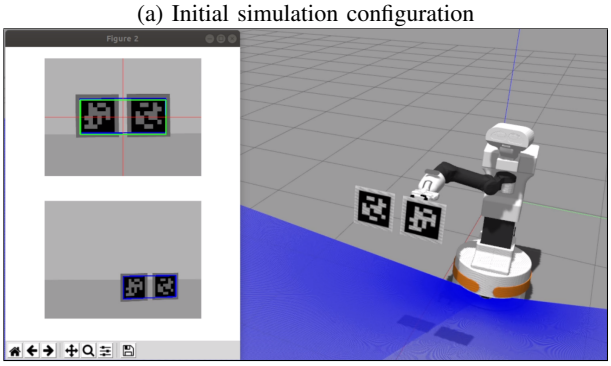
In this first section, the introduced control scheme runs using *Gazebo* simulator. Figure 6 presents an initial simulation setup example and the final robot configuration obtained with the presented VPC controller. The simulator allows us to obtain realistic scenarios. The goal is to compare different approaches to emphasize the relevance of the key points introduced in the scheme. For the following simulation results sections, the initial mobile base pose is $(0,0,0)$ and $\alpha_w = 1$. This configuration is representative of a generic case.

1) Visual task realization: Figures 7a and 7b present the error between the image moments and their desired values with and without CasADI. As one can see, the first figure clearly illustrates the accurate execution of the visual task, as the errors vanish. In contrast, Fig. 7b depicts the outcomes obtained when the gradient computation is based on finite differences. It exhibits much higher errors, thus highlighting the inability to precisely position.

2) Stability: Figure 8a shows the evolution of the error between the $I_{TC}{}^{th}$ predicted visual features and their desired values. At the beginning of the servoing $I_{TC} = N_p = 10$. Next, when the constrained piece of trajectory is considered small enough, I_{TC} is decremented by 1 (see III-C3d), and the prediction-reference constraint is now on the previous piece of predicted trajectory. This change in the I_{TC} value is represented by the vertical red lines. Finally, an error value close to zero indicates that the prediction-reference constraint is respected. The error between $\hat{S}(p+I_{TC})$ and S^* being null or close to null over the whole servoing and despite the numerous shifts, it can be concluded that the prediction-reference constraint is respected. Note that the relaxed velocity constraint allows

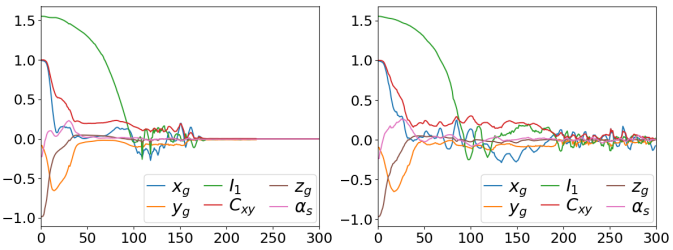


(a) Initial simulation configuration



(b) Final simulation configuration

Fig. 6: Gazebo simulation



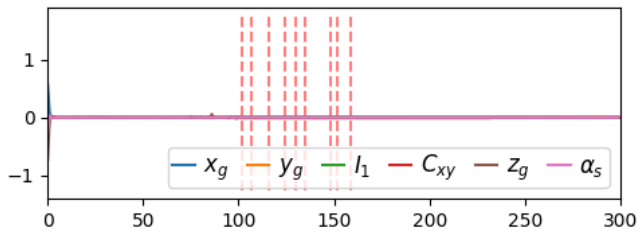
(a) $S-S^*$ evolution with CasADi (b) $S-S^*$ without CasADi

Fig. 7: Task realization results with and without CasADi

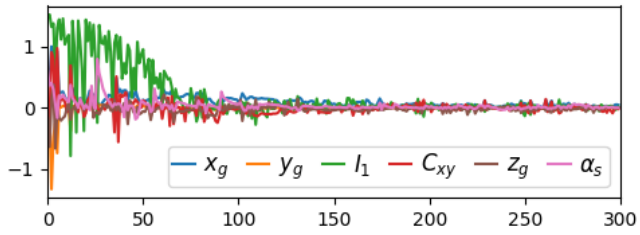
dealing with the initial configuration when the initial and desired poses are significantly different.

Figure 8b reiterates the results obtained without CasADi, emphasizing the challenge of meeting the prediction-reference constraint. This scenario demands numerous iterations for the solver to compute a solution satisfying all constraints, which is challenging within a reasonable time frame (< 200ms) without CasADi. The prediction-reference constraint error value is large at the beginning of the servoing when the arm is tucked and remains non-null until the end. It is impossible to rely on the positioning constraint set to guarantee closed-loop stability.

Thus, these results show the efficiency of the proposed approach. Furthermore, symbolical gradient computation appears to be an essential element in the experimental setups to compute a solution respecting the constraint within a time period compatible with the real-time control of a robotic system.

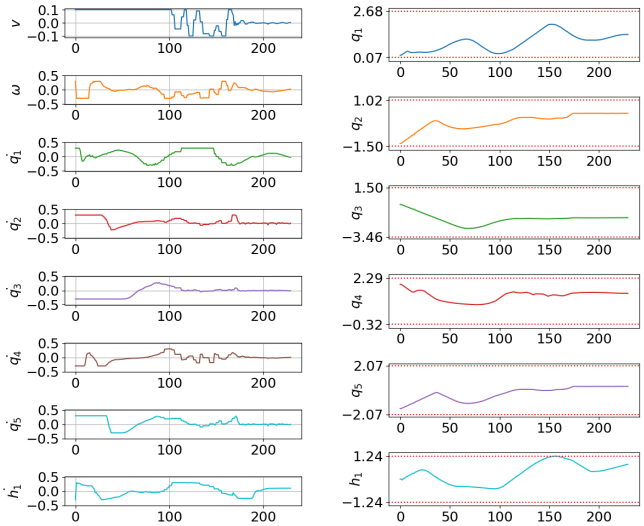


(a) Evolution of last $\hat{S}(p+I_{TC})-S^*$ with CasADi



(b) Evolution of last $\hat{S}(p+I_{TC})-S^*$ without CasADi

Fig. 8: Evolution last $\hat{S}(p+I_{TC})-S^*$ with and without CasADi



(a) Velocities evolution (b) Joint values evolution

Fig. 9: Joints and commands evolution

3) Joints and commands evolution: Finally, Fig. 9 depicts the evolution of velocities and joint angles. Despite a relaxed constraint, the velocities applied to the robot remain within the specified boundaries. As for the joint angles, they are kept within their limits, thanks to the manipulability measure.

B. Simulation results – Visual task convergence: Logarithmic vs command decrease constraint

The positioning constraint set has been defined to ensure the convergence of the visual task while maximizing manipulability. In this section, the logarithm-based method proposed in this paper and the command-based approach presented in [23] are compared.

First, let us recall the constraint set process by looking at Fig. 10, which depicts the simulated behavior following the concept explained in Fig. 5. The constrained prediction I_{TC} is initialized to $N_p = 10$. The proposed approach satisfies the terminal constraint since iteration 2 thanks to the well-defined and well-resolved optimization problem. The contribution of the relaxed input constraint is highlighted in Fig. 10a. Next, the constraint (30) imposes the transformation between the last two predicted poses to decrease (see Fig. 10b, 10c, and 10d) and the constrained prediction is kept to its current value until these predictions, i.e. the 9^{th} and 10^{th}, are close enough to be unified. When the logarithm becomes smaller than the threshold δ_{log} (see Fig. 10d), the current constraint set no longer influences the optimization, and the constraint configuration is updated by shifting the terminal constraint, i.e. $I_{TC} = 9$, as seen on Fig. 10e.

In this context, the evolution of the value I_{TC} is interesting because it measures the convergence rate (see Fig. 11). Faster achievement of $I_{TC} = 1$ corresponds to an earlier completion of the visual task. Due to the distinct nature of the prediction-prediction decrease constraint logarithm-based in Fig. 11a and command-based in Fig. 11b, two different behaviors are observed: on the one hand, the switches are started earlier, and on the other hand the interval between two switches is shorter in Fig. 11a compared to Fig. 11b. This results in a shorter servoing duration for the first case. This underlines the effectiveness of the logarithm-based constraint in contrast to the command-based approach. Indeed, quantifying the distance between two poses is inherently more accurate when utilizing an operational space measure than a joint space one.

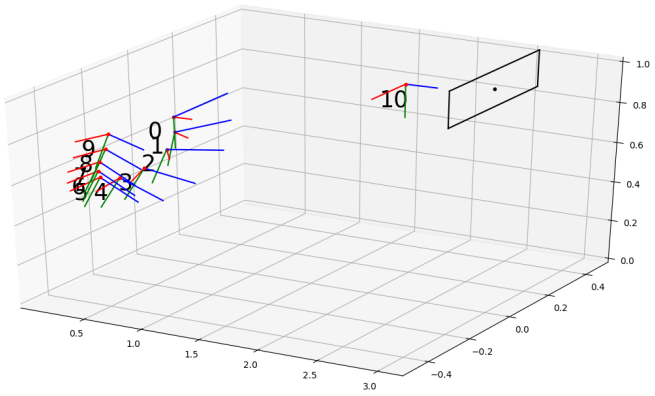
This analysis is confirmed by comparing Fig. 12 with Fig. 7a. It indicates that the execution of the visual task is significantly slower when employing the command decrease function.

C. Simulation results – Manipulability measure analysis

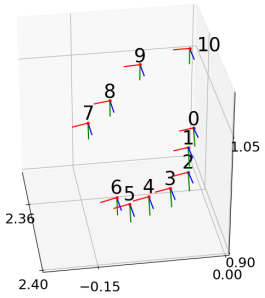
This section aims to investigate the impact of the choice of manipulability measure. Four scenarios are taken into account:

- $C1$: Without manipulability, i.e. $K_w = 0$
- $C2$: With w'_a only, i.e. $\alpha_w = 1$
- $C3$: With w'_{b+a} only, i.e. $\alpha_w = 0$
- $C4$: With w'_a and w'_{b+a} , with $\alpha_w = 0.1$

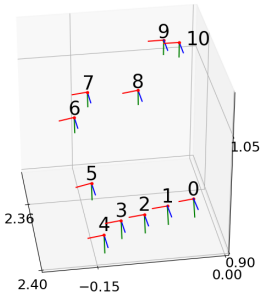
Figures 13a, 13b and 13c respectively present the w_a, w_{b+a}, and P evolution obtained for each case. These figures clearly illustrate that the $C2$ and $C3$ cases are indeed the scenarios where w'_a and w'_{b+a} are maximized, respectively, as anticipated. They also reveal that in the $C4$ case, the expected trade-off between both manipulabilities is achieved. Additionally, it is noteworthy that the evolution of w'_{b+a} in $C1$ and $C3$ yields similar results in terms of maximization. Nevertheless, in Fig. 13c, it is obvious that P decreases for the $C1$ and $C3$ cases. This can be attributed to the joint q_4 approaching its limit. The performances are similar for each measure concerning $C1$ and $C3$. This is, however, specific to the considered scenario. These two outcomes underline the significance of retaining the term w'_a in F_w. Consequently,



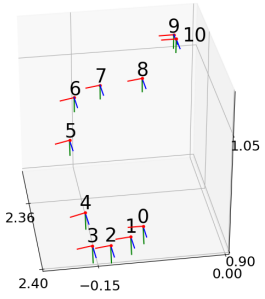
(a) Iteration 2



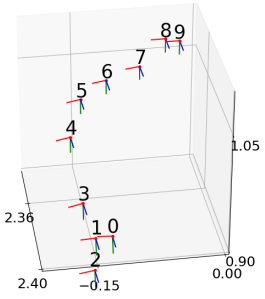
(b) Iteration 92



(c) Iteration 97

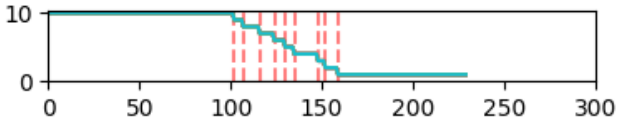


(d) Iteration 102

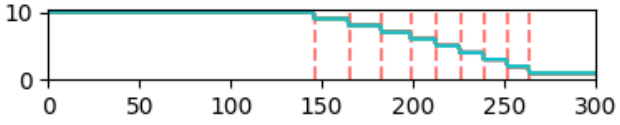


(e) Iteration 103

Fig. 10: Positioning constraint set shift results



(a) With logarithmic decrease constraint



(b) With command decrease constraint

Fig. 11: I_{TC} evolution

opting for $K_w = 0$ or $\alpha_w = 0$ is not the most appropriate choice. Now, concluding is more challenging concerning $C2$ and $C4$ cases. The performances appear comparable in the studied simulation scenario, requiring a more comprehensive analysis. Only a slight difference in the measure w'_a can be observed: the impact of this difference will be highlighted in the robot configuration trajectory in the experimental results section.

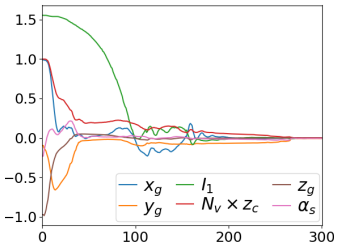
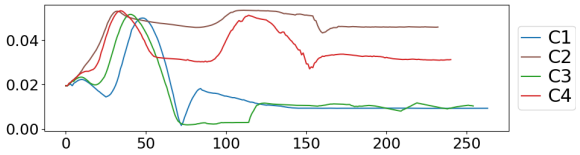
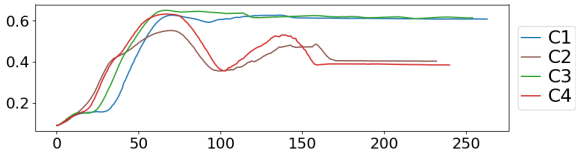


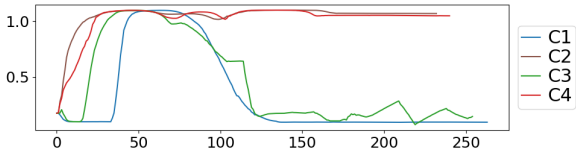
Fig. 12: $S - S^*$ evolution with command decrease constraint



(a) w_a evolution



(b) w_{b+a} evolution



(c) P evolution

Fig. 13: Manipulability measures evolution

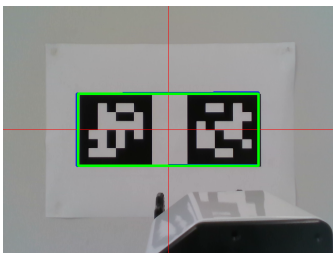
D. Experimental results

The previous simulation results highlight two setups with equivalent performances: cases $C2$ and $C4$. They have thus been implemented and tested on a robotic platform, with an initial mobile base pose $\chi_b = [0,0,0]^T$. Moreover, two other scenarios with $\alpha_w = 1$ and different initial mobile base poses have been conducted to demonstrate the proposed method's robustness and flexibility: $C2_l$ and $C2_r$, respectively, with $\chi_b = [0,+0.8,0]^T$ and $\chi_b = [0,-0.8,0]^T$. Complete trajectories are available in the video attached to this paper.

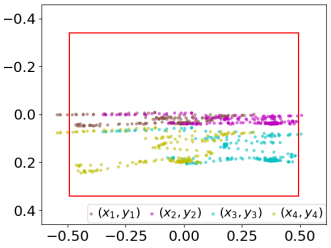
1) Visual task realization: In this scenario, the robotic system successfully achieves the task. The robot starts with a tucked arm, and the head camera can only see the landmark (cf. Fig. 14a). Nevertheless, the controller manages to drive the robot to the desired pose defined in the end-effector camera image (cf. Fig. 14b and 15). Moreover, Fig. 14c and 14d respectively display the interest points trajectory in the head

and end-effector images, confirming that the visual task is correctly performed. Also, it can be pointed out that the visual features are temporarily lost in the end-effector image[3], but thanks to the head camera, the robot can continue executing the task. However, Fig. 14c shows that the visibility constraint may sometimes be violated, as the visual features may leave the head camera field of view. This problem is due to the optimization process, which may terminate before satisfying all constraints because of the incorporated time-out. To deal with this issue, this constraint has been set up conservatively to avoid the loss of visual features.

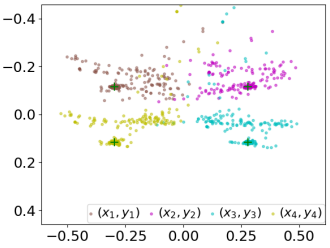
(a) Initial image - Head camera



(b) Final image - End-effector camera



(c) Points trajectory - Head camera



(d) Points trajectory - End-effector camera (The green crosses indicate the desired values)

Fig. 14: Task realization results

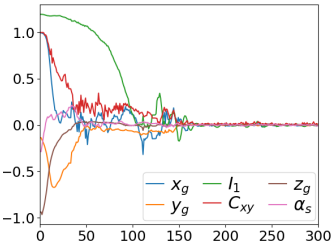
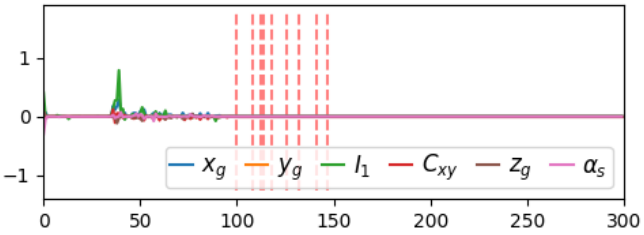
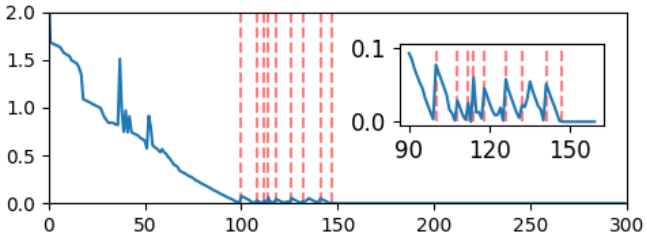


Fig. 15: $S - S^*$ evolution

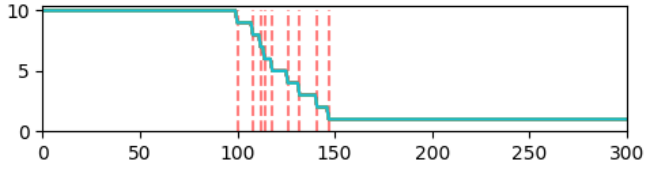
2) Stability: In this section, we analyze the performance of the positioning constraint set. In Fig. 16a, it is shown that the error between $S(p + I_{TC})$ and S^* is close to zero, indicating that the terminal constraint is generally satisfied, except in a few exceptional cases. This issue arises again because the optimization process sometimes halts and provides a solution that does not encompass the entire constraint set due to a timeout. However, the controller quickly corrects the latter and does not disturb the positioning constraint set process. In Fig. 16b, the impact of the prediction-prediction constraint can be seen. Indeed, the $\|\log_6(H_{I_{TC}})\|$ evolution shows that it is constantly decreasing. Moreover, it exhibits a triangular shape due to the I_{TC} update (cf. Fig. 16c) when it reaches a small value.



(a) Evolution last $\hat{S}(p + I_{TC}) - S^*$



(b) $\|log_6(H_{I_{TC}})\|$ evolution



(c) I_{TC} evolution

Fig. 16: Stability and convergence results

3) Robot behavior analysis: Figures 17, 18, 19, and 20 present screenshots of the robot, the end-effector camera images, and the head camera images trajectories, respectively for scenarios $C2$, $C4$, $C2_l$, and $C2_r$.

We can first analyze the first part of the robot trajectory in Fig. (a), (b), (c), and (d) for each case. This primary part is similar for all scenarios and aims to unfold the arm. This behavior is produced for two main reasons. First, the configuration with the tucked arm has small manipulability measure values due to the proximity to joint limits. The choice of the manipulability index used in the control scheme does not significantly impact this part of the movements. Second, the visual features and the weighting matrix Q_S are defined to prioritize correcting the end-effector camera orientation. The use of decoupled visual features allows us to influence the cartesian trajectory through the matrix Q_S, and this behavior can thus be slightly modified. Figures (i) - (l), and (o) - (r), present the end-effector and head image trajectories during this phase, respectively, and highlight the handling of the initial unavailability of target points in the end-effector camera by

[3]Let us recall that the visibility constraint is set on the head camera only.

using the head camera information. At the end of this first phase, the robot reaches configurations that balance the two tasks defined by F_{vs} and F_w. The system keeps indeed high movement capabilities while performing the visual task.

Figures (e), (f), and (g) depict the middle part of the movement. This latter starts when the robot comes closer to the target while still getting some room to maneuver. It emphasizes the controller's capability to use the system's redundancy to achieve its best tasks by coordinating the mobile base and the arm. When the end-effector gets closer to the desired pose, the robot uses its internal DoFs to perform the secondary task. It can be seen that the torso of the robot is rotating around the first joint q_1 to increase the manipulability. This aspect is even more noticeable when the initial position of the mobile base is far away from the final one. This behavior lasts longer for $C2_r$ than $C2$, and for $C2$ than $C2_l$, mostly justifying the longer duration of the entire trajectory. Figures (m) and (s)) present image examples for both cameras at this time: the target becomes generally visible by the end-effector camera, but the arm keeps its full motion range thanks to the visibility constraint design only set up on the head camera.

Finally, the last part of the trajectory illustrates the employment of the positioning constraint set to prioritize the visual task over the manipulability maximization. Only small movements are generated, which are quite similar for $C2$, $C2_l$, and $C2_r$. Figures (h) show the final configuration of the robot for each case. There is a noticeable difference between $C4$ and $C2$ cases where a more extended arm deployment is obtained in $C4$ because the mobile base's contribution is considered. During this part, the end-effector camera always sees the target (Fig. (n)). The visual data are thus used by the controller, avoiding reprojection errors and allowing an accurate positioning.

V. Conclusion

In this work, we have designed a multi-camera VPC strategy to control a mobile manipulator. The considered task consists of positioning the end-effector camera with respect to a given landmark. The proposed control law is fed with specific visual features, allowing the avoidance of perspective projection singularities while obtaining a nice decoupling of the camera DoFs. The visual features are computed using both the head and wrist cameras to deal with the risk of visual feature loss. From a control point of view, the method copes with several important challenges: (i) the large displacements, which in turn induce a large prediction horizon and question the stability; (ii) the large number of DoFs, which implies a large search space when optimizing, and a high redundancy leading to possible non-suitable configurations and undesired behaviors; (iii) the processing time. One of the key elements of the approach is the proposed time-varying positioning constraint set. Indeed, it prioritizes the vision-based task against the manipulability while avoiding local minima and guaranteeing closed-loop stability despite a large prediction horizon. Furthermore, we have also implemented the optimization problem using a symbolic representation to deal with the processing time. The strategy has been thoroughly simulated and evaluated using ROS and Gazebo, highlighting the best parameter choice. These latter have been implemented on our TIAGo robot. The obtained results demonstrate both the interest and efficiency of the proposed approach. In the future, we plan to extend this new framework to handle the presence of obstacles and realize more complex tasks involving both navigation and manipulation skills using vision.

References

[1] G. Colucci, L. Tagliavini, A. Botta, L. Baglieri, and G. Quaglia, "Decoupled motion planning of a mobile manipulator for precision agriculture," *Robotica*, vol. 41, no. 6, p. 1872–1887, 2023.

[2] Y. Qin, A. Escande, F. Kanehiro, and E. Yoshida, "Dual-arm mobile manipulation planning of a long deformable object in industrial installation," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 3039–3046, 2023.

[3] F. Pastor, F. J. Ruiz-Ruiz, J. M. Gómez-de Gabriel, and A. J. García-Cerezo, "Autonomous wristband placement in a moving hand for victims in search and rescue scenarios with a mobile manipulator," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 871–11 878, 2022.

[4] Y. Ma, F. Farshidian, and M. Hutter, "Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 149–12 155.

[5] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Trans. on Robotics*, vol. 26, no. 5, pp. 933–939, October 2010.

[6] F. Chaumette and S. Hutchinson, "Visual servo control, part 1 : Basic approaches," *Robotics and Automation Mag.*, vol. 13, no. 4, 2006.

[7] L. Grüne and J. Pannek, "Nonlinear model predictive control," in *Nonlinear Model Predictive Control*. Springer, 2017, pp. 45–69.

[8] A. Paolillo, T. S. Lembono, and S. Calinon, "A memory of motion for visual predictive control tasks," in *International Conference on Robotics and Automation*, 2020.

[9] F. Fusco, O. Kermorgant, and P. Martinet, "Integrating features acceleration in visual predictive control," *IEEE Rob. and Autom. Letters*, 2020.

[10] I. Mohamed, G. Allibert, and P. Martinet, "Sampling-based mpc for constrained vision based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2021)*, 2021.

[11] K. Zhang, Y. Shi, and H. Sheng, "Robust nonlinear model predictive control based visual servoing of quadrotor uavs," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 700–708, 2021.

[12] D. Pérez-Morales, O. Kermorgant, S. Domínguez-Quijada, and P. Martinet, "Multisensor-based predictive control for autonomous parking," *IEEE Transactions on Robotics*, 2021.

[13] A. Durand-Petiteville and V. Cadenat, "Advanced visual predictive control scheme for the navigation problem," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 2, pp. 1–21, 2022.

[14] S. Heshmati-alamdari, A. Eqtami, G. C. Karras, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A self-triggered position based visual servoing model predictive control scheme for underwater robotic vehicles," *Machines*, vol. 8, no. 2, p. 33, 2020.

[15] S. Norouzi-Ghazbi, A. Mehrkish, M. M. Fallah, and F. Janabi-Sharifi, "Constrained visual predictive control of tendon-driven continuum robots," *Robotics and Autonomous Systems*, vol. 145, p. 103856, 2021.

[16] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, 2020.

[17] M. Giftthaler, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3411–3417.

[18] 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.

[19] R. Colombo, F. Gennari, V. Annem, P. Rajendran, S. Thakar, L. Bascetta, and S. K. Gupta, "Parameterized model predictive control of a non-holonomic mobile manipulator: A terminal constraint-free approach," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1437–1442.

[20] S. S. Martínez, J. G. Ortega, J. G. Garcia, A. S. Garcia, and J. de la Casa Cárdenas, "Visual predictive control of robot manipulators using a 3d tof camera," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2013, pp. 3657–3662.

[21] M. Logothetis, G. C. Karras, S. Heshmati-Alamdari, P. Vlantis, and K. J. Kyriakopoulos, "A model predictive control approach for vision-based object grasping via mobile manipulator," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[22] H. Bildstein, A. Durand-Petiteville, and V. Cadenat, "Visual predictive control strategy for mobile manipulators," in *2022 European Control Conference (ECC)*, 2022, pp. 1672–1677.

[23] ——, "Multi-camera visual predictive control strategy for mobile manipulators," in *Accepted in 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2023. [Online]. Available: https://drive.google.com/file/d/1VgznusL-4HsKGYhRcrc3vmtc-aORqrBG/view?usp=share_link

[24] ——, "Enhanced visual predictive control scheme for mobile manipulator," in *2023 European Conference on Mobile Robots (ECMR)*. IEEE, 2023, pp. 1–7.

[25] O. Tahri, F. Chaumette, and Y. Mezouar, "New decoupled visual servoing scheme based on invariants from projection onto a sphere," in *2008 IEEE International Conference on Robotics and Automation*, 2008.

[26] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

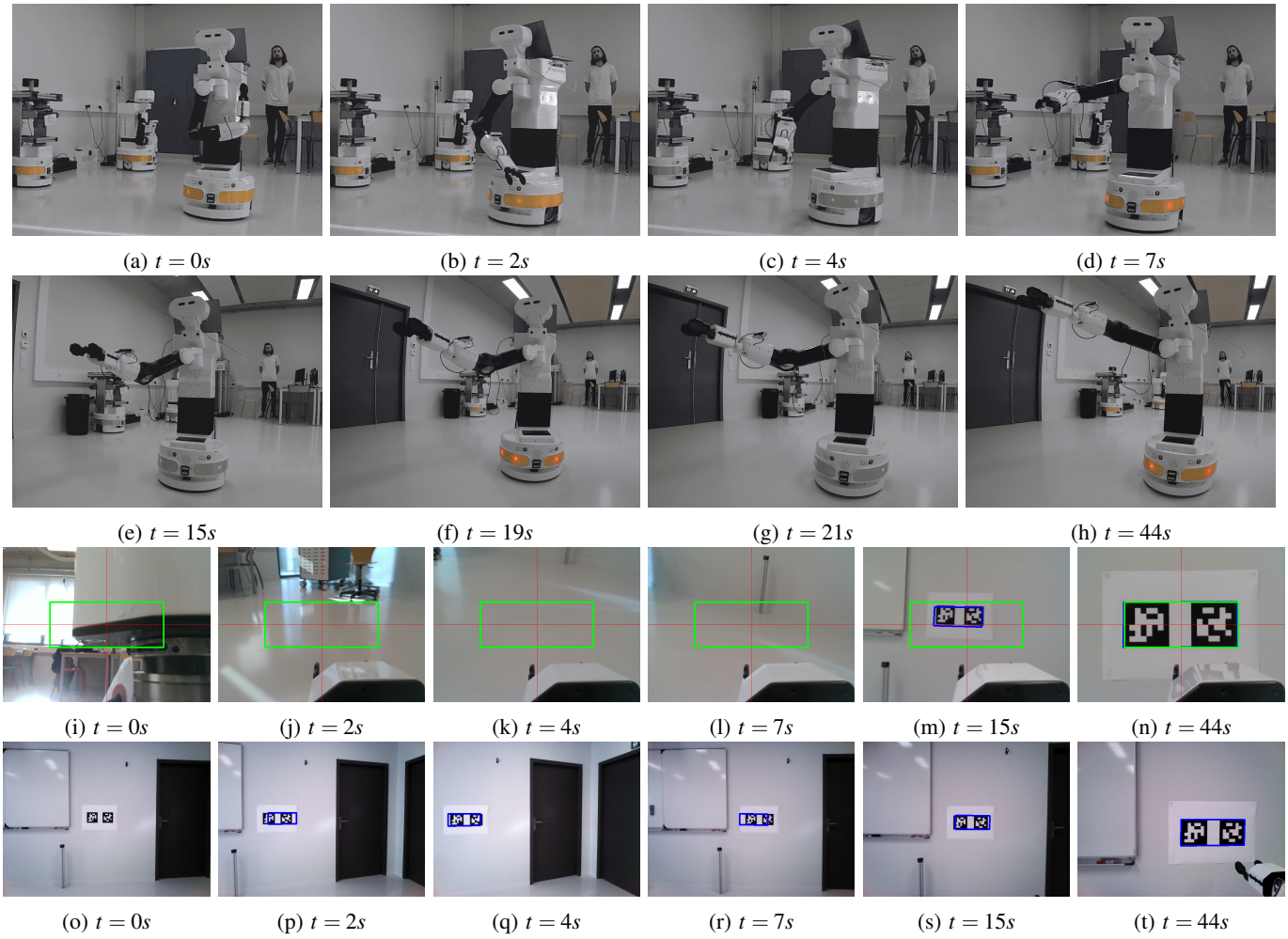
[27] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[28] N. BJ and K. PK, "Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance," *The International Journal of Robotics Research*, vol. 14, pp. 255–269, 1995.

[29] P. Scokaert, D. Mayne, and J. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *IEEE Trans. Autom. Control*, p. 648–654, 1999.

[30] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[31] J. Carpentier, F. Valenza, N. Mansard *et al.*, "Pinocchio: fast forward and inverse dynamics for poly-articulated systems," https://stack-of-tasks.github.io/pinocchio, 2015–2021.



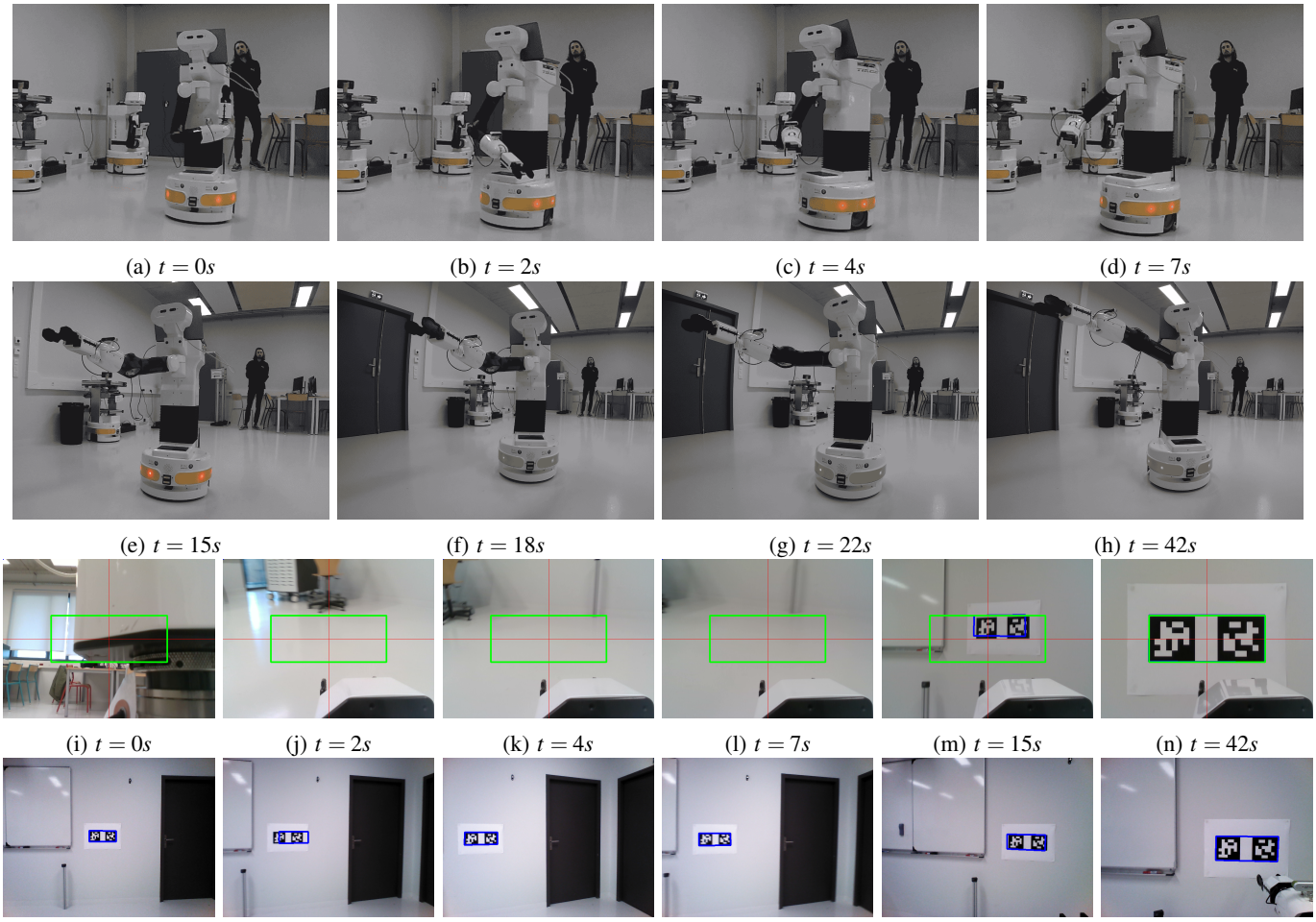
(a) $t = 0s$ (b) $t = 2s$ (c) $t = 4s$ (d) $t = 7s$

(e) $t = 15s$ (f) $t = 19s$ (g) $t = 21s$ (h) $t = 44s$

(i) $t = 0s$ (j) $t = 2s$ (k) $t = 4s$ (l) $t = 7s$ (m) $t = 15s$ (n) $t = 44s$

(o) $t = 0s$ (p) $t = 2s$ (q) $t = 4s$ (r) $t = 7s$ (s) $t = 15s$ (t) $t = 44s$

Fig. 17: $C2$ robot trajectory $(a)-(h)$, end-effector camera information $(i)-(n)$, head camera information $(o)-(t)$



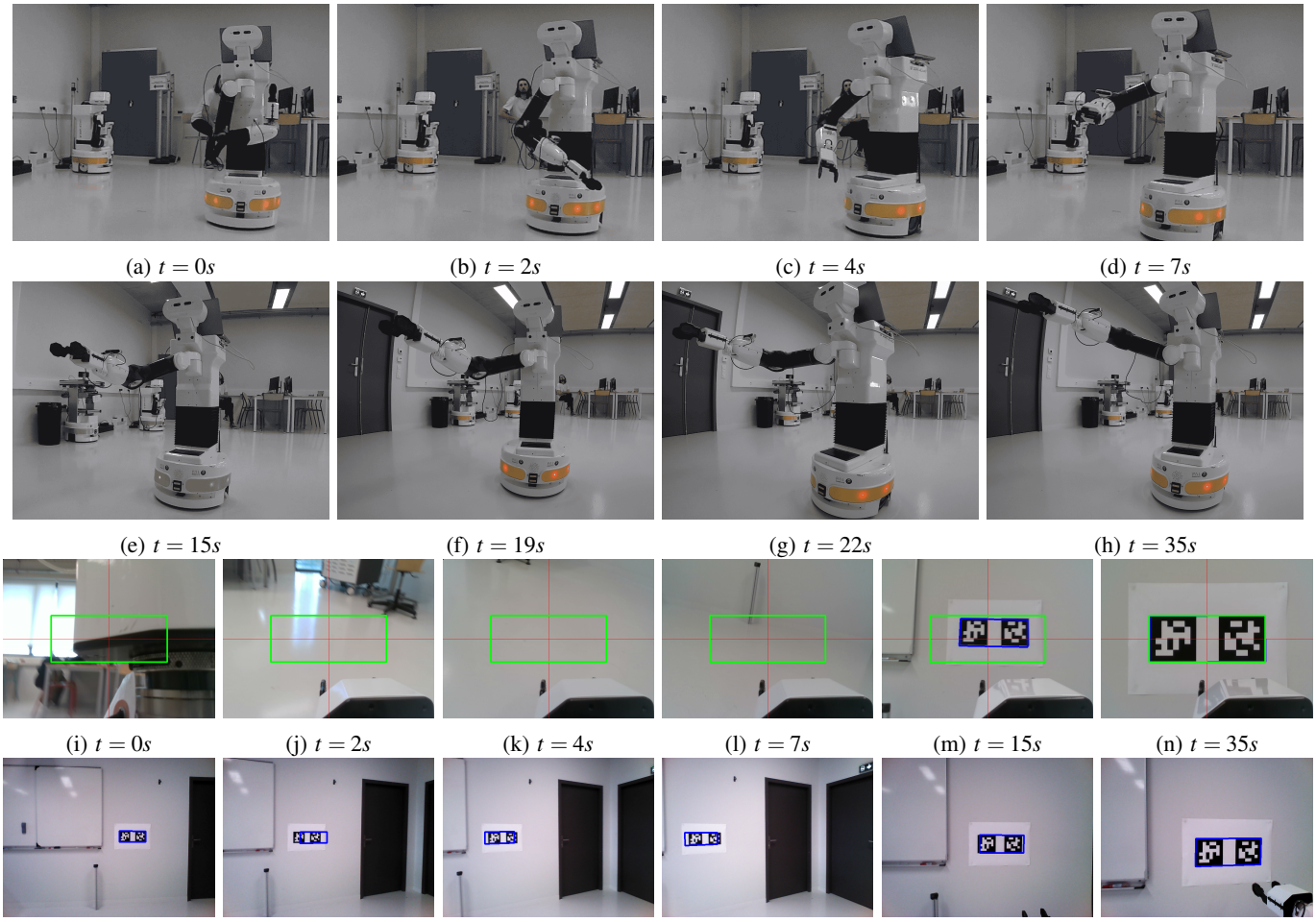
(a) $t = 0s$ (b) $t = 2s$ (c) $t = 4s$ (d) $t = 7s$

(e) $t = 15s$ (f) $t = 18s$ (g) $t = 22s$ (h) $t = 42s$

(i) $t = 0s$ (j) $t = 2s$ (k) $t = 4s$ (l) $t = 7s$ (m) $t = 15s$ (n) $t = 42s$

(o) $t = 0s$ (p) $t = 2s$ (q) $t = 4s$ (r) $t = 7s$ (s) $t = 15s$ (t) $t = 42s$

Fig. 18: $C4$ robot trajectory $(a)-(h)$, end-effector camera information $(i)-(n)$, head camera information $(o)-(t)$



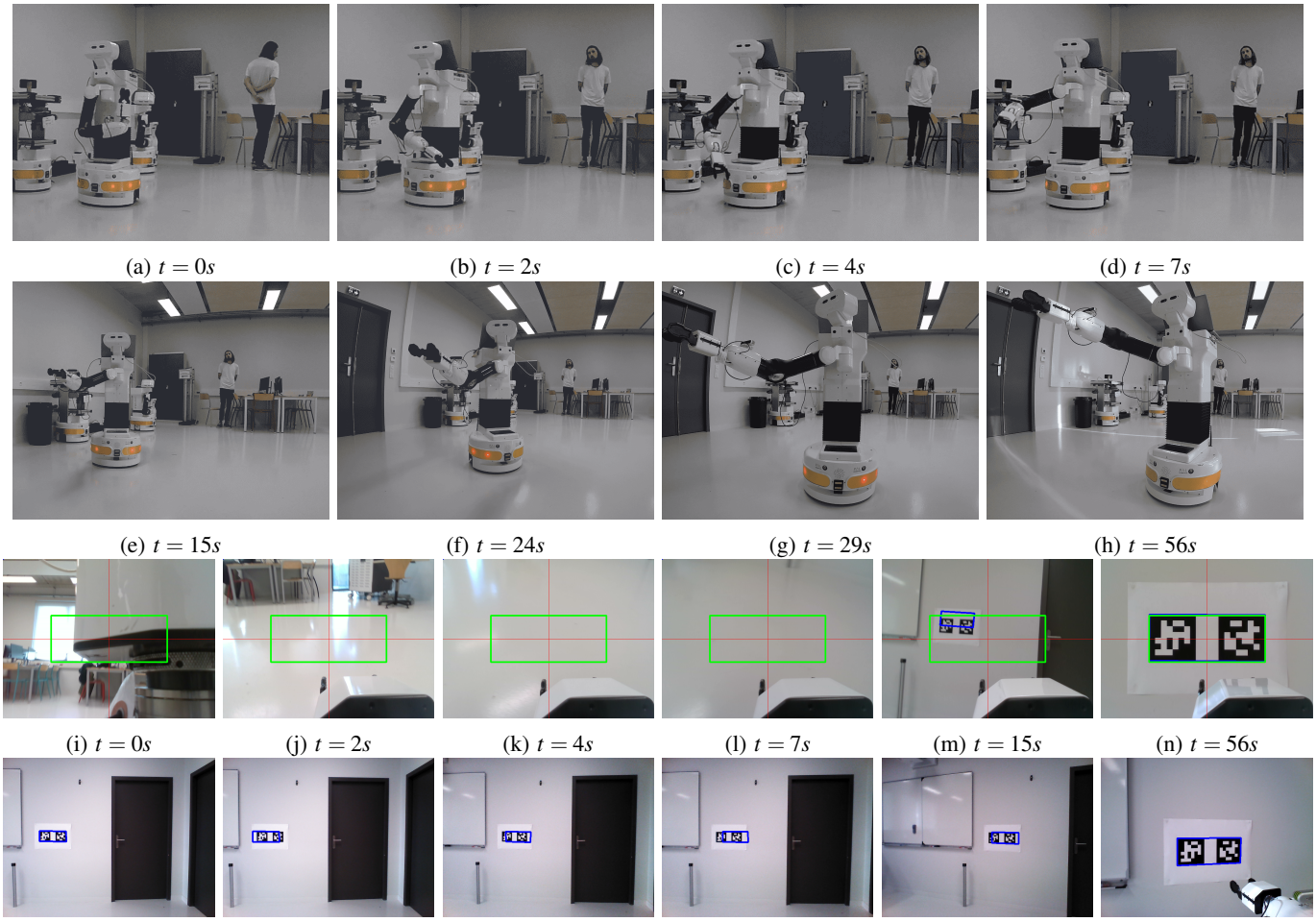
(a) $t = 0s$ (b) $t = 2s$ (c) $t = 4s$ (d) $t = 7s$

(e) $t = 15s$ (f) $t = 19s$ (g) $t = 22s$ (h) $t = 35s$

(i) $t = 0s$ (j) $t = 2s$ (k) $t = 4s$ (l) $t = 7s$ (m) $t = 15s$ (n) $t = 35s$

(o) $t = 0s$ (p) $t = 2s$ (q) $t = 4s$ (r) $t = 7s$ (s) $t = 15s$ (t) $t = 42s$

Fig. 19: $C2_l$ robot trajectory $(a)-(h)$, end-effector camera information $(i)-(n)$, head camera information $(o)-(t)$



(a) $t = 0s$ (b) $t = 2s$ (c) $t = 4s$ (d) $t = 7s$

(e) $t = 15s$ (f) $t = 24s$ (g) $t = 29s$ (h) $t = 56s$

(i) $t = 0s$ (j) $t = 2s$ (k) $t = 4s$ (l) $t = 7s$ (m) $t = 15s$ (n) $t = 56s$

(o) $t = 0s$ (p) $t = 2s$ (q) $t = 4s$ (r) $t = 7s$ (s) $t = 15s$ (t) $t = 56s$

Fig. 20: $C2_r$ robot trajectory $(a)-(h)$, end-effector camera information $(i)-(n)$, head camera information $(o)-(t)$