2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

Multi-camera Visual Predictive Control Strategy for Mobile Manipulators

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

***Abstract*—This work aims at designing a visual predictive control (VPC) scheme for a mobile manipulator equipped with two cameras. The task consists in accurately positioning the end-effector camera while starting a few meters away from the desired pose with a tucked arm. Three challenges are addressed in this paper: the initial unavailability of the visual features, the arm singularities together with the closed-loop stability, and the final positioning accuracy. The first one is dealt with by choosing image features extracted from both cameras and by suitably switching between them, the second one is tackled through a suitable manipulability measure introduced in the cost function, and the two last ones are fulfilled via the definition of an enhanced terminal constraint. The proposed approach has been validated experimentally on TIAGo robot. The obtained results show its relevance and its efficiency.**

I. Introduction

In this paper, we propose a multi-camera Visual Predictive Control (VPC) strategy to position a mobile manipulator end-effector. VPC [1] is the fusion between Nonlinear Model Predictive Control (NMPC) [2] [3] and Image-Based Visual Servoing (IBVS) [4]. The resulting control strategy combines the advantages of IBVS, *i.e.,* reactivity and absence of metric localization [5], with the ones of NMPC, *i.e.,* the possibility to take into account constraints such as joints limits and camera field of view during the minimization process. Over the last years, numerous VPC-based controllers were developed to control robotic systems such as a camera mounted on a robotic arm [6] [7] [8] [9], a quadrotor UAV [10], a mobile robot [11][12], an autonomous underwater vehicle [13] or a tendon-driven continuum robot [14]. Regarding mobile manipulators, NMPC strategies usually do not express the task in the image space. For example, in [15] the task is defined using the end-effector pose, while in [16], [17] and [18], the objective function relies on the generalized coordinates. Cameras can be used as the main sensor to control mobile manipulators but the task is not defined in the image space. In [19] and [20] the objective function is defined in the pose space, and the current pose is respectively estimated using a time of flight camera and an RGB-D camera. However, these methods need a very efficient 3D reconstruction to reach an accurate pose after convergence [21]. To our knowledge, the works presented in [22] and [23] are the only ones to consider a VPC strategy to control a mobile manipulator. In [22], a point-based hierarchical MPC is used to control an underwater manipulator vehicle, but stability issues are not taken into account. In [23], image moments obtained from a camera mounted on the end-effector of a mobile manipulator are used to control the whole system. A terminal constraint guarantees closed-loop stability while the last velocity constraints are relaxed to ensure feasibility. Simulations show that the system accurately positions the end-effector while dealing with the actuator joint limits and the camera field of view. However, the landmark must lie in the arm camera's field of view from the beginning of the task. This leads to an unsuitable motion of the robot which must move towards the target with its stretched arm, thus increasing the risk of collision, the vibrations, and the difficulty to switch towards the next potential manipulation task. In this context, using a second camera offers a complementary point of view to perform more general missions, where the robot has to start with a tucked arm.

This paper thus aims at extending the work presented in [23] with the design of a multi-camera VPC strategy and its implementation on the TIAGo robotic platform. This robot is equipped with two cameras located on the head and on the arm wrist. Here, the considered task consists in positioning the end-effector at a desired pose defined in the image space of its camera. The robot starts a few meters away from it with its arm tucked. To overcome the initial visibility problem, we rely on the head camera to compute the visual features and to project them into the space of the end-effector camera. In this way, this latter can be controlled to orientate itself towards the landmark. However, this projection might suffer from singularities due to the initial end-effector camera pose which does not allow the camera to perceive the features. To manage this issue, we propose a two-step process. First, while the landmark is not visible, the visual features are projected on an image sphere [24], *i.e.,* without projection singularities, and the whole system is controlled using cues based on spherical image moments [25]. Next, once the landmark becomes visible, the features are computed directly in the end-effector image and the robot is controlled relying on planar image moments, as these features can be more easily weighted to obtain an accurate final pose [23].

The described approach is implemented in a VPC framework allowing us to take into account constraints. Thus, in this paper, we present how the minimization problem is expressed in order to switch from spherical image moments to planar ones, as well as the constraints dealing with the actuator joint limits and camera field of view. Furthermore, we have also extended the objective function with a manipulability term [26] allowing the joints to stay away from singularities and

[†]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

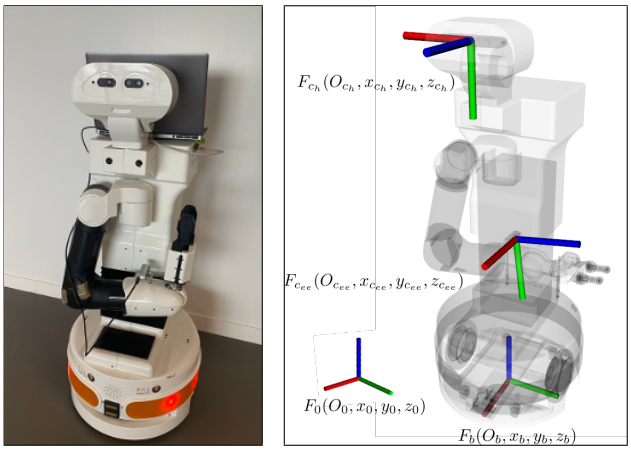
thus avoid the above-mentioned issues about collisions and vibrations. It will also make the switch to another potential manipulation task easier. Finally, we have also adapted the terminal constraint and the input constraints to handle properly the stability and the final end-effector accuracy.

The rest of the paper is organized as follows. First, the different models are introduced before detailing the proposed VPC strategy and its experimental validation on TIAGo robot. Finally, the obtained results are thoroughly discussed.

II. Preliminaries

A. Robotic system description and modeling

In this paper, we aim at positioning a camera embedded on the end-effector of a mobile manipulator relatively to a given landmark. The system is the TIAGo robot from PAL Robotics (see Fig. 1a) made of an upper body embedded on a differential mobile base. The former is composed of a 2 DoF head and a 7 DoF arm. It is equipped with two RGB-D cameras respectively fixed on the head and the wrist. Thus, this latter is controlled using only 5 DoF ($n_a = 5$), and the former uses only the yaw joint of the head ($n_h = 1$).



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

Fig. 1: The robotic system

First, we introduce the following frames: $F_0(O_0, x_0, y_0, z_0)$, $F_b(0_b, x_b, y_b, z_b)$, $F_{c_h}(O_{c_h}, x_{c_h}, y_{c_h}, z_{c_h})$ and $F_{c_{ee}}(O_{c_{ee}}, x_{c_{ee}}, y_{c_{ee}}, z_{c_{ee}})$ respectively as the world, mobile base, head camera, and wrist camera frames (see Fig. 1b). Relations for both cameras are indexed with generic c, while specific relations are detailed with c_h or c_{ee}.
The mobile base pose and its control vector are defined as:

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

where X, Y and θ are respectively the base coordinates in F_0 and the angle between F_b and F_0. v and ω are the linear and rotational velocities along x_b and around z_b. The arm configuration and its control vector are expressed as

$$\chi_a = \left[q_1, q_2, q_3, q_4, q_5\right]^T, u_a = \left[\dot{q_1}, \dot{q_2}, \dot{q_3}, \dot{q_4}, \dot{q_5}\right]^T \tag{2}$$

where q_i is the i^{th} joint angle and $\dot{q_i}$ is the i^{th} joint velocity. The same reasoning holds for the head configuration and its control vector:

$$\chi_h = \left[h_1\right]^T, u_h = \left[\dot{h_1}\right]^T \tag{3}$$

Thus, the mobile manipulator pose and its control vector are:

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

B. Projection method and visual features

1) Perspective projection (pp): The classical pinhole camera model is using a perspective projection to convert the coordinates of a 3D point $\boldsymbol{X_c} = \left[X_c, Y_c, Z_c\right]^T$, expressed in the camera frame, to the ones in the image frame.

$$\begin{bmatrix} x_i \\ y_i \\ Z_c \end{bmatrix} = \begin{bmatrix} \frac{1}{Z_c} & 0 & 0 \\ 0 & \frac{1}{Z_c} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \tag{5}$$

The drawbacks of this projection method are firstly the presence of a singularity at $Z_c = 0$, but also the discontinuity of projection when passing from $Z_c > 0$ to $Z_c < 0$ and vice-versa.

In the paper, the visual features vector is denoted S, completed with subscripts to indicate the projection method (pp for perspective and sp for spherical), and superscripts to specify the type of visual features (ip for interest points and m for moments). Classically, when relying on the perspective projection, the visual feature vector S_{pp}^{ip} is composed by the coordinates (x_i, y_i) of four interest points. This leads to:

$$S_{pp}^{ip} = \left[x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4\right]^T \tag{6}$$

[23] showed a suitable visual features vector based on 2D image moments is:

$$S_{pp}^{m} = \left[x_n, y_n, a_n, s_x, s_y, \alpha\right]^T \tag{7}$$

The normalized coordinates of the gravity center x_n and y_n, and the normalized area a_n are respectively closely related to the x, y, and z translation error, while the features s_x and s_y, derived from the image skewness, and the orientation α corresponding to the orientation of the ellipse obtained with image moments of an order less than 3 are respectively closely related to x, y, and z orientation error. These features thus offer good decoupling properties to control each DoF of the task. The calculations are more detailed in [23].

2) Spherical projection (sp): The spherical projection is used in this work to avoid the inherent singularity of the classical perspective projection. It is important to note that this projection is only virtual. It consists of the projection of the 3D points $\boldsymbol{X_c}$ on the unit sphere centered in O_c.

$$\left[x_c, y_c, z_c\right]^T = \boldsymbol{X_c} / \|\boldsymbol{X_c}\| \tag{8}$$

From this spherical image, a similar method relying this time on 3D moments can be used. However, [25] shows it is more difficult to get the same decoupling properties using 3D

moments. If O is the observed object and O_{sp} its spherical projection, these latter are defined by :

$$m_{i,j,k} = \sum_{O_{sp}} x^i y^j z^k \tag{9}$$

From [25], tests and intuition, an adequate visual features vector has been designed :

$$S_{sp} = \left[x_g, y_g, I_1, N_v \times z_c, z_g, \alpha_{sp}\right]^T \tag{10}$$

The coordinates of the gravity center x_g and y_g are respectively mainly related to the x and y translation errors, and sometimes related to the x and y rotation errors depending on the camera configuration. I_1 is a feature obtained from a suitable combination of 3D moments which is mainly related to the z translation error. The second part of the S_{sp} vector is designed using a different logic. Indeed, as for the perspective projection, controlling the x_c and y_c orientations is more complicated. Only one feature is retained to fulfill this aim: the cross product between the normal vector N_v of the target plane and the axis z_c: $N_v \times z_c$. Next, the orientation α_{sp} already used in [25] is closely related to the z orientation, thus playing an analog role to α with the stereographic projection. Finally, the last coordinate of the gravity center z_g only allows preventing the symmetric configuration (z_c looking the other way) to be considered. However, it must be noted that the major drawback of this projection method is the positioning accuracy which is in practice very hard to obtain.

In this work, both types of visual features are smartly used in the controller to avoid each drawback: S_{sp} is used to bring the robot into a suitable configuration for the perspective projection, and S_{pp} is used afterward to get an accurate convergence.

C. Manipulability and joint limits avoidance

For the issues mentioned earlier, it is necessary to avoid singularities and joint limits. To do so, it is proposed to use a specific metric denoted by w' which allows combining the envelope of a joint limits penalty function P with the manipulability index w as follows: $w' = Pw^2$ where :

$$P = 1 - \exp\left(-k \prod_{i=0}^{5} \frac{(q_i - q_{imax})(q_{imin} - q_i)}{(q_{imax} - q_{imin})}\right) \tag{11}$$

$$w = \det\left(J_{red}(\chi_a) J_{red}(\chi_a)\right)^T \tag{12}$$

$J_{red}(\chi_a)$ is the reduced Jacobian only taking into account translation velocities. This reduction is needed because only 5 joints are controlled. The elements q_{imax} and q_{imin} define the minimal and maximal joint limits and k is a positive constant. w' tends to 0 when the robot comes closer to singularities or joint limits.

III. Visual predictive control

A. The VPC scheme

As mentioned before, VPC is the result of coupling NMPC with IBVS. It thus shares characteristics from these two particular control techniques. As NMPC, it is the solution to a constrained optimization problem. More precisely, it consists in finding an optimal control sequence $U^*(\cdot)$ that minimizes a cost function J_{N_p} over a N_p steps prediction horizon under a set of user-defined constraints $C(U(\cdot))$. The obtained optimal control sequence is a N_c-dimensional vector where N_c is called the control horizon. It means that the N_c^{th} first predictions of the N_p long prediction horizon are computed using independent control inputs, while the remaining ones are all obtained using a unique control input equal to the N_c^{th} element of $U(\cdot)$. Similarly to IBVS, the cost function is defined in the image space. It is expressed as the sum of the quadratic error between the predicted visual features vector $\hat{S}$ and the desired ones S^* over the horizon N_p.

The optimal problem is then defined as follows:

$$U^*(\cdot) = \min_{U(\cdot)} \left(J_{N_p}(S(k), U(\cdot))\right) \tag{13}$$

with

$$J_{N_p}(S(k), U(\cdot)) = \sum_{p=k+1}^{k+N_p} F(p) \tag{14}$$

and

$$F(p) = \left[\hat{S}(p) - S^*\right]^T Q_S \left[\hat{S}(p) - S^*\right] + K_w / w'(p) \tag{15}$$

subject to

$$\hat{S}(k) = S(k), \hat{w}'(k) = w'(k) \tag{16a}$$

$$\hat{S}(p+1) = f(\hat{S}(p), U(p)) \tag{16b}$$

$$\hat{w}'(p+1) = g(\hat{w}'(p), U(p)) \tag{16c}$$

$$C(U^*(\cdot)) \leq 0 \tag{16d}$$

where $U^*(\cdot) = [u^*_{mm}(k), \ldots, u^*_{mm}(k+N_c-1)]$ is the computed optimal control and k represents instant $t_k = kT_s$, T_s being the prediction sampling period. f, g and $C(U^*(\cdot))$ respectively denote the prediction models and the inequality set of constraints (see next section).

Q_S is a diagonal matrix that allows weighting the error $S - S^*$ and thus prioritizing specific DoF against others. The efficient use of such a matrix has been made possible by using image moments instead of point-wise visual features as classically done in the VPC literature. K_w is also a weighting factor that balances the manipulability maximization with the visual task. Once the problem is solved, only $u^*_{mm}(k)$ is applied to the robot and the process is repeated. The previous optimization results are used to warm-start the solver.

B. The models

1) The prediction models: Two prediction models f are needed, one for each camera. These latter are obtained with the same global and exact method as in [23]. Two main steps are followed. First, the mobile base frame - camera frame relation expressed by the homogeneous transformation matrix ${}^bH_c$ is used to map the points from the camera frame to the mobile base frame. This latter is obtained using the forward kinematics model and thus only depends on the arm configuration χ_a if the end-effector camera is considered, and only on the head configuration χ_h for the head one. Next, the

relation between two mobile base poses at different instants, relying on the exact integration of the kinematic models of a differential base, gives the matrix ${}^{b_k}H_{b_{k+1}}$.

The prediction model for the points in camera frames is then given by:

$$\overline{\boldsymbol{X_c}}(k+1) = {}^{c}H_b(k+1)\,{}^{b_{k+1}}H_{b_k}\,{}^{b}H_c(k)\overline{\boldsymbol{X_c}}(k) \tag{17}$$

where the bar indicates homogeneous coordinates. Finally, g is got straightforwardly from a simple integration of χ_a.

2) The re-projection model: The information of the head camera needs to be projected in the end-effector camera frame when the latter cannot see the target. This is done using the homogeneous transformation matrix ${}^{c_{ee}}H_{c_h}$ which depends on χ_a and χ_h.

C. The projection method switch

As we mentioned earlier, the proposed strategy consists in first using the head camera to compute the visual features when the arm is tucked, *i.e.* when the end-effector camera cannot perceive the target. The visual features are then expressed in the camera end-effector frame using the spherical projection and then used to compute a control vector aimed at driving the end-effector to a pose for which the target can be seen. Once the pose is reached, the image captured by the end-effector camera is used to compute the visual features with the perspective projection method. These visual data are then used to compute the control vector driving the end-effector at the desired pose. It is then necessary to develop a switching method defining what projection method has to be used to compute the predicted visual features.

We first define an array $\mathcal{P}$ made of N_p cells. The n^{th} cell contains the projection method that has to be used to predict the visual features $\hat{S}(p)$, $\forall n \in [\![1,N_p]\!]$ and $p = k+n$, such as

$$\begin{cases} \hat{S}(p) = \hat{S}_{sp}(p) & \text{if } \mathcal{P}(n) = sp \\ \hat{S}(p) = \hat{S}_{pp}(p) & \text{if } \mathcal{P}(n) = pp \end{cases} \tag{18}$$

All cells are initialized with the sp value, meaning that the spherical projection is used for the N_p predictions. Next, our goal is to perform a smooth switch of projection method by substituting one by one the sp values by the pp ones. To do so, we first define a delimiter $p_{sp} \in [\![0,N_p]\!]$ such as:

$$\begin{cases} \mathcal{P}(n) = sp & \text{if } n \leq p_{sp} \\ \mathcal{P}(n) = pp & \text{if } n > p_{sp} \end{cases} \forall n \in [\![1,N_p]\!] \tag{19}$$

In the beginning, $p_{sp} = N_p$ to be consistent with the initial values of $\mathcal{P}$. Next, once the ${p_{sp}}^{th}$ predicted visual features are within an area belonging to the field of view of the end-effector camera, their projection mode is switched to pp. This behavior is encoded as follows:

$$p_{sp} = p_{sp} - 1 \text{ if } F(k+p_{sp}) < \delta_{switch} \tag{20}$$

where δ_{switch} is a user-defined threshold and $F(k+p_{sp})$ is calculated using (15). The process is summarized in the frame C of Fig. 2, where lines on arrows indicate conditions.

D. The enhanced terminal constraint (ETC)

The ETC is a constraint triptych that enhances the concept of TC III-D1 when working with a limited prediction horizon and non-optimal solver. The relaxation III-D3 allows the feasibility, i.e the respect of the TC is possible despite the horizon prediction limit, and the command decrease constraint III-D2 allows actually to reach this computed configuration. The constraints triptych is added when the sole perspective projection is used for all predictions.

1) The terminal constraint (TC): The TC [2] usually imposes that the last predicted visual features vector is equal to the desired one. A small necessary modification is made here: the TC is initially introduced for the last prediction and is shifted during the control:

$$\|\hat{S}^m_{pp}(k+p_{TC}) - S^{m*}_{pp}\| = 0 \tag{21}$$

where the p_{TC} is the constrained prediction index. Initially, $p^{th}_{TC} = N_p$. Of course, the TC must remain *terminal* and all predictions after the p^{th}_{TC} one have null command, this leads to (22):

$$u_{mm}(p) = 0,\ \forall p \geq k + p_{TC} - 1 \tag{22}$$

The TC is necessary for two reasons. First, it guarantees the closed-loop stability of an NMPC scheme. Second, it forces the realization of the positioning task to avoid a compromise with manipulability maximization.

2) The command decrease constraint: The use of local, and thus sub-optimal solver could make the control law stuck in a local minimum. For example, the predicted trajectory may realize the positioning task at the prediction constrained by the TC, but the first piece of trajectory could be null and the robot never reaches this pose [12]. The process to pull the robot out of these minima is again presented in Fig. 2 in the frame D: a decrease constraint is set up on the $(p^{th}_{TC} - 1)$ command until this latter is null. At this moment, the TC is shifted to the $(p_{TC} - 1)^{th}$ prediction. This process is repeated until $p_{TC} = 1$ so that the command applied to the robot actually makes it reach the pose satisfying the TC.

3) The velocity constraints: To respect the TC when the displacement to the reference configuration is too long with respect to the prediction horizon, the velocity constraints of the last inputs need to be relaxed [23]. This approach leads to the following set of constraints for the mobile base velocities:

$$\begin{bmatrix} u_{mm}(p) - u_{u|t} \\ u_{l|t} - u_{mm}(p) \end{bmatrix} \leq 0, \forall p \in [\![k, k+N_c-N_r-1]\!]$$
$$\begin{bmatrix} u_{mm}(p) - u_{u|r} \\ u_{l|r} - u_{mm}(p) \end{bmatrix} \leq 0, \forall p \in [\![k+N_c-N_r, k+N_c-1]\!] \tag{23}$$

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 limits of the actuator, and $u_{l|r}$ and $u_{u|r}$ are respectively the lower and upper relaxed boundaries.

E. The visibility constraints

In the context of visual servoing, the target must always remain visible. The following constraint allows guaranteeing

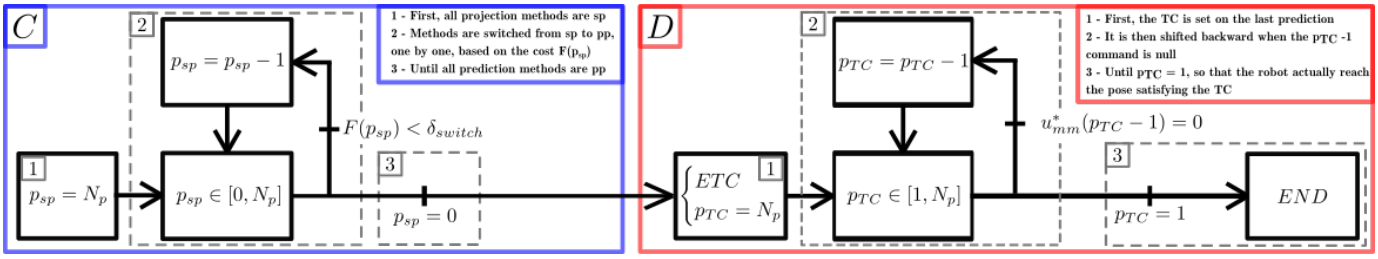


Fig. 2: Projection method switch (*C*) and ETC (*D*) processes scheme.

that the visual cues do not leave the camera's field of view. Depending on the prediction's projection method, these constraints are applied on the end-effector (*ee*) camera or the head (*h*) one:

$$\begin{cases} \begin{bmatrix} S^{ip}_{pp|h}(p) - S_{u|h} \\ S_{l|h} - S^{ip}_{pp|h}(p) \end{bmatrix} \leq 0 & \text{if } \mathcal{P}(n) = sp \\ \begin{bmatrix} S^{ip}_{pp|ee}(p) - S_{u|ee} \\ S_{l|ee} - S^{ip}_{pp|ee}(p) \end{bmatrix} \leq 0 & \text{if } \mathcal{P}(n) = pp \end{cases} \tag{24}$$

$\forall n \in [\![1, N_p]\!]$ and $p = k + n$, where $S_{l|ee}$, $S_{u|ee}$, $S_{l|h}$ and $S_{u|h}$ are respectively the lower and upper image boundaries of the end-effector and head camera.

F. The joint limits constraints

Finally, it is also necessary that the arm joints never exceed their lower and upper boundaries χ_{al} and χ_{au} defined by the elements q_{imax} and q_{imin} which leads to the constraints:

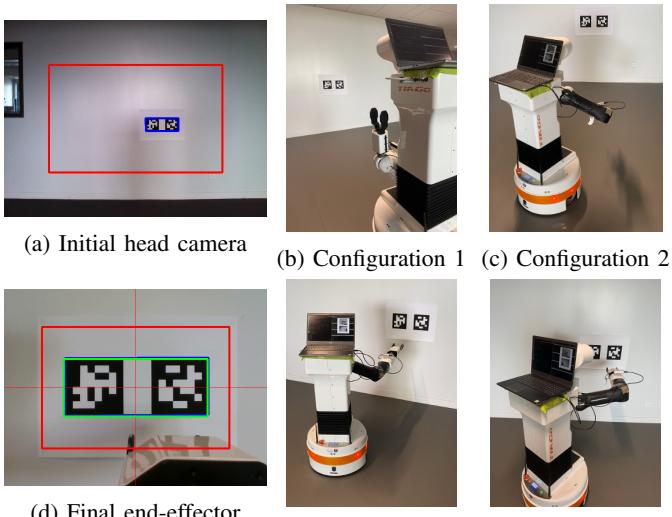
$$\begin{bmatrix} \chi_a(p) - \chi_{au} \\ \chi_{al} - \chi_a(p) \end{bmatrix} \leq 0, \forall p \in [\![k+1, k+N_p]\!] \tag{25}$$

IV. Results

This section presents experimental results to evaluate the proposed strategy. The VPC scheme is run on a TIAGo robot. All algorithms are implemented using the C++ language and the optimization problem is solved with the SLSQP solver from the NLopt package [27]. Matrices ${}^{b}H_c$ and ${}^{b_k}H_{b_{k+1}}$ are obtained with Pinocchio [28], a rigid body dynamics library. All tests are performed on an Intel Core i7-10850H and the VPC runs at a frequency of 5Hz. The solver timeout is set to $0.15s$, N_p and N_c are fixed to 10 steps with a sampling time $T_s = 0.4s$. The target is a rectangle centered in $(3, 0, 1.08625)$ and the initial robot pose is $(0, 0.3, 0)$ in F_0 with the arm tucked as shown on Fig. 3b. The camera and the mobile base have to travel about 2m to reach the target. The bounds on the mobile base linear and angular velocities are respectively equal to ± 0.1 m/s and ± 0.3 rad/s. 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 $\chi_{hl} = [-1.24]$. The matrix $Q_S(p) = diag(1,1,1,1,1,1)$ if $\mathcal{P}(p) = sp$, $Q_S(p) = diag(1,1,1,10,10,1)$ if $\mathcal{P}(p) = pp$. $K_w = 0.001$ and $k = 1e4$. Finally, when the ETC is added, $N_r = 1$.

A. Task realization

Figure 3 presents the robot trajectory with the initial (3b) and final (3f) configurations, and two other intermediate ones (3c) and (3e), which indicate the manipulability is indeed large as the arm is never stretched out. It also shows the head camera's initial view (3a) and the end-effector final view (3d), for which the visual task is achieved.



(a) Initial head camera (b) Configuration 1 (c) Configuration 2

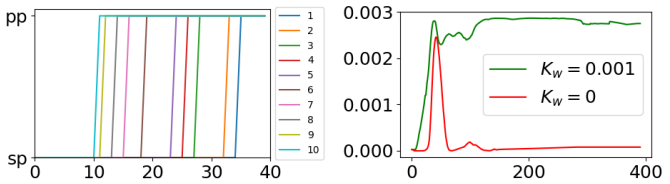
(d) Final end-effector camera (e) Configuration 3 (f) Configuration 4

Fig. 3: Experimentation snapshots.

The switching method evolution is shown in Fig. 4a. Every prediction is initially computed using the spherical projection method. As the predicted visual features fall within the end-effector camera's field of view, the projection method gradually changes to the perspective one. For example, at the 10^{th} iteration the projection method switches for the 10^{th} prediction. The transition is complete at the 35^{th} iteration when the method for the first prediction switches. From now on, there is no risk of projection singularity anymore and the spherical projection method becomes irrelevant. All prediction costs are entirely expressed with the perspective projection and ETC constraints triptych is introduced.

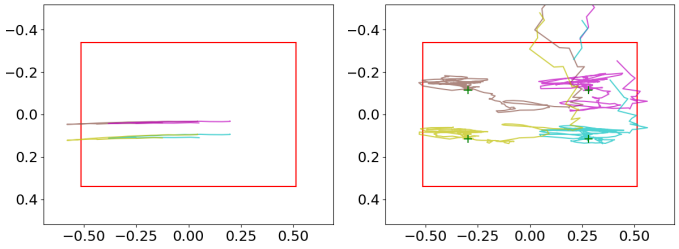
The switch divides thus the control into two parts. First, the spherical projection is used based on the head camera information projected in the wrist camera frame to bring the robot basically in a configuration where the end-effector camera has the target in sight. This leads to the second part, where the perspective projection can be used, allowing to control the end-effector camera with precision. Figure 5

follows this logic and presents the interest points trajectories in the corresponding image and the visual features vector error evolution of the first control part on the left, and of the second part on the right.

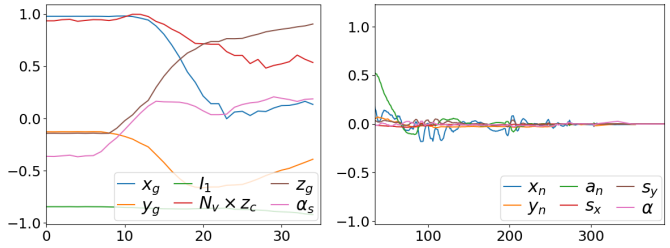


(a) Prediction method evolution (b) w' evolution

Fig. 4: Task realization results - Part 1



(a) Points trajectory - Head camera (b) Points trajectory - End-effector camera



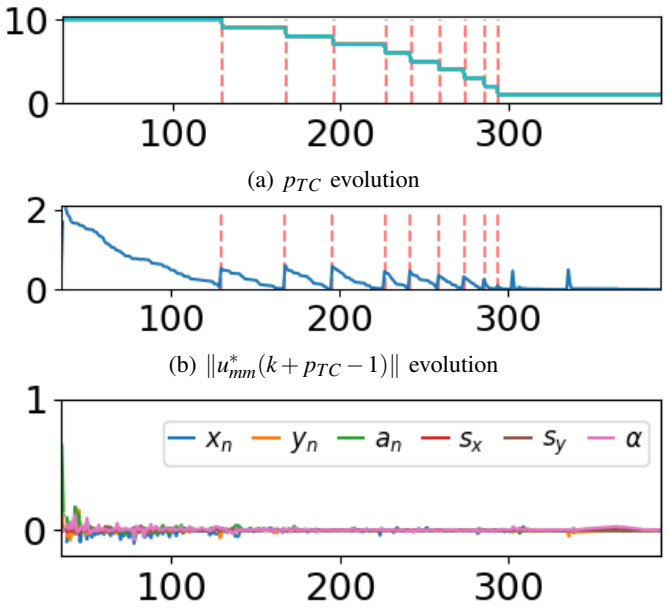
(c) $S - S^*$ evolution - sp (d) $S - S^*$ evolution - pp

Fig. 5: Task realization results - Part 2

In Fig. 5c, the feature z_c is approaching the value 1, indicating this part of the control is indeed bringing the end-effector camera facing the target. In Fig. 5b and 5d, it can be seen that the visual task is correctly performed. Indeed the controller manages to drive the camera to make the interest points reach their desired values (the green crosses). This is achieved by vanishing the error between the image moments and their desired values. In parallel of this positioning task, the controller also needs to maximize the manipulability w'. Figure 4b, plotting its evolution with and without its consideration in the cost F, shows this secondary task is also greatly achieved: w' is significantly greater with than without its consideration.

B. Stability and convergence

The stability and the precise convergence results are obtained thanks to the constraints ETC triptych. Figure 6a illustrates the p_{TC} evolution: the TC constraint is initially set up to the N_p^{th} prediction where it stays for a long time because of the input relaxation. The TC is progressively shifted prediction after prediction at the moment the $u^*_{mm}(p_{TC}-1)$ norm is almost zero. This may be possible only thanks to the command decrease constraint because of the non-optimal solution. This scheme creates thus an $u^*_{mm}(p_{TC}-1)$ norm with triangular shape (Fig. 6b) synchronized with the p_{TC} evolution. The last figure 6c shows the error between the last predicted image moments and their desired values, such that an error close to zero means the terminal constraint is respected. As it can be seen in the figure, the TC is only punctually not respected. The solver being set up with a timeout, the optimization process might stop and deliver a solution not dealing with the whole set of constraints. In Fig. 5a and 5c a similar scenario can be observed for the field of view constraints. To deal with this issue, this constraint has been set up in a conservative way to avoid the loss of visual features.



(a) p_{TC} evolution

(b) $\|u^*_{mm}(k+p_{TC}-1)\|$ evolution

(c) Evolution last $\hat{S}_m - \hat{S}^*_m$

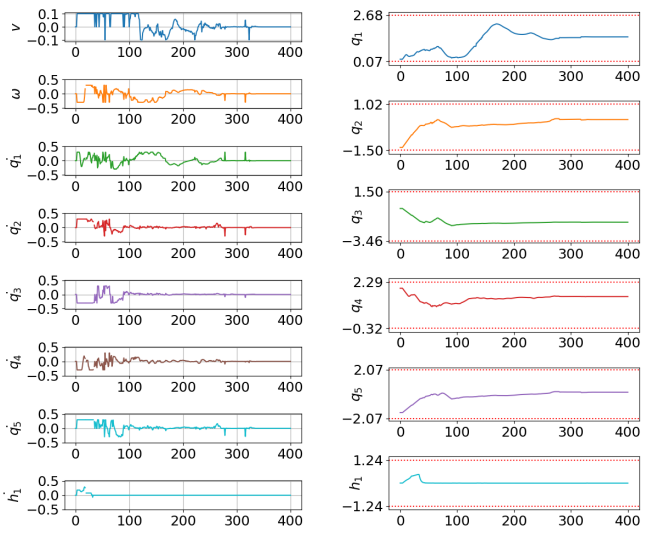
Fig. 6: Stability and convergence results

C. Joints and commands evolution

Finally, Fig. 7 shows the velocities and joint angles evolution. For both, their values remain within the given boundaries despite the use of a relaxed constraint to guarantee feasibility. Moreover, the joint angles stay away from their limits thanks to the manipulability measure.

V. Conclusion

In this paper, we have proposed a novel multi-camera VPC scheme allowing to control a mobile manipulator. The task is defined by a given pose expressed in the image space of the end-effector camera. By designing a criterion based on specific visual features extracted from both vision systems and a manipulability metric, the proposed control strategy allows overcoming two main issues: (i) the initial visibility problem if the robot has to start with a tucked arm; (ii) the singularities which may occur in particular configurations (e.g., stretched arm). It also guarantees stability and final positioning accuracy thanks to adapted constraints. It thus becomes possible to



(a) Velocities evolution (b) Joint values evolution

Fig. 7: Joints and commands evolution

avoid undesired motions, to reduce the collision risk and the vibrations, and make the switch to another manipulation task easier. The approach has been implemented on the TIAGo robot and the obtained experimental results show its relevance for efficiently controlling mobile manipulators. Based on these results, we plan to increase the use of constraints to handle the presence of obstacles and to allow multi-camera visibility management.

REFERENCES

[1] 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.

[2] F. Allgower, R. Findeisen, Z. K. Nagy *et al.*, "Nonlinear model predictive control: From theory to application," *Journal-Chinese Institute Of Chemical Engineers*, vol. 35, no. 3, pp. 299–316, 2004.

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

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

[5] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*, D. Kriegman, G. . Hager, and A. Morse, Eds. LNCIS Series, No 237, Springer-Verlag, 1998, pp. 66–78.

[6] C. Copot, C. Lazar, and A. Burlacu, "Predictive control of nonlinear visual servoing systems using image moments," *IET control theory & applications*, vol. 6, no. 10, pp. 1486–1496, 2012.

[7] A. Paolillo, T. S. Lembono, and S. Calinon, "A memory of motion for visual predictive control tasks," in *International Conference on Robotics and Automation*, no. CONF, 2020.

[8] F. Fusco, O. Kermorgant, and P. Martinet, "Integrating features acceleration in visual predictive control," *IEEE Robotics and Automation Letters*, 2020.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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, pp. 1–6.

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

[22] J. Gao, X. Liang, Y. Chen, L. Zhang, and S. Jia, "Hierarchical image-based visual serving of underwater vehicle manipulator systems based on model predictive control and active disturbance rejection control," *Ocean Engineering*, vol. 229, p. 108814, 2021.

[23] 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.

[24] T. Hamel and R. Mahony, "Robust visual servoing for under-actuated dynamic systems," 01 2000.

[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, pp. 3238–3243.

[26] 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.

[27] S. G. Johnson, "The nlopt nonlinear-optimization package," 2020. [Online]. Available: http://github.com/stevengj/nlopt

[28] 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.