Enhanced Visual Predictive Control Scheme for Mobile Manipulator

Hugo Bildstein, A. Durand-Petiteville, Viviane Cadenat

Enhanced Visual Predictive Control Scheme for Mobile Manipulator

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

***Abstract*—This paper proposes a multi-camera visual predictive control strategy for a mobile manipulator allowing to position the end-effector camera with respect to a landmark. Several issues are considered: (i) the visual landmark possible loss during navigation, (ii) the realization of large displacements which implies a large prediction horizon and impacts the closed-loop stability, (iii) the robot's high redundancy which may lead to a large search space and potential non-relevant solutions, (iv) the processing time. To cope with these challenges, 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 constraints allowing to prioritize the former against the latter. The strategy has been simulated and compared using ROS and Gazebo, showing its efficiency.**

I. Introduction

In this paper, we tackle the problem of controlling a mobile manipulator using a multi-camera Visual Predictive Control (VPC) [1] scheme. VPC combines the advantages of Image-Based Visual Servoing (IBVS) [2], *i.e.,* reactivity and absence of metric localization, with the ones of Nonlinear Model Predictive Control (NMPC) [3], *i.e.,* the possibility to take into account constraints such as joints limits and camera field of view during the minimization process. For these reasons, numerous VPC schemes were designed to control robotic arms [4] [5] [6], quadrotor UAVs [7], mobile robots [8], or autonomous underwater vehicles [9]. However, concerning mobile manipulators, NMPC schemes usually express the task using the end-effector pose [10] or the generalized coordinates [11] [12] [13]. Cameras are sometimes used to control mobile manipulators but the task is not defined in the image space [14] [15]. In such cases, the end-effector pose estimation accuracy has a significant impact on the control performances [2].

The design of a VPC scheme to control a mobile manipulator brings together the challenges related to mobile robots and robotic arms. First, the whole system contains many degrees of freedom, leading to a large search space for the NMPC optimization problem. We must then rely on an efficient solver in order to compute an optimal solution in a very short time. Next, the system is redundant and the end-effector pose can be obtained with an infinite number of configurations. However, these configurations are not equally suitable for the task to perform, and it is necessary to be able to select the most relevant ones. Then, if the mobile manipulator is equipped with a single camera to perform both navigation and manipulation tasks, it is challenging to keep the landmark in the field of view while performing an efficient trajectory. Thus, it might be interesting to consider a second camera to guarantee landmark visibility. 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 closed-loop system and it might be necessary to use large prediction horizons. To our knowledge, the works presented in [16] and [17] are among the few ones tackling some of the aforementioned challenges. In [16] the nominal VPC scheme was introduced, while [17] 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, we present a VPC scheme taking into account the aforementioned challenges. First, the robot is equipped with 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. Moreover, the positioning task is defined in the optimization problem using image moments [18], which facilitates the mapping between the task and the pose spaces. In addition to the positioning task, we also include in the optimization problem a measure of manipulability. This latter must deal with the redundancy of the robotic system by promoting configurations far from singularities. We then propose to extend the problem by adding a set of constraints, such as the classical visibility and joint limits constraints, similarly to [16]. Finally, we present the positioning constraint set guaranteeing the end-effector positioning despite the use of a local server and the presence of the manipulability measure in the optimization cost function. This method first includes the prediction-reference equality constraint, which is a modified version of the terminal constraint [3]. Next, the velocity constraint on the last predicted step is relaxed to ensure the problem's feasibility. Finally, we include a novel logarithm-based [19] 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 successfully achieve the task.

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

[†]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

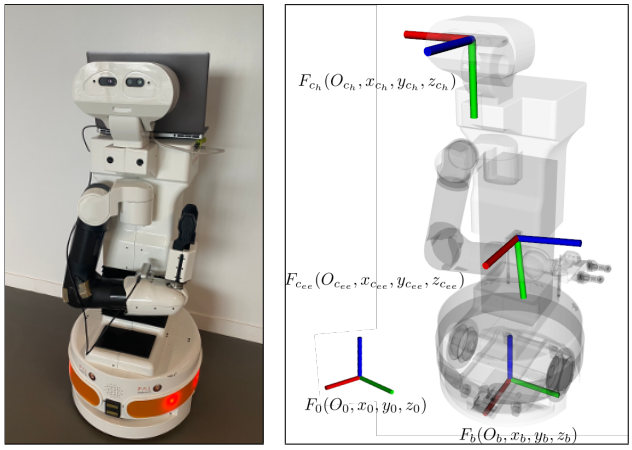
[‡]A. Durand-Petiteville is with Universidade Federal de Pernambuco UFPE, Departamento de Engenharia Mecânica, Av. da Arquitetura, 50740-550, Recife - PE, Brazil adrien.durandpetiteville[at]ufpe.br

979-8-3503-0704-7/23/$31.00 ©2023 IEEE

II. Preliminaries

A. Robotic system description and modeling

The objective is to position a camera mounted on the end-effector of a mobile manipulator relatively to a specific landmark. The considered system is the TIAGo robot from PAL Robotics, which consists of an upper body attached to a differential mobile base (cf. Fig. 1a). The upper body includes a 2-degree-of-freedom (DoF) head and a 7-DoF arm, with two RGB-D cameras fixed on the head and wrist. As a result, the wrist camera is operated using only 5 DoF ($n_a = 5$), while the head camera only utilizes the yaw joint ($n_h = 1$).



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

Fig. 1: The robotic system

First, we introduce four frames denoted as $F_0(O_0, x_0, y_0, z_0)$, $F_b(0_b, x_b, y_b, z_b)$, $F_{c_h}(O_{c_h}, x_{c_h}, y_{c_h}, z_{c_h})$, and $F_{c_{ee}}(O_{c_{ee}}, x_{c_{ee}}, y_{c_{ee}}, z_{c_{ee}})$, which respectively correspond to the world, mobile base, head camera, and end-effector camera frames (cf. Fig. 1b). In the sequel, the generic symbol c will be used to represent the relations for both cameras, the subscripts h or ee being indicated only when necessary. 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:

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

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

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

Now, it remains to express the end effector camera motion. Denoting by J_a and J_b the jacobian matrices of the arm and the mobile basis, the end effector camera kinematic screw $T_{C \in R_C/R}$ can be expressed as follows:

$$T_{C \in R_C/R} = \bar{J}_{b+a} \cdot \begin{bmatrix} u_b^T & u_a^T \end{bmatrix}^T \tag{5}$$

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

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

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

${}^{c_{ee}}X_b$ is the action matrix of the homogeneous transformation matrix ${}^{c_{ee}}H_b$ with:

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

${}^{c_{ee}}R_b$, ${}^{c_{ee}}t_b$ 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. Spherical projection method and visual features

Now, we focus on the choice of visual features allowing us to characterize the landmark. As classically done, we extract N interest points from the image provided by the camera. We can thus define a first visual feature vector S_{ip} made of the coordinates (x_i, y_i) of the N interest points of the landmark[1].

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

If 2D points are often used in visual servoing, it has been shown that they induce a strong DoF coupling, which may be an issue with complex systems such as a mobile manipulator. In this context, some works have exhibited an interest in considering the spherical projection model and using 3D moments [18]. To consider this approach, it is necessary to determine the 3D point position (X_i, Y_i, Z_i) in the camera frame. As the robot is equipped with RGB-D cameras, Z_i is available. Considering a normalized focal distance, X_i and Y_i can be deduced using the perspective projection model:

$$\begin{bmatrix} X_i \\ Y_i \end{bmatrix} = \begin{bmatrix} Z_i & 0 \\ 0 & Z_i \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{10}$$

The spherical projection consists in the projection of the 3D points $\boldsymbol{X}_i = \left[X_i, Y_i, Z_i\right]^T$ on the unit sphere centered in O_c:

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

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

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

From them, we have built the following visual features vector allowing us to obtain a good DoF decoupling, thus making easier the tuning of certain parameters of the control law [16]:

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

[1]Generally, it is necessary to consider at least 4 points to control the whole camera motion.

where (x_g, y_g, z_g) are the landmark gravity center coordinates, I_1 is a suitable combination of 3D moments, N_v the normal vector to the target plane and α_{sp} the orientation of the object projection around z_c. More details can be found in [16] and [18]. Using this projection method and the proposed visual features allows: (i) avoiding the inherent singularity around $Z_c = 0$ of the classical perspective projection and (ii) obtaining a nice decoupling DoF behavior [18], which will ease the tuning of the control law.

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

As previously mentioned, our robot is equipped with two cameras fixed on the end-effector and on the head. The task consists in positioning the first one with respect to a landmark. However, using only the visual features provided by this latter camera may lead to an undesired behavior: the arm will be stretched towards the landmark during the navigation, inducing vibrations and perturbations. To avoid this issue and allow motions with a tucked arm, we propose to project the visual information of the head camera in the end-effector camera frame. This re-projection is done using the homogeneous transformation matrix ${}^{c_{ee}}H_{c_h}$ which depends on χ_a and χ_h. 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. The control law will then be fed using the visual features either directly provided by the wrist camera as mentioned above or recomputed from the data issued by the head vision system thanks to the re-projection model.

III. The multi-camera VPC strategy

Now, we focus on our VPC strategy. We first state the considered optimal control problem before detailing the required different elements and constraints.

A. The VPC control problem

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 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 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 $U(\cdot)$. Now, let us focus on J_{N_p}. It is made of two terms. The first one F_{vs}, similarly to IBVS, explicitly depends on the visual features S and is expressed as the weighted quadratic error between the predicted visual features vector $\hat{S}$ and the desired ones S^*. The weighting is done through a diagonal positive definite matrix denoted by Q_S which allows to prioritize specific DoF against others. This matrix can be easily tuned thanks to the nice decoupling properties of the considered visual features vector S. The second term, F_w, is intended to improve the manipulability of the arm and of the entire mobile manipulator. It is defined by weighting two dedicated indices w'_a and w'_{b+a} through a gain $\alpha_w > 0$. These indices, which tend to zero when the robot comes closer to singularities and joint limits, will be defined in the next section. Finally, the balance between F_w and F_{vs} is performed through a dedicated gain denoted by $K_w > 0$. This leads to the following optimal control problem:

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

with

$$J_{N_p}(S(k), U(\cdot)) = \sum_{p=k+1}^{k+N_p} F(p) \tag{15}$$

and

$$F(p) = F_{vs}(p) + K_w F_w(p) \tag{16}$$

$$F_{vs}(p) = \left[\hat{S}(p) - S^*\right]^T Q_S \left[\hat{S}(p) - S^*\right] \tag{17}$$

$$F_w = \alpha_w / \hat{w}_a(p) + (1 - \alpha_w) / \hat{w}_{b+a}(p) \tag{18}$$

subject to

$$\hat{S}(k) = S(k) \tag{19a}$$

$$\hat{\chi}_a(k) = \chi_a(k) \tag{19b}$$

$$\hat{S}(p+1) = f(\hat{S}(p), U(p)) \tag{19c}$$

$$\hat{\chi}_a(p+1) = g(\hat{\chi}_a(p), U(p)) \tag{19d}$$

$$C(U^*(\cdot)) \leq 0 \tag{19e}$$

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). 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 prediction models

Two prediction models f and g are needed. The first one, f, is obtained with the global and exact method used in [16]. It consists of first computing the homogeneous transformation matrix ${}^bH_c$ between F_b and F_c thanks to the forward kinematic model. Then, the exact integration of the mobile base kinematic model is used to determine matrix ${}^{b_k}H_{b_{k+1}}$ which connects two successive mobile robot poses. The prediction model for the points in camera frames is given by [16]:

$$\overline{\boldsymbol{X}}_i(k+1) = {}^{c_{k+1}}H_{b_{k+1}} {}^{b_{k+1}}H_{b_k} {}^{b_k}H_{c_k} \overline{\boldsymbol{X}}_i(k) = H(k) \overline{\boldsymbol{X}}_i(k) \tag{20}$$

where the bar indicates homogeneous coordinates. The second prediction model, g, corresponds to the integration of the robotic arm kinematic model.

C. The manipulability indices

1) Manipulator manipulability: To address the aforementioned issues of singularities and joint limits, a specific metric called w'_a [20] is proposed. This metric combines the envelope of a joint limits penalty function P with the classical manipulability index w_a, as shown by the following equation:

$$w'_a = P w_a^2 \tag{21}$$

where:

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

$$w_a = \det\left(J_a^{red}(\chi_a)\, J_a^{red}(\chi_a)^T\right) \tag{22b}$$

q_{imax} and q_{imin} define the minimal and maximal joint limits and k is a positive constant. J_a^{red} is the jacobian J_a of the arm reduced to take into account only translation velocities. This reduction is needed because only 5 joints are controlled. Thus, w'_a tends to 0 when the robot comes closer to singularities or joint limits. This term has thus to be maximized.

2) Mobile manipulator manipulability: We now define a metric taking into account the mobile base, which affects the manipulability of the entire robot. From the previous reasoning, we propose the following measure:

$$w'_{b+a} = P w_{b+a}^2 = P \ \det\left(\bar{J}_{b+a}^{red}(\chi_a)\, \bar{J}_{b+a}^{red}(\chi_a)^T\right)^2 \tag{23}$$

where P is given by (22a), w_{b+a} being deduced from the entire system jacobian $\bar{J}_{b+a}^{red}$. As w'_a, this term tends to 0 when singularities and joint limits are close.

D. The visibility constraint

In the context of visual servoing, it is crucial to ensure that the target remains within the camera's field of view at all times. In the proposed framework, the visibility constraint is always set on the head image, allowing the arm to keep full freedom of movement. To achieve this, the following constraint is applied to the head image to guarantee that the visual cues do not exceed the image's limits:

$$\begin{bmatrix} S_{ip}^{c_h}(p) - S_u \\ S_l - S_{ip}^{c_h}(p) \end{bmatrix} \leq 0, \forall p \in [\![k+1, k+N_p]\!] \tag{24}$$

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

E. The joint limits constraints

Next, it is also essential 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:

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

F. The positioning constraint set

The positioning constraint set is necessary for three reasons. First, it guarantees the closed-loop stability of the NMPC scheme. Second, it forces the realization of the positioning task to avoid a compromise with manipulability maximization. Third, it prevents the robot from being stuck in a local minimum that might appear when relying on local, and thus sub-optimal, solvers. In this section, we first present the three constraints comprised in this set, and we next detail their use.

1) The prediction-reference equality constraint: Inspired by the terminal constraint method [3], we propose a prediction-reference equality constraint. It imposes that given predicted visual features are equal to the reference ones. This is expressed in the following form:

$$\|\hat{S}(k + p_{TC}) - S^*\| = 0 \tag{26}$$

where the p_{TC} is the constrained prediction index.

2) The prediction-prediction decrease constraint: We now propose a second constraint imposing the transformation between two predicted poses to decrease. To do so, we first define $H_{p_{TC}}$ as the transformation matrix between the pose at the predicted instant $k + p_{TC} - 1$ and the one at $k + p_{TC}$. 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 $se(3)$ [19]:

$$\nu = \log_6(H) \tag{27}$$

In this work, $H = H_{p_{TC}}$ is used in its homogeneous transformation matrix form and ν in its 6 dimensional motion vector form. Actually, 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:

$$\|log_6(H_{p_{TC}})\| < min_{log} - \delta_{min} \tag{28}$$

where $H_{p_{TC}} = H(k + p_{TC} - 1)$, and min_{log} represents the smallest $\|\log_6(H_{p_{TC}})\|$ value observed up to the current instant. δ_{min} is introduced to force a minimum decrease, inspired by [21]. It must be large enough to speed up the convergence but small enough to let the solver focus on the tasks.

3) The velocity constraints: To provide the necessary large prediction horizon, the velocity constraints of the last inputs can be relaxed [8]. This approach leads to the following set of constraints for the mobile manipulator velocities:

$$\begin{bmatrix} u_{mm}(p) - u_{u|t} \\ u_{l|t} - u_{mm}(p) \end{bmatrix} \leq 0, \forall p \in [\![k, k+N_c-N_r-1]\!]$$
$$\begin{bmatrix} u_{mm}(p) - u_{u|r} \\ u_{l|r} - u_{mm}(p) \end{bmatrix} \leq 0, \forall p \in [\![k+N_c-N_r, k+N_c-1]\!] \tag{29}$$

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.

4) Using the constraints: At the beginning of the servoing, p_{TC} is set up to N_p. On the one hand, constraint (26) forces the last predicted features to be equal to the desired ones, guaranteeing that the task is feasible. It should be noted that this constraint can be respected thanks to constraint (29). On the other hand, constraint (28) imposes the transformation between the last two predicted poses to decrease, prioritizing the visual task and preventing the robot from being stuck in a local minimum.

p_{TC} is maintained to its current value until the logarithm becomes smaller than a threshold δ_{log}, meaning the poses predicted at instants $k+p_{TC}-1$ and $k+p_{TC}$ are close enough to be considered equal. At this moment, the current constraint does not have an impact on the optimization anymore, and the constraint configuration must be updated by applying $p_{TC} = p_{TC}-1$. This process is repeated until $p_{TC} = 1$ so that the command applied to the robot actually makes it reach the desired pose.

IV. Results

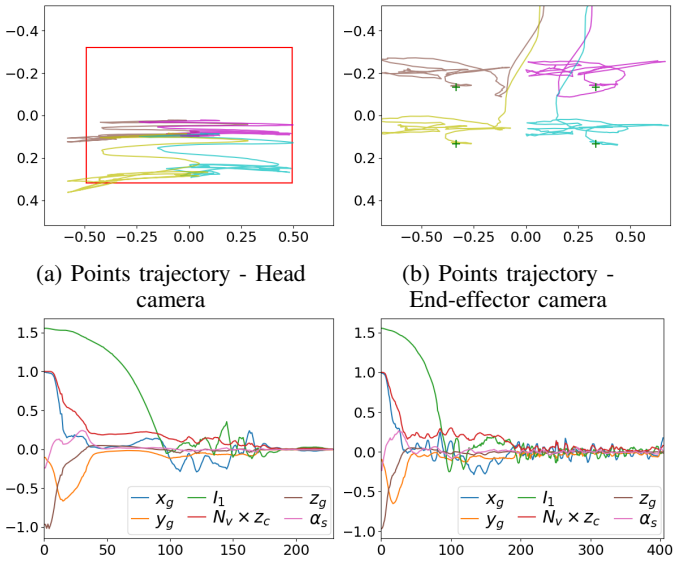
This section presents simulation results (cf. video) to evaluate the efficiency of the proposed approach. It is divided into three parts. In the first one, the presented VPC scheme is run and analyzed. Next, the scheme is tested with two different decrease constraints: one is based on the command, as in [17], and the other on the proposed logarithmic constraint to highlight its relevance. The last part analyses the influence of the manipulability measure.

All algorithms are implemented using the C++ language and the optimization problem is solved with the SLSQP solver from the NLopt package. All gradients are symbolically computed with the CasADi software [22] offline, and only evaluated online. Matrices ${}^{b}H_c$ and ${}^{b_k}H_{b_{k+1}}$ are obtained with Pinocchio [23], 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 $(3,0,1.08625)$. The camera and the mobile base have to travel about 2m to reach the target. The arm is initially tucked. 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 $\chi_{hl} = -1.24$. Matrix $Q_S(p)$ is the identity matrix, while $N_r = 1$. Time units of the plots are the control loop iterations.

A. Proposed scheme results

In this section, the approach is tested with the initial robot pose $(0,0,0)$. Additional results for other initial configurations are available in the supplementary video.

1) Visual task realization: Fig. 2b and 2c show that the visual task is correctly performed. Indeed, the controller successfully drives the camera to bring the interest points to their desired values (indicated by the green crosses), which is realized by vanishing the error between the image moments



(a) Points trajectory - Head camera

(b) Points trajectory - End-effector camera

(c) $S - S^*$ evolution

(d) $S - S^*$ without CasADi

Fig. 2: Task realization results

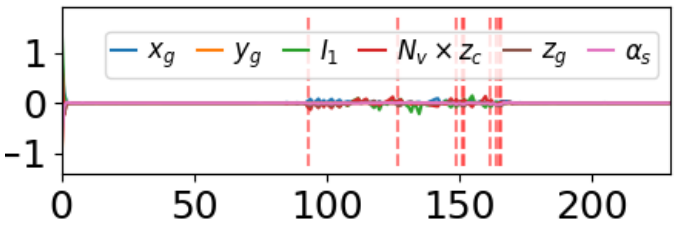
and their desired values. However, Fig. 2a shows that the visibility constraint may be sometimes violated, as the visual features may leave the head camera field of view. This problem is due to the optimization process which may terminate before satisfying all constraints because of the incorporated time-out. To deal with this issue, this constraint has been set up in a conservative way to avoid the loss of visual features. Finally, Figure 2d provides the results obtained without the use of the symbolic gradient computation with CasADi. This figure shows that an accurate positioning cannot be achieved.

2) Stability: Figure 3a displays the error between the ${p_{TC}}^{th}$ predicted image moments and their desired values, where a small error implies that the terminal constraint is satisfied. This figure clearly illustrates that the proposed scheme enables the satisfaction of the constraint despite the initial irrelevant arm configuration regarding the visual task and the large distance between the initial and desired poses. Red vertical lines represent the shifts realized to accelerate the convergence and are more detailed in section IV-B. Figure 3b presents again the obtained results without CasADi, and illustrates the difficulty to satisfy the terminal constraint when the arm is tucked. Indeed, it requires many iterations of the solver to calculate a solution respecting all the constraints, which cannot be done in a reasonable time ($< 200ms$) without CasADi.

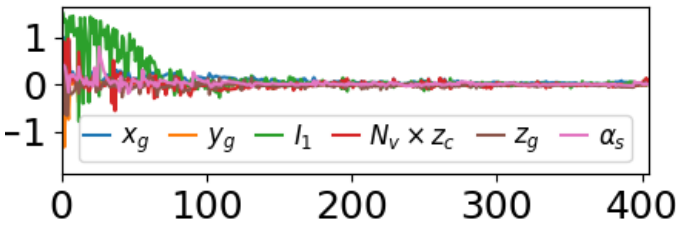
3) Joints and commands evolution: Finally, Fig. 4 shows the velocities and joint angles evolution. The values of the former remain within the given boundaries despite the use of a relaxed constraint. Concerning the joint angles, they stay away from their limits thanks to the manipulability measure.

B. Visual task convergence: Logarithmic vs command decrease constraint

To ensure the visual task convergence over the manipulability maximization, the positioning constraints set is needed.

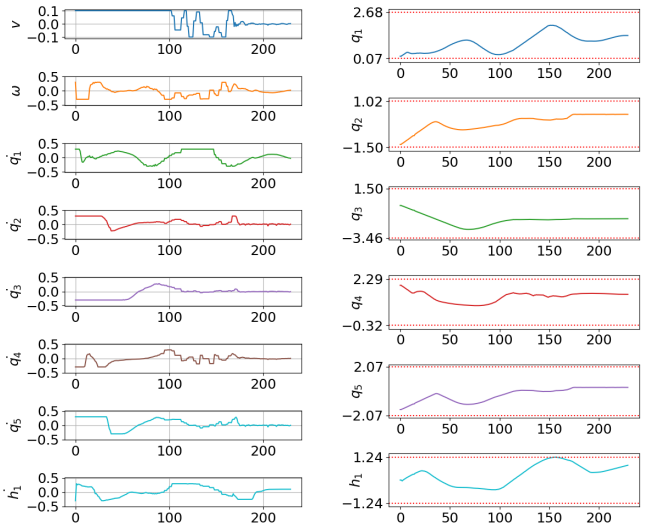


(a) Evolution of last $\hat{S}_m - \hat{S}^*_m$ with CasADi



(b) Evolution of last $\hat{S}_m - \hat{S}^*_m$ without CasADi

Fig. 3: Evolution last $\hat{S}_m - \hat{S}^*_m$

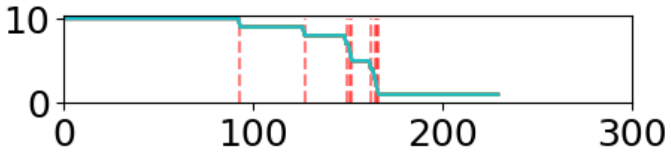


(a) Velocities evolution (b) Joint values evolution

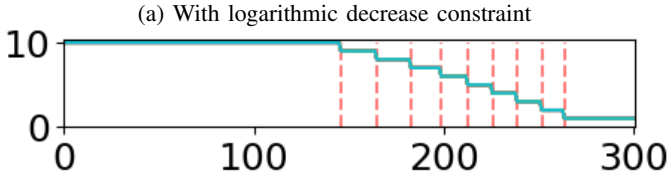
Fig. 4: Joints and commands evolution

In this section, we compare the logarithm-based method presented in this paper with the command-based one presented in [17]. To do so, we focus on the evolution of p_{TC} which quantifies the convergence rate (see Fig. 5). Indeed, the faster the value of p_{TC} is equal to zero, the faster the visual task will be completed. For both methods, the prediction-reference equality constraint is initially applied to the N_p^{th} prediction, which remains unchanged for a long time due to the input relaxation. Next, the prediction-reference equality is progressively shifted to the previous prediction until it is positioned on the first prediction, *i.e.,* $p_{TC} = 0$. However, the prediction-prediction decrease constraint being different, logarithm-based for Fig. 5a and command-based for Fig. 5b, we observe two different behaviors. In Fig. 5a, the switches are started earlier than in Fig. 5b, and the servoing is shorter. This highlights the efficiency of the logarithm-based constraint over the command-based one.

Remark: As can be seen in Fig. 3a, the prediction-reference equality constraint value is not exactly equal to zero. The chosen value to trigger the switch of the prediction-reference equality constraint has an impact on the shifting process. A smaller value smooths the transitions but increases the convergence time, while a larger one speeds up the convergence time but leads to rough changes in the optimization problem that might lead to the non-respect of the constraints.



(a) With logarithmic decrease constraint



(b) With command decrease constraint

Fig. 5: p_{TC} evolution

Figure 6 can be directly confronted to figure 2c. It shows that the visual task realization is much slower using the command decrease function.

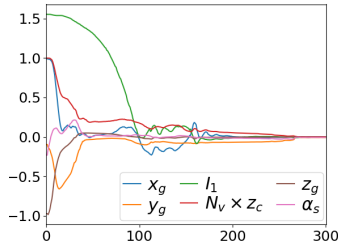


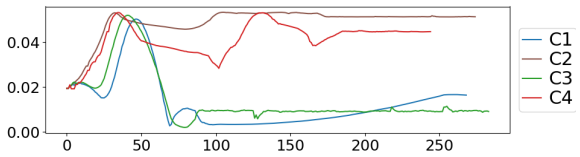
Fig. 6: $S - S^*$ evolution with command decrease constraint

C. Manipulability measure analysis

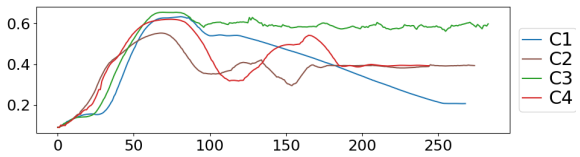
This last section studies the influence of the manipulability measure choice. Four cases are considered:

Figures 7a, 7b and 7c respectively presents the w_a, w_{b+a}, and P evolution obtained for each case. These figures clearly show that the C2 and C3 cases are indeed the scenarios where w'_a and w'_{b+a} are respectively maximized, as expected. They also demonstrate that, in the C4 case, the trade-off between both manipulabilities is reached, again as expected. It can also be noted that the evolution of w'_{b+a} in C3 and C4 leads to similar results in terms of maximization. However, Fig. 7c shows that P drops for the C3 case. It can be shown that this is due to the joint q_4 which is coming close to its limit, thus inducing convergence issues that can be observed in the video. The same behavior for P can be seen for the C1 case except it occurs lately. These two results highlight the importance of keeping the term w'_a in F_w. Thus choosing $K_w = 0$ or $\alpha_w = 0$ does not appear to be the most relevant choice. Now, regarding

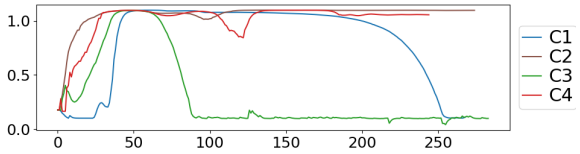
C2 and C4 cases, it is more difficult to draw a conclusion. Indeed, performances are similar in the studied simulation scenario and a more thorough analysis should be conducted.



(a) w_a evolution



(b) w_{b+a} evolution



(c) P evolution

Fig. 7: Manipulability measures evolution

V. Conclusion

This work proposes a multi-camera VPC strategy to control a mobile manipulator. It benefits from two complementary vision systems to perform a task consisting in positioning the end-effector camera with respect to a given landmark. The proposed approach allows to deal with several challenges: (i) the large displacements which imply a large prediction horizon and question the stability, (ii) the large number of DoFs which induces a large search space when optimizing, and a high redundancy leading to possible non-suitable configurations and undesired behaviors, (iii) the processing time. It relies on a cost function depending on both the visual features and the manipulability coupled with several constraints. Among them, an original positioning constraint set allows prioritizing the vision-based task against the manipulability while avoiding local minima and guaranteeing closed-loop stability despite a large prediction horizon has been introduced. In addition, to deal with the processing time, we have also implemented the optimization problem using a symbolic representation. The strategy has been simulated using ROS and Gazebo and compared to our previous work, thus demonstrating its interest and efficiency. In the future, we plan to extend this new framework to handle the presence of obstacles.

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

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] ——, "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

[18] 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.

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

[20] 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.

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

[22] 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.

[23] 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.