Visual Predictive Control Strategy for Mobile Manipulators

H Bildstein, A Durand-Petiteville, Viviane Cadenat

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. It consists in combining image-based visual servoing with model predictive control to benefit from the advantages of both control structures. Two challenges are addressed in this paper: the choice of the visual features and the closed-loop stability. The first ones rely on image moments to improve the end effector positioning precision. The second one is tackled through a terminal constraint coupled with suitable input constraints to reduce the computational burden. Simulation results using ROS and Gazebo validate the proposed approach.**

I. INTRODUCTION

In this paper it is proposed to design a Visual Predictive Control (VPC) strategy to control a mobile manipulator. VPC [1] is the fusion between Image-Based Visual Servoing (IBVS) [2] and Nonlinear Model Predictive Control (NMPC) [3] [4]. The obtained control scheme thus combines the advantages of IBVS, i.e., reactivity and absence of metric localization [5], with the ones of NMPC, i.e., ability to explicitly deal with constraints such as control inputs boundaries or camera field of view limits. Over the last decade, the interest for VPC-based controllers has grown and numerous schemes were designed to control different robotic systems: a camera mounted on a robotic arm [6] [7] [8], a flying camera [9] [10], a mobile robot [11], an autonomous underwater vehicle [12] or a tendon-driven continuum robot [13]. However, MPC strategies aiming at controlling a mobile manipulator are usually not defined in the image space. For example, in [14] the cost function represents the end-effector pose error and is then minimized using a Sequential Linear Quadratic Model Predictive Control. Another approach consists in relying on the generalized coordinates of the system to express the cost function, such as in [15], [16] and [17]. Cameras are sometimes used as the main sensor to control mobile manipulators but the task is not defined in the image space. For example, [18] and [19] respectively use a time of flight camera and a RGB-D camera to retrieve the object pose, then both rely on a pose-based cost function to achieve the task. To our knowledge, the work presented in [20] is the only one to consider a VPC strategy to control a mobile manipulator. It relies on a hierarchical MPC using points as visual features to control an underwater manipulator vehicle, but it does not deal with issues related to stability.

[†]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}@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@ufpe.br

When considering a VPC scheme to control a mobile manipulator, it is necessary to take into account two challenges. The first one is the choice of the visual features and their prediction model. Indeed, it is well known for classical IBVS, that the type of visual features has a significant impact on the trajectories in both the Cartesian and image spaces [2]. For a VPC scheme, this choice might seem to be less important, as it is possible to influence the computed trajectories via the definition of the optimization problem. However, a mobile manipulator being redundant and the point visual features strongly dependent, it is challenging to efficiently weight the cost function to increase the accuracy of the end-effector pose. For this reason, it is proposed in this work to rely on image moments as the visual features to define the cost function. These visual features are independent and it is then straightforward to weight the cost function in order to increase the pose accuracy in a given direction. Second, autonomous robotic systems represent a challenge regarding the closed-loop stability. Indeed, they usually have to perform large displacements in environments possibly cluttered with *a priori* unknown obstacles[1]. From an optimal problem point of view, the presence of obstacles induces frequent updates of the constraints, leading to significant modification of the optimal trajectory. For this reason, it is then not possible to rely on dual mode methods and a terminal constraint has to be included to guarantee the closed-loop stability [3]. Moreover, the prediction horizon has to be large enough to allow the predicted trajectory to reach goals, increasing the computational burden. Here, the constraints on the last velocities are relaxed as in [21] to obtain a trajectory dealing with the terminal constraint and the computational burden.

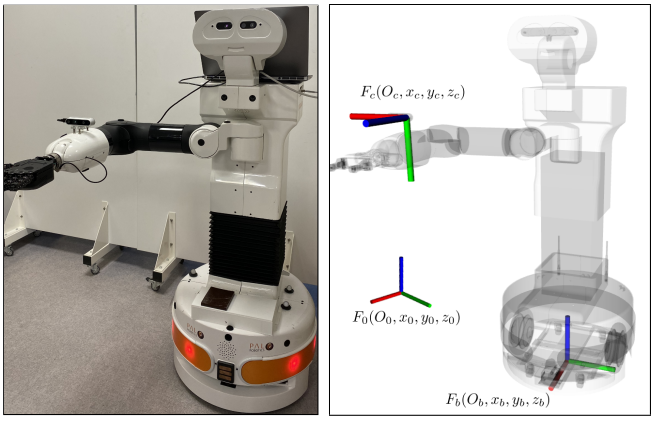
In this work, it is proposed to control a mobile manipulator by relying on a VPC scheme (i) using image moments as visual features to increase the end-effector position accuracy, (ii) including a terminal constraint to guarantee the closed-loop stability, (iii) relaxing some constraints on the velocities to reduce the computational burden, and (iv) including a constraint on the camera field of view to guarantee the success of the task. The rest of the paper is organized as follows. First, the different models are introduced. Second, the VPC strategy aiming at controlling a mobile manipulator is detailed. Next, results obtained by simulation are presented and discussed in order to highlight the relevance of the proposed approach. Finally, the future steps aiming at further developing the proposed approach are highlighted.

[1]In this preliminary work, the environment is obstacle free.

II. Preliminaries

A. Robotic system description and modeling

In this paper, we aim at controlling a stereo camera embedded on a mobile manipulator with respect to a given landmark. The camera is an Intel Realsense D435 providing RGB-D data. The system is the TIAGo robot from PAL Robotics (see Figure 1a). It consists of a 7 degrees of freedom (DOF) arm embedded on a differential mobile base. The camera is mounted on the wrist of the arm, thus only 5 DOF of the arm are used to control it ($n_a = 5$).



(a) The TIAGo robot with an Intel Realsense camera mounted on the forearm

(b) The robot model with the world frame, base frame and camera frame

Fig. 1: The robotic system

To model the robotic system, the following frames are introduced: $F_0(O_0, x_0, y_0, z_0)$ as the world frame, $F_b(0_b, x_b, y_b, z_b)$ as the mobile base frame and $F_c(O_c, x_c, y_c, z_c)$ as the camera frame (see Figure 1b). 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. Thus, the mobile manipulator pose and its control vector are:

$$\chi_{mm} = \left[\chi_b^T, \chi_a^T\right]^T, u_{mm} = \left[u_b^T, u_a^T\right]^T \tag{3}$$

B. Visual features

In this work, the task is defined in the image space by a set of visual features S characterizing the considered landmark. Thus, the goal is to make vector S converge to its desired value S^*. Classically – and in most VPC schemes – S is composed the coordinates (x_i, y_i), expressed in the image space, of four landmark interest points. This leads to:

$$S_{ip} = \left[x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4\right]^T \tag{4}$$

However, these features suffer from one major drawback: they are coupled and do not offer intuitive solutions to control the 6 DOF of the task [22]. It is thus difficult to deal with the x_c and y_c orientation errors using only interest points coordinates. Image moments can provide better information related to position and orientation errors. If O is the observed object and O_p its projection in the image, image moments m_{ij} and centered moments μ_{ij} are defined by:

$$m_{ij} = \iint_{O_p} x^i y^j \, dx dy \tag{5}$$

$$\mu_{ij} = \iint_{O_p} (x - x_g)^i (y - y_g)^j \, dx dy \tag{6}$$

where $x_g = m_{10}/m_{00}$ and $y_g = m_{01}/m_{00}$. Combinations of such moments allows to build interesting sets of visual features. Here, 6 adequate visual features designed in [23] and [22] have been used:

- The quantity $a_n = Z^* \sqrt{\frac{a^*}{a}}$, which is closely related to the z-translation error, where the area $a = m_{00}$, a^* and Z^* are respectively the desired area and depth.
- The normalized coordinates $x_n = a_n x_g$ and $y_n = a_n y_g$ of the center of gravity which are respectively closely related to the x-translation and y-translation errors.
- The orientation $\alpha = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$ related to z-orientation error. It corresponds to the orientation of the ellipse obtained with moments of order less than 3.
- The features s_x and s_y, respectively related to the x-orientation and y-orientation errors, and defined as:

$$\begin{cases} s_x = (c_2 c_3 + s_2 s_3)/K \\ s_y = (c_2 s_3 - s_2 c_3)/K \end{cases} \tag{7}$$

where $c_1 = \mu_{20} - \mu_{02}$, $s_1 = 2\mu_{11}$, $c_2 = \mu_{03} - 3\mu_{21}$, $s_2 = \mu_{30} - 3\mu_{12}$, $c_3 = c_1^2 - s_1^2$, $s_3 = 2 s_1 c_1$, $I_1 = c_1^2 + s_1^2$, $I_3 = \mu_{20} + \mu_{02}$, and $K = I_1 I_3^{3/2} / \sqrt{a}$.

The final visual features vector is then defined by:

$$S_m = \left[x_n, y_n, a_n, s_x, s_y, \alpha\right]^T \tag{8}$$

Moments m_{ij} and μ_{ij} can be computed from the interest points of arbitrary polygons [24], which leads to $S_m = g(S_{ip})$ where g is the corresponding mapping function.

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 of a constrained optimal 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 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 equals 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:

$$U^*(\cdot) = \min_{U(\cdot)} \left(J_{N_p}(S_m(k), U(\cdot)) \right) \tag{9}$$

with

$$J_{N_p}(S(k), U(\cdot)) = \sum_{p=k+1}^{k+N_p} \left[\hat{S}_m(p) - S_m^* \right]^T Q_S \left[\hat{S}_m(p) - S_m^* \right] \tag{10}$$

subject to

$$\hat{S}_{ip}(k) = S_{ip}(k) \tag{11a}$$
$$\hat{S}_{ip}(p+1) = f(\hat{S}_{ip}(p), U(p)) \tag{11b}$$
$$\hat{S}_m(p) = g(\hat{S}_{ip}(p)) \tag{11c}$$
$$C(U^*(\cdot)) \leq 0 \tag{11d}$$

where $U^*(\cdot) = [u^*_{mm}(k), \ldots, u^*_{mm}(k+N_c-1)]$ is the computed optimal control and k represents time $t = kT_s$, T_s being the control sampling period. f and $C(U^*(\cdot))$ respectively denote the prediction model and the inequality set of constraints (see next section). Q_S is a diagonal matrix which allows to weight the error $S - S^*$ and thus to prioritize 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. 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 model

As shown in Equation 11, it is mandatory to design a model computing the predicted visual features $\hat{S}_m$. As it is more straightforward to obtain a prediction model for point-wise visual features, it is proposed to first compute $\hat{S}_{ip}(p+1) = f(\hat{S}_{ip}(p), U(p))$, then to deduce $\hat{S}_m$ from the relation $S_m = g(S_{ip})$. The mapping g being defined in II-B, we now present the 3 necessary steps to obtain f.

1) The image plane - camera frame relation: First, the metric coordinates in the image frame $\left[x_i, y_i, Z_c, 1\right]^T$ of a point of interest can be mapped to their corresponding pixel coordinates $\left[u, v, Z_c, 1\right]^T$ using the following relation:

$$\begin{bmatrix} u \\ v \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 & u_0 \\ 0 & f_y & 0 & v_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ Z_c \\ 1 \end{bmatrix} \tag{12}$$

where f_x and f_y are the focal lengths, and (u_0, v_0) the coordinates of the image center, both expressed in pixel[2]. Next, the classical pinhole camera model is used to obtain the coordinates $\left[X_c, Y_c, Z_c, 1\right]^T$, expressed in the camera frame, from the ones in the image frame.

$$\begin{bmatrix} x_i \\ y_i \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{Z_c} & 0 & 0 & 0 \\ 0 & \frac{1}{Z_c} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = {}^iH_c \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{13}$$

2) The mobile base frame - camera frame relation: The coordinates expressed in the camera frame, $\left[X_c, Y_c, Z_c, 1\right]^T$, are mapped to the ones expressed in the mobile base frame, $\left[X_b, Y_b, Z_b, 1\right]^T$, via the homogeneous transformation matrix ${}^bH_c$. This latter is obtained using the forward kinematics model and thus only depends on the arm configuration χ_a.

$$\left[X_b, Y_b, Z_b, 1\right]^T = {}^bH_c(\chi_a) \left[X_c, Y_c, Z_c, 1\right]^T \tag{14}$$

3) The relation between two robot poses at different instants: To obtain such a relation, the kinematic models of a differential robot and of an arm are integrated over T_s. During this interval, v and ω are constant. It leads to:

$$\begin{cases} X(k+1) &= X(k) + \frac{v(k)}{\omega(k)} \left(\sin(\theta(k) + \omega(k)T_s) - \sin(\theta(k)) \right) \\ Y(k+1) &= Y(k) - \frac{v(k)}{\omega(k)} \left(\cos(\theta(k) + \omega(k)T_s) - \cos(\theta(k)) \right) \\ \theta(k+1) &= \theta(k) + \omega(k)T_s \end{cases} \tag{15}$$

when $\omega \neq 0$. For $\omega = 0$, the solution is straightforward. Let us define $\Delta X = X(k+1) - X(k)$, $\Delta Y = Y(k+1) - Y(k)$ and $\Delta\theta = \theta(k+1) - \theta(k)$. Thus, the homogeneous matrix between two successive mobile frames $F_b(k)$ and $F_b(k+1)$, denoted ${}^{b_k}H_{b_{k+1}}$, is defined as follows:

$${}^{b_k}H_{b_{k+1}} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & 0 & \Delta X \\ \sin(\Delta\theta) & \cos(\Delta\theta) & 0 & \Delta Y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

Finally, the prediction model f for point-wise visual features is given by:

$$\hat{S}_{ip}(k+1) = {}^iH_c(k+1) {}^cH_b(k+1) {}^{b_{k+1}}H_{b_k} {}^bH_c(k) {}^cH_i(k) \hat{S}_{ip}(k) \tag{17}$$

C. The terminal constraint (TC)

In this work, the closed-loop stability is ensured by a zero terminal equality constraint [3]. It imposes that the last predicted visual features vector is equal to the desired one. The respect of the TC guarantees the existence of a trajectory leading from the current state to the desired one, thus insuring the recursive feasibility.

$$\|\hat{S}_{mi}(k+N_p) - S^*_{mi}\| = 0, \forall i \in [\![1,6]\!] \tag{18}$$

As a strict equality constraint cannot be satisfied, the constraints are converted into inequalities by defining a small enough threshold δ_{tc}.

$$\|\hat{S}_{mi}(k+N_p) - S^*_{mi}\| - \delta_{tc} \leq 0, \forall i \in [\![1,6]\!] \tag{19}$$

[2]In this work the image coordinates are measured on a plane obtained with an unity focal, *i.e.,* when $Z_c = 1$.

D. The velocity constraints

These constraints aim at taking into account the physical boundaries of the actuators. With the number of prediction steps, they define the maximal length of the predicted trajectory. However, in order to respect the TC, the predicted trajectory has to be long enough. In the mobile manipulation context, it might represent a challenge due to the distance the mobile base has to cover to reach the goal, while the number of prediction steps cannot be too large to minimize the computational burden. Thus, to insure the TC to be satisfied all the time while taking into account the physical boundaries of the actuators, it is proposed to constrain the velocities of the first part of the inputs according to the physical boundaries, and to relax the constraints for the second part of the inputs. This approach leads to the following set of constraints for the mobile base velocities:

$$\begin{aligned} \begin{bmatrix} u_{mm}(i) - u_{u|t} \\ u_{l|t} - u_{mm}(i) \end{bmatrix} \leq 0, \forall i \in [\![k, k+N_c-N_r]\!] \\ \begin{bmatrix} u_{mm}(i) - u_{u|r} \\ u_{l|r} - u_{mm}(i) \end{bmatrix} \leq 0, \forall i \in [\![k+N_c-N_r, k+N_c]\!] \end{aligned} \tag{20}$$

where $i \in [\![k, k+N_c]\!]$, 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 actuators limits, and $u_{l|r}$ and $u_{u|r}$ are respectively the lower and upper relaxed boundaries. $N_c \gg N_r$, so that the command applied to the robot respects the actuators limits, while the TC can be ensured.

E. The visibility constraints

In the context of visual servoing, the target must remain always visible. The following constraint allows to guarantee that the visual cues do not leave the camera field of view.

$$\begin{bmatrix} S_{ip}(i) - S_u \\ S_l - S_{ip}(i) \end{bmatrix} \leq 0, \forall i \in [\![k+1, k+N_p]\!] \tag{21}$$

where S_l and S_u are the lower and upper image boundaries.

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}, which leads to the following constraints:

$$\begin{bmatrix} \chi_a(i) - \chi_{au} \\ \chi_{al} - \chi_a(i) \end{bmatrix} \leq 0, \forall i \in [\![k+1, k+N_p]\!] \tag{22}$$

IV. Results

This section presents simulation results allowing to evaluate the proposed approach and to show its efficiency. To do so, the results are divided into two parts. In the first one, the VPC scheme is run in a 2D simulator in order to highlight the relevance of the image moments over the point-wise visual features. In the second one, the Gazebo software is used to simulate the considered framework in a more realistic context.

All algorithms are implemented using the C++ language and the optimization problem is solved with the SLSQP solver from the NLopt package [25]. The matrix ${}^bH_c$ is obtained with Pinocchio [26], a rigid body dynamics library, which is using a URDF model of the robot. 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 is fixed to 10 steps with a sampling time[3] $T_s = 0.4s$. The target is a rectangle centered in $(3, 0, 1.08625)$ and the initial mobile base pose is $(0, 0, 0)$ in F_0 as shown on Figure 3a. The camera and the mobile base have to travel about 2m to reach the target. Finally, the bounds on the mobile base linear velocity is equal ± 0.1 m/s, while the angular velocities ones are fixed to ± 0.1 rad/s. The minimal and maximal joint limits are given by: $\chi_{au} = [2.68, 1.02, 1.50, 2.29, 2.07]$ and $\chi_{al} = [0.07, -1.50, -3.46, -0.32, -2.07]$.

A. Residual with visual feature vector S_{ip} vs S_m

This first set of simulations aims at showing the advantage of relying on moment visual features over point ones for a redundant systems such as a mobile manipulator. Indeed, as explained in [23], the use of point visual features makes challenging the regulation of the x_c or y_c orientation errors. To compare the two approaches, it is proposed to show the errors for DOF. To do so, we rely on a 6D cost scoring the difference between the resulting frame and the reference one. In this work, the score is expressed as:

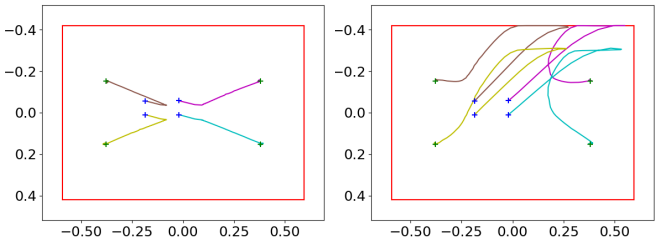
$$R_H = \log_6(H^{-1}H^*) \tag{23}$$

where H is the frame obtained with the VPC controller, H^* the reference one, and log_6 is the function mapping the group SE(3) to its Lie algebra [27]. Actually, R_H is a 6D motion vector corresponding to the velocity that should be applied during $1s$ to move from the reference frame - H^* - to the frame defined by H.

For this first test, $N_c = N_p = 10$, $N_r = 0$, $Q_S = I_6$ and the TC is not included to obtain results allowing to compare the choice of the visual features. As it is shown in Figures 2a and 2b, the VPC controller manages to drive the camera in order to make the visual features converge towards the desired ones (green crosses in the figure) for both configurations. However, the trajectory obtained in the image space is significantly shorter for the point-wise visual features than for the image moment ones. For this second case, the visibility constraint prevents the visual features from leaving the same field of view. Regarding the final pose accuracy, it can be seen in Figures 2g and 2h that the use of image moments allows to obtain a significantly more accurate result. It is confirmed by the results presented in Figures 2c and 2d, where the vector R_H corresponding to S_m is significantly smaller than the S_{ip}'s one, showing the interest of using the moments. After convergence with S_{ip}, $\|R_H\| \leq 1\text{e}{-1}$, while with S_m, $\|R_H\| \leq 1.5\text{e}{-2}$. The benefit of choosing the adequate visual features vector is almost a ratio of 10 on the R_H norm. In Figure 2c it can be seen that the difficulty is coming from ω_x. Moreover, it can be seen that the evolution of image moments (Figure 2f) follows the same trend as

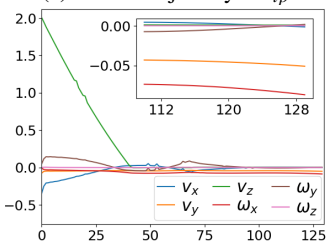
[3]This sampling time is only used for prediction, the control loop is running at 5Hz.

R_H (Figure 2d), while the one of the points (Figure 2e) is strongly different (Figure 2c). Thus, it is straightforward to tune the weighting matrix Q_S to prioritize any direction when relying on image moments as visual features.

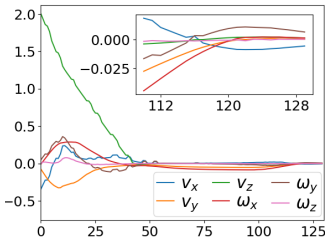


(a) Points trajectory - S_{ip}

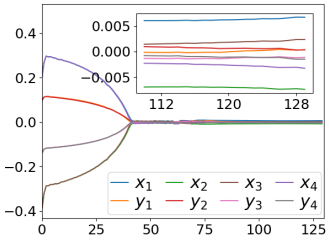
(b) Points trajectory - S_m



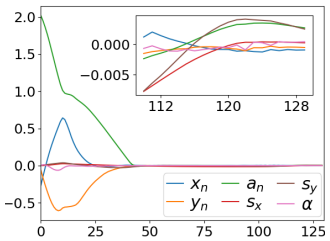
(c) R_H score evolution - S_{ip}



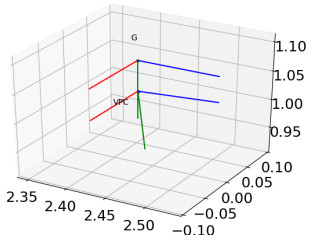
(d) R_H score evolution - S_m



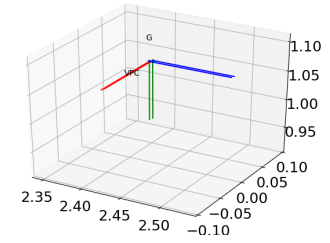
(e) S_{ip} - S^*_{ip} evolution



(f) S_m - S^*_m evolution



(g) Final frames - S_{ip}

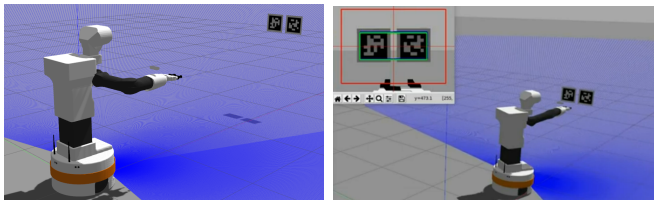


(h) Final frames - S_m

Fig. 2: First test: Comparison of the visual servoing task execution with S_{ip} on the left and S_m on the right

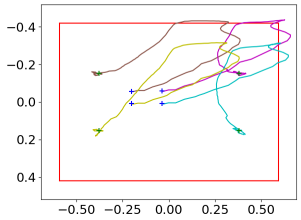
B. Results with Gazebo

In this section the proposed framework is tested with the Gazebo simulator. This latter allows to simulate robots in complex environments and to obtain more realistic scenarios, close to experimental conditions (see figure 3). The target is represented by aprilTags [28] which will be detected by a perception node. Moreover, the TC is included to deal with the closed-loop stability. Next, the prediction parameters are setup to $N_c = 9$ and $N_r = 1$ to obtain a sufficiently long predicted trajectory guaranteeing the problem feasibility. Finally, the weighting matrix is chosen as $Q_S = diag(1,1,1,10,10,1)$ to prioritize ω_x and ω_y. The obtained results are presented in Figures 4 and 5[4].

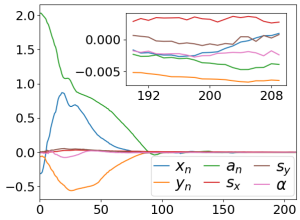


(a) Initial pose (b) Final pose

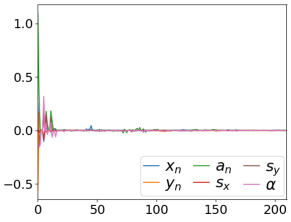
Fig. 3: TIAGo robot and the landmark in Gazebo.



(a) Points trajectory



(b) Evolution of S_m

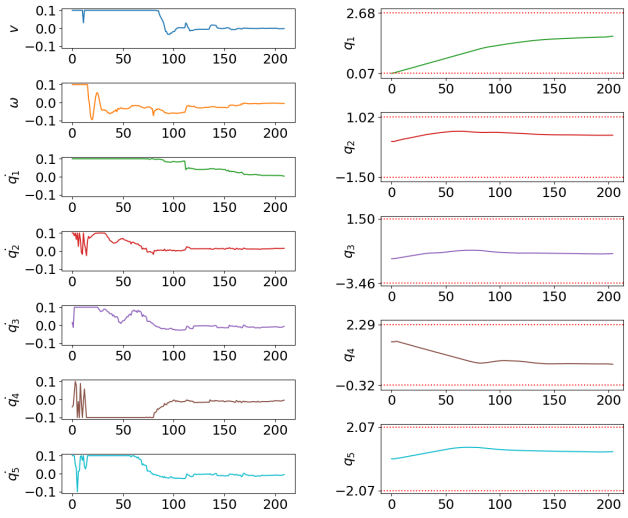


(c) Evolution last predicted moments

Fig. 4: Second test: Simulation results with Gazebo - part 1

In Figure 4a it can be seen that the task is correctly achieved. Indeed the VPC controller manages to drive the camera to make the visual features reach their desired values (the green crosses) corresponding the camera goal position (see Figure 3b). This is achieved by vanishing the error between the image moments and their desired values (see Figure 4b). In parallel to the positioning task, the controller has to deal with several constraints. For example, Figure 4c shows the error between the last predicted image moments and their desired values. An error close to zero means that the terminal constraint is respected. As it can be seen in the figure, the use of a relaxed input constraint allows to respect such a constraint despite the initial large distance between the current pose and the desired one. It can be noticed that the TC is initially and punctually not respected. The solver being setup with a timeout, the optimization process might stop and deliver a solution not dealing with the whole set of constraints. In Figure 4a a similar scenario can be observed for the field of view constraint. To deal with this issue, this

[4]A video of the resulting simulation can be found at `https://youtu.be/jjhEfflvCNc`.



(a) Velocities evolution (b) Joint values evolution

Fig. 5: Second test: Simulation results with Gazebo - part 2

constraint has been setup in a conservative way allowing the visual features to not leave the camera field of view. Finally, Figure 5 shows the velocities and joint angle evolution. For both, their values stay within the given boundaries despite the use of a relaxed constraint to guarantee the feasibility.

V. Conclusion

In this work, a VPC scheme has been designed to control a mobile manipulator. It relies on image moments to increase the end-effector position accuracy and ease the tuning of a weighting matrix to prioritise a given direction. It also ensures stability thanks to the introduction of a terminal constraint and to the definition of relaxed input constraints allowing to reduce the computational burden. The obtained results are promising, and highlight the interest and validity of the proposed approach. Several future extensions are considered: trajectory improvement (*e.g.,* [21]), additional constraints dealing with dynamical obstacles, dynamic weighting of Q_S, and validation on an experimental testbed.

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] 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.

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

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

[8] 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.

[9] S. Heshmati-alamdari, G. K. Karavas, A. Eqtami, M. Drossakis, and K. J. Kyriakopoulos, "Robustness analysis of model predictive control for constrained image-based visual servoing," in *2014 IEEE Int. Conf. on Robotics and Automation*, May 2014, pp. 4469–4474.

[10] A. Mcfadyen, P. Corke, and L. Mejias, "Visual predictive control of spiral motion," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1441–1454, 2014.

[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] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] A. Durand-Petiteville and V. Cadenat, "Visual predictive control scheme for a mobile robot navigating in a cluttered environment," in *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*. IEEE, 2020, pp. 1–6.

[22] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 713–723, 2004.

[23] O. Tahri and F. Chaumette, "Determination of moment invariants and their application to visual servoing," Ph.D. dissertation, INRIA, 2003.

[24] C. Steger, "On the calculation of arbitrary moments of polygons," *Munchen Univ., Munchen, Germany, Tech. Rep. FGBV-96-05*, 1996.

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

[26] 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.

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

[28] 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.