Grasping objects with a cable-driven parallel robot designed for transfer operation by visual servoing

Remi Ramadour, François Chaumette, Jean-Pierre Merlet

Grasping objects with a cable-driven parallel robot designed for transfer operation by visual servoing

Rémy Ramadour[1,2]   François Chaumette[1]   Jean-Pierre Merlet[2]

Abstract— **Our objective is to extend the assistance functionalities of the cable-driven parallel robot** `Marionet-Assist`, **designed principally for transfer operation, by allowing it to grasp usual objects (knives, box of medicines, phone, ⋯) by using visual servoing. Our crane robot has a configuration that provides three translational d.o.f., and a camera was added to its end-effector. In order to compute the translational velocity sent to the robot controller, the area and center of gravity in the image of the object to be grasped are used. Experimental results are presented. They show the robustness of our scheme with respect to modeling errors and an excellent positioning accuracy allowing grasping.**

I. Introduction

We will live longer, but will we live better ? The global ageing tendency of our populations raises political, economical and sociological issues. It recently was granted as a priority by several governments and administrations, thus becoming a scientific and technological challenge. A better life begins by means of autonomy and follows by preventing risks instead of treating their consequences. The Large-scale Initiative Action *Personally Assisted Living* gathers multiple Inria teams and several partners in order to offer technological solutions improving the autonomy of disabled and elderly people, as well as their quality of life. For example, daily tasks such as going into a bathtub, to the bathroom, may be painful and physically demanding and they also can lead to falling, which is one of the ten first death causes for elderly people. Fragility acts as an auto-catalyst : it sums up to itself, and the more you are, the more you will eventually become.

In order to avoid such situations, `Marionet-Assist` was designed, built and integrated in a full-scaled apartment (see. Fig.1). `Marionet-Assist` is a cable-driven parallel robot (CDPR) which provides services such as walking-aid, lifting people, and may also be used to collect information concerning for example the health of the user. The parallel structure allows for an easy lift of elderly people and has less intrusivity and a much more lighter design than any serial one. The limited workspace which is the usual drawback of parallel robots has been overcome by choosing a cable-driven mechanism [8], [13] with motorized drums that are used that can coil or uncoil the wires [1].

To achieve lifting and transfer tasks, only three translational degrees of freedom (d.o.f.) are required. For a CDPR, N+1 wires are required to control N d.o.f. if gravity is not used because wires can pull but not push [10]. In our case we use 4 wires that are attached at the same point of the platform

[1]INRIA Rennes-Bretagne Atlantique-IRISA, Rennes 35042, France (e-mail: francois.chaumette@irisa.fr)
[2]INRIA Sophia-Antipolis, Sophia-Antipolis 06902, France (e-mail: jean-pierre.merlet@inria.fr, remy.ramadour@inria.fr)

while the coiling systems are located in the ceiling so that, when not in use, the platform can move to the ceiling and will be almost invisible, with the wires running along the walls. When appropriately located the 4 wires system allows to have a total reachable workspace that almost covers a room, although only three wires will be under tension simultaneously [4], which increases the complexity of the kinematics that will be presented in Section III.

Besides being a mobility assistance device, `Marionet-Assist` remains a manipulator robot able to grasp and move objects. Prior to the modeling and conception phases, several interviews were performed with concerned people, caregivers, gerontologists and territorial authorities. It helped us to define guidelines for our devices, and highlighted also manipulating object as an important need: grabbing fallen keys, or cleaning a table are daily-life situations that convey their risk of accident for fragile persons. However, if the actual accuracy of `Marionet-Assist` is sufficient when it comes to lift a person or providing a walking-aid, a better one is required for grasping daily-life objects: we need an absolute accuracy better than 0.01 meters, while the current value is about 0.10 meters. Teleoperating the system to achieve such a task may be possible, but some situations require another approach: the end-user may not have the necessary motor and/or cognitive skills, the target may be out of sight, ...

A camera was then installed on the platform of `Marionet-Assist` as an extra-sensor to use visual servoing functionalities. Control of a CDPR using vision has been addressed in [11] for a planar CDPR actuated by three cables: using visual servoing in an eye-to-hand configuration, the authors were able to compensate the inaccuracy of the robot and achieve a successful positioning of the end-effector. Vision-based control has also been used in order to simplify the kinematic model of a CDPR [7], but the proposed method involves multiple cameras in an eye-to-hand configuration, which is not satisfying in our situation for several reasons: size of the workspace, cost and intrusivity. So, despite of their interesting results, it can not be implemented here. For similar reasons, 3D sensors like the Kinect were not considered in this work: generally oversized in comparison of the information required, this kind of sensor have the major drawback of their excessive data weight. In our work, a single webcam is used, in an eye-in-hand configuration. The experiments presented in the last section will show that by using visual servoing with a closed-loop kinematic model, the accuracy of `Marionet-Assist` improves significantly and fulfill our need, as the control scheme does not depend on the exact location of both the end-effector and the target.

In order to obtain a visual servoing scheme, we have to extract data from a series of images and measuring features that can be related to the displacement of the camera. In the literature, it often relies on key points or geometrical features [6], [9]. But, as mentioned in [3], a lot of daily-life objects does not present such features (consider a spoon, or a phone). They nevertheless ultimately need them in their own scheme. Recent works have exhibited 2D moments as a good alternative [14]. 2D moments are particularly adapted to non-textured objects without distinct shape, but they remain relevant in other cases: discrete moments could still be used with methods involving detection and matching of key points, or photometric moments [2], which seem a very promising alternative. 2D moments have two major advantages: they are easy to compute, and as it will be presented, using an adequate combination of low order 2D moments (typically computed from the area and the center of gravity), the corresponding interaction matrix has an extremely simple and linear form [5], [14]. We will then use them to control the 3 translational d.o.f. and estimate the final rotation around the Z-axis when required. But it ultimately relies on a good segmentation of the targeted object to estimate features, which remains a drawback on our situation and will be discussed in the final section.

II. MARIONET-ASSIST

(a) Global view of the apartment   (b) View of the platform

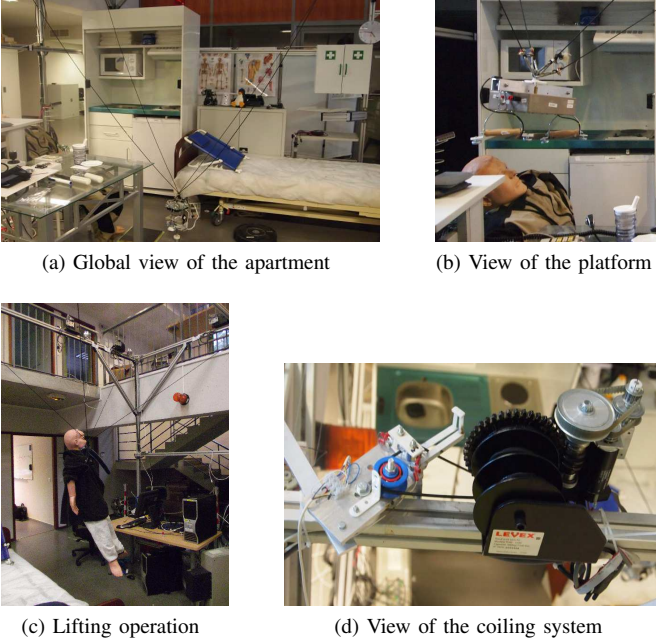(c) Lifting operation   (d) View of the coiling system

Fig. 1: The MARIONET-ASSIST wire-driven parallel robot used for transfer operations

Marionet-Assist is a modular robot, and for this experiment we use only four wires whose winches are installed in the top corners of the flat and are connected at the same point B_0 on the platform. Hence, the robot has three d.o.f. and allows to perform transfer operations in almost any point of the flat. Very low elasticity Kevlar wires are used, that can be coiled and uncoiled on motorized drums. Wire lengths, that

are essential for the robot control, are estimated through the rotation of the drum motors (see Fig. 1d). Furthermore, small aluminium foils have been glued at regular known points on the wire and can be detected at the output point of the winches: this allows one to update the current wire length as the lengths estimation based on the rotation of the drum may diverge because of the variability of the coiling process.
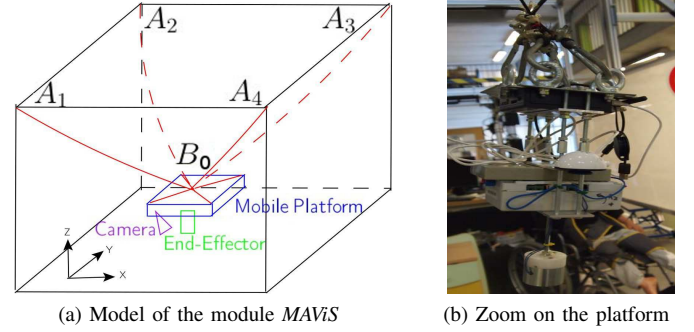

(a) Model of the module *MAViS*   (b) Zoom on the platform

Fig. 2: The MAViS (Marionet-Assist module for Visual Servoing)

III. KINEMATIC ANALYSIS OF Marionet-Assist

The mobile platform of a CDPR is connected to a fixed frame by m wires. The length of each cable is actuated by a motorized drum, which allows to move the platform with n d.o.f.. If A_i denotes the output point of the wire i and B_i its attachment point on the platform, we have the following relations :

$$\rho_i \geq ||B_i A_i|| = ||A_i C + \mathcal{R} C B_{i_r}|| \tag{1}$$

where C is an arbitrary point located in the platform (usually the center of mass), $\mathcal{R}$ the rotation matrix between the robot frame and the platform frame, B_{i_r} the location of B_i measured in the platform frame, and ρ_i the effective length of the wire. When the wire has a positive tension (meaning it is not slack), the inequality becomes an equality.

However, wires may be elastic and their length from a control viewpoint l_i can differ from the measured length ρ_i. The relation between the two values and the wire tension can be linearly approximated by:

$$\tau_i = k(\rho_i - l_i) \tag{2}$$

where k is a coefficient embedding physical characteristics of the wire (elasticity, mass, . . .), and τ_i the tension in wire i. In order to compute the required length, we thus have to obtain the tension in each wire.

This can be achieved by solving the mechanical equilibrium, given by the following equation:

$$\mathcal{F} = \mathbf{J}^{-T} \tau \tag{3}$$

where $\mathcal{F}$ stands for the vector of external forces applied to the platform, and τ the vector of tensions in wires. The matrix

$m \times n$ matrix $\mathbf{J}^{-1}$ is called the inverse kinematic Jacobian, whose rows are given by:

$$\left(\frac{A_i B_i}{||A_i B_i||} \quad \frac{CB_i \times A_i B_i}{||A_i B_i||} \right) \tag{4}$$

In our case, all the wires are attached on the same point B_0, and we select $C = B_0$. Controlling the orientation is thus not allowed, but the kinematic Jacobian becomes much simpler. Yet, controlling the orientation is not desired for this particular robot, as the rotation has to be free when it comes to help to move peoples.

However, some inaccuracies in the calibration and the modeling of the CDPR induce errors in the estimation of the positioning. There will be errors in pre-positioning and in estimating the location of both the end-effector and the target. For instance, in order to asses the repeatability of the platform without visual servoing functionalities, we have requested the robot to reach a location X^1 previously measured in the fixed reference frame, then a second location X^2 and to go back to X^1, and finally to reach a location X^3 and to go back to X^1 again. Despite showing a good repeatability relatively to the size of the workspace (maximum L_2-error $= 0.048$ meter only for this experiment, for a workspace of $4mx3mx3m$, which is quite sufficient for its original purpose), the results given in Table I show that the accuracy in the positioning the robot is not sufficient for grasping an object in an open-loop mode when its location is known. The absolution positioning accuracy has been empirically estimated to 0.10 meters, which has to be improved. Furthermore in many cases the object will be only roughly located.

	X_0^1	X_1^1	X_2^1
position(m)	(1.53,1.00,1.21)	(1.51,1.00,1.23)	(1.57,1.02,1.19)
error(m)	(0,0,0)	(-0.02,0.00,0.02)	(0.04,0.02,-0.02)

TABLE I: Repeatability errors in positioning after two trajectory requests for the robot without visual servoing

Hence we had to consider control schemes that will both allow to compensate the kinematic uncertainties of the robot and provide the grasping location. Visual-servoing functionalities allows a finer tuning of the positioning, provided a good servoing scheme. The choice was made to have an image-based approach over a position-based for several reasons, amongst which the fact that we will later consider a semi-supervised approach (see conclusions), which implies to operate on unknown targets, thus without any prior model of the target.

As the visual-servoing scheme that we use will send velocity commands, a kinematic modeling of the platform is required.

If we neglect the elasticity of the wires, in the current configuration, the inverse and forward kinematics are straightforward and admit a single solution. The velocity $\dot{\rho}$ of the wires are related to the translational velocity $\mathbf{v}$ of the platform by the following relation:

$$\dot{\rho} = \mathbf{J}^+ \mathbf{v} \tag{5}$$

where $\mathbf{J}^+$ is the 4×3 pseudo-inverse kinematic Jacobian whose rows are the vectors $\frac{A_i B_0}{\rho_i}$, with A_i being the location

of the ith wire output point (attached to the base), $C = B_0$ being the wire attachment point on the platform, and ρ_i the real length of the wire i.

This relation allows one to design a simple velocity control scheme that enables the platform to move along an arbitrary trajectory at a specified velocity. But computing the Jacobian requires to have an estimation of the location of B_0 and every length of wires with positive tension. So the numerical computing of the Jacobian will only be an approximation. Also, some rotational motions may be seen as perturbations due to the dynamic of the system and mechanical uncertainties, which could prevent a grasping task to success. Moreover, we do not know how many and which wires are in tension, implying uncontrolled changes in the configurations. But the experiments presented in this paper will show that using vision-based control allows us to overcome these drawbacks and that they remain without significant consequence on the control scheme.

IV. OBJECTIVE

Our goal is to grasp and move a target from a location A to a location B. The whole process breaks down into five steps:

1) at first the user specifies a target and an area in the room: the manipulator moves to this area.
2) then the target is detected, involving image processing.
3) the manipulator positions itself using visual servoing such that the object is centered in the image frame and at a depth of 0.20 meters.
4) as the relative location of the target relatively to the end-effector is known with a sufficient accuracy at the end of the visual servoing step, the platform moves toward its final position and grasp the target.
5) finally, the user specifies a second area which corresponds to where the target must be put down.

In our experiments, the visual servoing is made on the handle of a knife, and we use an electromagnet added to the platform in order to grasp the blade.

(a) Template of the object of interest

(b) Initial image
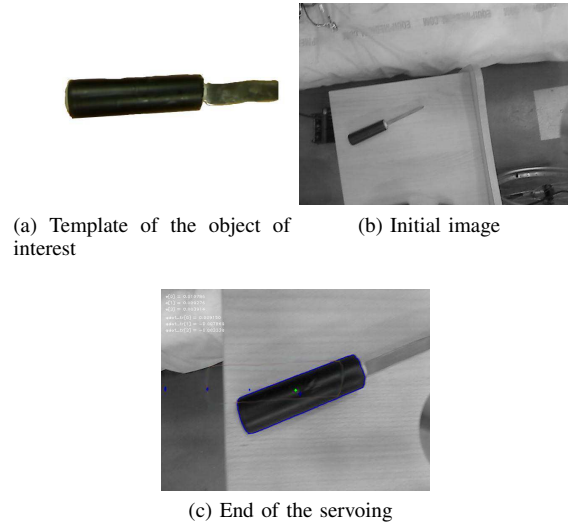

(c) End of the servoing

Fig. 3: Initial image and template

Fig.3a shows the template of the target (taken at the desired depth, close to the desired orientation), used for the detection and the measurement of the desired values of features depending on the target (such as the area in our case). Fig.3b shows a typical example of the initial image provided by the camera. Fig.3c shows the image acquired at the convergence of the visual servoing. We can note that the handle is indeed centered in the image and has the same area than the template depicted on Fig.3a.

Finally, as the blade is to be grasped, moments of order 2 are used to estimate the orientation of the object (see Fig.3c) and to provide a final displacement in order to achieve the grasping task.

V. VISUAL SERVOING

Visual servoing is a classical approach to control the robot motions in closed loop with respect to visual data [6], [9]. In this part, we will first recall the basics of this approach. Then, we will give the set of features used in our experiments, involving image moments [5]. Finally, we will present a special combination of those moments taken from [14], and expose the reasons why it is the most pertinent choice in our situation.

A. Basic concepts

First step consists in selecting a set $\mathcal{M}$ of primitives to measure in the image, easy to extract and track. From these image measurements, a set of visual features is designed, denoted by $\mathbf{s}$. We want these features to converge toward a certain desired value, denoted $\mathbf{s}^*$, from which the following error vector $\mathbf{e}$ is defined:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* \qquad (6)$$

This error has to be regulated to zero in order to converge toward our objective. A classic and simple method consists in specifying an exponential and decoupled decreasing:

$$\dot{\mathbf{e}} = -\lambda \, \mathbf{e} \qquad (7)$$

The parameter λ is a positive gain, which allows us to control the time to convergence of the system.

As the purpose is to extract from these features a way to control the robot motions, we have to model the existing relation between the variation of the error and the control input. This relation is obtained from the matrix $\mathbf{L_e}$, linking the variation of the error to the camera velocity $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$, where $\boldsymbol{v} = (v_x, v_y, v_z)$ and $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ are respectively the translational and the rotational velocity components expressed in the camera frame. Note that, in the following, we consider that only the three translational d.o.f. are controlled, so the output of the control scheme reduces to $\mathbf{v} = \boldsymbol{v} = (v_x, v_y, v_z)$.

This relation is expressed as follow:

$$\dot{\mathbf{e}} = \mathbf{L_e}\mathbf{v} \qquad (8)$$

Here, we have $\mathbf{L_e} \in \mathcal{M}_{m,n}$, where m denotes the number of features involved, and n is the number of d.o.f. controlled (here $n = 3$).

Ultimately, using (6), (7) and (8), the control law is given by:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L_e}}^+ (\mathbf{s} - \mathbf{s}^*) \qquad (9)$$

where $\widehat{\mathbf{L_e}}^+$ stands for the pseudo-inverse of an estimation of $\mathbf{L_e}$. Note that when the interaction matrix is square and full-rank, $\widehat{\mathbf{L_e}}^+$ is then equal to $\widehat{\mathbf{L_e}}^{-1}$. For further investigation on visual servoing, the reader is invited to refer to [6].

What remains important here is the necessity to carefully choose a set of features allowing us to obtain an adequate interaction matrix $\mathbf{L_e}$, linear when it is possible.

As will be described in the following part, these requirements are fulfilled by using the area and the center of gravity of the target as primitives, from which a set of features and a linear interaction matrix can be determined for our particular configuration.

B. 2D moments

Once the object of interest has been segmented in the image, moments are features easy to compute and present some nice geometrical properties. We recall that, in this paper, we are only concerned with the three translational motion.

The definition of a 2D moment of order (i, j) is $m_{ij} = \sum_{(x,y)\in\mathcal{T}} x^i y^j$, where (x, y) are the coordinates of the pixels belonging to $\mathcal{T}$, corresponding to the image of the target.

As shown in [5], only the area and the coordinates of the center of gravity are required to control the three translational d.o.f.. They can be easily obtained from the moments m_{00} (corresponding to the area of the target), m_{01} and m_{10} of order 0 and 1 (which, divided by m_{00}, give us the center of gravity of $\mathcal{T}$):

$$\begin{cases} a & = & m_{00} \\ x_g & = & m_{10}/m_{00} \\ y_g & = & m_{01}/m_{00} \end{cases} \qquad (10)$$

But using $\mathbf{s} = (x_g, y_g, a)$ implies to compute an estimation of the depth at each iteration, which one would want to avoid.

The new set of features proposed in [14] is defined by $\mathbf{s} = (x_n, y_n, a_n)$ with $x_n = x_g a_n$, $y_n = y_g a_n$ and $a_n = Z^*\sqrt{\frac{a^*}{a}}$ where Z^* is the value of the depth at the desired pose. In that case, the interaction matrix resolves as $\mathbf{L_e} = -\mathbf{I}_3$.

The choice of the coefficient of normalization $Z^*\sqrt{a^*}$ in a_n was made to provide the constant term equal to -1 in the diagonal of the matrix. We can note that this coefficient is constant and that the current value of the depth Z does not appear anywhere and thus has not to be estimated. In this case, it is very easy to demonstrate the Lyapunov global asymptotic stability of the system [6]. With this method, both linearity of the system and features efficiency are achieved. The control law resolves in:

$$\begin{cases} v_x & = & -\lambda \, x_n \\ v_y & = & -\lambda \, y_n \\ v_z & = & -\lambda \, (a_n - a_n^*) \end{cases} \qquad (11)$$

The relation between the camera frame and the end-effector frame is coarsely approximated by the matrix $^r\mathbf{R}_c =$

Diag(1, -1, 1) corresponding to a rotation of π radian around the X-axis.

Thus the final control law is expressed as following:

$$\dot{\rho}_i = -\frac{\lambda}{\rho_i} \begin{pmatrix} b_x - a_{i_x} \\ b_y - a_{i_y} \\ b_z - a_{i_z} \end{pmatrix}^{\top} \begin{pmatrix} x_n \\ -y_n \\ -(a_n - a_n^*) \end{pmatrix} \quad (12)$$

for each wire with positive tension.

VI. EXPERIMENT AND RESULTS

Due to the nature of the cable-driven parallel mechanism, small errors in the orientation of the platform around each axis can occur. Nevertheless, the rotation matrix $^r\mathbf{R}_c$ was fixed to the values given in Section V, which is then only an approximation. The results will show that it has no significant effect on the success of the visual servoing.

Also, the precision in the position of the small aluminium foils glued to the wires is ± 0.01 meters, which can induce small errors on the positioning. The choice of a closed loop control scheme allows to counterbalance these errors which then does not have any significative impact on the success of the process.

Note that for the visual servoing implementation that we now present, we used the library ViSP[1] [12].

A video illustrating our results is provided as supplementary material.

A. Study case

Controlling only 3 d.o.f., the chosen features are $\mathbf{s} = (x_n, y_n, a_n)$ and their desired values $\mathbf{s}^* = (0, 0, 0.20)^2$. In a fixed reference frame, the target (center of gravity of the handle) coordinates are $(2.03, 1.22, 0.80)$. The estimated initial position of the camera is $(1.90, 1.15, 1.64)$ and its final position, after the visual servoing step, should be $(2.03, 1.22, 1.00)$.

Using a constant gain λ in (9), we encountered two major issues: at the beginning of the visual servoing, the first velocity commands were too high, and the target was sometimes lost because of the large displacement between two successive images ; at the end, the value of the velocity commands were too low to allow the actionning of the motor drums, so the cables were desynchronized and the visual servoing failed at the end. The library ViSP allows to express the gain λ as a function of the norm of the error in order to decrease the time to convergence and avoid the above drawbacks.

More precisely, introducing $\lambda_\infty = \lim_{\|\mathbf{e}\| \to \infty} \lambda(\|\mathbf{e}\|)$ and $\lambda_0 = \lim_{\|\mathbf{e}\| \to 0} \lambda(\|\mathbf{e}\|)$, the function

$$\lambda(\|\mathbf{e}\|) = ae^{-b\|\mathbf{e}\|} + c$$

is designed, where $a = \lambda_0 - \lambda_\infty$, $b = \lambda'(0)/a$ and $c = \lambda_\infty$, leading the velocity to have an almost linear decay. For the experiments presented here, we chose the following parameters: $a = 0.95, b = 10, c = 0.05$.

In Fig.4, the 3D trajectory of the camera attached to the end-effector and different projections are represented. The positioning before grasping is successful, despite of the uncertainties

[1] http://www.irisa.fr/lagadic/visp/visp.html
[2] All distances and coordinates are expressed in meter.

(a) 3D Trajectory
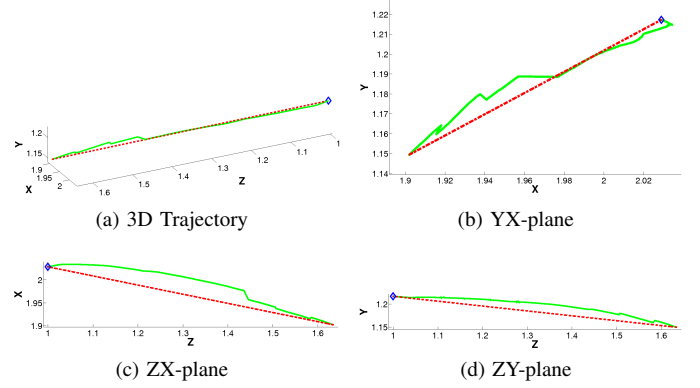

(b) YX-plane

(c) ZX-plane

(d) ZY-plane

Fig. 4: Trajectory (legend: green plain lines stands for the real trajectory, red dashed lines for the theoretical trajectory, blue square indicates the final coordinates)

on the locations of both the object and the camera, prior to the visual servoing process and after. As a value of 0.01 meter was selected as the shutoff parameter, we can infer that we have an accuracy better than one centimeter for each direction, which is of course a significative improvement, and is sufficient for the purpose of our work. It can be noticed that the manipulator does not follow an ideal trajectory, which should be a straight line. However, the Fig.4b and 4d show that the curvature is due to the manipulator reaching the desired state more quickly in some coordinates (X and Y) than the other (Z). That is for the most part due to the choice of a linear decay of the velocity and the selected parameters: the function reacts to the norm of the error which is very close to the norm of $(a_n - a_n^*)$. Thus, the gain mostly adapts itself relatively to the error in depth and reaches the (X, Y) coordinates faster.

(a) Error versus iteration number (legend: red dashed line stands for x_n, green crosses for y_n and blue diamonds for $(a_n - a_n^*)$)

(b) Velocity computed for each d.o.f. (legend: red dashed line stands for $\mathbf{v}_x$, green crosses for $\mathbf{v}_y$ and blue diamonds for $\mathbf{v}_z$)
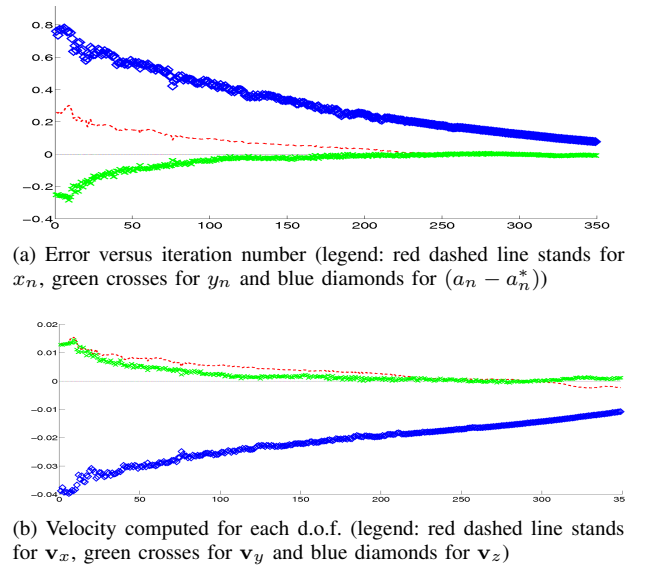
Fig. 5: Velocity and errors

In this case, the visual servoing process took 18 seconds and 349 iterations. The time for one iteration of the control loop is lower than 40 milliseconds, the remaining time consisting in

communication between the platform and the controller and recording of informations for further studies.

B. General context

In order to test both accuracy and repeatability of our scheme, we started the visual servoing part from nine different locations: the eight corners of a cube and a location in which the robot had only to move along the Z-axis. Given the position of the target in the fixed reference frame, those eight corners were chosen such as the camera was initially positioned at $(\pm 0.06, \pm 0.05, 0.50)$ and $(\pm 0.06, \pm 0.05, 0.60)$ meter from the target. The target was initially positioned in $(1.53, 1.00, 1.01)$, so the final position of the camera should be $(1.53, 1.00, 1.21)$ at the end of the third step, because of our choice of s^*.

n^o	Initial position	Final position	Iterations	Time
1	(1.54,1.00,1.60)	(1.65,0.94,1.18)	207	18s
2	(1.48,0.96,1.63)	(1.62,0.97,1.16)	200	20s
3	(1.48,1.06,1.63)	(1.59,1.02,1.21)	75	19s
4	(1.61,0.94,1.60)	(1.56,1.06,1.22)	224	17s
5	(1.60,1.06,1.63)	(1.58,0.98,1.20)	207	20s
6	(1.48,0.96,1.53)	(1.54,0.99,1.22)	91	15s
7	(1.48,1.06,1.50)	(1.60,1.00,1.16)	201	16s
8	(1.60,0.97,1.51)	(1.55,0.94,1.19)	206	17s
9	(1.60,1.06,1.53)	(1.57,0.98,1.21)	119	14s

TABLE II: Estimated initial and final positions (in meters), number of iterations and total duration of the visual servoing.

As can be seen in Table II, there are small positioning errors right at the beginning of the visual servoing, due to the system inaccuracies. As the robot does not know its exact location, final positions seem far from where they should be. Nevertheless, the end-effector reaches its desired location with an accuracy less than 0.01 meter in each and every case, which means that an error of 0.10 meter in the robot model has no effect on the vision-based control, showing the robustness of our scheme. That proves that, in order to obtain the accuracy requested to achieve the grasping task, visual servoing is a necessary step. For those nine cases, the L_1-error between the real trajectory and the expected one has its mean equal to 0.02 meters, and its maximum is 0.04 meters, which also indicates that our results have a better accuracy than the one exhibited by the robot alone. It can also be noticed that even if the number of iterations is very variable, the whole time of the process remains globally the same. A lower number of iterations generally happens when the target is lost. When it occurs, the manipulator goes on with the old velocity command, and when the object is eventually detected again, the frequency at which the commands are sent is restored.

VII. Conclusions

Aiming to use a CDPR in order to grasp and move objects, we added visual servoing capabilities to the existing `Marionet-Assist`. Our cable-driven parallel robot was initially designed to transfer people through an entire room, but the relatively low accuracy that is need to transfer people is not sufficient for grasping objects with roughly known location. Using simple image moments, we were able to control the manipulator and successfully grasp a specific object, moving it from a place to another. The experiments showed that using vision-based control increases the accuracy and allows the grasping task. They also show that our method is robust with respect to errors such as rotation and positioning errors induced by the displacement of the platform and uncertainties on model parameter values, such as the estimation of the length of the cable and the orientation of the platform.

Our objectives are now to try other configurations for the `Marionet-Assist`, some of them adding the control of the rotation around the Z-axis. Also, we rely on a good segmentation process prior to the visual servoing process, and there are yet no fully reliable methods, as we need an 100% rate of success. We aim to implement a semi-supervised method which will feed a learning process adapted to each singular situation and environment. Finally, we are working on a scheme allowing us to control which wires will be in tension, thus planning smoother trajectory and avoiding brutal configuration changes.

References

[1] J. Albus, R. Bostelman, and N. Dagalakis. The nist robocrane. *J. of Robotic Systems*, 10(5):709–724, july 1993.

[2] Manikandan Bakthavatchalam, François Chaumette, and Eric Marchand. Photometric moments: New promising candidates for visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'13*, pages 5521–5526, Karlsruhe, Allemagne, May 2013.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002.

[4] Marco Carricato and Jean-Pierre Merlet. Direct geometrico-static problem of under-constrained cable-driven parallel robots with three cables. In *ICRA*, pages 3011–3017, 2011.

[5] F. Chaumette. Image moments : a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4):713–723, Aug. 2004.

[6] François Chaumette and S. Hutchinson. Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, 2006.

[7] T. Dallej, M. Gouttefarde, N. Andreff, M. Michelin, and P. Martinet. Towards vision-based control of cable-driven parallel robots. In *IEEE International Conference Intelligent Robots and Systems, IROS'11*, volume on CD ROM, pages 2855–2860, 2011.

[8] M. Hiller et al. Design, analysis and realization of tendon-based parallel manipulators. *Mechanism and Machine Theory*, 40(4):429–445, april 2005.

[9] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12:651–670, Oct. 1996.

[10] S. Kawamura et al. Development of an ultrahigh speed robot falcon using wire drive system. In *IEEE Int. Conf. on Robotics and Automation*, pages 215–220, Nagoya, 25-27 may 1995.

[11] H. Kino et al. Robust pid control using adaptive compensation for completely restrained parallel -wire driven parallel robots: translational systems using the minimum number of wires under zero-gravity condition. *IEEE Trans. on Robotics*, 23(4):803–811, august 2007.

[12] E. Marchand, F. Spindler, and François Chaumette. ViSP for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4):40–52, 2005. Special issue on Software Packages for Vision-Based Control of Motion, P. Oh, D. Burschka (Eds.).

[13] S. Perreault and C.M. Gosselin. Cable-driven parallel mechanisms: application to a locomotion interface. *ASME J. of Mechanical Design*, 130(10):102301–1/8, october 2008.

[14] O. Tahri and F. Chaumette. Image moments : generic descriptors for decoupled image-based visual servo. In *IEEE International Conference on Robotics and Automation, ICRA'04*, volume 2, pages 1185–1190, 2004.