

Visual-based feedback control of Casting Manipulation

Adriano Fagiolini*, Hitoshi Arisumi† and Antonio Bicchi*

*Interdepartmental Research Center “E. Piaggio”
Faculty of Engineering - University of Pisa
Via Diotisalvi 2, 56126 Pisa, Italy
{a.fagiolini, bicchi}@ing.unipi.it

†ISRI/AIST-STIC/CNRS Joint Japanese-French Robotics Lab.
Intelligent Systems Research Institute, AIST
Umezono 1-1-1, Tsukuba, Ibaraki, 305-8568, Japan
h-arisumi@aist.go.jp

Abstract—In this paper, we present a method to control casting manipulation by means of real-time visual feedback. Casting manipulation is a technique to deploy a robotic end-effector at large distances from the robot’s base, by throwing the end-effector and controlling its ballistic flight using forces transmitted through a light tether connected to the end-effector itself. The tether cable can also be used to retrieve the end-effector and exert forces on the robot’s environment. Previous work has shown that casting manipulation is able to catch objects at a large distance, proving it viable for applications such as sample acquisition and return, rescue, etc. In previous experiments, the position of the target object was known exactly. In this paper, we present a first attempt at closing a real-time control loop on casting manipulation using visual feedback of moving targets. As accurate planning methods developed for off-line open-loop planning cannot be used in real-time, we develop a simplified model and control algorithm, whose effectiveness is demonstrated through experiments.

Index Terms—Casting manipulation, moving target, visual feedback, real-time planner.

I. INTRODUCTION

ONE of the basic features required in robotic manipulation is a large workspace. Conventional manipulators consist of only rigid links. Thus, to expand a robot’s workspace is not a task to be easily accomplished. One possible way is to extend the length of each link or to add extra links [1], [2], but this results in an increased weight of the robot and, in most cases, a decreased performance. Another means is to use mobile platforms [3]. Nevertheless, this approach is dependent on the land properties. As a matter of fact, the space that the robot can reach can be restricted by, e.g., depressions, humps, or obstacles.

For solving this problem, the casting manipulation was introduced [4]. This robotic manipulation consists of a technique to deploy a robotic end-effector at large distances from the robot’s base, by throwing the end-effector and controlling its ballistic flight using forces transmitted through a light tether connected to the end-effector itself. The tether cable can also be used to retrieve the end-effector and exert forces on the robot’s environment. Previous work has shown that casting manipulation is able to catch objects at a large distance, proving it viable for applications such as sample acquisition and return, rescue,

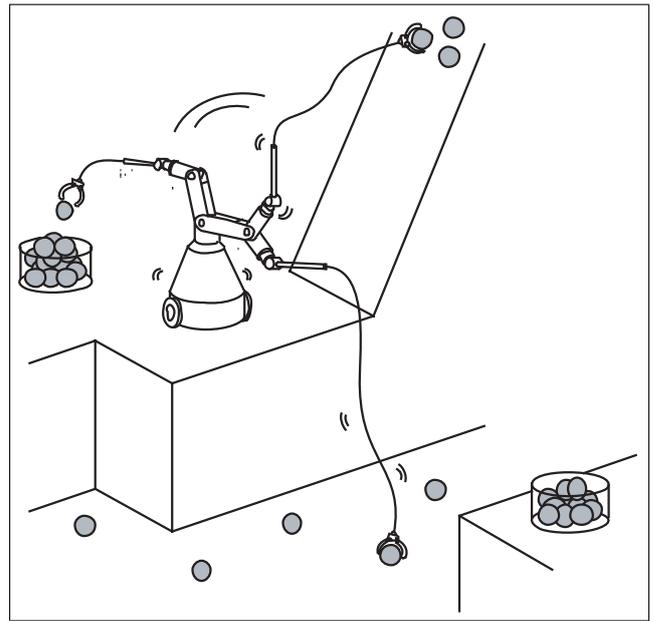


Fig. 1. A casting manipulator used for a pick-and-place task. The robotic end-effector is deployed at large distance from the robot’s base, by throwing it and controlling its ballistic flight. In this way, even if the robot has a relative simple mechanical structure and control, it may be used to retrieve far objects.

etc. Figure 1 shows a casting manipulator used for a pick-and-place task.

In this paper, we present a first attempt at closing a real-time control loop on casting manipulation using visual feedback of moving targets. As accurate planning methods [10], [11] developed for off-line open-loop planning cannot be used in real-time, we develop a simplified model and control algorithm, whose effectiveness is demonstrated through experiments. The paper is organized as follows. The analytical model of a two-link casting manipulator is obtained in Section II. Section III describes the vision system as far as concerned with the algorithms used to extract the target position and to calibrate the camera. Successive Section IV presents the control algorithm, and Section V describes the experimental setup and then shows the results of the experiments. Finally, the closing Section VI summarizes the work achievements and discusses about possible future developments.

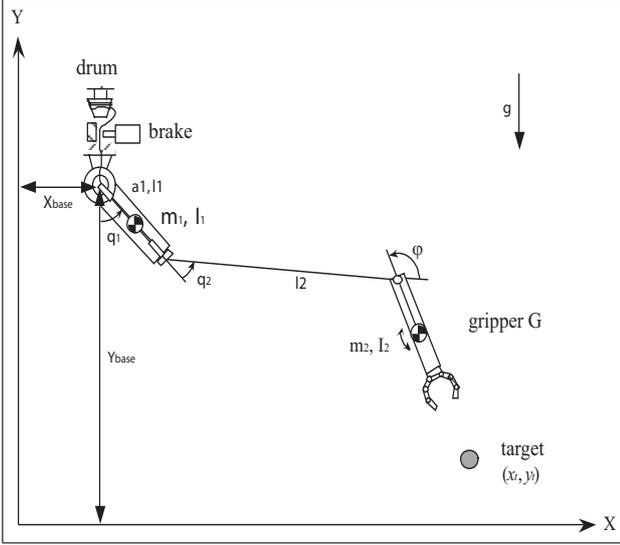


Fig. 2. Analytical model of a two-link planar casting manipulator. The first link is rigid and actuated, whereas the second is flexible (like a string) and presents no actuation. A gripper is used to catch the target object and a braking mechanism is employed to control its ballistic flight.

II. MODELLING

Consider the casting manipulator depicted in figure 2 which consists of a two-link planar manipulator. The first link L_1 is rigid, whereas the second one, L_2 , is composed of a flexible material (like a string) and a gripper mounted at the end of it. Both joints are rotational, but the first one is actuated by the torque τ , whilst no actuation is present at the second one. The robot is then provided with a braking mechanism, preventing the second link to unroll. As far as concerned with the nomenclature, let I_1 and m_1 be the inertia and the mass of link L_1 , respectively. Let also a_1 be the length of this link and l_1 be the distance of its center of mass from the first joint. We take no account of the weight of the second link (string). In such a case, I_2 and m_2 coincide with the gripper inertia and mass, respectively. Furthermore, the distance l_2 of the center of mass of link L_2 from the second joint equals its length a_2 . Besides, let q_1 and q_2 be the joint variables, and g the gravity acceleration. Finally, define (x_0, y_0) as the position of the gripper center of mass and φ its orientation with respect to the \hat{x} axis.

The phases of operation of this casting manipulator are depicted in figure 3. During the first phase, link L_1 is swung so that the gripper attains an adequate amount of energy to reach the target object. Afterwards, the gripper is thrown at a suitable time towards the target by releasing the string. By stopping multiple times the string unrolling with the braking mechanism, the posture (position and orientation) of the gripper is adjusted during its flight so as to catch the target object. Once it has been caught, the string is reeled up and the target object is retrieved.

Focusing on the swinging phase, it is firstly necessary to obtain a model of the system. This task is complicated by the presence of the flexible component. However, if the robot is controlled in such a way that link L_2 never becomes loose, it behaves as a manipulator with only rigid

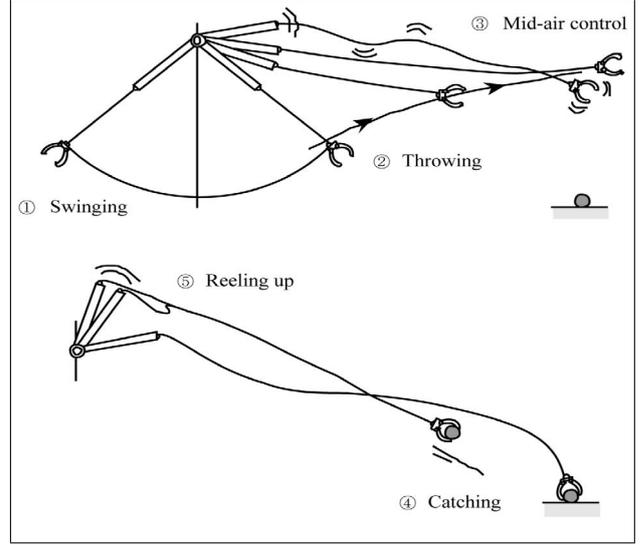


Fig. 3. Operation phases of casting manipulation

links. Under this assumption, its dynamics is describes by¹

$$\begin{pmatrix} \tau \\ 0 \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} + \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} + \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

$$b_{11} = m_1 l_1^2 + m_2 (a_1^2 + l_2^2 + 2 a_1 l_2 c_2) + I_1 + I_2,$$

$$b_{12} = I_2 + m_2 (l_2^2 + a_1 l_2 c_2),$$

$$b_{21} = b_{12},$$

$$b_{22} = I_2 + m_2 l_2^2,$$

$$c_{11} = -m_2 a_1 l_2 s_2 \dot{q}_2,$$

$$c_{12} = -m_2 a_1 l_2 s_2 (\dot{q}_1 + \dot{q}_2),$$

$$c_{21} = m_2 a_1 l_2 s_2 \dot{q}_1,$$

$$c_{22} = 0,$$

$$g_1 = m_1 g l_1 s_1 + m_2 g (a_1 s_1 + l_2 s_{12}),$$

$$g_2 = m_2 g l_2 s_{12}.$$

As explained in [4], [5], the first link is swung so that q_2 follows a series of cycloidal trajectories with a suitable amplitude. For this purpose, a computed torque control is realized as follows. The system dynamics may be written as

$$\begin{cases} \ddot{q}_1 = \frac{b_{22}}{\Delta} (\tau - c_{11} \dot{q}_1 - c_{12} \dot{q}_2 - g_1) + \frac{b_{12}}{\Delta} (c_{21} \dot{q}_1 + c_{22} \dot{q}_2 + g_2), \\ \ddot{q}_2 = -\frac{b_{21}}{\Delta} (\tau - c_{11} \dot{q}_1 - c_{12} \dot{q}_2 - g_1) - \frac{b_{11}}{\Delta} (c_{21} \dot{q}_1 + c_{22} \dot{q}_2 + g_2), \end{cases}$$

having defined $\Delta = b_{11} b_{22} - b_{12}^2$. Then, it is possible to take advantage of control τ to linearize the second joint dynamics by choosing

$$\tau = c_{11} \dot{q}_1 + c_{12} \dot{q}_2 + g_1 - \frac{b_{11}}{b_{21}} (c_{21} \dot{q}_1 + c_{22} \dot{q}_2 + g_2) - \frac{\Delta}{b_{21}} u$$

¹The following abbreviations are used: $c_i = \cos(q_i)$, $s_i = \sin(q_i)$, $c_{ij} = \cos(q_i + q_j)$ and $s_{ij} = \sin(q_i + q_j)$.

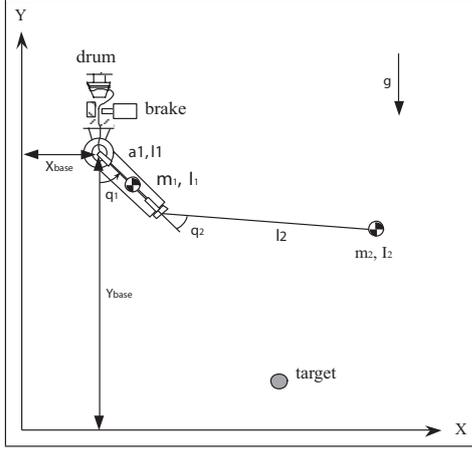


Fig. 4. Simplified analytical model of a two-link planar casting manipulator. The gripper is replaced by a mass point since we are not considering the orientation of the robot's end-effector.

in which the new input control $u \in \mathbb{R}$ is introduced. The system dynamics with respect to u has the form

$$\begin{cases} \ddot{q}_1 = f_1(q_1, q_2, u), \\ \ddot{q}_2 = u. \end{cases}$$

where f_1 is an opportune nonlinear function. Let $q_{2_d}(t)$ be the desired trajectory of the second joint. By selecting the control u as

$$u = \ddot{q}_{2_d} + K_v(\dot{q}_{2_d} - \dot{q}_2) + K_p(q_{2_d} - q_2),$$

the error $e_2 = q_{2_d} - q_2$ asymptotically converges to zero.

During this initial swinging phase, the gripper state is related to the instantaneous robot configuration by the following relation:

$$\begin{cases} x_0 = x_{base} + a_1 s_1 + a_2 s_{12}, \\ y_0 = y_{base} - a_1 c_1 - a_2 c_{12}, \\ \varphi = q_1 + q_2 + \frac{\pi}{2}, \\ \dot{x}_0 = a_1 c_1 \dot{q}_1 + a_2 c_{12} (\dot{q}_1 + \dot{q}_2), \\ \dot{y}_0 = a_1 s_1 \dot{q}_1 + a_2 s_{12} (\dot{q}_1 + \dot{q}_2), \\ \dot{\varphi} = \dot{q}_1 + \dot{q}_2. \end{cases} \quad (1)$$

Therefore, the gripper state at the throwing time is obtained by (1) evaluating the joint variables, q_1 and q_2 , and their first temporal derivatives at the same time. By doing so, we assess also the gripper state at the beginning of its flight.

According to our intention to focus in the first place only on reaching a target object in motion, the robot's end-tip orientation φ may not be considered. From now on, refer to figure 4 in which the gripper is replaced with a mass point. Similarly to what was obtained above, the mass point state at the throwing time is related to the manipulator state and is given by the relation

$$\begin{cases} x_0 = x_{base} + a_1 s_1 + a_2 s_{12}, \\ y_0 = y_{base} - a_1 c_1 - a_2 c_{12}, \\ \dot{x}_0 = a_1 c_1 \dot{q}_1 + a_2 c_{12} (\dot{q}_1 + \dot{q}_2), \\ \dot{y}_0 = a_1 s_1 \dot{q}_1 + a_2 s_{12} (\dot{q}_1 + \dot{q}_2). \end{cases}$$

Once the mass point has attained enough energy, it is thrown towards the target object by releasing the string. Without any control action, the mass point goes through an arc of parabola subjected only to the gravity acceleration g . Let t be the time elapsed from the throwing. The instantaneous position of the mass point is then

$$\begin{cases} x(t) = x_0 + \dot{x}_0 t, \\ y(t) = y_0 + \dot{y}_0 t - \frac{1}{2} g t^2, \\ \dot{x}(t) = \dot{x}_0, \\ \dot{y}(t) = \dot{y}_0 - g t, \end{cases}$$

and its landing position is

$$x_{land-max} = x_0 + \frac{\dot{x}_0}{2g} \left(\dot{y}_0 + \sqrt{\dot{y}_0^2 + 2g y_0} \right). \quad (2)$$

This point also represents the furthest point that can be reached. Thus, if the target moves beyond $x_{land-max}$ it cannot be reached, unless it subsequently comes closer to the manipulator. We neglect disturbances such as air resistance and friction caused by string motion.

III. VISION SYSTEM

When dealing with a moving target, it is necessary to measure its instantaneous position $\mathbf{p}_{target} = (x_t, y_t)^T$. For this reason, a camera-based vision system is realized. Refer to section V for a portrayal of the underlying hardware and software setup. For the time being, focus on the algorithms for position detection and camera calibration.

A. Detection of the target position

To extract the information about the target position from the camera data, various techniques were analyzed. Among them the feature tracking and the pattern matching were considered. Despite of their wide usage in several applications, it was decided to use a contour-detection-based algorithm because of its simple realization and robustness². Such an algorithm looks out the frame for markers with specific characteristics, like a certain number of circles in it, and then returns the position (u, v) of a specific point, e.g., the center of the biggest circle.

B. Camera calibration

The information thus far extracted by the camera data actually represents the target position mapped onto the image plane. Hence, it is now essential to calculate the real-world coordinate (x_t, y_t) by conversion.

Consider a set of points spread over the space visible from the camera. Let $\mathbf{p}_i^{(image)} = (u_i, v_i, 1)^T$ be the position of the i -th point, in homogenous coordinate, on the image plane detected by the algorithm described above, and $\mathbf{p}_i^{(real)} = (x_i, y_i, 1)^T$ the corresponding one in the real world. The coordinate conversion is generally

²This algorithm has not been published yet and it represents part of the work of Eng. Antonio Danesi, Eng. Daniele Fontanelli, and Eng. Vincenzo Scordio.

accomplished by using a vector function (homography map) such that

$$\mathbf{p}_i^{(real)} = \mathcal{H}(\mathbf{p}_i^{(image)}).$$

To choose and identify the function \mathcal{H} is in general a complex task. Nevertheless, if the motion of the object to be tracked is limited to a plane, the conversion may be performed with an adequate precision assuming the existence of a linear relation between plane coordinate and real-world one, that is using a homography map which consists of a matrix. Therefore, the conversion from image to real coordinate is obtained by using the formula

$$\mathbf{p}_i^{(real)} = \mathcal{H} \mathbf{p}_i^{(image)}. \quad (3)$$

It can be shown that \mathcal{H} has the form

$$\mathcal{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix},$$

and it is completely identified as soon as the eight parameters h_{ij} are somehow evaluated. This is exactly the purpose of the camera calibration procedure. Equation (3) may be explicitly written as

$$\begin{cases} x_i = h_{11} u_i + h_{12} v_i + h_{13}, \\ y_i = h_{21} u_i + h_{22} v_i + h_{23}, \\ 1 = h_{31} u_i + h_{32} v_i + 1. \end{cases} \quad (4)$$

In practice, it is required to assess the position (u_i, v_i) on the image plane, via the algorithm described in the section above, of only four points. It is clear that equations (4) hold for each of them. In such a condition, one can put them all together and get the system

$$\begin{pmatrix} x_1 \\ y_1 \\ 0 \\ x_2 \\ y_2 \\ 0 \\ x_3 \\ y_3 \\ 0 \\ x_4 \\ y_4 \\ 0 \end{pmatrix} = \begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_1 & v_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_2 & v_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_3 & v_3 \\ u_4 & v_4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_4 & v_4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_4 & v_4 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix}$$

which may be written in a more compact form as

$$\mathbf{p} = \mathcal{M} \mathbf{h} \quad (5)$$

where matrix \mathcal{M} is not squared and, hence, has no inverse. Therefore, solving (5) by the least-squares method, it is possible to evaluate the unknown parameters of the homogenous map with the formula:

$$\mathbf{h} = \mathcal{M}^\dagger \mathbf{p}. \quad (6)$$

To sum up, the calibration of the camera consists of the following steps:

- ▷ Set $m \geq 4$ markers onto the plane where the mass point moves on;

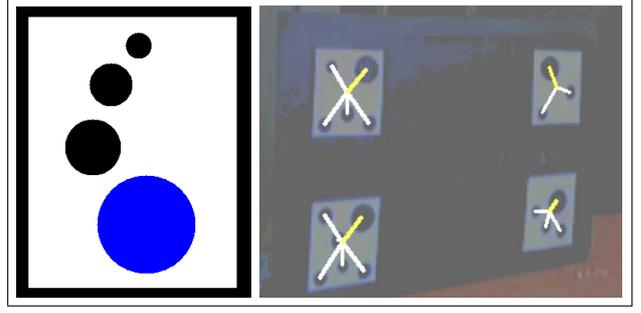


Fig. 5. The image on the left represents one of the markers used both for object identification and for camera calibration. The picture on the right is a snap-shot of the calibration process which shows that the vision system is able to recognize all of the markers settled.

- ▷ Place the camera so that it can see at least 4 markers;
- ▷ Identify the visible markers and detect their positions;
- ▷ Use equation (6) to evaluate the parameters \mathbf{h} .

In figure 5 is reported a snap-shot of the calibration process, showing that the vision system is able to recognize all of the markers settled. Once the calibration is performed, the camera must not be moved (unless it is calibrated again) and the target position is obtained as follows:

- ▷ Detect the target position (u, v) on the image plane;
- ▷ Convert this position to real-world coordinate with the relation

$$\begin{cases} x_t = h_{11} u + h_{12} v + h_{13}, \\ y_t = h_{21} u + h_{22} v + h_{23}, \end{cases}$$

using the estimated value of \mathcal{H} .

IV. VISUAL FEEDBACK AND REAL-TIME REPLANNING

In [10], [11] we proposed the method of multiple braking control as a technique to catch a still object by way of casting manipulation. The idea to control the posture of the gripper while flying, by stopping multiple times the last link unrolling, could be still applied. Nonetheless, its implementation has a drawback. In fact, the optimizing algorithms used to calculate the mid-air control parameters have a too long computation time and, hence, may not be used for a real-time planning which relies on the current target and gripper position. Thus, we focus for the time being only on the issue of reaching the object and we propose for that aim an alternative approach.

To start with, consider a still target and refer to figure 6. The target object is reached in the following way. In the first place, link L_1 is swung so that the mass point attains enough energy to reach a point further than the initial target position, x_t . At a suitable time, when passing by point A , the mass point is thrown towards the object. This is achieved by stopping the motion of link L_1 and releasing the string which begins to unroll. If no other action would be undertaken, the mass point would go through an arc of parabola, subjected only to the gravity acceleration g , and land in position (2). On the contrary, the mass point is left in free-flight only at the beginning. When it passes from point B , the string length l equals the distance

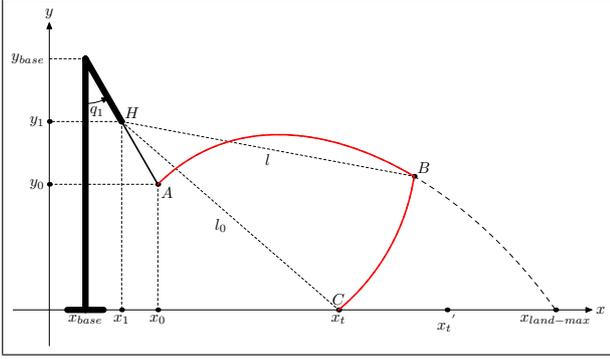


Fig. 6. Reaching a still object by a single braking. The mass point is thrown towards the target object once it has attained enough energy to reach it. When the length of the string equals the distance between the rigid link end and the target, the string is braked and the mass point is constrained to go through an arc of circumference which leads it directly onto the target.

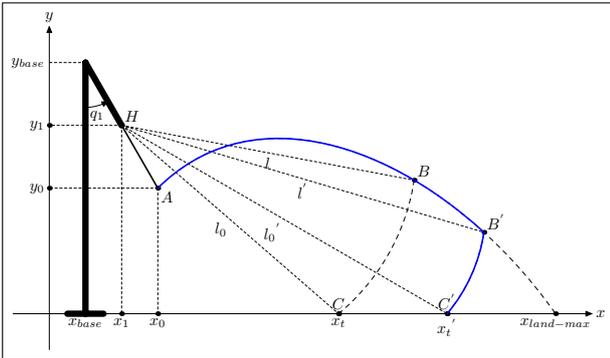


Fig. 7. In case of a receding target, the braking instant is delayed.

l_0 between point H and the instantaneous target position $(x_t, 0)^T$. From that point on, the string unrolling is stopped by using the braking mechanism. Neglecting the string elasticity, the mass point is constrained to go through an arc of circumference, which leads it directly onto the target.

Consider now a target randomly moving within position 0 and $x_{land-max}$. Let x_t be its position at the throwing time and x'_t the current one, continuously measured by the vision system. Once the mass point is thrown, the control system waits for the time when the string length equals the distance between point H and the current target position $(x'_t, 0)^T$. At that time, the string unrolling is stopped by braking and the mass point is constrained to go through an arc of circumference and land onto the new target position. As a result, in case of a target receding from the robot (figure 7), the braking instant is delayed up to the time when the mass point passes by point B' . It is evident from the figure that this procedure fails if the target moves further than $x_{land-max}$. On the contrary, if the target comes nearer to the robot (say to position x''_t), the start of the braking is anticipated (figure 8) and the task is though successfully accomplished.

This control policy is clearly less general than previous ones, seeing that it does not take into account of the gripper orientation and that it assumes the gripper stops moving once the braking is started. However, its low and constant

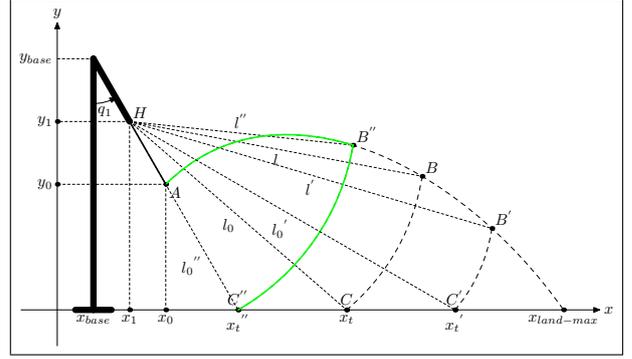


Fig. 8. In case of an approaching target, the braking instant is anticipated.

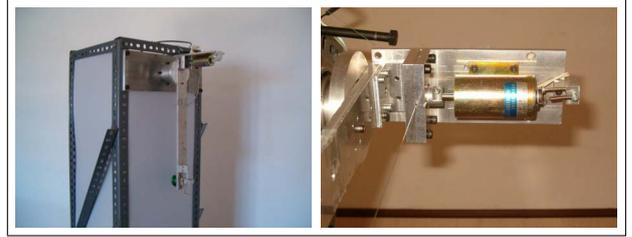


Fig. 9. The casting manipulator and the braking mechanism.

computational cost proves it viable to be used to replan the control in real-time.

V. EXPERIMENTS

To check the effectiveness of the proposed method, a casting manipulator, a camera-based vision system, and a control system were realized.

Examine firstly the robot realized. It consists of a two-link casting manipulator with a rigid link and a flexible one. Both joints are rotational, but the first one is actuated by a direct-drive motor whilst the second one presents no actuation. The analytical model of the manipulator has been determined in section II. Its geometrical and inertial parameters are reported in table I. An optical encoder with 81000 pulses per revolution was used to measure the first joint variable q_1 . As for the second joint variable q_2 , a potentiometer was used. Because of its noisiness, it was necessary to make use of a Butterworth filter in order to obtain a better measurement of q_2 . The robot has also a braking mechanism used to stop the unrolling of the second link. A ball was used to implement the mass point. Figure 9 shows the robot and a detail of the braking mechanism.

Consider now the vision and control systems, starting with the hardware description. The computer used was a Pentium[®] IV with clock frequency of 3.0 GHz. A US Digital PCI4ES card was used to read the encoder data and a National Instrument PCI6024E card was used for the potentiometer. For the vision, a USB Logitech[®] Orbit camera with frame-rate of 16 fps was used. From a software point of view, the underlying operating system was Microsoft Windows[®] XP and the algorithms, described in section III, for the extraction of the target position and the camera calibration, were implemented taking advantage of

TABLE I
SPECIFICATION OF THE CASTING MANIPULATOR REALIZED.

Parameter	Value	Unit
x_{base}	0.000	m
y_{base}	1.695	m
a_1	0.342	m
m_1	1.1105	kg
I_1	0.0216	$kg\,m^2$
a_2	0.495	m
m_2	0.084	kg
I_2	1.3440E-005	$kg\,m^2$

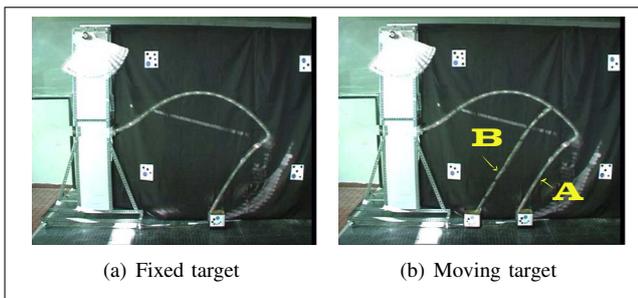


Fig. 10. Experimental ball's trajectory. The last part of the trajectory differs from an arc of circumference owing to the small but not negligible string elasticity which had not been considered in the model.

the OpenCV[®] library. All the source code was written in C++. Besides, the control of the direct-drive motor was scheduled with a cycle-time of 0.5 msec. The vision system is easy to set up, hard to fail in tracking the target, and shows high precision once calibrated. Besides, the calibration procedure is very quick and, hence, could be performed during the experiment if the camera were moved. The only observed demerit was the long sampling time (about 62 msec to obtain a new data position), which restricted the target velocity. However, this problem was due to the low cost camera used and it was not a limitation of the algorithm. To solve it, one could take advantage of a high speed vision system such as that described in [13], which can detect the position of a target every 1 msec.

Finally, two kinds of experiment were performed. The former ones were aimed at verifying the system performance in case of a stopped target, whose position was measured before throwing the ball. Figure 10(a) shows the experimental ball's trajectory, which apparently differs from what was expected by theory. This discrepancy was due to the string elasticity whose effect, not considered in the model, was enforced by the abrupt braking of the tether. That caused an impulsive tension in it which was then transmitted to the robot's end-effector.

The latter experiments, checked the effectiveness of the proposed method in case of a target approaching the robot's base. During such experiments, the target object position was firstly detected and then the ball was thrown. During the ball flight, the target object instantaneously moves to a newer position, before the string unrolling is stopped. Figure 10(b) shows two overlaid trajectories: trajectory *A* is obtained with a fixed target and trajectory *B* is obtained with a moving target. It is evident that an approaching target is reached by anticipating the braking start.

VI. CONCLUSION

In this paper, we started dealing with the issue of catching a moving object and presented a first attempt at closing a real-time control loop on casting manipulation. As accurate planning methods developed for off-line open-loop planning could not be used in real-time for their long computation time, we focused only on reaching the moving target and proposed for that a simplified model. A camera-based vision system was realized to detect the instantaneous position of the target object. By using the laboratory two-link casting manipulator, the effectiveness of the proposed control method was eventually demonstrated.

Future developments of this work will concern the control of the orientation of a gripper so that to be able also to grasp the moving target.

ACKNOWLEDGMENTS

Authors would like to acknowledge Giovanni Tonietti for his useful hints and assistance throughout this work, Antonio Danesi for his assistance in the realization of the vision system, Fabio Vivaldi for building up the experimental setup, and Riccardo Schiavi for his help in realizing the experiments.

REFERENCES

- [1] H. Mochiyama, E. Shimemura and H. Kobayashi, "Shape correspondence between a spatial curve and a manipulator with hyper degrees of freedom", *International Conference on Intelligent Robots and Systems*, 1998.
- [2] N. Takahashi et al. "Simulated and experimental results of dual resolution sensor based planning for hyper redundant manipulators", *International Conference on Intelligent Robots and Systems*, 1993.
- [3] O. Khatib, K. Yokoi, K. Chang, D. Ruspin, R. Holmberg and A. Casal, "Vehicle/Arm coordination and multiple mobile manipulator decentralized cooperation", *International Conference on Intelligent Robots and Systems*, 1996.
- [4] Hitoshi Arisumi, Tetsuo Kotoku, Kiyoshi Komoriya, "A study of Casting Manipulation (Swing Motion Control and Planning of Throwing Motion)", *International Conference on Intelligent Robots and Systems*, 1997.
- [5] Hitoshi Arisumi, Tetsuo Kotoku, Kiyoshi Komoriya, "Swing Motion Control of Casting Manipulation (Experiment of Swing Motion Control)", *International Conference on Robotics and Automation*, 1998.
- [6] Hitoshi Arisumi, Tetsuo Kotoku, Kiyoshi Komoriya, "Study on Casting Manipulation (Experiment of Swing Control and Throwing)", *International Conference on Intelligent Robots and Systems*, 1998.
- [7] Hitoshi Arisumi, Tetsuo Kotoku, Kiyoshi Komoriya, "Swing Motion Control of Casting Manipulation", *IEEE*, 1999.
- [8] Hitoshi Arisumi, Kiyoshi Komoriya, "Posture Control of Casting Manipulation", *International Conference on Robotics and Automation*, 1999.
- [9] Hitoshi Arisumi, Kiyoshi Komoriya, "Study on Casting Manipulation (Midair control of gripper by impulsive force)", *International Conference on Intelligent Robots and Systems*, 1999.
- [10] Hitoshi Arisumi, Kazuhito Yokoi, Kiyoshi Komoriya, "Casting Manipulation (Braking Control for Catching Motion)", *International Conference on Robotics and Automation*, 2000.
- [11] Hitoshi Arisumi, Kiyoshi Komoriya, "Catching Motion of Casting Manipulator", *International Conference on Intelligent Robots and Systems*, 2000.
- [12] K. Furuta, M. Yamakita, S. Kobayashi, "Swing up control of inverted pendulum", *Proceedings of the IEEE International Conference on Industrial Electronics, Control and Instrumentation*, 1991.
- [13] Akio Namiki, Takashi Komuro, Masatoshi Ishikawa, "High-speed sensory-motor fusion for robotic grasping, *Measurement Science and Technology*", Vol.13, pp.1767-1778 (2002).