Proceedings of the 1st Iberic Conference on Theoretical and Experimental Mechanics and Materials / 11th National Congress on Experimental Mechanics. Porto/Portugal 4-7 November 2018. Ed. J.F. Silva Gomes. INEGI/FEUP (2018); ISBN: 978-989-20-8771-9; pp. 1031-1042.

PAPER REF: 7451

A GANTRY ROBOT AUTOMATIC POSITIONING SYSTEM USING COMPUTATIONAL VISION

Nuno Beites[1], Manuel Dias[1], M.J.G.C.Mendes[1,2(*)], F. Carreira[1,3], F. Campos[1], J.M.F.Calado[1,3]

[1]Instituto Superior de Engenharia de Lisboa (ISEL), Instituto Politécnico de Lisboa, Lisboa, Portugal
[2]Centre for Marine Technology and Ocean Engineering (CENTEC), Lisbon, Portugal
[3]LAETA/IDMEC - Instituto de Engenharia Mecânica, Lisbon, Portugal
[(*)]*Email:* mmendes@dem.isel.ipl.pt

ABSTRACT

With the new industrial revolution (industry 4.0) under way, there is a need to automate/digitalize even the simplest processes to keep companies competitive, efficient and secure. This work proposes an automatic and robotic positioning system with position control through the image of video cameras. The objective is to propose this type of automatic system to companies aiming to improve robotic systems operations that are often controlled manually and with associated risks. The developed system was tested in simulation (using *Matlab®*) and in an experimental scale prototype.

Keywords: Automatic positioning system, gantry robot, computational vision, systems engineering, industry4.0.

INTRODUCTION

The fast development in the information and communication technology has caused a great impact in the fields of industrial automation and robotics [1]. This evolution changed industrial processes, bringing about more simplicity, adaptability and capacity to configure systems in the industrial environments [2]. Nowadays, markets demand rapid adaptations of processes, as well as improvements in productivity, systems safety and, essentially, people safety. Industrial automation and robotic systems allow task execution over 24 h/day, while maintaining performance and quality, and decreasing/eliminating worker’s contact with dangerous or unhealthy activities. Thus, industrial automation and robotics have become key concepts to improve both the manufacturing and the safety engineering. On other hand, the connectivity and communication between peripheral devices and controllers are key factors for the implementation of systems according to the concept of Industry 4.0, which has been reformulating how to control and manage industrial processes. Industry 4.0 combines aspects from the physical and virtual worlds, information technologies and cybernetic systems to create a new integrated worker-machine manufacturing environment [3].

Computer vision has been successfully applied in process automation, by adding sensory capabilities that help performing positioning tasks [1]. The enhanced perception about the environment and/or the objects brings efficiency to operations management since visual information can be used to change task execution and to choose the most appropriate action to each situation.

This work proposes an approach to control the position of a cartesian robot (gantry robot) with the aid of a computer vision system. We address the class of tasks in which the robot must process several objects placed in a nearly organized way on a horizontal plane below the robot. An example of this kind of industrial task and gantry robot is represented in Figure 1.

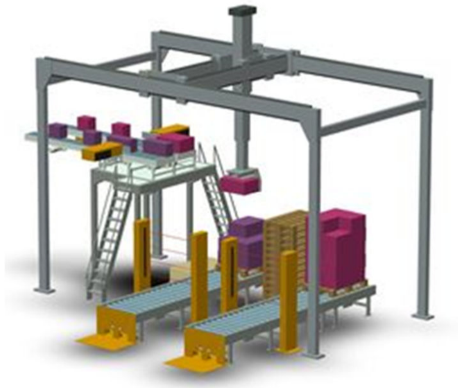

Fig. 1 - Palletizing gantry robot [3].

This paper is organized as follow: section 2 presents related work and robot applications that inspire our approach. Section 3 presents the proposed position control method to reach several circular targets. Section 4 presents the prototype and the experimental results. Finally, section 5 presents the conclusions and topics for future work.

RELATED WORK

In the last decades, developments in the robotics field have produced numerous visually guided robots to address different tasks. For example, Tong et al. [4] proposes the application of Image-Based Visual Servo (IBVS) control in a robot arm to organize surgical instruments.

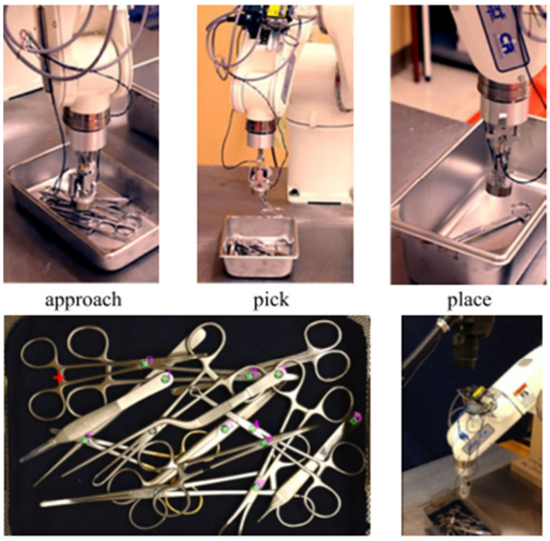

Fig. 2 - Pick and place of surgical instruments using IBVS control [4].

Fig. 3 - Prototype of a hole inspection system using PBVS control [5].

In that example, the robot identifies a set of surgical instruments, through computer vision, and organizes them in several containers, according to their characteristics (see Figure 2). Another approach is proposed in [5] where the authors use a Position-Based Visual Servo (PBVS) system to control the motion of a 6 degree of freedom robot arm. This robot is equipped with an endoscopic end effector to inspect the bore hole surface (see Figure 3). Regardless of the application, the goal of IBVS and PBVS is the minimization of an error, $e(t)$, obtained from an image captured at instant t, given by [6]:

$$e(t) = s - s', \tag{1}$$

where s is a vector of features derived from the image and s'represents the reference values for these variables, being s computed as follows:

$$s = k(m(t), a), \tag{2}$$

where $m(t)$ is a set of measurements performed on the image (e.g. Region of Interest coordinates or object centroids) and a is a set of parameters providing additional information about the vision system. When the target object is static s' is constant and only s changes, due to camera motion. According to this formulation, visual servoing schemes differ mainly in how s is designed.

In the control systems that apply IBVS s is defined as a set of points directly obtained from the image. In this approach, function k(m(t),a) allows transforming image measurements m(t) (in pixels), to feature measurements (in some other unit), using the intrinsic camera parameters a. Although simple, this approach must be carefully applied when fast movements are expected, because in this case there is the risk of capturing consecutive images with small overlap. Thus, the IBVS approach is considered more suitable in applications where it is necessary to overcome calibration errors or image noise, due to its lower sensitivity to these factors [6, 7, 8, 9, 10, 11, 12, 13].

In contrast to IBVS, PBVS uses the camera pose with respect to another reference coordinate frame to compute s. This requires that the camera pose is estimated, using the set of image measurements $m(t)$, the intrinsic camera parameters and the 3D model of the observed object. In these cases, $k(m(t), a)$ is often defined as the transformation that computes the camera pose, with a gathering the intrinsic camera parameters and the 3D object model [6, 7, 8, 9, 10, 12, 13].

METHODOLOGY

In this work it was addressed the task of sequentially driving a robot towards several objects, with the aid of visual information. The purpose is to develop an image-based robot control system that brings the end-effector onto each object, hereafter referred as *target*, to perform an operation of interest such as picking, painting, soldering, etc. A few assumptions regarding the system under study are made: first, a camera is attached to the robot's end effector; second, the robot has a Cartesian kinematic structure or, alternatively, can be controlled in the Cartesian space. Finally, we assume that the targets are coarsely placed along a line that is parallel to one of the world frame axes, a condition that is valid in many automated operations. For the purpose of this work, the targets are represented as circles placed at the ground plane and detected in the image by the application of the Hough transform [14]. Figure 4 illustrates the benchmark system under study, a gantry robot, and defines the coordinate frames attached to the world and to the camera.

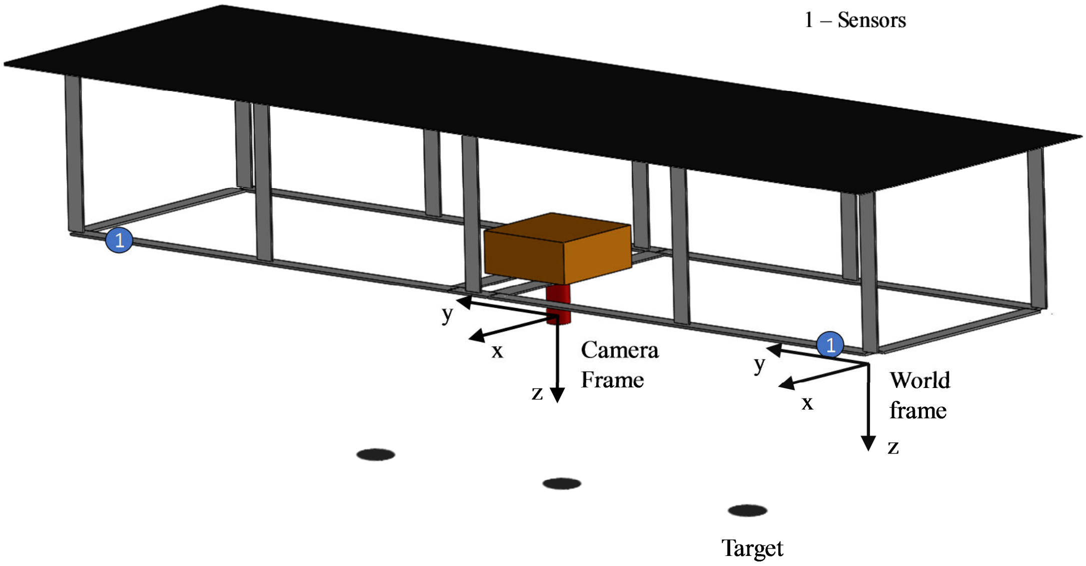

Fig. 1 - Gantry robot coordinate frames.

Control Logic

To accomplish the task in hand, three modes of operation were defined, namely, *Search*, *Approach* and *Processing*. In the *Search* mode, the camera is moved along the y axis at constant speed, until a new target is detected. In the *Approach* mode, the inputs to the gantry axes are produced by a suitable controller, designed to position the camera over the new target. Once the *Approach* step is completed, the system enters the *Processing* mode, in which the robot performs the operation on the target object. Since in this work no specific task was considered, the *Processing* step is simply simulated as a pause for a fixed amount of time, which would allow carrying out a specific task.

The algorithm that manages the execution of the task is depicted in the flowchart of the Figure 5. As described in the flowchart, at each iteration, a new image is captured and a target detector is applied, both in the *Search* and *Approach* modes. After updating the motor commands, the conditions for mode transition are evaluated as described next. Let (x_t^c, y_t^c) be the coordinates of the target in the image reference frame and w, h respectively be the width and height of the image, in pixels. The transition from the *Search* mode to the *Approach* mode occurs when a new target is detected, i.e., when a target is found in the green region of the image plane depicted in Figure 6(a). Notice that by limiting this region to values $y_t^c > h/2 + 2$ we are excluding targets close to or below the center of the image. This strategy is intended to exclude those targets which have already been processed from being searched for and processed repeatedly.

The *Approach* mode terminates when the camera position is within an acceptable distance from the target. In terms of image features, this condition was defined as:

$$\left|y_t^c - \frac{h}{2}\right| < 1 \wedge \left|x_t^c - \frac{w}{2}\right| < 1 \tag{3}$$

According to this expression, the transition from *Approach* to *Processing* occurs when the target is inside the blue area depicted in Figure 6(b). The control logic defined in Figure 5 commands the sequence of the *Approach*, *Search* and *Processing* operations and repeats this sequence until a predefined number of targets has been processed.

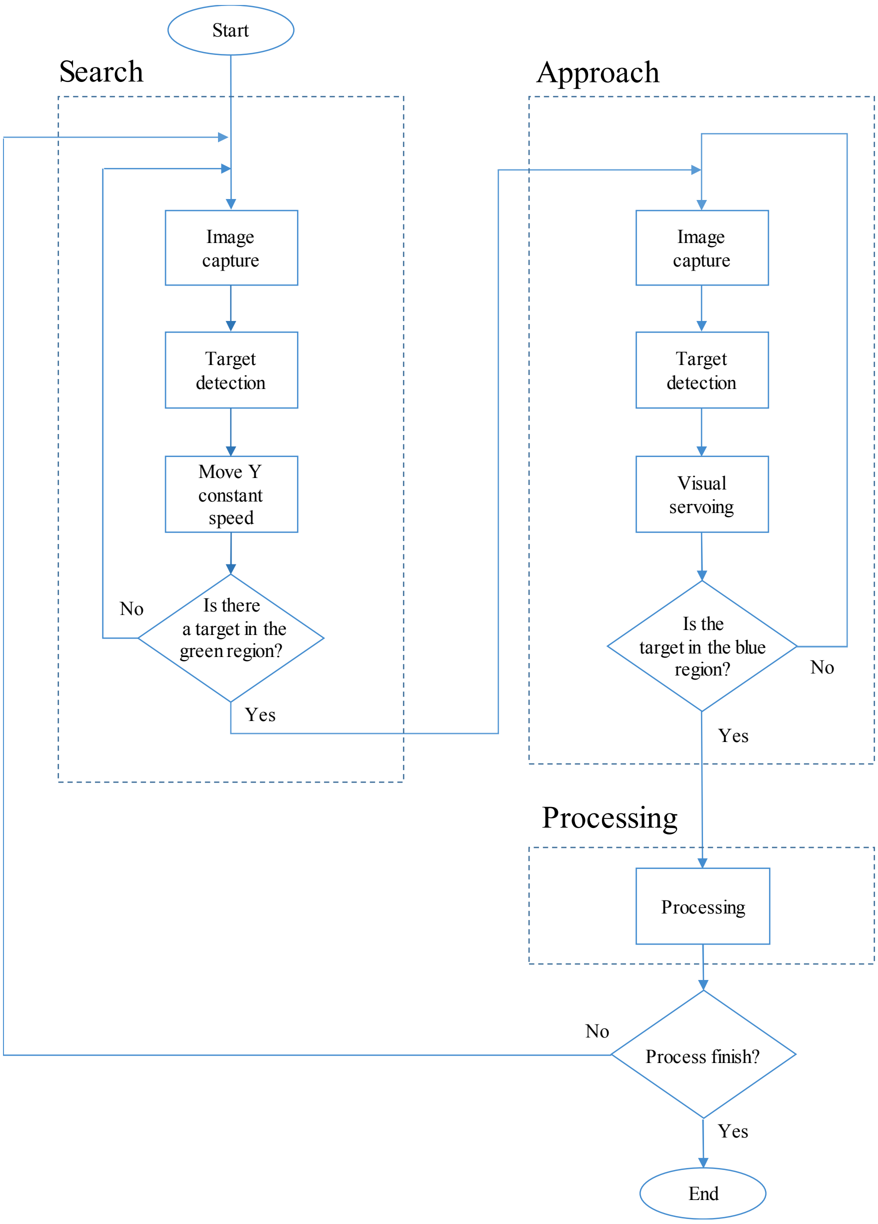

Fig. 5 - Flowchart of the vision algorithm.

Simulation Control Loop

In the *Approach* mode, a visual servoing control loop was applied, to place the camera over the target. It is assumed that the camera optical axis is vertical, implying that the camera is centered on the target if the target image is at the center of the image. Therefore, the control objective may be defined as the minimization of the distance between the target, with coordinates (x_t^c, y_t^c), and the image center, with coordinates $(w/2,\ h/2)$. To this end, a controller of the image-based servoing type [6] was implemented in simulation (see Figure 7). This is a controller characterized by the motor inputs being derived from an error directly measured in the image. Specifically, the error, E, and control, U, signals are two-valued vectors given by:

$$E = [w/2, h/2] - [x_t^c, y_t^c] \quad (4)$$

$$U = K_p \times E \quad (5)$$

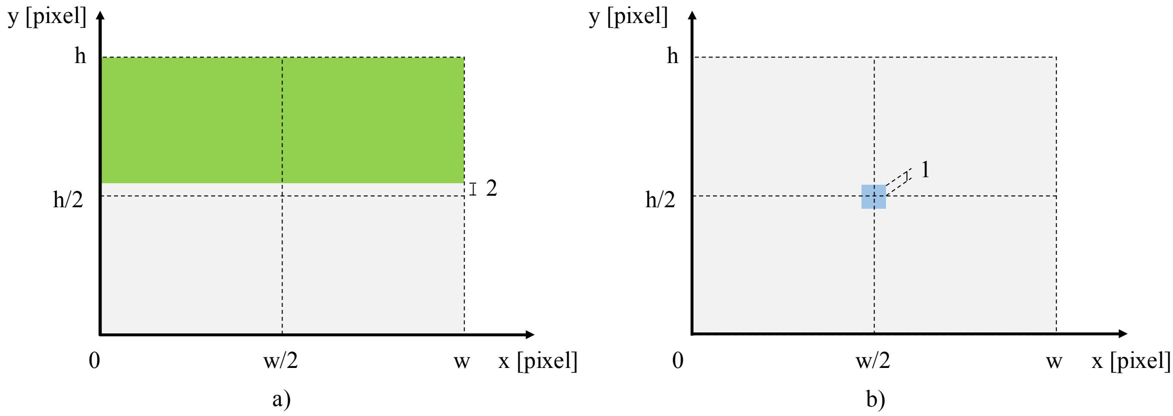

Fig. 6 - "Decision" regions in the image plane for: a) the *Search* mode, b) the *Approach* mode.

The last equation applies a proportional control law, with K_p being a diagonal matrix with equal gains applied to the x and y axis.

The simulation model was developed in Matlab/Simulink®, with the use of the Virtual Reality toolbox to interact with a 3D virtual environment. For our experiments, a 3D model of the system was developed (see Figure 3), featuring a configurable set of targets and a moving virtual camera.

Simulation results

To validate the control logic and the visual servoing loop, several simulation tests were performed. Here we describe a typical scenario where three targets were placed along the line $x = -0.5$ m, $z = 0.5$ m with y values of 1.5, 4 and 6.5 m, in the world reference frame. The initial position of the camera is at the origin of this frame. Also, in this simulation a proportional gain of $k_{px} = k_{py} = 6$ mm/s.pixel and an image acquisition framerate of 20 Hz were used.

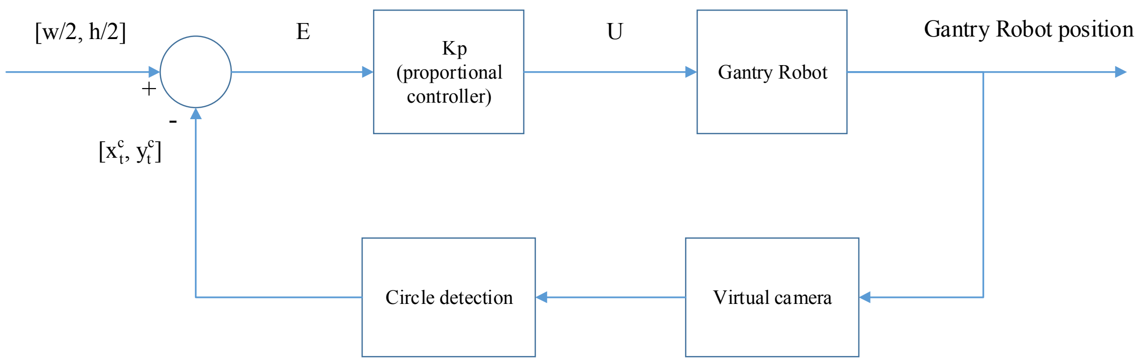

Fig. 7 - Block diagram of the image-based servo controller - in simulation.

Figures 8-(a), (c) shows the trajectory of the camera in the x and y coordinates and Figures 8-(b), (d) shows the speed in the two directions. These results validate the proposed control logic in commanding the sequence of operations to process the given targets. Concretely, Figure 8(c) shows that the algorithm succeeds in searching each target, moving the camera towards it and stopping for processing. In this Figure, labels 1, 2 and 3 identify each phase for the second target. Figure 8(a) demonstrates the adjustment produced in the x axis to eliminate the initial deviation in this direction. This adjustment is first applied at 2.5 s, when the first target is detected. The visual servoing towards this target brings the error to a low value, corresponding to the 1 pixel admissible error in the image plane. Nevertheless, the *Approach* mode applied to the second target brings about a further reduction in this error.

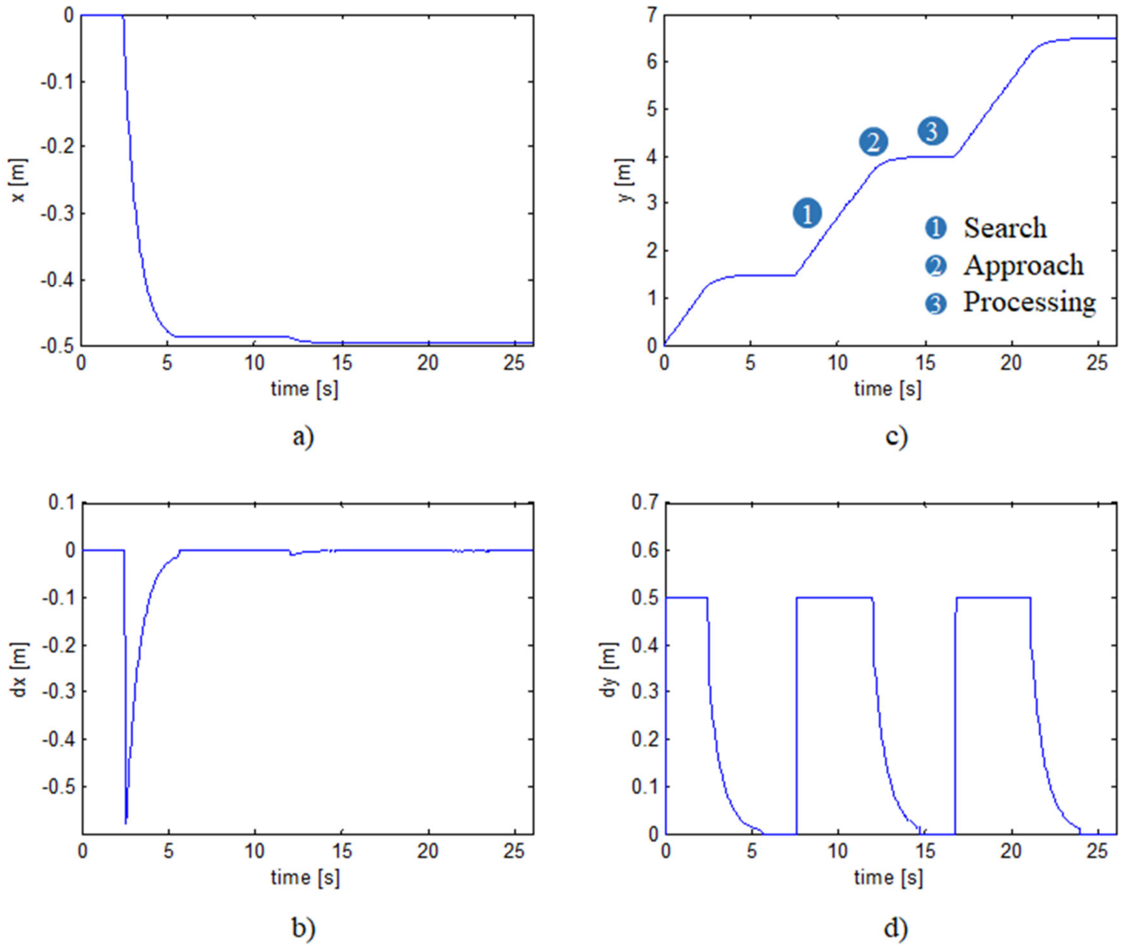

Fig. 8 - Camera *x* position (a) and speed (b), *y* position (c) and speed (d) - in simulation.

EXPERIMENTAL PROTOTYPE TESTS AND RESULTS

After algorithm validation using the Matlab® simulation environment, a gantry robot experimental prototype was constructed at scale 1:10 (see Figure 9). The structural elements (E) - aluminum bars, belts, screws, etc. and the electric/electronic devices (motors, control boards, etc.) were selected and purchased from the company Vslot-Europe [15]. As electric/electronic components the prototype has a 24 V power supply (17 A and 400 W) and, the motors used for the gantry axes actuation are stepper motors, Nema 17 (B) and Nema 23 (C), for the belt drives (*y* and *z* axes) and screw drive (*x*), respectively. These stepper motors have an accuracy of 200 steps per revolution, or 1,8 °/ step. For the control of stepper motors, the CNC xPRO Controller Stepper Driver V3 (D) was used. This controller architecture is based on the ATmega328p Arduino and allows to directly send instructions (rotation angles) to the stepper motors from the Arduino's Integrated Development Environment (IDE). The camera (A) used was an HP USB Webcam with 5MP and a maximum frame rate of 30.

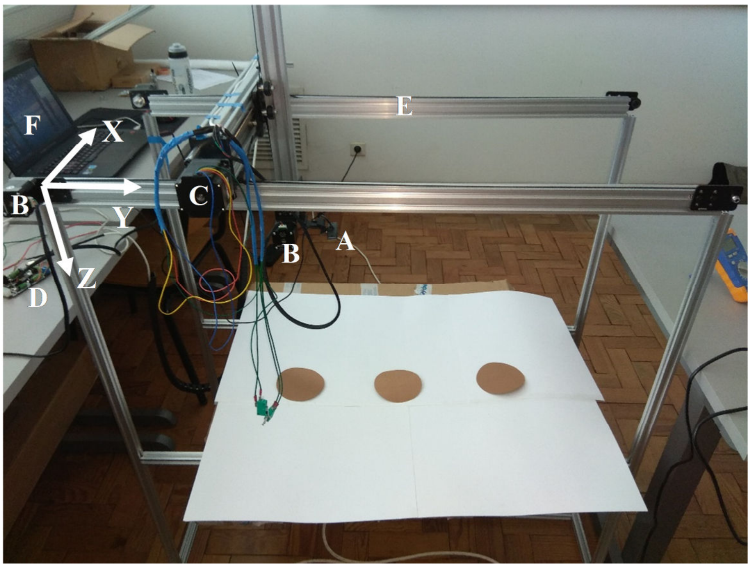

Fig. 9 - Gantry robot experimental prototype.

For this experimental prototype a control and supervision architecture was developed (as shown in Figure 10) to control the stepper motors, to process camera images and to allow the local and remote supervision (using a web browser). Figure 10 shows the interfaces between the different components in the architecture.

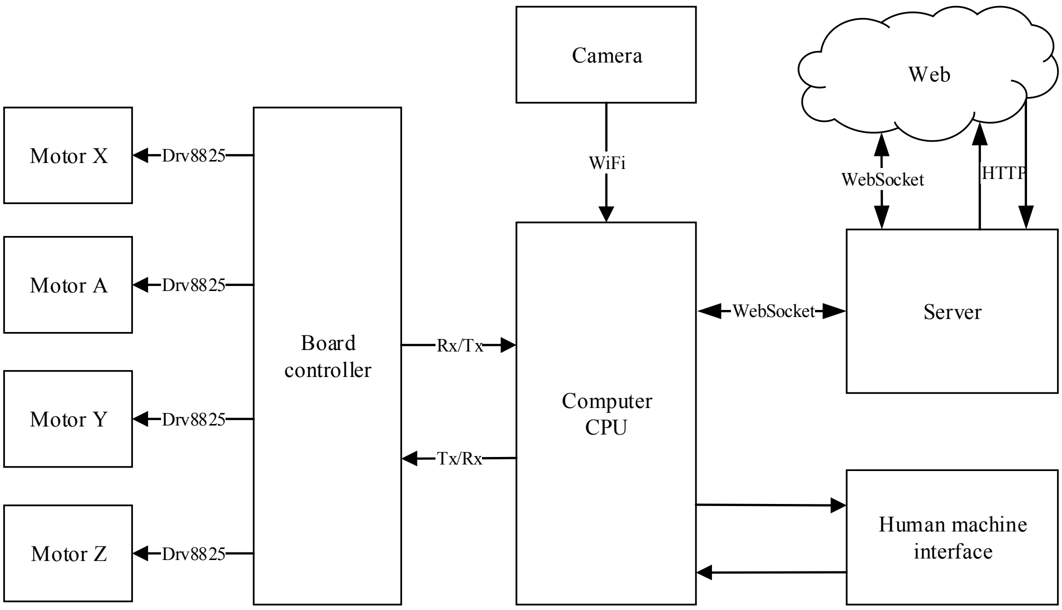

Fig. 10 - Control and supervision system architecture - experimental prototype.

Experimental Control Loop

After validation of the image-based position control loop (in simulation), a similar control loop (Figure 11) was used to control the experimental prototype.

In the experimental prototype it was not possible to implement the proportional action, since whenever an instruction is sent to the controller, the speed and the distance to be traveled (motion parameters) are set constant. The motor will act until the motion is complete, unless the following instruction opposes the movement being performed. On the other hand, in a proportional control scheme, a speed change in the same direction may be required. This is not possible with the current prototype because, for a speed change to be made, the controller needs to interrupt the previous instruction, change the parameters and send another instruction. Since this causes a discontinuous movement, our approach was to always send instructions at the same speed. Concretely, we use a positive constant velocity in the *y* axis in the *Search* mode, and constant speed in *x*/*y* axes in the *Approach* mode, with varying direction, depending on the error signal. As described, the control law applied in the *Approach* mode is that of an ON-OFF controller (see Figure 11).

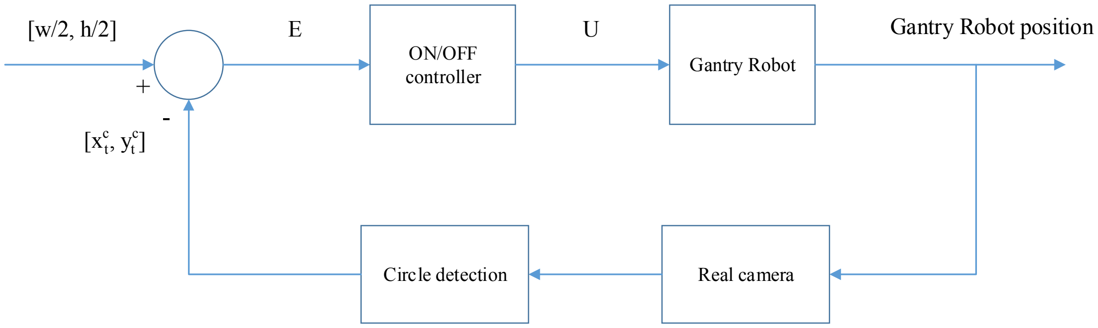

Fig. 11 - Block diagram of the image-based position control - experimental prototype.

To experimentally validate the control logic and the visual control loop, several tests were performed with the same three targets, at scaled positions, in the world reference frame.

Experimental Results

Figures 12-(a), (c) show the trajectory of the camera in the *x* and *y* coordinates and Figures 12-(b), (d) shows the speed in the two directions. These results also validate experimentally the proposed control logic in commanding the sequence of operations to process the given targets.

In Figure 12(a), zones (1) relate to the target search phase and zones (2) represent the position corrections applied in the *x* direction to approximate the center of the target with the center of the image. In this example, two adjustments were made, the first one of 45 mm to the right of the initial position and the second, in the opposite direction, of about 25 mm.

In Figure 12(c), the several phases of the *y* motion are represented, with zones (1) representing the target *Search* phase, zones (2) representing the *Approach* phase and the zones (3) representing the *Processing* phase.

CONCLUSIONS

This work proposed an approach to control the position of a gantry robot using a computer vision system. The approach aims at tasks in which a number of objects are displayed on a horizontal plane within the robot's workspace, and each object must be processed by the robot. By adopting a few assumptions, namely regarding the placement of the targets and the camera orientation, a very simple control system was developed. The simulation and experimental results validated the approach with respect to both the control logic and the image-based control loops.

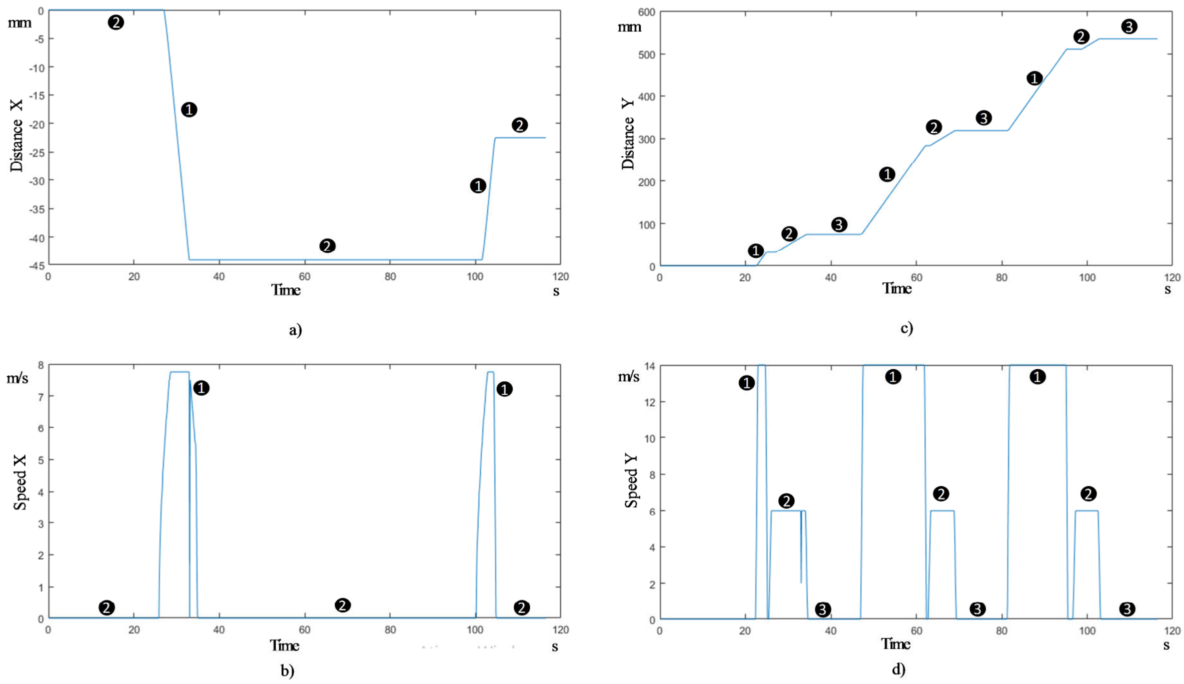

Fig. 12 - Camera *x* position (a) and speed (b), *y* position (c) and speed (d) - experimental prototype.

Due to its simplicity, this kind of control system can be effortlessly adopted in industrial applications where automatic and accurate positioning are needed. Moreover, the introduction of a vision system to control an industrial operation has the addition benefits that the same system may also be used to perform quality control and safety tasks, thus reducing risks to human operators.

In future work, the current prototype should be modified to include continuous controlled servos in place of stepper motors, to allow testing the image-based servo controller in real experiments. Also, the accuracy of the system should be studied in relation to the resolution of the camera. In this context, higher resolution cameras should be tested in the control loop, and an industrial application task should be used as a testbed. The next steps in this work also include the system integration in an internet network, allowing for remote monitoring and action, under the approach of industry 4.0.

ACKNOWLEDGMENTS

This work was supported by the 2nd edition of the Annual Contest for Projects of Research, Development, Innovation and Artistic Creation (IDI&CA) of Instituto Politécnico de Lisboa, under project grant IPL/2017/ROBOTCLEAN/ISEL.

REFERENCES

[1]-F. Civerchia, S. Bocchino, C. Salvadori, E. Rossi, L. Maggiani, and M. Petracca, "Industrial Internet of Things monitoring solution for advanced predictive maintenance applications," J. Ind. Inf. Integr., vol. 7, pp. 4-12, 2017.

[2]-B. Zhang, J. Wang, G. Rossano, C. Martinez, and S. Kock, "Vision-guided robot alignment for scalable, flexible assembly automation," 2011 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2011, pp. 944-951, 2011.

[3]-LANGHAMMER, "Gantry Robots PRO03, PRO04," 2018. [Online]. Available: https://www.langhammer.de/en/products/gantry-robots-pro03pro04.html. [Accessed: 26-Jul-2018].

[4]-Y. Xu, X. Tong, Y. Mao, W. B. Griffin, B. Kannan, and L. A. Derose, "A vision-guided robot manipulator for surgical instrument singulation in a cluttered environment," Proc. - IEEE Int. Conf. Robot. Autom., pp. 3517-3523, 2014.

[5]-G. Biegelbauer and M. Vincze, "3D vision-guided bore inspection system," Proc. Fourth IEEE Int. Conf. Comput. Vis. Syst. ICVS'06, vol. 2006, no. Icvs, p. 22, 2006.

[6]-F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," IEEE Robot. Autom. Mag., vol. 13, no. 4, pp. 82-90, 2006.

[7]-T. Shen, J. Yang, Y. Cai, D. Li, and G. Chesi, "Visual servoing with cylinders: Reaching the desired location following a straight line," Chinese Control Conf. CCC, no. 1, pp. 11183-11188, 2017.

[8]-M. Gridseth, K. Hertkorn, and M. Jagersand, "On Visual Servoing to Improve Performance of Robotic Grasping," Proc. -2015 12th Conf. Comput. Robot Vision, CRV 2015, pp. 245-252, 2015.

[9]-G. Palmieri, M. Palpacelli, M. Battistelli, and M. Callegari, "A comparison between position-based and image-based dynamic visual servoings in the control of a translating parallel manipulator," J. Robot., vol. 2012, 2012.

[10]-S. Huang, Y. Yamakawa, T. Senoo, and M. Ishikawa, "A direct visual servo scheme based on simplified interaction matrix for high-speed manipulation," 2012 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2012 - Conf. Dig., pp. 1950-1955, 2012.

[11]-G. J. Garcia, C. A. Jara, J. Pomares, and F. Torres, "Direct visual servo control of a robot to track trajectories in supervision tasks," 11th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2010, no. December, pp. 1434-1439, 2010.

[12]-C. Cai, E. Dean-le, N. Somani, and A. Knoll, "3D Image-based Dynamic Visual Servoing with uncalibrated Stereo Cameras."

[13]-M. T. Hussein, “SURVEY PAPER A review on vision-based control of flexible manipulators,” Adv. Robot., vol. 29, no. 24, pp. 1575-1585, 2015.

[14]-J. Illingworth and J. Kittler, “The adaptive Hough transform,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 5, pp. 690-698, 1987.

[15]-Vslot-Europe, “Vslot-Europe - By Open Technologies,” 2018. [Online]. Available: http://vslot-europe.com/. [Accessed: 25-Jul-2018].