24th COBEM - 2017

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-0822 CONTRIBUTING TO THE IMPLEMENTATION OF A VISION-BASED CONTROL STRATEGY FOR A PLANAR PARALLEL MANIPULATOR

Fernanda Thaís Colombo
João Vitor de Carvalho Fontes
Maíra Martins da Silva
Universidade de São Paulo, Escola de Engenharia de São Carlos, SP, Brazil
fernanda.colombo@usp.br, joao.fontes@usp.br and mairams@sc.usp.br

Abstract. *The control of parallel manipulators require the use of complex strategies due to the coupling of the kinematic chains. Classical control strategies for robotic manipulators have to be revisited when dealing with the design of parallel kinematic manipulators. Among these strategies, two strategies should be considered: (i) joint space computed torque control and (ii) Cartesian space computed torque control. The first strategy (i) requires the calculation of inverse kinematics of the manipulator since the feedback control is performed by using the actual position of the active joints. This complexity imposes serious limitations for the control design. The second strategy (ii) requires the measurement of the end-effector's pose since because the feedback control strategy is based on this measurement. Thus, although mathematically simpler, this choice imposes a significant technical challenge: the measurement of the end-effector. In this work, two different aspects of strategy (ii) are treated for a 3$\underline{R}$RR, a planar parallel manipulator. Firstly, the technical details of the measurement of the end-effector's pose using servoing are given demonstrating the viability of the Cartesian space computed torque control. Secondly, an extensive numerical study is carried out for the evaluation of different control approaches: (a) joint space control, (b) Cartesian space control with servoing and (c) High-authority Low-authority control considering feedback of joints' position and end-effector's pose values. The objective of this numerical study is to furnish insights for the experimental implementation of strategy (ii). The results of both studies demonstrate the potential of the exploiting visual servoing for controlling parallel manipulators.*

Keywords: *parallel planar manipulator, visual servoing, control strategies.*

1. INTRODUCTION

Classical control strategies for robotic manipulators have to be revisited when dealing with the design of parallel kinematic manipulators (Paccot *et al.*, 2009). Among these strategies, (i) joint space computed torque control and (ii) Cartesian space computed torque control stand out. The first strategy (i) requires the calculation of inverse kinematics of the manipulator since the feedback control is performed by using the actual position of the active joints. This strategy, which is widely employed in serial manipulators, may not be suitable for parallel manipulators due to the complexity of the required calculations. It can impose some difficulties to the controller design, such as lack of speed and performance, stability issues, among others. The second strategy (ii) requires the measurement of the position of the end-effector since the feedback control strategy is based on this variable. Thus, although mathematically simpler, this choice imposes a significant technical challenge: the measurement of the pose of the end-effector.

In this work, two different aspects of strategy (ii) are treated. Firstly, the technical details of the measurement of the end-effector's pose using visual servoing are given, demonstrating the viability of the Cartesian space computed torque control (Paccot *et al.*, 2008). Secondly, an extensive numerical study is carried out for the evaluation of different control approaches: (a) joint space control, (b) Cartesian space control with servoing and (c) High-authority Low-authority control considering feedback of joints' position and end-effector's pose values.

2. MANIPULATOR DESIGN

The parallel manipulator under study is the 3$\underline{R}$RR manipulator, illustrated in Fig. 1. This planar manipulator consists of 3 kinematic chains composed of one active revolute joint $\underline{R}$ and two passive revolute joints RR. Details on the inverse kinematic and dynamic models of this manipulator can be found in (Fontes and da Silva, 2016).

A prototype of the 3$\underline{R}$RR manipulator is located in the Dynamic Laboratory at EESC/USP. The active joints are driven by *Maxon EC60* brushless motors linked to *Maxon GP52C* planetary gear that provides a nominal torque of 0.82 Nm at 1200 rpm. Each motor has a *Maxon EPOS2 50/5* driver that is able to sample up to 1 kHz using the embedded PID positioning controller (MaxonMotor, 2017). The visual servoing system is equipped with the *oCam-5CRO-U* digital camera that is able to capture up to 120 frames per second for certain image resolutions (Odroid-Withrobot, 2017).

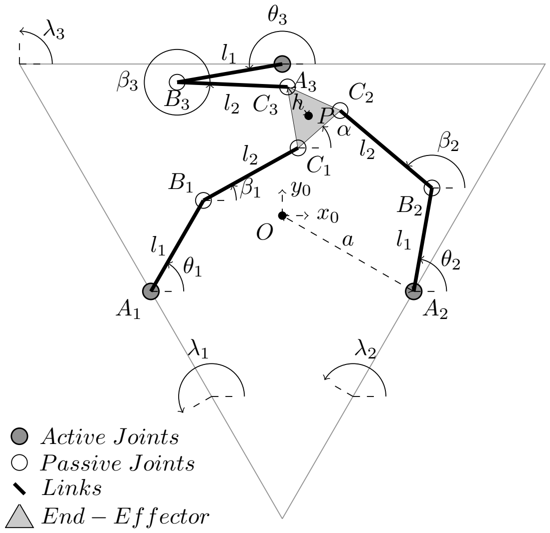

Figure 1. Model of the 3<u>R</u>RR manipulator.

3. MEASUREMENT OF THE END-EFFECTOR'S POSE USING COMPUTER VISION

The position $X = [x, y, \alpha]$ of the end-effector of the 3<u>R</u>RR manipulator can be directly measured by analyzing an image captured by a digital camera that is placed statically above the manipulator (Hutchinson *et al.*, 1996). In this project, a color USB 3.0 camera was used, which was able to capture 58 frames per second with a 640 x 480 resolution while a computer vision program was running. Each frame is analyzed in real-time using the *OpenCV* library in order to detect the positions (R_x, R_y) and (G_x, G_y) of the center of the red and green circular markers fixed at the end-effector's surface, respectively. A simplified description of the computer vision code written in *C++* to compute the position of those markers in the images captured can be seen in the Algorithm described in Fig. 2.

```
Open video capture device
loop
    Get image frame
    Remove distortion from the image using the intrinsic parameters of the camera
    Rotate the undistorted image in order to align the coordinate systems of the manipulator and the camera
    Split the image in $RGB$ color model into 3 single channels $r$, $g$ and $b$

    Apply the following filter to obtain the binary image $JustRed = ((1.2 * r - 0.5 * g - 0.5 * b) > 110)$
    Detect all contours in $JustRed$
    Draw the biggest contour, filling its interior, in the properly sized blank image $RedContour$
    For each non-zero pixel in $RedContour$, obtain its location in the $x$ and $y$ directions
    For each direction, sum all the locations found, obtaining $\sum R_x$ and $\sum R_y$
    Then divide $\sum R_x$ and $\sum R_y$ by the number of non-zero pixels to determine $R_x$ and $R_y$, respectively

    Create a rectangular region of interest $GreenROI$ around the pixel located at $R_x$ and $R_y$ of the image captured
    Convert $GreenROI$ from $RGB$ to $HSV$, extracting the $H$ channel
    Copy the $H$ of $GreenROI$ to a blank image with same size of the image captured
    Apply the following filter to obtain the binary image $JustGreen = ((H > 35)\&(H < 85)\&(g - r > 15))$
    Detect all contours in $JustGreen$
    Draw the biggest contour, filling its interior, in the properly sized blank image $GreenContour$
    For each non-zero pixel in $GreenContour$, obtain its location in the $x$ and $y$ directions
    For each direction, sum all the locations found, obtaining $\sum G_x$ and $\sum G_y$
    Then divide $\sum G_x$ and $\sum G_y$ by the number of non-zero pixels to determine $G_x$ and $G_y$, respectively
end loop
```

Figure 2. Algorithm for the computation of the positions (R_x, R_y) and (G_x, G_y) of the red and green markers.

The images resulting from the steps 3, 4 and 5 of the Algorithm 2 can be seen in Fig. 3. The bending of some edges in the original image (a) is no longer seen in the undistorted one (b). The x direction of the manipulator's coordinate system is now aligned to the horizontal direction of the rotated image (c), a necessary step to obtain the correct pose α of the end-effector.

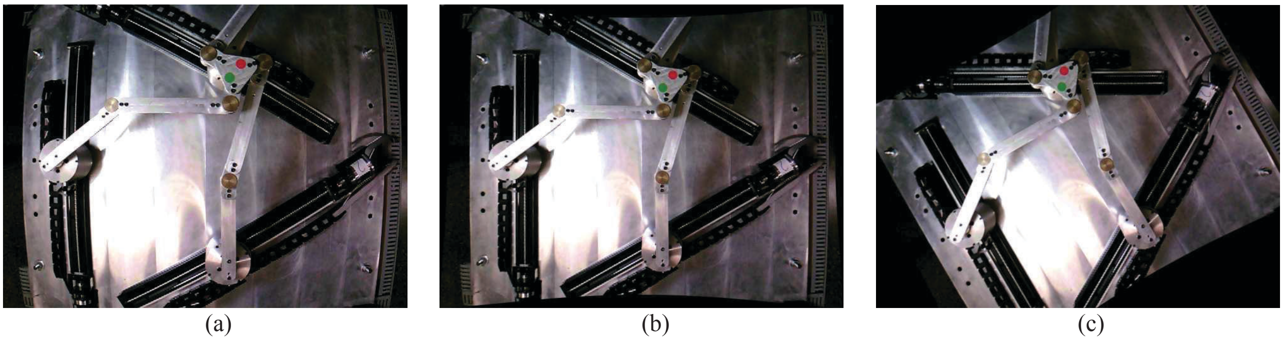
(a) (b) (c)

Figure 3. (a) Original image captured by the camera; (b) undistorted image; and (c) undistorted and rotated image.

After applying both Red and Green filters, the binary images $JustRed$ and $JustGreen$ are determined, respectively, as can be seen in Fig. 4 (a) and (c). Noticeably in the image $JustGreen$ (a), the green marker was not the only contour that passed through the green filter designed, so a function that calculates which contour is the biggest and draws it in a blank image was also done. So, the images $GreenContour$ and $RedContour$, seen in Fig. 4 (b) and (d), show that each one of the markers were successfully detected. Moreover, to improve the frame rate captured, the Green filter is only applied to a 100x100 pixels square region of interest around the center of the red marker (R_x, R_y). To find the pixel corresponding to the center of each marker, the location of each non-zero pixel is summed and then divided by the number of non-zero pixels, as described in the Algorithm in Fig. 2.

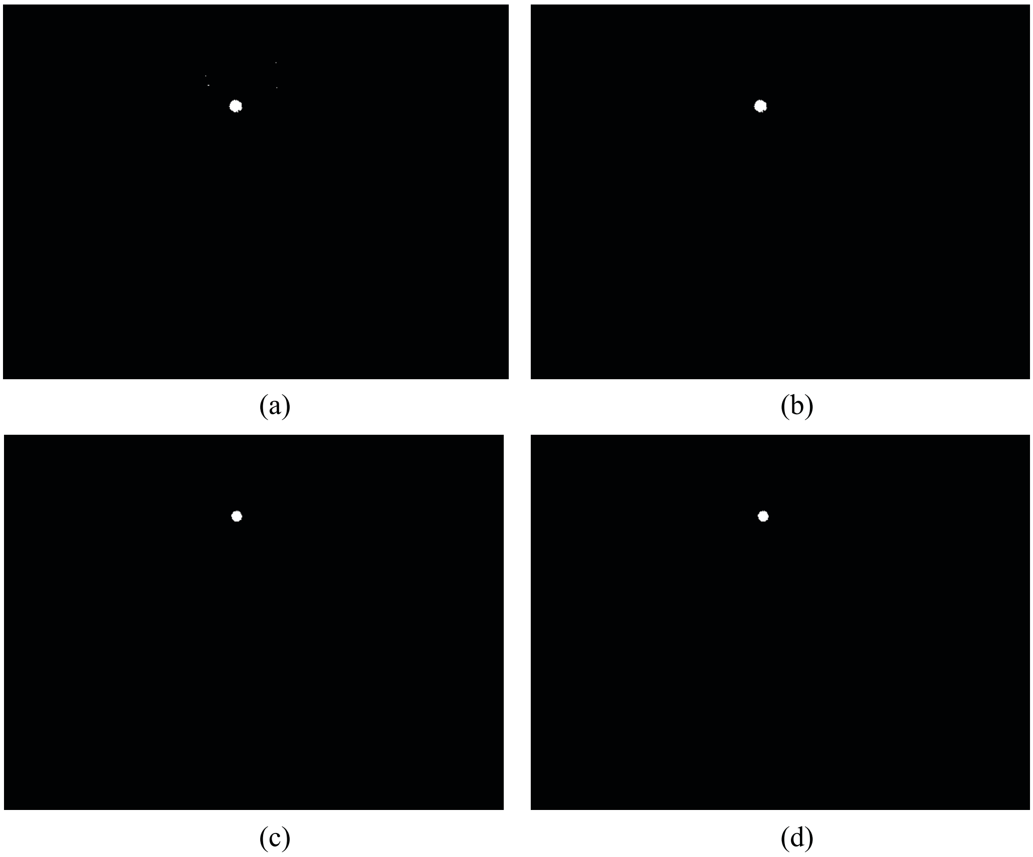
(a) (b)
(c) (d)

Figure 4. (a) $JustGreen$, (b) $GreenContour$, (c) $JustRed$ and (d) $RedContour$ images.

The shape of the end-effector, that can be seen in more detail in Fig. 5, corresponds to an equilateral triangle, in which the point P is located at its center. The points G and R, that are the centers of the green and red markers, respectively, are at the same distance from the point P and the segment line $\overline{GR}$ is parallel to the side $\overline{C_1C_2}$.

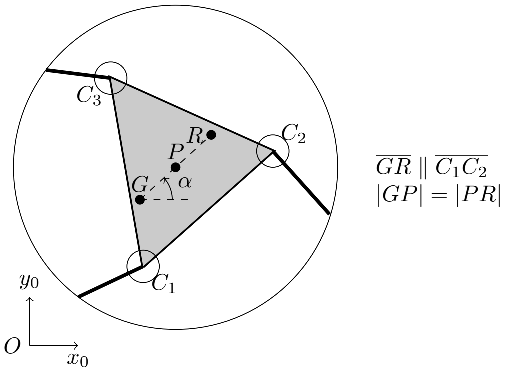

Figure 5. End-effector of the 3RRR manipulator detailed.

The position of the end-effector $X = [x, y, \alpha]$ in each frame captured by the camera can be then found using the points (G_x, G_y) and (R_x, R_y) corresponding to the center of the green and red markers, respectively, according to Eq. 1.

$$\begin{aligned} x = x_p &= \frac{G_x + R_x}{2} \\ y = y_p &= \frac{G_y + R_y}{2} \\ \alpha &= \arctan\left(\frac{G_y - R_y}{G_x - R_x}\right) \end{aligned} \tag{1}$$

4. CONTROL STRATEGIES FOR THE 3RRR MANIPULATOR

The closed-loop control of a robotic manipulator can be classified according to the use of the position of the joints $q = [\theta_1, \theta_2, \theta_3]$, in the joint space, or the position of the end-effector $X = [x, y, \alpha]$, in the Cartesian space, as feedback (Paccot *et al.*, 2007). Usually, the trajectory of the manipulator is defined by the desired position of its end-effector over time.

To control a manipulator in the joint space, the desired trajectory, given by X_d, must be converted to a trajectory in terms of the angular position of the joints q_d, using the inverse kinematics IKM. This strategy can be seen in Fig. 6.

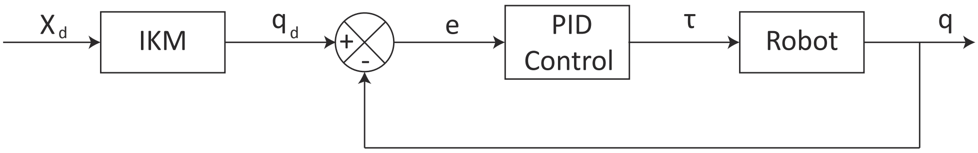

Figure 6. Joint space control.

On the other hand, the feedback signal in the Cartesian space control, seen in Fig. 7, is based on the position of the end-effector, that can be experimentally measured by the computer vision's algorithm described above. This makes the computation of the forward kinematics unnecessary, a complex task for a parallel manipulator. For this control strategy is necessary to compute the inverse of the Jacobian matrix J^{-1} of the manipulator.

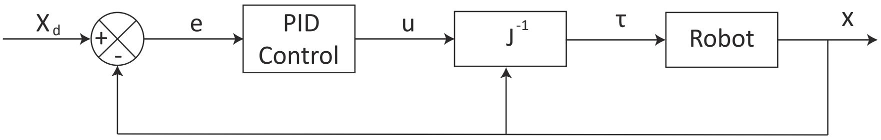

Figure 7. Cartesian space control.

Finally, the High-Authority Low-Authority control can be seen as the superposition of the control strategies in both joints and Cartesian space, as seen in Fig. 8. Since the camera used in this work has a slower data acquisition rate than the motor's encoders, the Cartesian space control has High-Authority while the joint space Control has Low-Authority.

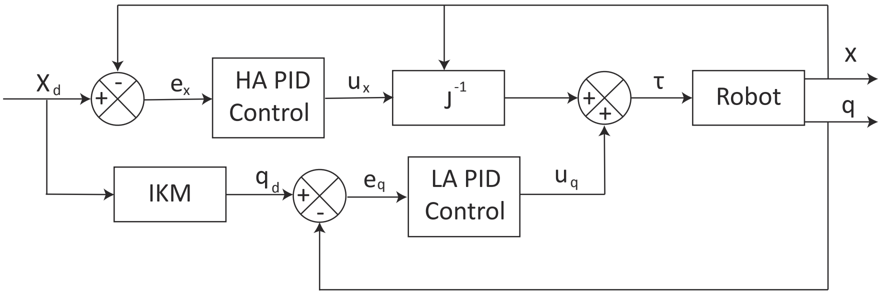

Figure 8. High-Authority Low-Authority control.

5. RESULTS AND DISCUSSION

5.1 Experimental measurement of the end-effector's pose using Computer Vision

The stability and performance of the Algorithm 2 was tested for 29 different positions of the end-effector. During this test, the end-effector was kept still and the green and red markers were detected using computer vision and the frame rate measured was 58 in average with a 640 x 480 image resolution. The average position of the center of the red and green markers, for each one of the 29 positions of the end-effector tested, can be seen in Fig. 9 as the red and green circles connected by a blue line.

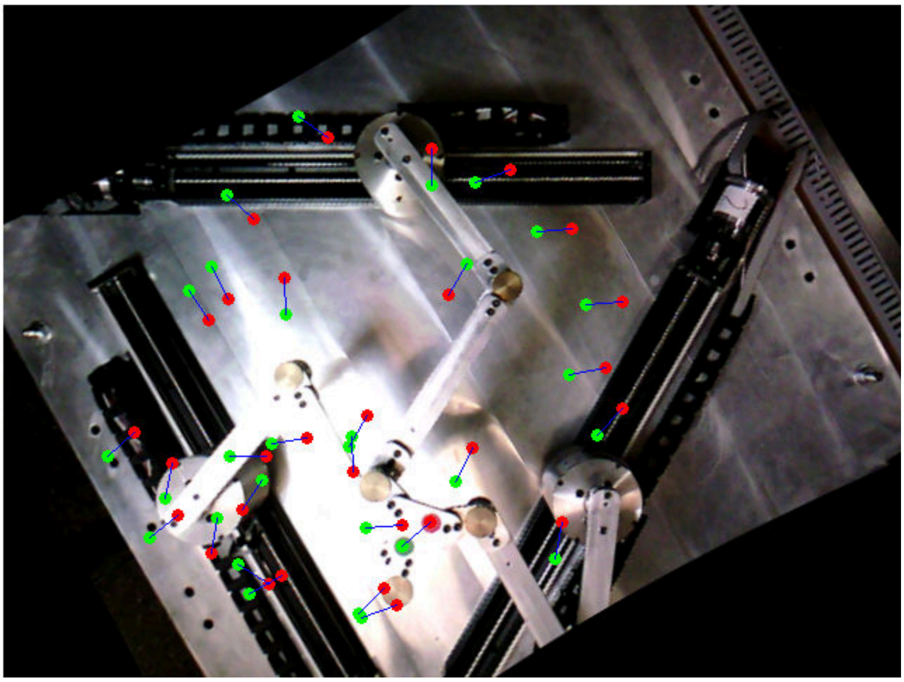

Figure 9. Position of the red and green markers detected for 29 different still positions of the end-effector.

For each one of the 29 still positions of the manipulator tested, the positions (R_x, R_y) and (G_x, G_y) of the center of the red and green markers, respectively, were saved during $20s$. The standard deviations of each R_x, R_y, G_x, and G_y data sets, seen in Fig. 10 (a), are small, showing that the data points are very close to the mean value of each data set. The difference between the maximum and minimum values of each data set, shown in Fig. 10 (b) are also small (less than 1.2 pixels), which indicates that the Algorithm 2 proposed in this project constantly detects the correct pixel corresponding to the center of the markers.

5.2 Numerical evaluation of the proposed Control Strategies

Each one of the control strategies described previously was evaluated numerically by *Simulink - Simscape Multibody* environment in *Matlab*, that can simulate the kinematics and the dynamics of the 3$\underline{\text{R}}$RR manipulator as the Plant of the control scheme under investigation. The PID gains were found using manual tuning. The performance of each strategy was then compared considering that the end-effector $X = [x \text{ in mm}, y \text{ in mm}, \alpha \text{ in rad}]^T$ should go from $X_0 = [0, 0, 0]^T$

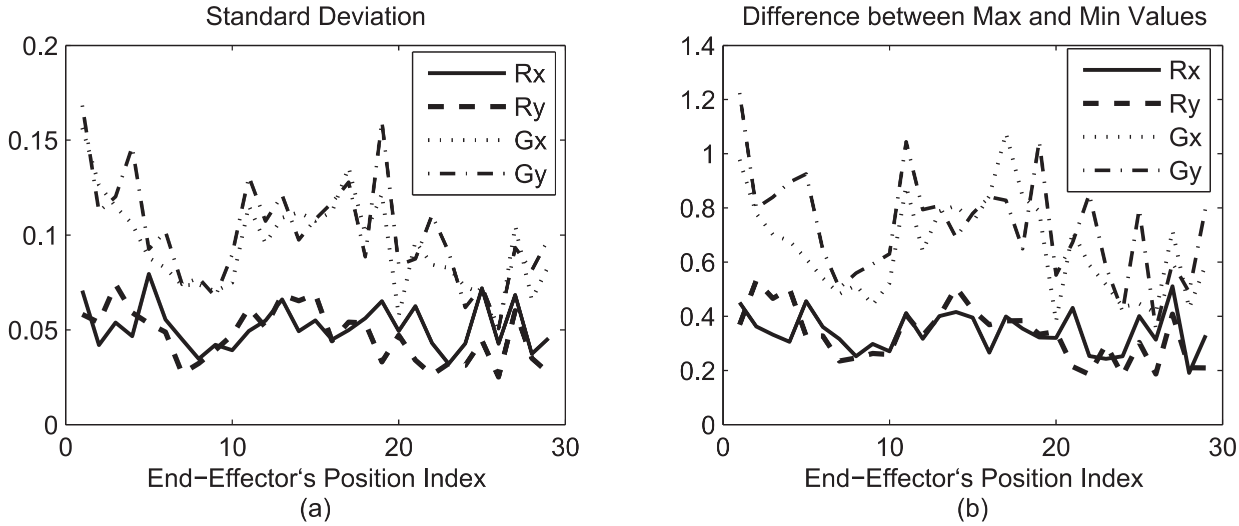

Figure 10. (a) Standard Deviation and (b) Difference between Max and Min values of the (R_x, R_y) and (G_x, G_y) detected for 29 different still positions of the end-effector.

to $X_1 = [50, 30, 0]^T$, in $\Delta t = 2s$, even though a disturbance was being applied to it. The motion of the end-effector in the $x - y$ plane and the change of its pose overtime can be seen, for each control type, in Fig. 11 (a) and (b), respectively.

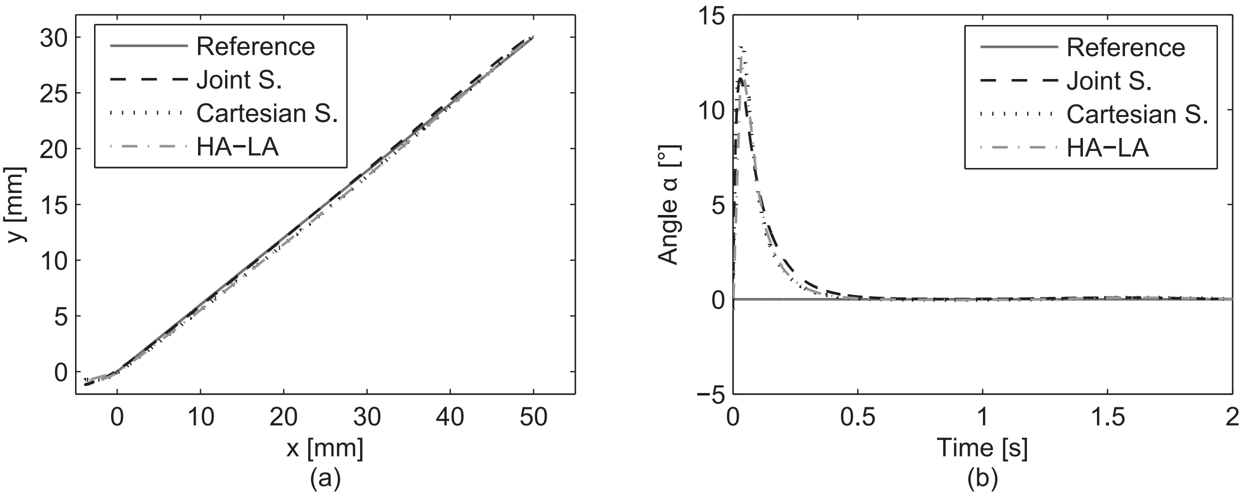

Figure 11. Comparison of the end-effector's position (a) and pose (b) obtained by the Control Strategies in the joint space, in the Cartesian space and High-Authority Low-Authority.

The three strategies compared show similar satisfactory behaviors in the simulation environment. With these positive results, these controls approaches will be implemented experimentally, in order to validate them and point out the benefits and drawback of each strategy.

6. CONCLUSIONS

Important aspects for implementing a vision-based control strategy for the 3$\underline{R}$RR, a planar parallel manipulator, are treated in this manuscript. The computer vision algorithm proposed was successful at detecting the position of the centers of the red and green markers, proving that the Cartesian space control strategy using visual servoing is viable, with a sampling rate of almost 60 fps. Moreover, three different control strategies were numerically evaluated using *Simulink*, an environment capable of simulating the kinematics and dynamics of the manipulator as the physical plant of the control systems modeled. These theoretical PID gains used in the control strategies in the joint space, in the Cartesian space and High-Authority Low-Authority, can be used in the future as an initial guess for applying these strategies experimentally to the prototype of the manipulator.

7. ACKNOWLEDGEMENTS

This research is supported by FAPESP 2014/01809-0 and by CNPq 405569/2016-5. F.T. Colombo is thankful for her scholarship FAPESP 2015/25936-4.

8. REFERENCES

Fontes, J.V. and da Silva, M.M., 2016. "On the dynamic performance of parallel kinematic manipulators with actuation and kinematic redundancies". *Mechanism and Machine Theory*, Vol. 103, pp. 148–166.

Hutchinson, S., Hager, G.D. and Corke, P.I., 1996. "A tutorial on visual servo control". *IEEE transactions on robotics and automation*, Vol. 12, No. 5, pp. 651–670.

MaxonMotor, 2017. `https://www.maxonmotorusa.com/maxon/view/product/control/Positionierung/347717`. EPOS2 50/5, Digital positioning controller, 5 A, 11 - 50 VDC - Specifications. Accessed: 2017-09-20.

Odroid-Withrobot, 2017. `https://github.com/withrobot/oCam/tree/master/Products/oCam-5CRO-U`. OCam: 5MP USB 3.0 Color Camera. Accessed: 2017-09-20.

Paccot, F., Andreff, N. and Martinet, P., 2007. "Revisiting the major dynamic control strategies of parallel robots". In *Control Conference (ECC), 2007 European*. IEEE, pp. 4377–4384.

Paccot, F., Andreff, N. and Martinet, P., 2009. "A review on the dynamic control of parallel kinematic machines: Theory and experiments". *The International Journal of Robotics Research*, Vol. 28, No. 3, pp. 395–416.

Paccot, F., Lemoine, P., Andreff, N., Chablat, D. and Martinet, P., 2008. "A vision-based computed torque control for parallel kinematic machines". In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, pp. 1556–1561.

9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.