An Optimized Control Strategy for Image-Based Visual Servoing Using Polar Coordinates

Liang Li [1,a], Junqi Luo [2,b], Zhenyu Zhang [1,c]
Jiyuan Liu[1,d], Shanjun Zhang[3,e], Liucun Zhu[2,f,*]

[1] College of Mechanical and Marine Engineering, Beibu Gulf University, Guangxi, China;

[2] Institute of Advanced Science and Technology, Beibu Gulf University, Guangxi, China;

[3] Research Institute for Integrated Science, Kanagawa University, Kanagawa 259-1293, Japan.

[a] LiLiang543@outlook.com, [b] l.junqi@hotmail.com, [c] zhenyu_zhang001@163.com, [d] jyLiu2000@163.com, [e] chiyoz01@jindai.jp, [f] lczhu@bbgu.edu.cn

Abstract. Image-based visual servoing is a crucial technique in robotic systems that enables accurate manipulation and control through visual feedback. In scenarios involving substantial rotational motion or changes in orientation, traditional Cartesian-based visual servoing methods may encounter limitations in terms of control performance. To address this challenge, we propose a novel approach that use coordinates for visual servoing control. By reparametrizing the control inputs in terms of polar radius and polar angle, our method offers a more intuitive and efficient way to handle rotations and angular variations. In addition, we further study the polar-based IBVS control by conducting experiments using a Proportional (P) controller as well as the Levenberg-Marquardt (LM) algorithm respectively, and the experimental results indicate that the proposed method demonstrates a faster convergence speed and better robustness.

Keywords: visual servoing; polar coordinates; robot control; robustness.

1. Introduction

The Image-Based Visual Servoing (IBVS) system, also known as feature-based visual servoing, involves capturing image features of the observed target under the current pose[1]. These features are then compared with the desired image features to calculate deviations, resulting in corresponding feature error values. Through an IBVS controller, control commands for the robot's joint angles are computed to adjust the end-effector's pose, thereby reducing image feature errors and accomplishing localization or tracking tasks in the visual servoing system[2].

However, due to Cartesian coordinates using three-dimensional position and orientation parameters to describe the target, it might involve intricate conversions and computations for rotation, especially when rotating around specific axes. The polar coordinate system offers greater robustness against variations in the position and orientation of the target[4]. To a certain extent, it can reduce reliance on camera intrinsic parameters such as focal length and principal point. This is because polar coordinate measurements are based on angles and distances, rather than absolute pixel coordinates[5]. The motion of the target can involve nonlinear transformations, such as rotations. Polar coordinates naturally accommodate this nonlinearity, as they transform rotational changes into simple variations in polar angles[3], [6]. Cartesian coordinates might require more intricate mathematical processing to address nonlinear motions.

In contrast, polar coordinates offer a more natural way to handle rotation and directional changes, as they partition the motion parameters into polar radius and angle, representing distance and angle, respectively[7]. This representation allows for easier adaptation to rotational motion and directional changes without the need for complex coordinate transformations.

We incorporate the benefits of polar coordinates into our proposed approach. The structure of the paper is as follows：In Section 2, IBVS is introduced, focusing on the image Jacobian of the polar and cartesian form. Next, the experimental results and the experiment platform are given In Section 3. Finally, conclusions are given in Section 4.

2. Formulation of Image Jacobian for IBVS

In the image, a 2-D point with coordinates $\boldsymbol{p} = (x, y)$ is projected in the camera frame by a 3-D point with coordinates $\boldsymbol{P} = (X, Y, Z)$[8].

$$\begin{cases} x = X/Z = \dfrac{(u - c_u)}{f\alpha} \\ y = Y/Z = \dfrac{(v - c_v)}{f} \end{cases} \#(1)$$

The coordinates of the image point are represented as $\boldsymbol{m} = (u, v)$ in pixels. The camera intrinsic parameters, denoted as $\boldsymbol{a} = (c_u, c_v, f, \alpha)$, encompass c_u and c_v as the principal points, f as the focal length, and $\boldsymbol{a}$ as the pixel dimension ratio[9]. Here, we simplify by setting $\boldsymbol{p} = (x, y)$ as the image plane coordinates of the point. Taking the time derivative of the projection equations (1),we obtain:

$$\begin{cases} \dot{x} = \dfrac{\dot{X}}{Z} - \dfrac{X\dot{Z}}{Z^2} = \dfrac{\dot{X} - x\dot{Z}}{Z} \\ \dot{y} = \dfrac{\dot{Y}}{Z} - \dfrac{Y\dot{Z}}{Z^2} = \dfrac{\dot{Y} - y\dot{Z}}{Z} \end{cases} \#(2)$$

Where $\dot{x}$ and $\dot{y}$ represent their velocities on the image plane. The equation allows us to link the velocity of the 3-D point $\boldsymbol{P} = (X, Y, Z)$ to the camera's velocity.

$$\dot{P} =- v_c - \omega_c \times P \Leftrightarrow \begin{cases} \dot{X} =- v_x - \omega_y Z + \omega_z Y \\ \dot{Y} =- v_y - \omega_z X + \omega_x Z \\ \dot{Z} =- v_z - \omega_x Y + \omega_y X \end{cases} \#(3)$$

Injecting (3) in (2), and grouping terms we obtain:

$$\begin{cases} \dot{x} =- \dfrac{v_x}{Z} + \dfrac{x v_z}{Z} + xy\omega_x - (1 + x^2)\omega_y + y\omega_z \\ \dot{y} =- \dfrac{v_y}{Z} + \dfrac{y v_z}{Z} - xy\omega_y + (1 + y^2)\omega_x - x\omega_z \end{cases} \#(4)$$

which can be written

$$\dot{\boldsymbol{p}} = \boldsymbol{J_c V_c} \#(5)$$

Where $\dot{\boldsymbol{p}} = (\dot{x}, \dot{y})$, $\boldsymbol{V_c} = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]$ represent the camera velocity and the interaction matrix $\boldsymbol{J_c}$ related to $\boldsymbol{p}$ is

$$\boldsymbol{J_c} = \begin{bmatrix} -\dfrac{1}{Z} & 0 & \dfrac{x}{Z} & xy & -(1 + x^2) & y \\ 0 & -\dfrac{1}{Z} & \dfrac{y}{Z} & 1 + y^2 & -xy & -x \end{bmatrix} \#(6)$$

In the matrix $\boldsymbol{J_c}$, the value Z is the depth of the point relative to the camera frame. Therefore, any control scheme that uses this form of the interaction matrix must estimate or approximate the value of Z[10]. Similarly, the camera intrinsic parameters are involved in the computation of x and y.However, a degeneracy occurs for the polar representation when a point is at the origin. In order to overcome this singularity, the coordinates (ρ, θ) can be used[11].our point feature $\boldsymbol{\varphi} = (\rho, \theta)$, comprising the radius of the feature point with respect to the optical center. The polar radius and polar angle are given by:

$$\rho \ = {}^{1}\!/_{Z}\sqrt{x^2 + y^2}\#(7)$$

$$\theta = \tan^{-1}\frac{y}{x}\#(8)$$

Where ρ is the polar radius and θ is the polar angle. The two feature representations are related by:

$$u = r\cos\theta\,, v = r\sin\theta\,\#(9)$$

After simple calculations, the analytical form of the Jacobian $\boldsymbol{J_p}$ of feature $\boldsymbol{\varphi} = (\rho, \theta)$ is given by:

$$\boldsymbol{J_p} \ = \begin{bmatrix} \frac{c}{Z} & \frac{s}{Z} & -\frac{\rho}{Z} & -(1+\rho^2)s & (1+\rho^2) & 0 \\ -\frac{s}{\rho Z} & \frac{c}{\rho Z} & 0 & -\frac{c}{\rho} & -\frac{s}{\rho} & 1 \end{bmatrix}\#(10)$$

Where $c = \cos\theta$ and $s = \sin\theta$, which can be written

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{J_p V_c}\#(11)$$

Therefore

$$\boldsymbol{V_c} =- \boldsymbol{\lambda}\widehat{\boldsymbol{J_p^+}}(\boldsymbol{\varphi} - \boldsymbol{\varphi}^*)\#(12)$$

Where $\boldsymbol{\varphi}^*$ is the expectation of image features and $\widehat{\boldsymbol{J_p^+}}$ is the pseudoinverse of the image Jacobian. $\boldsymbol{\lambda}$ represents the control gain.

3. Experiment Preparation And Results

3.1 Experiment Setup

Our hardware setup comprises a UR3 6-degree-of-freedom robot configured with the eye-in-hand configuration. And an Intel Realsense D435i depth camera. On the software side, we use open-source libraries, including Python and OpenCV, integrated with the Robot Operating System (ROS) for development purposes.

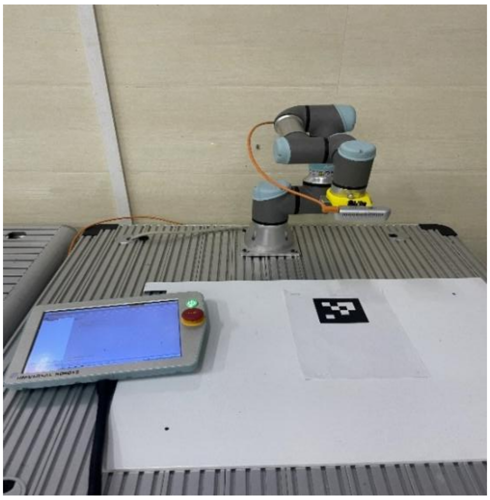


Fig1. the components of the experimental setup.

Table 1 containing experimental parameters, including control gains, image resolution and the focal length of camera. The selection of control gains comes with specific requirements. Through experimentation, it's more appropriate for the control gain to be within the range from 0 to 1.

Consequently, a step size of 0.1 was chosen for experimentation, leading to the discovery that a control gain of 0.2 is more suitable.

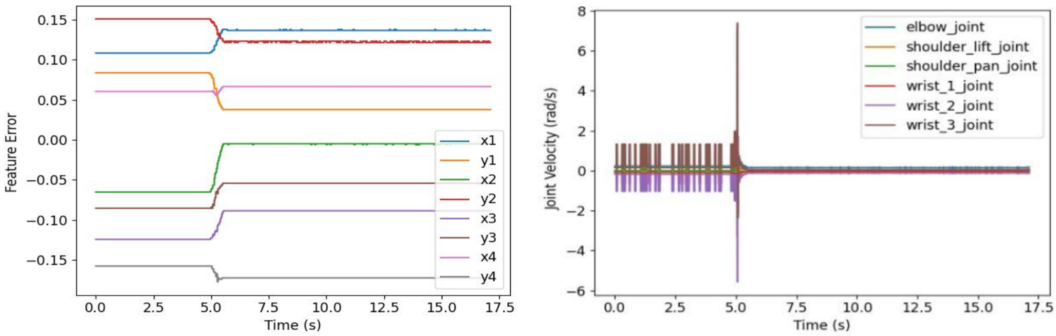
Table 1. Experiment Parameter

control gain(λ)	0.2
focal length(mm)	608.8
image resolution	640 x 480

3.2 Experiment 1: Comparison of Polar and Cartesian Coordinates Under Large Rotations

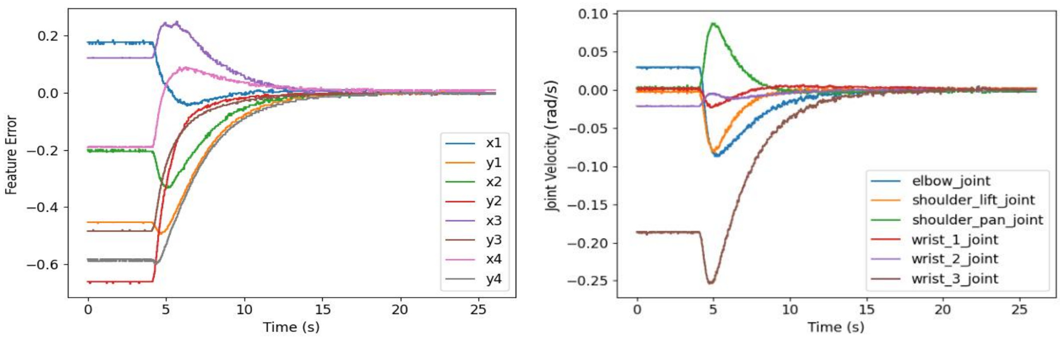
In this experiment, we examined the performance of the visual servoing system when dealing with substantial rotations. Specifically, we compare the effectiveness of polar and Cartesian coordinates in handling these rotations.

Firstly, we chose an initial position and a desired position. These selections do not need to fulfill any special requirements; the primary consideration is to ensure that moving the robotic arm from the initial position to the desired position generates a substantial amount of rotational motion. The initial positions of the joint angles are $(81.08°, -101.38°, 134.9°, -120.67°, -91.87°, 348.84°)$ and the desired positions of the joint variables are $(98.95°, -91.81°, 126.1°, -120.87°, -88.6°, 39.04°)$.



(a) (b)

Fig2. Cartesian-based IBVS with large rotations. (a) curve of feature error. (b) curve of joint velocity.



(a) (b)

Fig3. Polar-based IBVS with large rotations. (a) curve of feature error. (b)curve of joint velocity.

In the Fig.2, it depicts a situation where extensive rotational movements are executed using Cartesian coordinates. In the Fig2(a), the linear trend of the image error indicates a failure in error convergence. Additionally, in the Fig2(b), Initially, the joint velocity exceeded 1 rad/s, and by the

5th second, it reached around 5 rad/s. This indicates a convergence failure. The subsequent curve smoothing after the 5th second is due to safety concerns, we had to stop transmitting joint velocity information to the robotic arm.

Fig.3 shows a scenario involving extensive rotational motions using polar coordinates. (a) displays the convergence behavior of the image Jacobian under polar coordinates, with a relatively faster convergence taking about 10 seconds, resulting in decent convergence performance. (b)presents the joint velocities of the robotic arm. The significant difference in the brown curve compared to other curves is evident. The brown curve represents the end-effector's motion velocity, providing evidence of good results even during large rotational movements.

By comparing Fig.2 and Fig.3, it is apparent that in cases of large rotations, Polar-based IBVS proves to be more robust than Cartesian-coordinate-based IBVS. And the entire convergence process took only around 10 seconds.

3.3 Experiment 2: Comparison between P Controller and LM Algorithm

Based on the above experiments, we can conclude that in scenarios involving significant rotations, the use of polar coordinate form of IBVS yields better convergence. However, P controllers exhibit issues of slow convergence speed and low robustness. Therefore, the following experiments will compare the P controller with the LM algorithm. In this experiment, we conducted a comparative study between P controller and the LM algorithm for control purposes. The aim was to evaluate their performance in a controlled environment and assess their effectiveness in different scenarios. The initial positions of the joint variables are $(78.53^\circ, -98.12^\circ, 128.23^\circ, -112.76^\circ, -90.15^\circ, 346.25^\circ)$ and The desired positions of the joint variables are $(77.72^\circ, -105.54^\circ, 105.38^\circ, -82.77^\circ, -87.94^\circ, 3.53^\circ)$.

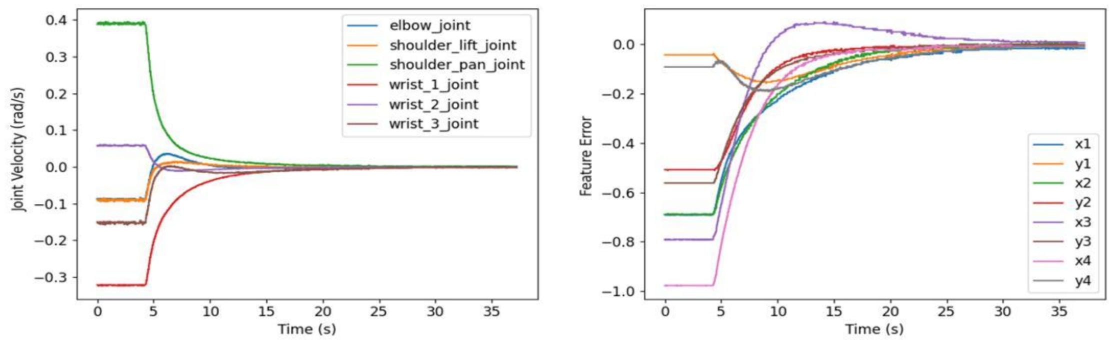


Fig4. Polar-based IBVS with P controller.

Fig.4 shows the experimental results of using P controller in polar coordinate form. During the initial five seconds of the curve, we can observe a relatively flat trend, we had not yet transmitted the joint velocities from the PC to the robotic arm. Additionally, we observed that the feature error is dimensionless. This is because, in the case of polar coordinates, image features are represented by polar radius and polar angle, unlike Cartesian coordinates where features can be expressed in millimeters. This distinction, however, does not impact our experiment. Subsequently, we can deduce that the robotic arm took around 10 seconds to converge, indicating that the achieved performance is still not highly satisfactory. The joint velocity of the robotic arm ranges from -0.35 rad /s to 0.4 rad /s, which falls within a relatively safe speed range.

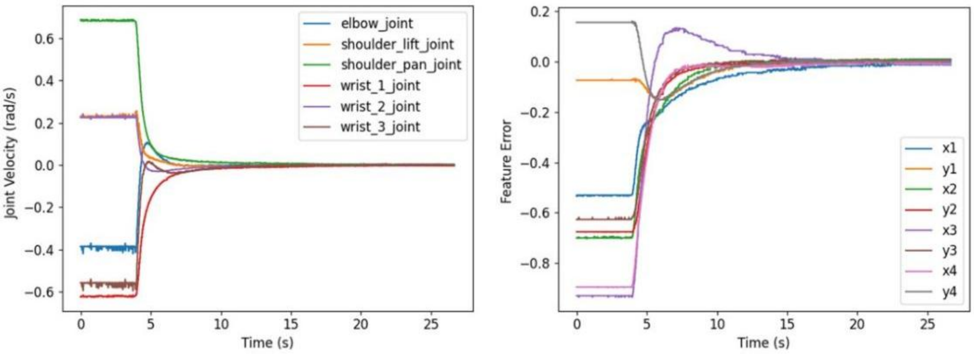


Fig5. Polar-based IBVS with LM algorithm.

Fig.5 displays the experimental results of using the LM algorithm in polar coordinate form. Similar to the previous case, the linear trend in the initial five seconds is attributed to the delay between the robotic arm and the PC, which doesn't affect the experimental outcomes. Additionally, due to image processing performance, slight fluctuations are observed in the image features. However, these fluctuations do not significantly impact the joint velocity of the robotic arm, demonstrating the robustness and disturbance resistance of this algorithm.

Furthermore, the entire convergence process took only around 6 seconds, which was an improvement for servo control. The joint velocity of the robotic arm ranged from -0.6 rad /s to 0.7 rad /s, falling within a safe speed range and fulfilling the requirement for rapid convergence. Therefore, based on the results depicted in Figures 4 and 5, it can be observed that the LM algorithm exhibits faster convergence speed compared to the P controller. Under the same conditions, compared to using P controller, the LM algorithm exhibited a faster convergence rate, achieving an improvement of nearly 65% over the P controller.

4. Conclusion

In this paper, we primarily focused on two aspects. Firstly, we compared the performance of IBVS in polar and Cartesian coordinate forms, particularly when dealing with significant rotational quantities. The results indicated that convergence failed when using Cartesian coordinates, whereas polar coordinates demonstrated superior convergence and robustness, especially in cases involving substantial rotations.

Secondly, building upon the polar-coordinate-based IBVS, we conducted a comparison between P controller and the Levenberg-Marquardt (LM) algorithm. The outcomes revealed that the LM algorithm exhibited stronger disturbance resistance and faster convergence speed. This suggests that the LM algorithm offers promising performance in the realm of visual servo control. The next step will apply the proposed method to practical robotic grasping tasks.

Acknowledgment

This work was financially supported by special Fund for Bagui Scholars of the Guangxi Zhuang Autonomous Region NO.2019A08.

References

[1] F. Chaumette and S. Hutchinson, 'Visual servo control. I. Basic approaches', *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.
[2] S. Hutchinson, G. D. Hager, and P. I. Corke, 'A tutorial on visual servo control', *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
[3] A. Ghasemi, P. Li, and W.-F. Xie, 'Adaptive Switch Image-based Visual Servoing for Industrial Robots', *Int. J. Control Autom. Syst.*, vol. 18, no. 5, pp. 1324–1334, May 2020.

[4] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, ‘Comparison of Basic Visual Servoing Methods’, *IEEEASME Trans. Mechatron.*, vol. 16, no. 5, pp. 967–983, Oct. 2011.
[5] G. Allibert and E. Courtial, ‘Switching controller for efficient IBVS’, in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal: IEEE, Oct. 2012, pp. 1695–1701.
[6] N. R. Gans and S. A. Hutchinson, ‘Stable Visual Servoing Through Hybrid Switched-System Control’, *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 530–540, Jun. 2007.
[7] G. Ye, W. Li, H. Wan, and H. Lou, ‘Novel two-stage hybrid IBVS controller combining Cartesian and polar based methods’, in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2015, pp. 397–402.
[8] B. Espiau, F. Chaumette, and P. Rives, ‘A New Approach to Visual Servoing in Robotics’.
[9] E. Marchand, F. Spindler, and F. Chaumette, ‘ViSP for visual servoing: a generic software platform with a wide class of robot control skills’, *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, Dec. 2005.
[10] Yun-Hui Liu, Hesheng Wang, Chengyou Wang, and Kin Kwan Lam, ‘Uncalibrated visual servoing of robots using a depth-independent interaction matrix’, *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 804–817, Aug. 2006.
[11] P. I. Corke, F. Spindler, and F. Chaumette, ‘Combining Cartesian and polar coordinates in IBVS’, in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA: IEEE, Oct. 2009, pp. 5962–5967.