Int. J. Advanced Design and Manufacturing Technology, 2023, Vol. 16, No. 1, pp. 45-56

DOI: 10.30486/admt.2022.1924605.1256 ISSN: 2252-0406 https://admt.isfahan.iau.ir

A New Visual Servoing Method for Grasping and Assembling Objects using Stereo Image Based Feedback

Mahmoud Jeddi[1]
Faculty of Material and Manufacturing Technologies
Malek Ashtar University of Technology, Tehran, Iran
E-mail: m.jedi100@mut.ac.ir

Ahmad Reza Khoogar[2, *]
Faculty of Material and Manufacturing Technologies
Malek Ashtar University of Technology, Tehran, Iran
E-mail: khoogar@mut.ac.ir
* Corresponding author

Received: 25 February 2021, Revised: 8 June 2021, Accepted: 17 June 2021

Abstract: In this paper, an eye-in-hand stereo image-based visual serving controller for industrial 6 degrees of freedom manipulator robots is presented. The visual control algorithms mostly use the relationship between camera speed and changes in image features, to determine the end-effector movement path. One of the main problems of the classical IBVS method is the inability to estimate the distance of the object related to the camera, which requires peripheral equipment such as a laser rangefinder to estimate the depth. In this study, two cameras were mounted on the end-effector of a 6 DOF manipulator robot. The distance of the object to the camera is estimated by the equations associated with the epipolar plane, and the interaction matrix is updated at any time. For increasing response speed, the image interaction matrix was divided into two separate parts related to translational and rotational motion, and it was found that only the translational motion part is affected by distance. The control method separates the camera motion into three-stage based on pure rotation, pure translation, and hybrid motion, which has a better time response compared to the classical IBVS control methods. Additionally, a method for position prediction and trajectory estimation of the moving target in order to use in a real-time grasping task is proposed and developed using Recursive Least Square as the trajectory estimators in the image plane. The simulation results show that the proposed method increases the system response speed and improves the tracking performance.

Keywords: Feature Matching, Image Interaction Matrix, Recursive Least Square, Stereo Image Based Visual Servoing, 6 DOF PILZ Robot

Biographical notes: Mahmoud Jeddi received his MSc in Mechanical Engineering from University of Tabriz 2012. He is currently PhD student in the department of Mechanical Engineering at the Malek Ashtar University of Technology, Tehran, Iran. His current research interests include Robotic, Control and Vision. **Ahmad Reza Khoogar** is Associate Professor of Mechanical Engineering at the Malek Ashtar University of Technology, Tehran, Iran. He received his PhD in Mechanical engineering from The University of Alabama, USA in 1989. His current research interests include Robotic, Control and Artificial Intelligence.

Research paper

1 INTRODUCTION

In recent years, a wide variety of applications regarding autonomous robot behavior in unstructured and unknown environment have been developed. There is a part of vision-based robotic research area called visual servo [1]. Visual servoing guide robots using the vision information. The visual servo is a framework to formalize the vision-based feedback control as a dynamical system. This framework provides rigorous evaluation for developing vision-based control systems, for example, controllability, and asymptotic stability, region of stability, robustness and sensitivity. Since these terms are familiar for control engineers visual servoing became a powerful tool for designing vision-based robotic systems. Visual servoing (VS) is a mature robotic technique having wide applications such as target/feature tracking, manipulator robot grasping [2]. The task in visual servoing, is to control the pose of the robot's end-effector, relative to the target, using visual features extracted from an image of the goal object. Existing VS schemes can be classified as image-based VS (IBVS), position-based VS (PBVS), and hybrid approaches [3]. These two methods are classified based on how the image is used to guide the robot. In PBVS, using image data the position of the end effector relative to the object is estimated and this estimated position is used to generate the robot control signal. In, IBVS image features are used directly to guide the robot. A control signal is generated to guide the robot so that the current features move towards the desired features. Visual servoing system can be single camera or stereo. The camera is usually positioned in either eye-to-hand or eye-in-hand mode. It is called the eye-in-hand where the camera is mounted on the end effector of the robot, but it is called the eye-to-hand when the camera is in a fixed position. This paper presents the eye-in-hand stereo IBVS method to control the end effector of 6 DOF manipulator PILZ robot. In IBVS and PBVS, the tracking error is defined in the image and the Cartesian space respectively, while in hybrid approaches, the error is defined in both spaces [4]. Particular interest in this paper is object pose estimation in IBVS using Recursive Least Square method. The visual control algorithms mostly use the relationship between camera speed and changes in image features (the image interaction matrix), to determine the robot end-effector path. One of the disadvantages of the classic IBVS method is the inability to estimate the depth of the target relative to the camera, which causes the robot to be ineffective in some situations or reduce system response speed. In this study, two cameras were mounted on the end-effector of a 6 DOF manipulator robot. The distance of the object to the camera is estimated by the equations associated with the epipolar plane, and the interaction matrix is updated at any time. For increasing response speed, the image interaction matrix was divided into two separate parts related to translational and rotational motion, and it was found that only the translational motion part is affected by distance. The control method separates the camera motion to three stages based on pure rotation, pure translation, and hybrid motion, which has a better time response compared to the classical IBVS control methods. In this research, it is assumed that two cameras are installed on the robot end-effector. Using the epipolar plane, the depth of the object relative to the camera is estimated at all times. The proposed method can be used for moving and stationary purposes. In some cases, due to the high speed of the moving target, the robot is not able to track and capture it. Therefore, using prediction methods, future positions of features in the image space can be estimated. In this paper, the position of the moving target in the image space was estimated using RLS, and the use of this method could improve the response speed of the system. Another problem with classical IBVS methods is the lack of detection of placement in regular shapes. Using a stereo system solves this problem. The proposed controller also solves the inherent problem of the classic IBVS method, which rotates 180 degrees around the camera axis. The paper is organized as follows. The main contributions are explicitly stated in Section 2. A traditional Image Based Visual Servoing and stereo image based visual servoing scheme are stated in Section 3. In this section the mathematical theory of monocular IBVS is presented and developed. Then, by extending the equations for the stereo approach, the image interaction matrix is calculated, while the description of the Recursive Least square is presented in Section 4. So, position prediction and trajectory estimation of the moving target are added to the proposed stereo-IBVS and make the process faster in a real-time grasping task a method. In Section 5, an intelligent hybrid visual control scheme incorporating developed controllers is explained in details. The proposed approach is evaluated in various simulation scenarios provided. Followed by Section 6 which summarizes findings and the contributions of this study.

1.1. Review of Previous Work

Many studies and methods have been proposed to improve the classical IBVS method. In this method, the control goal is to match the current properties with the desired properties on the image plane. Most studies have reported long convergence times that are not acceptable in industrial applications. In industrial applications, the IBVS method is effective and usable when the system response time is fast and stable. It is possible to reduce the system response time in the IBVS method by increasing the control law gain, but this method has limitations because increasing the gain causes robot instability and unwanted shakings [5]. In addition, traditional IBVS systems are stable only in a limited area around the desired position, and also when the desired features are far away from the initial features state, the convergence time is long and the singularity points of the image may cause the robot motion to be failed [6]. Xie et al. proposed the idea of using the switching control method in IBVS [7]. In this method, the controller switches in two modes of rotation and translation. By considering the fact that the image interaction matrix is strongly influenced by depth, they used a laser system to determine the depth of the features. In this study, it was shown that switching schemes can solve some of the inherent problems of the IBVS method, such as the inability to rotate 180 degrees around the center of the camera or get stuck in local minima, etc. Also, in this research, all camera parameters are assumed to be known and certain, and the condition for changing between different modes is based on the norm of the features error, which is predefined. This criterion is not directly related to motion separation. In fact, a switch-based control method is required that ensures stability and has a more effective criterion for switching between rotation and translation modes to meet the needs of industrial applications [7]. Another issue with the IBVS method is the dependence of the control system performance on the accuracy of the camera calibration. Although many studies have improved the tracking performance of the IBVS method by using image moments as features or trajectory optimization, none of them considered the camera parameters uncertain and assumed that the camera was well calibrated. In addition, path optimization methods take relatively considerable time to solve the optimization problem in each movement step, which is not suitable for industrial environments [8]. Camera parameters can be obtained by the calibration process. However, in some cases, especially in industrial applications, this is not possible and these parameters may change during an operation [9]. However, most studies in this field are based on the design of the kinematics-based controllers. In other words, they consider the robot as a precise positioning system that has negligible dynamics. Kinematic-based controllers ignore system dynamics and implement a simpler control process. Using a kinematics-based visual servo system is easier than using a dynamics-based control system. However, dynamic-based methods use the robot dynamic model to design the controller, and therefore can provide better control performance in terms of convergence time and guarantee stability compared to the kinematic-based control method. Ghasemi et al. Proposed an adaptive switch control method based on the robot dynamic model for the IBVS system. In this study, three states of pure rotation, pure translation, and hybrid movement were selected as control modes. A different control gain was considered for each control stage. The results showed that the adaptive switch control method has a faster time response and better stability than the two switch methods IBVS and traditional IBVS. However, in this study, the depth of the features is considered as a predefined parameter and the monocular IBVS approach was used for the Vision system, which is unsuitable in industrial environments with uncertain variables [10]. Undoubtedly, one of the most effective human tools in understanding the world and recognizing it is the sense of sight. Understanding the three-dimensional properties of a landscape and finding the geometric information of the objects in it, is one of the most important areas of research in machine vision [11]. By the growth of control systems, in the future we will see smart devices that, while being able to detect objects, route, estimate and maintain distance from passers-by or other devices, and can themselves implement the corresponding route operations [12]. With the increasing use of 3D images for different locations, different sensors are used to simulate high quality visual information [13]. Stereo vision is one of the visual sensors that extracts depth by having at least two images of a scene and based on the adaptation of the stereo [14]. Stereo vision is an emerging method that is inspired by the human visual system and calculates the three-dimensional coordinates of objects using images from left and right cameras and the corresponding stereo equations and algorithms [15]. In robotic applications, the use of stereo vision allows the robot to obtain information about the structure and geometric shape of objects and their location, both relative to the robot and to other objects [16]. The stereo vision model consists of an object and two cameras with a parallel or intersecting optical axis or other layouts that are used to achieve a three-dimensional position of the surrounding environment [17]. Payeur proposed a method based on neural network to predict the trajectory in a robotic environment in real time. His method used the six most recent measurements of the object coordinates as inputs [18]. Hideki Hashimoto describes a control scheme for a robotic manipulator system that uses visual information to position and orientate the end-effector. The control system directly integrates visual data into the servoing process without subdividing the process into determination of the position and orientation of the work piece and inverse kinematic calculation. The feature of the control scheme is the use of neural networks for the determination of the change in joint angles required in order to achieve the desired position and orientation. The proposed system is able to control the robot so that it can approach the desired position and orientation from arbitrary initial ones [19]. It should be noted that the first three columns of the image interaction matrix are related to the translation motion of the end effector and the next three columns are related to its rotational motion. If the motion is divided into translation and rotation parts, the rotational motion is independent of the depth parameter, while the translation motion is dependent on the depth, which is calculated using the stereo technique at any time and placed in the interaction matrix. This technique gives the chance of overcoming the nonlinearity created by the depth parameter and allows us to use adaptive methods to estimate the camera parameters in stereo eye in hand mode. Therefore, in this paper, the idea of switching between different modes of motion of the robot will be used in such a way that the three modes of pure rotation, pure translation and a hybrid motion for fine tuning will be used. The remaining paper is organized as follows: in section 2, a methodology is presented. In section 3, the adaptive switch controller for stereo IBVS considering the three movement states is designed. In section 4, simulation results are presented and finally, conclusion remarks are given and the advantage of the presented method is explained in section 5.

2 CONTRIBUTIONS OF PAPER

- An image-based visual servoing (IBVS) approach based on stereo vision has been presented and mathematically discussed and compared to the case of Monocular IBVS.
- The method for stacking the proper image interaction matrices for the case of image based stereo visual servoing has been developed for two cases of parallel and non-parallel cameras and the exact depth information has been extracted from the geometry of the vision system and used in image interaction matrices.
- A method for trajectory estimation of a moving object has been proposed to predict the position of the object which is used in a stereo image-based visual servoing for a real-time grasping procedure. The system dynamics of the object has been modeled in both linear and nonlinear description in image plane instead of 3- D space. object pose estimation in S_IBVS using Recursive Least Square method.
- For increasing response speed, the image interaction matrix was divided into two separate parts related to translational and rotational motion, and it was found that only the translational motion part is affected by distance. The control method separates the camera motion to three stages based on pure rotation, pure translation, and hybrid motion, which has a better time response compared to the classical IBVS control methods.

3 PROPOSED METHODOLOGIES

The task in visual servoing is to control the pose of the robot’s end-effector, relative to the goal, using visual features extracted from an image of the goal object. In this paper, the goal is to optimize the assembly of robotic parts without the need for an operator. To increase the accuracy of assembling, the two cameras are mounted on the robot's end-effector to detect the position and direction of the parts. The processed information is sent to the robot controller for decision. Object trajectory in image planes is predicted by RLS filter to increase convergence velocity. Proposed methodology is shown in “Fig. 1”.

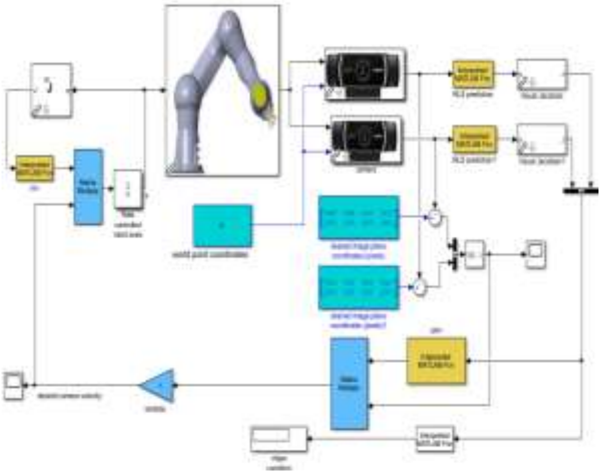

Fig. 1 Using estimation of object positions in stereo image-plane for an image-based servoing approach to grasp a moving target by 6 Dof PILZ robot.

As shown in “Fig .1”, the object in the 3-D world viewed by stereo cams and recursive least square filter estimates the trajectory of object in both image planes. Because the cameras are mounted to the robot end-effector, so the output of the images is a motion gradient and the image interaction matrix must be obtained. Using the inverse of the image interaction matrix as well as the difference between the observed points and the desired point and multiplying these two by the system gain, the desired output speed can be obtained at any time. By multiplying the desired output speed by the inverse of the robot Jacobean and integrating the response, the desired angles for the joints can be calculated.

3.1. Dynamical Model of The Stereo-IBVS

Imagine a camera mounted on the end effector and move with a body velocity $\vartheta = (v, \omega)$ in the world frame and observe a world point P with camera relative coordinates $P = (X, Y, Z)$. The velocity of the point relative to the camera frame is

$$\dot{\boldsymbol{P}} = -\boldsymbol{\omega} \times \boldsymbol{P} - \boldsymbol{v} \quad (1)$$

Which can write in scalar form as:

$$\begin{aligned} \dot{X} &= Y\omega_x - Z\omega_y - v_x \\ \dot{Y} &= X\dot{\omega}_z - Z\omega_x - v_y \\ Z &= X\omega_y \dot{-} Y\omega_x - v_z \end{aligned} \quad (2)$$

The perspective projection for normalized image-plane coordinates could write as:

$$\boldsymbol{x} = \frac{\boldsymbol{X}}{\boldsymbol{Z}}\ ,\boldsymbol{y} = \frac{\boldsymbol{Y}}{\boldsymbol{Z}} \tag{3}$$

Using the quotient rule drive the temporal derivative as:

$$\dot{\boldsymbol{x}} = \frac{\dot{X}Z - X\dot{Z}}{Z^2}\ ,\dot{\boldsymbol{y}} = \frac{\dot{Y}Z - Y\dot{Z}}{Z^2} \tag{4}$$

With placement $X = xZ$ and $Y = yZ$ in "Eq. (2)" then write in matrix form as:

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{y}} \end{pmatrix} = \begin{bmatrix} -\frac{1}{\mathrm{Z}} & \mathbf{0} & \frac{\mathrm{x}}{\mathrm{Z}} & \mathbf{xy} & -(\mathbf{1}+\mathbf{x}^2) & \mathbf{y} \\ \mathbf{0} & -\frac{1}{\mathrm{Z}} & \frac{\mathrm{y}}{\mathrm{Z}} & \mathbf{1}+\mathbf{y}^2 & -\mathbf{xy} & -\mathbf{x} \end{bmatrix} \begin{bmatrix} \mathbf{v_x} \\ \mathbf{v_y} \\ \mathbf{v_z} \\ \boldsymbol{\omega}_\mathbf{x} \\ \boldsymbol{\omega}_\mathbf{y} \\ \boldsymbol{\omega}_\mathbf{z} \end{bmatrix} \tag{5}$$

Which maps camera spatial velocity to feature velocity in normalized image coordinates. If normalized image-plane coordinates change to the pixel coordinates:

$$\boldsymbol{u} = \frac{\boldsymbol{f}}{\boldsymbol{\rho_u}}\boldsymbol{x} + \boldsymbol{u_0}\ ,\boldsymbol{v} = \frac{\boldsymbol{f}}{\boldsymbol{\rho_v}}\boldsymbol{y} + \boldsymbol{v_0} \tag{6}$$

In Eq. (6), f describes the focus length and(u_0, v_0) is principal point. By definition $\bar{u} = u - u_0$ and $\bar{v} = v - v_0$, Eq. (6) could be rearranged as $x = \frac{\rho_u}{f}\bar{u}$, $y = \frac{\rho_v}{f}\bar{v}$ by considering ρ_u, $\rho_v$and $f$is constant parameters, so the temporal derivative related to the pixel coordinates is:

$$\boldsymbol{x} = \frac{\boldsymbol{\rho_u}}{\boldsymbol{f}}\bar{\boldsymbol{u}}\ , \boldsymbol{y} = \frac{\boldsymbol{\rho_v}}{\boldsymbol{f}}\bar{\boldsymbol{v}} \tag{7}$$

And substituting "Eq. (7) and Eq. (5)" into Eq. leads to:

$$\begin{pmatrix} \dot{\bar{\boldsymbol{u}}} \\ \dot{\bar{\boldsymbol{v}}} \end{pmatrix} = \begin{bmatrix} -\frac{\boldsymbol{f}}{\boldsymbol{\rho_u Z}} & \mathbf{0} & \frac{\bar{\boldsymbol{u}}}{\boldsymbol{Z}} & \frac{\boldsymbol{\rho_v}\overline{\boldsymbol{uv}}}{\boldsymbol{f}} & -\frac{\boldsymbol{f}^2 + \boldsymbol{\rho_u}^2\bar{\boldsymbol{u}}^2}{\boldsymbol{\rho_u f}} & \frac{\boldsymbol{\rho_v}\bar{\boldsymbol{v}}}{\boldsymbol{\rho_u}} \\ \mathbf{0} & -\frac{\boldsymbol{f}}{\boldsymbol{\rho_v Z}} & \frac{\bar{\boldsymbol{v}}}{\boldsymbol{Z}} & \frac{\boldsymbol{f}^2 + \boldsymbol{\rho_v}^2\bar{\boldsymbol{v}}^2}{\boldsymbol{\rho_v f}} & -\frac{\boldsymbol{\rho_v}\overline{\boldsymbol{uv}}}{\boldsymbol{f}} & -\frac{\boldsymbol{\rho_u}\bar{\boldsymbol{v}}}{\boldsymbol{\rho_v}} \end{bmatrix} \begin{bmatrix} \boldsymbol{v_x} \\ \boldsymbol{v_y} \\ \boldsymbol{v_z} \\ \boldsymbol{\omega_x} \\ \boldsymbol{\omega_y} \\ \boldsymbol{\omega_z} \end{bmatrix} \tag{8}$$

To simplify assume $\rho = \rho_u = \rho_v$ and $\bar{f} = {}^{f}/_{\rho}$. The Jacobian matrix could be simplified as:

$$\boldsymbol{J_p(p, Z)} = \begin{bmatrix} -\frac{\bar{\boldsymbol{f}}}{\boldsymbol{Z}} & \mathbf{0} & \frac{\bar{\boldsymbol{u}}}{\boldsymbol{Z}} & \frac{\overline{\boldsymbol{uv}}}{\bar{\boldsymbol{f}}} & -\frac{\bar{\boldsymbol{f}}^2 + \bar{\boldsymbol{u}}^2}{\boldsymbol{f}} & \bar{\boldsymbol{v}} \\ \mathbf{0} & -\frac{\bar{\boldsymbol{f}}}{\boldsymbol{Z}} & \frac{\bar{\boldsymbol{v}}}{\boldsymbol{Z}} & \frac{\bar{\boldsymbol{f}}^2 + \bar{\boldsymbol{v}}^2}{\boldsymbol{f}} & \frac{\overline{\boldsymbol{uv}}}{\bar{\boldsymbol{f}}} & -\bar{\boldsymbol{u}} \end{bmatrix} \tag{9}$$

"Eq. (7)" can write concise matrix form as:

$$\dot{\boldsymbol{P}} = \boldsymbol{J_p(p, Z)\vartheta} \tag{10}$$

Where, J_p, is the 2 × 6 image Jacobian matrix for a point feature at coordinate p and camera distance$Z$. Jacobean matrix $J_p(p, Z)$ could be divided in two parts of translation and angular:

$$\boldsymbol{J_p(p, Z)} = \begin{bmatrix} \underbrace{\begin{matrix} -\frac{\bar{\boldsymbol{f}}}{\boldsymbol{Z}} & \mathbf{0} & \frac{\bar{\boldsymbol{u}}}{\boldsymbol{Z}} \\ \mathbf{0} & -\frac{\bar{\boldsymbol{f}}}{\boldsymbol{Z}} & \frac{\bar{\boldsymbol{v}}}{\boldsymbol{Z}} \end{matrix}}_{\frac{1}{Z}J_t} & \underbrace{\begin{matrix} \frac{\overline{\boldsymbol{uv}}}{\bar{\boldsymbol{f}}} & -\frac{\bar{\boldsymbol{f}}^2 + \bar{\boldsymbol{u}}^2}{\boldsymbol{f}} & \bar{\boldsymbol{v}} \\ \frac{\bar{\boldsymbol{f}}^2 + \bar{\boldsymbol{v}}^2}{\boldsymbol{f}} & \frac{\overline{\boldsymbol{uv}}}{\bar{\boldsymbol{f}}} & -\bar{\boldsymbol{u}} \end{matrix}}_{J_\omega} \end{bmatrix} \tag{11}$$

The "Eq. (11)" can write in brevity as:

$$\dot{\boldsymbol{P}} = \left(\frac{\mathbf{1}}{\boldsymbol{Z}}\boldsymbol{J_t(p, Z)}\ \vdots \boldsymbol{J_\omega(p, Z)}\right)\boldsymbol{\vartheta} \tag{12}$$

Substitute "Eq. (12)" into "Eq. (8)":

$$\begin{pmatrix} \dot{\bar{\boldsymbol{u}}} \\ \dot{\bar{\boldsymbol{v}}} \end{pmatrix} = \frac{\mathbf{1}}{\boldsymbol{Z}}\boldsymbol{J_t v} + \boldsymbol{J_\omega \omega} \tag{13}$$

Rearranging Eq. (13) in linear form:

$$\frac{\mathbf{1}}{\boldsymbol{Z}}\boldsymbol{J_t v} = \begin{pmatrix} \dot{\bar{\boldsymbol{u}}} \\ \dot{\bar{\boldsymbol{v}}} \end{pmatrix} - \boldsymbol{J_\omega \omega} \tag{14}$$

Writing "Eq. (12)" in compact form $A\theta = B$, we have a simple linear equation. Computing the image Jacobian requires knowledge of the camera intrinsic, the principal point, and focal length, but in practice, it is quite tolerant of errors in these.

3.2. Interaction Matrix for Stereo Vision

Consider the pair of cameras are look at an arbitrary point in the world. The projected point in each image plane is showed by $\{p_i(x_i, y_{i)}\ ,\ i = l, r\}$ so relative equation for projecting observed points in the left and right image planes could be written as:

$$\boldsymbol{x_l} = \frac{X + \frac{b}{2}}{Z}\quad , \boldsymbol{y_l} = \frac{Y}{Z} \tag{15-a}$$

$$\boldsymbol{x_r} = \frac{X - \frac{b}{2}}{Z}\quad , \boldsymbol{y_r} = \frac{Y}{Z} \tag{15-b}$$

Normalize the coordinates and describe “Eq. 15” in pixel dimensions:

$$\boldsymbol{x_l = \frac{u_l - u_0}{f^{*}\alpha}} \quad , \quad \boldsymbol{y_l = \frac{v_l - v_0}{f^{*}}} \tag{16-a}$$

$$\boldsymbol{x_r = \frac{u_r - u_0}{f^{*}\alpha}} \quad , \quad \boldsymbol{y_l = \frac{v_r - v_0}{f^{*}}} \tag{16-b}$$

In “Eq. (16)”, u_0 and v_0 are the coordinates of the camera principal point, f is the focal length, α is the ratio of the pixel dimensions where $\frac{dy}{dx} = \alpha$ and f^{*} is focal length described in pixel dimensions. Taking the time derivative of the perspective projection Equations:

$$\boldsymbol{x_l = \frac{\dot{X} + x_l\dot{Z}}{Z}} \quad , \ \boldsymbol{y_l = \frac{\dot{Y} + y_l\dot{Z}}{Z}} \tag{17-a}$$

$$\boldsymbol{x_r = \frac{\dot{X} - x_r\dot{Z}}{Z}} \quad , \ \boldsymbol{y_l = \frac{\dot{Y} - y_r\dot{Z}}{Z}} \tag{17-b}$$

The velocity of a feature point in an image p_I can be written related to the velocity of a feature point in a camera frame P^c as:

$$\boldsymbol{\dot{p} = J_c^I \dot{P}^c} \tag{18}$$

Where of $p_I = [p_l, p_r]$ and:

$$\boldsymbol{J_C^I} = \begin{bmatrix} \frac{\partial x_l}{\partial X} & \frac{\partial y_l}{\partial X} & \frac{\partial x_r}{\partial X} & \frac{\partial y_r}{\partial X} \\ \frac{\partial x_l}{\partial Y} & \frac{\partial y_l}{\partial Y} & \frac{\partial x_r}{\partial Y} & \frac{\partial y_r}{\partial Y} \\ \frac{\partial x_l}{\partial Z} & \frac{\partial y_l}{\partial Z} & \frac{\partial x_r}{\partial Z} & \frac{\partial y_r}{\partial Z} \end{bmatrix}^{T} = \begin{bmatrix} \frac{1}{Z} & 0 & \frac{1}{Z} & \frac{1}{Z} \\ 0 & \frac{1}{Z} & 0 & \frac{1}{Z} \\ -\frac{X+\frac{b}{2}}{Z^2} & -\frac{Y}{Z^2} & -\frac{X-\frac{b}{2}}{Z^2} & -\frac{Y}{Z^2} \end{bmatrix}^{T} \tag{19}$$

The velocity of P^c related to spatial camera velocity can be written as:

$$\boldsymbol{\dot{P}^c = -\omega_c \times P^c - v_c} \tag{20}$$

Solve “Eq. 20”:

$$\boldsymbol{\dot{P}^c} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} -\omega_y Z + \omega_x Y - v_x \\ -\omega_z X + \omega_x Z - v_x \\ -\omega_x Y + \omega_y X - v_Z \end{bmatrix} = \boldsymbol{\Lambda u_c} \tag{21}$$

Substituting Eq. 18 in Eq.21 can be written:

$$\boldsymbol{\dot{p}_I = J_c^I \dot{P}_c} \ \ \mapsto \ \boldsymbol{\dot{p}_I = J_c^I \Lambda u_c = J_{st} u_c} \tag{22}$$

Where, J_{st} is the stereo-vision image interaction which expresses the relation between a velocity of a feature point in an image $\dot{p}_I$, and a moving velocity of a camera u_c. Considering $X = \frac{b(x_l + x_r)}{2(x_l - x_r)}$, $Y = y_l \frac{b}{(x_l - x_r)}$ and $Z = \frac{b}{(x_l - x_r)}$ rewrite the stereo-vision image interaction matrix as:

$$\boldsymbol{J_{st}} = \begin{bmatrix} -\frac{a}{b} & 0 & x_l\frac{a}{b} & x_l y & -\left(1 + \frac{x_l(x_l + x_r)}{2}\right) & y \\ 0 & -\frac{a}{b} & y\frac{a}{b} & 1 + y^2 & -y\frac{(x_l + x_r)}{2} & -\frac{(x_l + x_r)}{2} \\ -\frac{a}{b} & 0 & x_r\frac{a}{b} & x_r y & -\left(1 + \frac{x_l(x_l + x_r)}{2}\right) & y \\ 0 & -\frac{a}{b} & y\frac{a}{b} & 1 + y^2 & -y\frac{(x_l + x_r)}{2} & -\frac{(x_l + x_r)}{2} \end{bmatrix} \tag{23}$$

Where, $a = x_l + x_r$ is called feature point disparity and $y = y_l = y_r$. Eventually, the stereo-vision image interaction matrix could be obtained with the velocities expressed in the camera frame and then transformed into the sensor frame [20].

$$\boldsymbol{\dot{p}_I} = \begin{bmatrix} \boldsymbol{J_l M_c^l} \\ \boldsymbol{J_r M_c^r} \end{bmatrix} \boldsymbol{u_c = J_{st} u_c} \tag{24}$$

Assume a camera spatial velocity be unit magnitude $v^T v = 1$, Due to “Eq. (24)”, write the camera velocity in terms of the pseudo-inverse $v = J^{+}\dot{p}$ where $J^{+} \in R^{2n\times 6}$ the Jacobian stack and $\dot{p} \in R^{2n}$ is the point velocities. Substitution this Eq’s yields the equation of an ellipsoid in the point velocity space.

$$\boldsymbol{p^{\cdot T} J^{+T} J^{+} \dot{p} = 1} \ \ \mapsto \ \ \boldsymbol{p^{\cdot T} (JJ^T)^{-1} \dot{p} = 1} \tag{25}$$

The eigenvectors of JJ^T define the principal axes of the ellipsoid and the singular values of J are the radii. The ratio of the maximum to minimum radius is given by the condition number of JJ^T and indicates the anisotropy of the feature motion. A high value indicates that some of the points have low velocity in response to some camera motions. Next section explained how to design a proper controller via selecting features.

3.3. Design Controller

In general, the relationship between changes in image features and camera speed can be written as follows:

$$\boldsymbol{\dot{s}(\mathrm{t}) = J_{st}(t) V_c(t)} \tag{26}$$

On the other hand, using the Jacobin robot matrix, the camera speed can be achieved:

$$\boldsymbol{V_c(t) = J_R(\mathrm{t})\dot{q}(t)} = \begin{bmatrix} \boldsymbol{V_{ct}(t)} \\ \boldsymbol{V_{cr}(t)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J_{Rt}(t)\dot{q}(t)} \\ \boldsymbol{J_{Rr}(t)\dot{q}(t)} \end{bmatrix} \tag{27}$$

Where, $J_R(\mathrm{t}) = [J_{Rt}(t) \quad J_{Rr}(t)]^{\boldsymbol{T}} \in \mathbb{R}^{6*1}$ is decomposed to the translational and rotational part, by considering “Eq. (26) and (27)” for the i^th features $\dot{s}_i(\mathrm{t})\epsilon\mathbb{R}^{2*1}$ as:

$$\dot{\boldsymbol{s}}_{\boldsymbol{i}}(\boldsymbol{t}) = [\boldsymbol{J}_{\boldsymbol{t}}(\boldsymbol{t}) \quad \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{t})]\begin{bmatrix}\boldsymbol{V}_{\boldsymbol{ct}}(\boldsymbol{t})\\ \boldsymbol{V}_{\boldsymbol{cr}}(\boldsymbol{t})\end{bmatrix} = \boldsymbol{J}_{\boldsymbol{t}}(\boldsymbol{t})\boldsymbol{V}_{\boldsymbol{ct}}(\boldsymbol{t}) + \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{t})\boldsymbol{V}_{\boldsymbol{cr}}(\boldsymbol{t}) \tag{28}$$

By expanding the “Eq. (28)” and placing the relation (27) in this equation, the relation between the features and the speed of the camera is obtained as follows:

$$\dot{\boldsymbol{s}}_{\boldsymbol{i}}(\boldsymbol{t}) = \boldsymbol{J}_{\boldsymbol{t}}(\boldsymbol{t})\boldsymbol{J}_{\boldsymbol{Rt}}(\boldsymbol{q}(\boldsymbol{t}))\dot{\boldsymbol{q}}(\boldsymbol{t}) + \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{t})\boldsymbol{J}_{\boldsymbol{Rr}}(\boldsymbol{q}(\boldsymbol{t}))\dot{\boldsymbol{q}}(\boldsymbol{t}) \tag{29}$$

Due to the three different stage of camera motion, the adaptive controller is designed based on the switch method. The first stage is only rotational, so the translational command is turned off. In the second stage, the translational movement is active and the rotational command is off. The third stage is the hybrid motion of rotation and translation.

In the first stage, the rotation command is off and controller decides just based on translational motion:

$$\dot{\boldsymbol{s}}_{\boldsymbol{i}}(\boldsymbol{t}) = \frac{\boldsymbol{1}}{\boldsymbol{Z}}\boldsymbol{J'}_{\boldsymbol{t}}(\boldsymbol{t})\boldsymbol{J}_{\boldsymbol{Rt}}(\boldsymbol{q}(\boldsymbol{t}))\dot{\boldsymbol{q}}(\boldsymbol{t}) \tag{30}$$

In “Eq. (30)” $J'_t(t) = ZJ_t(t)$. In the second stage, the camera is in pure rotation, so just the rotational part is turned on:

$$\dot{\boldsymbol{s}}_{\boldsymbol{i}}(\boldsymbol{t}) = \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{t})\boldsymbol{J}_{\boldsymbol{Rr}}(\boldsymbol{q}(\boldsymbol{t}))\dot{\boldsymbol{q}}(\boldsymbol{t}) \tag{31}$$

Finally, in the third stage the camera motion is due to both translation and rotation movement, so the rotation and translation command in this stage switched on:

$$\dot{\boldsymbol{s}}_{\boldsymbol{i}}(\boldsymbol{t}) = \frac{\boldsymbol{1}}{\boldsymbol{Z}}\boldsymbol{J'}_{\boldsymbol{t}}(\boldsymbol{t})\boldsymbol{J}_{\boldsymbol{Rt}}(\boldsymbol{q}(\boldsymbol{t}))\dot{\boldsymbol{q}}(\boldsymbol{t}) + \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{t})\boldsymbol{J}_{\boldsymbol{Rr}}(\boldsymbol{q}(\boldsymbol{t}))\dot{\boldsymbol{q}}(\boldsymbol{t}) \tag{32}$$

Depending on the position of the features relative to the desired position in the image space, the controller adjusts the movement of the cameras (and consequently the movement of the robot) and commands one of the mentioned positions.

If the current points of the corners of the quadrilateral are out of square shape, it is clear that the camera is farther away from the target object, so the translational mode is activated first. (See “Fig. 2”).

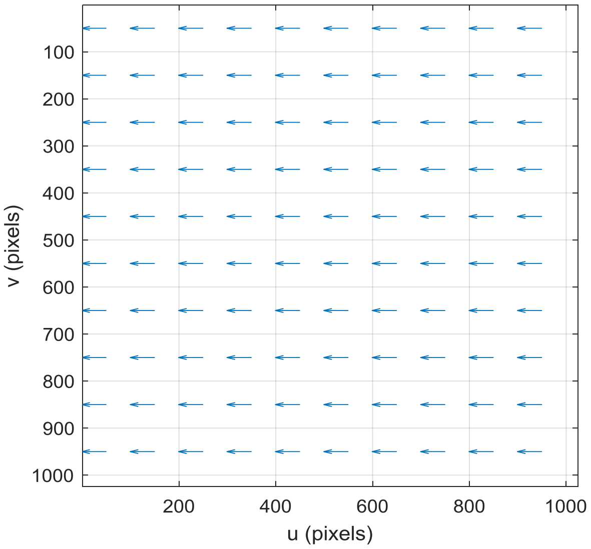

Fig. 2 Pure translation of camera movement.

If the target points differ from the desired points only in terms of orientation and positioning, only the rotation command is issued. (See “Fig. 3”).

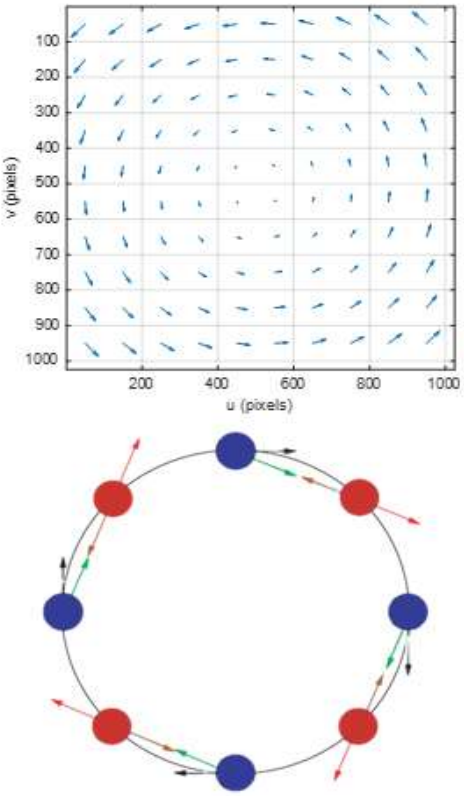

Fig. 3 Pure rotation of camera movement.

If they are slightly different in terms of position and rotation, then both rotation and translation commands are activated by the controller and this is called fine-tuning. (See “Fig. 4”).

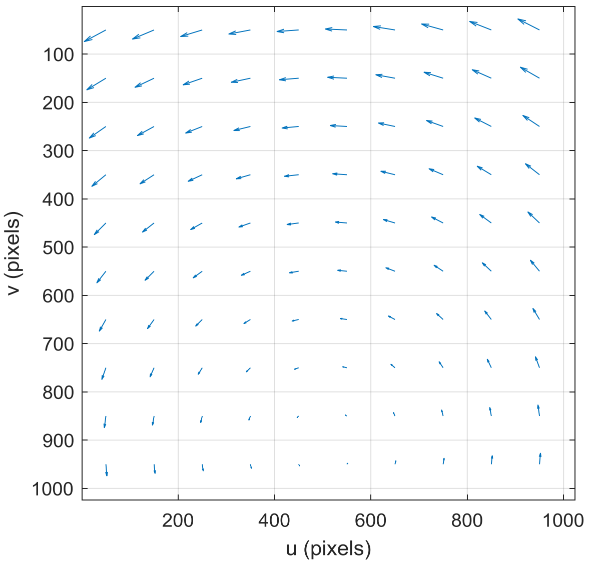

Fig. 4 hybrid movement of camera motion.

3.4. Moving Object Modeling

In order to provide essential position information for an image-based stereo visual servoing approach to grasp a moving object, it is possible to model the motion of the target in image-planes and to predict the trajectory and positions in a near future sequence. Based on the estimated velocity and acceleration of the moving object in right and left image planes and knowing the current position parameters, the estimated position of the object or the feature points in next instance, $(\hat{x}_k, \hat{y}_k)$ could be predicted as:

$$\boldsymbol{\hat{x}_k = \hat{x}_{k-1} + \hat{v}_{k-1}.\Delta T + \frac{1}{2}\hat{a}_{k-1}.\Delta T^2} \tag{35}$$

ΔT is the sampling period. In order to model the moving object in a recursive procedure, “Eq. (35)” should be expressed in form of the discrete time state transition and its observation models are as follows:

$$X_k = \emptyset_{k,k-1} X_{k-1} + W_{k-1} \tag{36}$$

$$Z_k = H_k X_k + v_k \tag{37}$$

Where, $X_k = [x_k, y_k, dx_k, dy_k]^T$ is the state vector, $Z_k = [x_k, y_k]^T$ is the measurement vector, $\boldsymbol{\emptyset_{k,k-1}}$ is the state transition matrix which represents the transition from one state vector $Xk - 1$ to the next vector Xk, Wk represents the process noises and are the measurement noises in both x and y direction. Hk is called observation matrix and represents the relationship between the measurement and the state vector. Now we are able to obtain the observation models as follows:

The measurement vector $Z_k = [x_k, y_k]^T$ is the actual position of an object or a feature point in x-y image planes in right and left cameras which could be obtained using the vision system.

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ dx_k \\ dy_k \end{bmatrix} + \begin{bmatrix} \gamma_x \\ \gamma_y \end{bmatrix} \tag{38}$$

$$\begin{bmatrix} x_k \\ y_k \\ dx_k \\ dy_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ dx_{k-1} \\ dy_{k-1} \end{bmatrix} + \begin{bmatrix} 0.5 * d^2(x_{k-1}) * \Delta t^2 \\ 0.5 * d^2(y_{k-1}) * \Delta t^2 \\ d^2(x_{k-1}) * \Delta t \\ d^2(y_{k-1}) * \Delta t \end{bmatrix} \tag{39}$$

4 TRAJECTORY ESTIMATION USING RECURSIVE LEAST SQUARE METHOD

Enough number of feature points projected in the camera plane are selected, a Recursive Least Squares method can be used to find the best estimation of the state variables of the object e.g., $X_k = [x_k, y_k, dx_k, dy_k]^T$ from the previous states data. The best estimation for time interval k can be computed as:

$$\boldsymbol{\hat{x}_k = \hat{x}_{k-1} + G_k.[Z_k - H_k \hat{X}_{k-1}]} \tag{40}$$

Where, G_k is the optimal gain matrix and H_k is the observation matrix. The gain matrix can be computed by $G_k = L_k H'_k$. L_k is the error covariance matrix for the estimation of the state of time interval k and can be expressed as:

$$\boldsymbol{L_k = (\emptyset_{k,k-1}^{-T} L_{k-1}^{-1} \emptyset_{k,k-1}^{T} + H_k^T H_k)^{-1}} \tag{41}$$

5 SIMULATION RESULTS

In this section the robotic stereo visual servoing system for PILZ robot is modeled. The effectiveness of the image-based stereo visual servoing system compared to the monocular system is validated. Then the system performance in a task of tracking and grasping a moving object is examined and the results for utilizing recursive least square method for predicting the position and trajectory of the moving target are presented and discussed. The object assumes to be a square cube and the corners of this cube are the object image features extracted in both cameras.

The stereo system consists of two parallel cameras which are located at a distance of b/2 with respect to the origin of the sensor frame. In order to keep at least 3 selected feature points on the object in both cameras' fields of views, during the approaching phase, the distance b is selected to be equal to 10 cm. The simulation results for 6 Dof PILZ robot and object points are shown in Fig. 5.

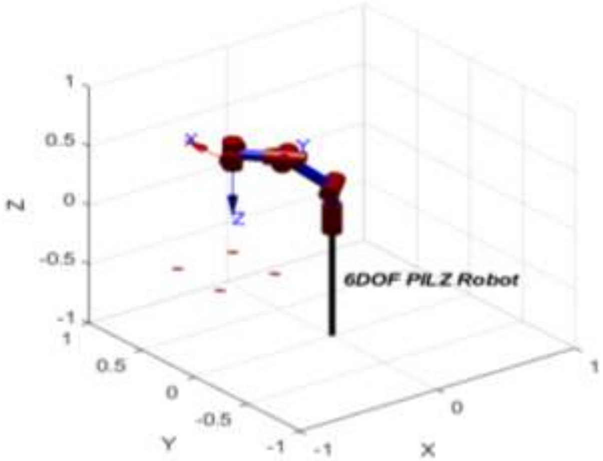

Fig. 5 6 Dof PILZ Robot and object simulation.

For the presented algorithm, a grasping algorithm with an object in a sinusoidal motion with a linear velocity of 4 cm/sec would be tested. As it was mentioned previously, the tracking and grasping task is performed by pre-defining desired positions for the object image features such that the robot moves and aligns the end-effector with the object and reaches towards it. (See "Fig. 6").

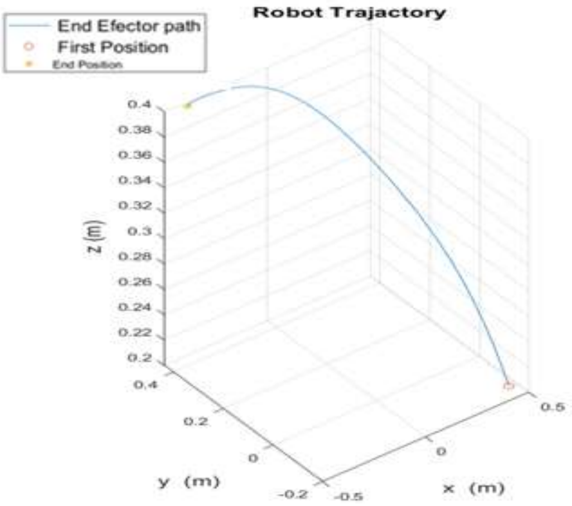

Fig. 6 The trajectory of end- effector in 3-D world for tracking the object.

Figure 7 illustrates the feature trajectory of the stereo image based visual servoing system with an RLS estimator for tracking and grasping a moving object in linear trajectory.

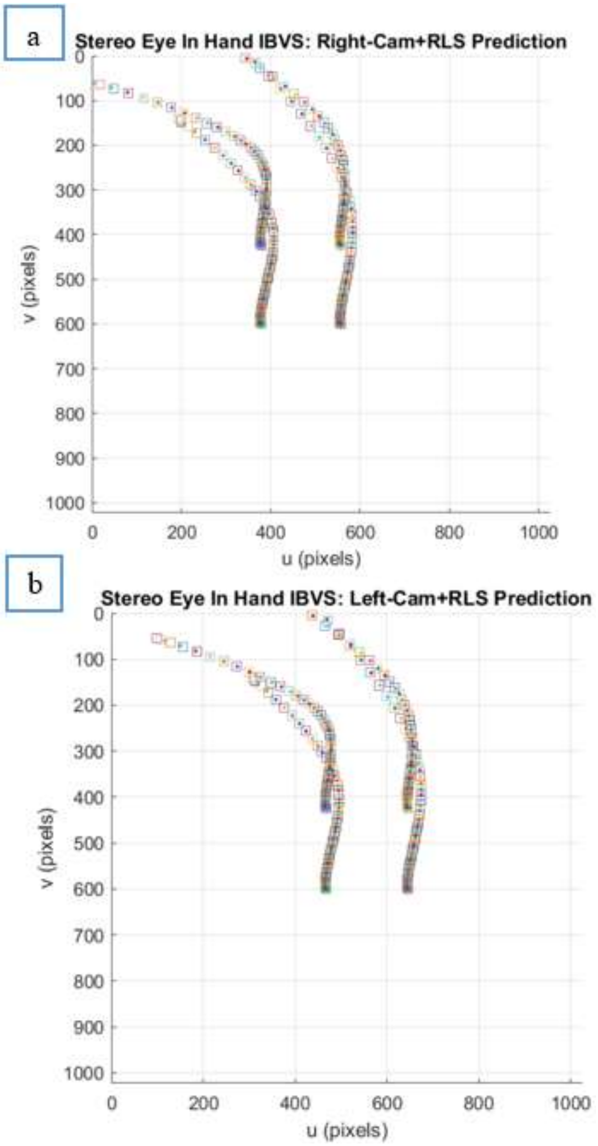

Fig. 7 Stereo IBVS system with parallel cameras behavior in a procedure of grasping a moving object using RLS method: (a): Image feature trajectories in right images, and (b): Image feature trajectories in left images.

From the simulation results shown in "Fig. 8 a and b", for a tracking and grasping task, the pixel error due to the measurement noise could be considerably reduced in both image plane by using Recursive Least Square (RLS) algorithms based on the moving object model.

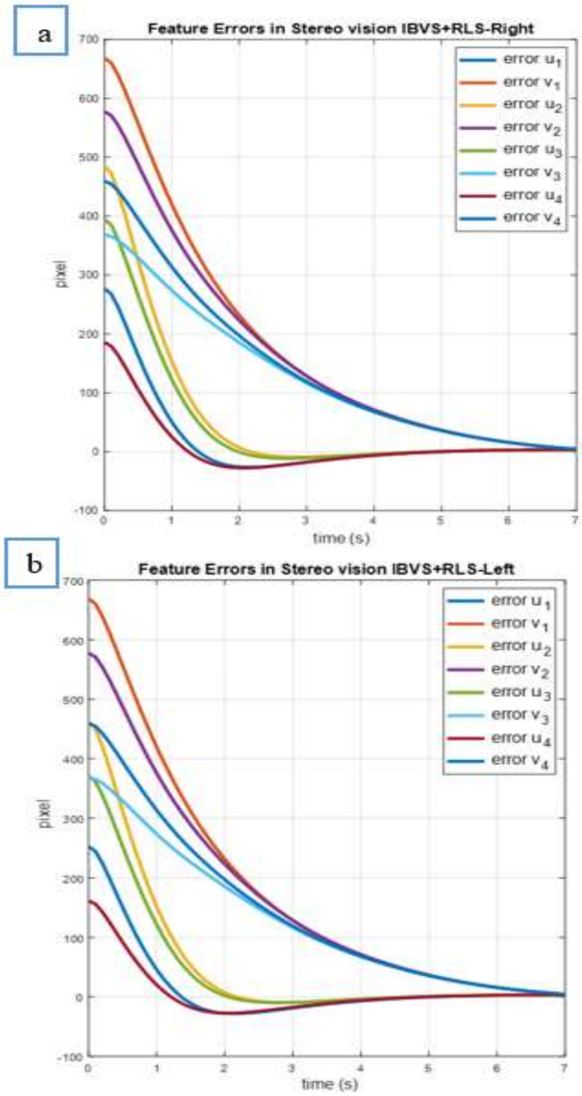

Fig. 8 Stereo IBVS system with parallel cameras behavior in a procedure of grasping a moving object using RLS method: (a): Image feature errors for right cameras, and (b): Image feature errors for left cameras.

The camera velocity components in the system with RLS prediction compared to the system without prediction, started with relatively low speeds. From the simulations shown in “Fig. 9”, it is quite considerable that in comparison with the monocular system, the trajectories of the points in both images plane are smoother in the case of using the RLS estimator and the camera velocity components do not include large oscillations which lead to fewer energy consumptions.

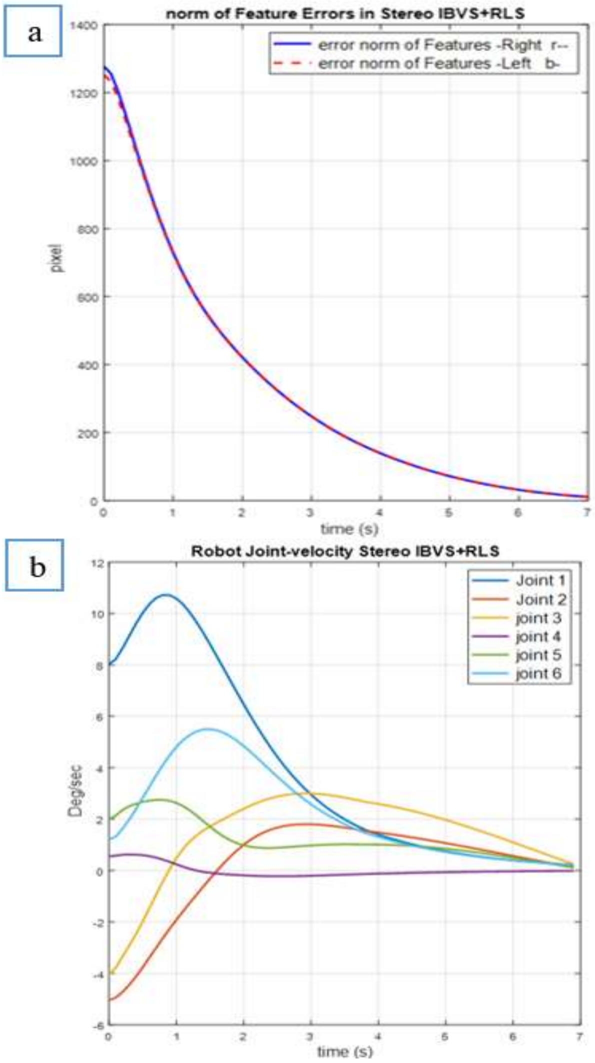

Fig. 9 Stereo IBVS system with parallel cameras behavior in a procedure of grasping a moving object using RLS method: (a): Norm of features errors in stereo IBVS, and (b): Camera frame velocity components.

In MATLAB software, the vision control system based on a single camera and stereo was simulated and the moving object was predicted by using the Recursive least squares. As mentioned in the previous sections, the image interaction matrix was divided into two parts, rotational and translational, according to the movement of the camera and the object. The rotational part is independent of the distance from the object to the camera, but to calculate the translation part, the distance from the object to the camera is required. The distance from the object to the camera is estimated at any given time by the epipolar plane. Separating the rotation and translation sections helped to reduce computations and speed up the system's time response. The simulation results are shown in “Table 1”.

Table 1 Comparison results for all the IBVS cases for tracking and grasping of a moving object

Vision system	Convergence time(s)	Maximum tracking error (pixel)	Maximum joint velocity (deg /sec)
Mono-IBVS	12.4	650	8.1
Stereo-IBVS	7.8	780	9.2
Mono-IBVS+ RLS	8.1	700	8.7
Stereo-IBVS+RLS	6.9	680	10.6

"Table 1" show that the convergence time is reduced using the stereo image based visual servoing and tracking and grasping task, the pixel error due to the measurement noise could be considerably reduced in both image plane by using Recursive Least Square (RLS) algorithms based on the moving object model.

6 CONCLUSIONS

This article presents a novel eye- in- hand image-based stereo visual servoing system for a real-time task of tracking and grasping a moving object in an uncalibrated environment. An image-based visual servoing (IBVS) approach based on stereo vision has been presented and mathematically discussed and compared to the case of Monocular IBVS. The monocular and stereo visual servoing system are simulated on 6 Dof PILZ robot. From the results, it can be inferred that in the single-camera based vision system, the system is slow in convergence and the overshoot is too high. Conversely, in the stereo control system, stereo-based convergence is faster and less response is observed than in single-camera mode. This is because in stereo vision, the Jacobean image matrices can be updated at any time using the calculated depth information. Therefore, it is possible to create the correct feedback command that leads to the stability of the vision-based control system. For increasing response speed, the image interaction matrix was divided into two separate parts related to translational and rotational motion, and it was found that only the translational motion part is affected by distance. The control method separates the camera motion to three stages based on pure rotation, pure translation, and hybrid motion, which has a better time response compared to the classical IBVS control methods.

It can be also inferred from the results that the case with the RLS estimator shows better tracking and convergence performance and has better behavior in end-effector 3-D trajectories.

REFERENCES

[1] Hashimoto, K., A review on vision-based control of robot manipulators, Adv. Robot. Int. J. Robot. Soc. Japan, Vol. 17, No. 10, 2003, pp. 969–991.

[2] Corke, P., Robotics, Vision and Control: Fundamental Algorithms In MATLAB®Second, Completely Revised, Vol. 118, 2017.

[3] Chaumette, F., Hutchinson, S., Visual Servo Control, I. Basic Approaches, IEEE Robot. Autom. Mag., Vol. 13, No. 4, 2006, pp. 82–90.

[4] Chaumette, F., Hutchinson, S., Visual Servo Control, II. Advanced Approaches [Tutorial], IEEE Robot. Autom. Mag., Vol. 14, No. 1, 2007, pp. 109–118.

[5] Keshmiri, M., Xie, W. F., and Mohebbi, A., Augmented Image-Based Visual Servoing of A Manipulator Using Acceleration Command, IEEE Trans. Ind. Electron., Vol. 61, No. 10, 2014, pp. 5444–5452.

[6] Chaumette, F., Potential Problems of Stability and Convergence in Image-Based and Position-Based Visual Servoing, In The Confluence of Vision and Control, Springer, 1998, pp. 66–78.

[7] Xie, W. F., Li, Z., Tu, X. W., and Perron, C., Switching Control of Image-Based Visual Servoing With Laser Pointer in Robotic Manufacturing Systems, IEEE Trans. Ind. Electron., Vol. 56, No. 2, 2008, pp. 520–529.

[8] Shu, T., Gharaaty, S., Xie, W., Joubair, A., and Bonev, I. A., Dynamic Path Tracking of Industrial Robots with High Accuracy Using Photogrammetry Sensor, IEEE/ASME Trans. Mechatronics, Vol. 23, No. 3, 2018, pp. 1159–1170.

[9] Shen, Y., Xiang, G., Liu, Y. H., and Li, K., Uncalibrated Visual Servoing of Planar Robots, In Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), Vol. 1, 2002, pp. 580–585.

[10] Ghasemi, A., Li, P., and Xie, W. F., Adaptive Switch Image-Based Visual Servoing for Industrial Robots, Int. J. Control. Autom. Syst., Vol. 18, No. 5, 2020, pp. 1324–1334.

[11] Ghasemi A., Xie, W. F., Decoupled Image-Based Visual Servoing for Robotic Manufacturing Systems Using Gain Scheduled Switch Control, In 2017 International Conference on Advanced Mechatronic Systems (ICAMechS), 2017, pp. 94–99.

[12] Liang, X., Wang, H., Liu, Y. H., Chen, W., and Zhao, J., A Unified Design Method for Adaptive Visual Tracking Control of Robots with Eye-In-Hand/Fixed Camera Configuration, Automatica, Vol. 59, 2015, pp. 97–105.

[13] Wang, H., Adaptive Visual Tracking for Robotic Systems Without Image-Space Velocity Measurement, Automatica, Vol. 55, 2015, pp. 294–301.

[14] Dong G., Zhu, Z. H., Kinematics-Based Incremental Visual Servo for Robotic Capture of Non-Cooperative Target, Rob. Auton. Syst., Vol. 112, 2019, pp. 221–228.

[15] Keshmiri M., Xie, W. F., Image-Based Visual Servoing Using an Optimized Trajectory Planning Technique, IEEE/ASME Trans. Mechatronics, Vol. 22, No. 1, 2016, pp. 359–370.

[16] Zheng, D., Wang, H., Chen, W., and Wang, Y., Planning and Tracking in Image Space for Image-

Based Visual Servoing of A Quadrotor, IEEE Trans. Ind. Electron., Vol. 65, No. 4, 2017, pp. 3376–3385.

[17] Zhao, Y., Xie, W. F., and Liu, S., Image-Based Visual Servoing Using Improved Image Moments in 6-DOF Robot Systems, Int. J. Control. Autom. Syst., Vol. 11, No. 3, 2013, pp. 586–596.

[18] Payeur, P., Le-Huy, H., and Gosselin, C. M., Trajectory Prediction for Moving Objects Using Artificial Neural Networks, IEEE Trans. Ind. Electron., Vol. 42, No. 2, pp. 147–158, 1995.

[19] Hashimoto, H., Kubota, T., Sato, M., and Harashima, F., Visual Control of Robotic Manipulator Based on Neural Networks, IEEE Trans. Ind. Electron., Vol. 39, No. 6, pp. 490–496, 1992.

[20] Martinet, P., Cervera, E., Stacking Jacobians Properly in Stereo Visual Servoing, In Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), Vol. 1, 2001, pp. 717–722.

[21] Kim, D. J., Lovelett, R., and Behal, A., Eye-in-Hand Stereo Visual Servoing of an Assistive Robot Arm in Unstructured Environments, In 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 2326–2331.