
Simulation of Visual Servo Control on CRS Catalyst-5t Robot System

Magdy M. Abdelhameed¹, Mohammed Marey² and Wael Sami Taie³

¹Mechatronics Engineering Department, Ain Shams University, Cairo, Egypt.

²Faculty of computer & information sciences, Ain Shams University, Cairo, Egypt

³Mechatronics Engineering Department, October 6 University, Cairo, Egypt

Received: 4-11-2014/ Revised: 8-12-2014 / Accepted: 28-12-2014

Abstract

This paper presents the position based visual servoing (PBVS) control system which govern the motion of the manipulator with an eye in hand configuration. The main core of the PBVS control scheme is the pose estimation process in which the relative pose (position and orientation) of the object with respect to the camera mounted on end-effector was estimated by two techniques (Classic POSIT and Kalman filter). CRS Catalyst-5t robot system has been selected for setting up the PBVS control system. The simulation software program has been built to evaluate the success of the simulated robot vision system in performing the required visual servoing tasks using the designed position based control scheme.

Key Words: Position based visual serving, pose estimation, POSIT and kalman filter.

Introduction

During the last decades, many researches were interested in the machines that mimic the human capabilities to gather information from the environment to interact in an autonomous way. In this aspect the machines structure has been designed to mimic some human limbs like arms and hands to perform their functions.

Arm robot has been already used in many industries because it can be more accurate than humans, it can perform multi tasks simultaneously, it can work continuously without fatigue or tiring and also it have a repeatable precision at all times. Arm robots have many applications in the industry like pick and place parts, welding, drilling and insertion of parts. It also has many applications in the medical field and it can perform well in the hazardous environment and inaccessible locations like underwater and space. But, arm robot cannot perform its work efficiently especially on unknown environments without supplying it with a large sensorial capacity.

Visual sense is one of the most human senses that provide sufficient information and non contact measurements from unknown environment. Without visual information, many arm robot tasks can be only performed in a well known environment where every object is known and can be found in a well known pose (position and orientation). Any error about the pose of the object or about the robot pose

will lead to task failure. Potential sources (such as gear backlashes, bending of the links, joints slippage, and poor fixture) would lead to the errors in the robot endeffector pose.

Vision based control system has been used to solve the above problem since the early work of Shirai and Inoue

who discussed how to use a visual feed back to compute the difference between the desired and the actual positions of the manipulator to correct the position of the robot and increase the task accuracy⁽¹⁾. typically visual sensing and the manipulation are used sequentially in an open loop fashion, "looking" then "moving". The success of this task depends directly on the calibration accuracy of the visual sensor and the robot manipulator. To enhance the overall accuracy of the operation visual feedback must be used with the manipulator in a closed loop fashion. This refers to visual servoing.

Visual servoing is the use of visual information in the feedback loop of the lowest level of a system control (usually robotic systems) with fast image processing to provide interactive behavior⁽²⁾. so visual servoing is considered a multi-disciplinary system that consists of vision system, robotics system, control system and computer system. The task of robotic visual servoing is to control the robot end-effector pose relative to the pose of the object being manipulated using real time visual feedback.

One of the main problems in position based visual servoing (PBVS) is to estimate the pose of the object frame with respect to the camera frame. The object features extracted from the image are used to estimate the relative pose of the object with respect to the camera. Using these data, the error between the current and the desired pose of the robot end-effector is calculated and the robot will modify its motion to decrease the value of this error, the task of PBVS is fulfilled when this error is vanished completely.

When using CRS Catalyst-5T robot which is a five degree of freedom robot in an unstructured environment with visual servoing control schemes installed in it, it will be important to build a simulator for this system in order to test the overall system modeling before installing a visual servoing system on the real robot that is why in this work we are interested in building such a simulator. In this paper, CRS Catalyst-5t robot system has been selected for setting up the PBVS control system, the results of the system using different pose estimation techniques will be compared.

System simulator setup

The system simulator for robot-camera platform consisting of the integration of CRS Catalyst-5T robot subsystem and vision subsystem will be described and discussed in this section.

a. Vision subsystem

The vision subsystem is represented by a camera projection model in which the image formation process is performed by mapping the object features from the 3D object frame to 2D image plane. Two techniques will be described in this section to model the image formation process, the perspective projection model and the scaled orthographic projection. Both techniques require the camera intrinsic parameters (the focal length f , inter-pixel spacing P_x, P_y along x_i, y_i axes and the image principle point O_i) and the camera extrinsic parameters (relative pose between the camera frame and the object frame) to be known.

1) The perspective projection model

To map the object feature point P_j^o from the object frame O^o to the image plane O_i the relative pose of the object frame with respect to the camera frame must be defined [5]. The relative pose is defined by $w = [X Y Z \phi \alpha \varphi]$ where X, Y and Z present the relative translation and ϕ, α and φ present the relative orientation angles roll, pitch and yaw respectively. The transformation of the point P_j^o from the object frame to the camera frame is given by:

$$P_j^c = T + R(\phi, \alpha, \varphi)P_j^o \tag{1}$$

Where

$$R(\phi, \alpha, \varphi) = \begin{bmatrix} C_\phi C_\alpha & C_\phi S_\alpha S_\varphi - S_\phi C_\varphi & C_\phi S_\alpha C_\varphi + S_\phi S_\varphi \\ S_\phi C_\alpha & S_\phi S_\alpha S_\varphi + C_\phi C_\varphi & S_\phi S_\alpha C_\varphi - C_\phi S_\varphi \\ -S_\alpha & C_\alpha S_\varphi & C_\alpha C_\varphi \end{bmatrix}$$

$$P_j^o = [X_j^o \ Y_j^o \ Z_j^o]^t, P_j^c = [X_j^c \ Y_j^c \ Z_j^c]^t, C_\theta = \cos\theta \text{ and } S_\theta = \sin\theta$$

Now the image point p_j^i is defined by

$$x_j^i = \frac{f X_j^c}{P_x Z_j^c}, \quad y_j^i = \frac{f Y_j^c}{P_y Z_j^c} \tag{2}$$

The scaled orthographic projection

The point p_j^i in figure 1, is the image point of the 3D object point P_j^o that is resulted from the nonlinear perspective projection model; the linear scaled orthographic projection (SOP) model can be used to approximate the previous model to get the image point p_j^i for the same 3d object point P_j^o [6].

$$p_j^i = \begin{bmatrix} x_j^i \\ y_j^i \end{bmatrix} = s \begin{bmatrix} X_j^c \\ Y_j^c \end{bmatrix}, \quad s = \frac{f}{Z_j^c} \tag{3}$$

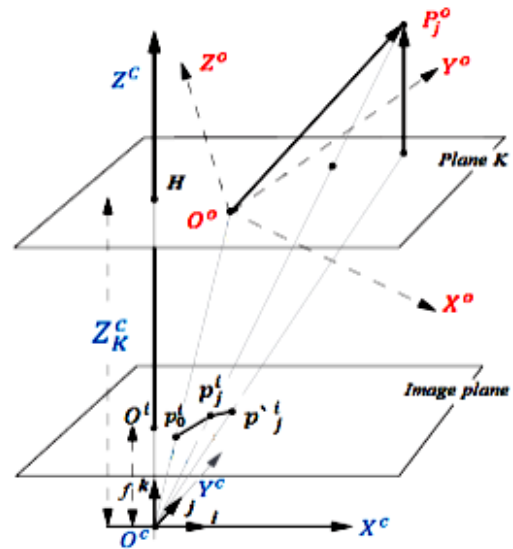


Figure 1: the perspective projection and scaled orthographic projection

CRS catalyst-5t robot kinematics

In this section the CRS Catalyst-5t Robot is described and its parameters required to build the robot simulator whose coordinate frames are illustrated in figure 2 are defined. This robot has 5 revolute joints that control the positioning of the end effector of the robot. Kinematic model of CRS

Catalyst-5t Robot was built using Denavit Hardenberg (DH) transformation matrix representation⁽³⁾, using robot schematic and the axis convention illustrated in figure 2. The kinematic parameters of the robot are the twist angle α_i , link length a_i , link offset d_i and finally the lower and upper joint limits θ_{imin} and θ_{imax} of the joint q_i are given in table 1.

Joint	α_i	a_i	d_i	θ_{imin}	θ_{imax}
0	90	0	.254	-180	180
1	0	.254	0	0	45
2	0	.254	0	0	120
3	-90	0	0	-45	45
4	0	0	0.0508	-75	75

Table 1: Kinematic parameters and joint limits, all angles in degrees, lengths in meters

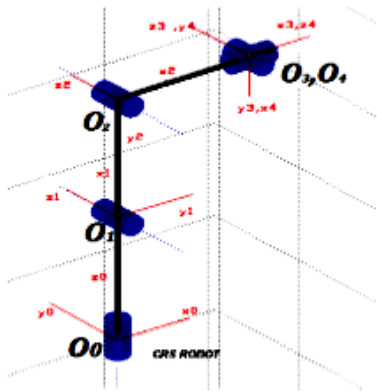


Figure 2: the details of coordinate frames used for robot

Assuming that the transformation from the end-effector frame to the base frame is given by

$$H = H_4^0 = \begin{bmatrix} R_4^0(q) & O_4^0(q) \\ 0 & 1 \end{bmatrix} \quad (3)$$

Where $R_4^0(q)$ is the rotation matrix, $O_4^0(q)$ is the translation vector, and $q = q_0, \dots, q_4$ T is the vector of joint variables.

The velocity relationships between the end effector with respect to the base frame and the joint velocity are determined by the robot jacobian J as shown

$$V = V_4^0 = \begin{bmatrix} v_4^0 \\ \omega_4^0 \end{bmatrix} = J \dot{q} \quad (4)$$

Where the component of V_4^0 are the linear velocity v_4^0 and the angular velocity ω_4^0 of the end-effector with respect to the base frame, \dot{q} is the velocity vector of all joints and J is the jacobian matrix defined by:

$$J = [J_0 \dots J_4] \quad (5)$$

$$J_i = \begin{bmatrix} J_{vt} \\ J_{\omega i} \end{bmatrix} = \begin{bmatrix} z_i \times (O_i^0 - O_0^0) \\ z_i \end{bmatrix} \quad (6)$$

Where J_{vt} , $J_{\omega i}$ are linear and the angular jacobian 3×1 column vectors

$$z_i = R_i^0 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

With $R_i^0 = R_1^0 \cdot \dots \cdot R_i^{i-1}$ and O_i^0 is the translation vector of the transformation matrix H_i^0 .

c. Camera robot configuration

There are two main configurations to combine the camera and the robot on visual servoing, the first configuration is called “eye in hand” in which the camera is mounted rigidly on the robot end effector as shown in figure 3, the transformation H^{EC} between the camera frame and the end effector frame is constant and must be defined to transfer the motion from the camera frame to the end effector frame.

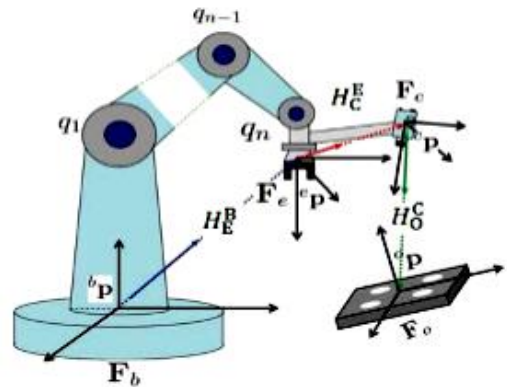


Figure 3: eye in hand configuration with the main frames of the robot set

The second configuration is called “eye to hand” where the camera place in the work space to monitor the object and the robot end effector, in this configuration the transformation HC E between the camera and robot end effector will be changed through the robot motion, so this transformation must be computed at each iteration. In this paper the eye in hand configuration has been used because it provides better viewing possibility with less probability of viewing occlusion due to the moving arm and target.

Pose estimation

Pose estimation is the problem of defining the relative position and orientation of the object with respect to the camera by using the image information and the 3D model geometry. The object feature points P_j is known from the object geometry, image feature coordinates are known from image measurements x_j^i, y_j^i , and the camera intrinsic parameters are defined by camera calibration, by combining equations (1) and (2) it will be clear that defining the vector of the pose parameters $w = [X, Y, Z, \theta, \alpha, \phi]$ is possible. The pose estimation problem will be solved using two techniques the first technique employs orthography and scaling with iteration while the second technique employs extended kalman filter.

1) Pose from orthography and scaling with iteration (POSIT)

In this method, the rotation matrix and the translation vector of the object frame with respect to the camera frame are calculated iteratively⁽⁴⁻⁷⁾; where the calculated approximate pose is converging to the actual pose in a few iterations. This method requires four or more feature points to be used. The rotation matrix is the matrix whose rows are the coordinates of the unit vectors i, j, k of the camera frame expressed in the object frame O^o .

$$R = \begin{bmatrix} i_x^o & i_y^o & i_z^o \\ j_x^o & j_y^o & j_z^o \\ k_x^o & k_y^o & k_z^o \end{bmatrix} \quad (8)$$

Using this rotation matrix the projection of the vector P_j^o on the unit vector i of the camera frame can be obtained by the dot product $(P_j^o \cdot i)$ where i is the first row in rotation matrix,⁽⁷⁾. To compute the rotation matrix, i and j needed to be computed in the object frame, the vector k could be obtained by the cross product $i \times j$. The translation vector T is the vector between the center of projection O^c and the origin O^o , the translation T vector is aligned with the line $O^o P_j^o$ and its value according to SOP is

$$T = \frac{Z_K^C}{f} O^o P_j^o \quad (9)$$

The unit vectors i, j and z component of the translation vector Z_K^C could be computed according to these fundamental equations

$$P_j^o \cdot I = x_j^i (1 + \epsilon_i) - x_j^b \quad (10)$$

$$P_j^o \cdot J = y_j^i (1 + \epsilon_i) - y_j^b \quad (11)$$

$$\epsilon_i = \frac{1}{Z_K^C} P_j^o \cdot K \quad (12)$$

The scaling factor of the projection $S = \frac{S_1 + S_2}{2}$

$$\text{Where } (I, J)^{1/2} = S_1, (J, J)^{1/2} = S_2 \quad (13)$$

The unit vectors i, j, k

$$i = I/S_1, j = J/S_2 \text{ and } k = i \times j \quad (14)$$

The component of the translation in z direction ($Z_K^C = f/s$) that is used to calculate the full translation vector using equation (9). In the first iteration, assume that $\epsilon_i(0) = 0$ and then re-compute the new ϵ_i the iteration until $|\epsilon_i(n) - \epsilon_i(n-1)| < \text{threshold}$. by the end of the iteration process, the rotation matrix will be defined by the final results of unit vector i, j, k and the translation vector will be calculated by substituting the final result of Z_K^C in equation (9).

2) Pose estimation using extended kalman filter

Kalman filter is based on defining the linear state model of the dynamic system and on the measurements defined on the output equation, the system state optimal estimation is based on the accuracy of the system dynamic model, the output model “define the relation between the state vector and the measurements” and on the assumption of representing the measurement noise with zero mean Gaussian noise with a know covariance⁽⁵⁾. The system state vector is chosen to be the relative pose parameters of the object with respect to camera frame and their respective velocities⁽⁶⁾.

$$w = [X, \dot{X}, Y, \dot{Y}, Z, \dot{Z}, \theta, \dot{\theta}, \alpha, \dot{\alpha}, \phi, \dot{\phi}] \quad (15)$$

The system dynamic model is written as

$$w_k = Aw_{k-1} + \gamma_k \quad (16)$$

Where A is block diagonal with 2x2 blocks of the form $\begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$, and γ_k a zero mean Gaussian noise with covariance known covariance Q, t is the time sample period, sample time step

The output model $Z_k = G(w_k) + \theta_k$ Where

$$G(w_k) = f\left[\frac{X_1}{P_x Z_1}, \frac{f Y_1}{P_y Z_1}, \frac{X_2}{P_x Z_2}, \frac{f Y_2}{P_y Z_2}, \frac{X_3}{P_x Z_3}, \frac{f Y_3}{P_y Z_3}, \frac{X_4}{P_x Z_4}, \frac{f Y_4}{P_y Z_4}, \frac{X_5}{P_x Z_5}, \frac{f Y_5}{P_y Z_5}\right] \quad (17)$$

θ_k a zero mean Gaussian noise with covariance known covariance R.

In kalman filter, the output model must be linear to get the optimal solution, the output model used in this case is clearly nonlinear, and so the output model must be linearized about the current state estimates at each sample step. Through the extended kalman filter algorithm, these equations must be applied

Prediction Equations:

$$w_{k,k-1} = A w_{k-1,k-1} \quad (18)$$

$$p_{k,k-1} = A p_{k-1,k-1} A^T + Q_{k-1} \quad (19)$$

Linearization:

$$C_k = \frac{\partial G(w_{k,k-1})}{\partial w_{k,k-1}} \quad (20)$$

Kalman gain:

$$K = p_{k,k-1} C_k^T ((R_k + C_k p_{k,k-1} C_k^T)^{-1}) \quad (21)$$

Estimate Update:

$$w_{k,k} = w_{k,k-1} + K (Z_k - G(w_{k,k-1})) \quad (22)$$

$$p_{k,k} = p_{k,k-1} - K C_k p_{k,k-1} \quad (23)$$

IV. Position based visual servoing control scheme

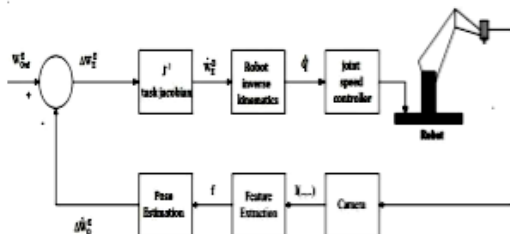


Figure 4: PBVS control scheme

In this section, the end effector motion is controlled in the desired end effector frame, the feedback signal is chosen to be the relative pose of the current end effector frame with respect to the desired end effector frame W_E^{Ed} , the target of this control scheme is to get $W_E^{Ed} = 0[8]$, using the pose

estimation methods, the relative current pose of the object with respect to the camera W_o^C , and the relative desired pose of the object with respect to camera W_o^{Cd} could be obtained. In order to estimate W_E^{Ed} , one must first calculate

$$T_E^{Ed} = (T_E^E \cdot T_o^{Cd}) \cdot (T_E^E \cdot T_o^C)^{-1} \quad (24)$$

Then we can extract the pose vector

$$W_E^{Ed} = [X_E^{Ed}, Y_E^{Ed}, Z_E^{Ed}, \theta_E^{Ed}, \alpha_E^{Ed}, \varphi_E^{Ed}] \quad (25)$$

The feedback signal W_E^{Ed} is converted to the total rotation angle form

$$\hat{W}_E^{Ed} = [X_E^{Ed}, Y_E^{Ed}, Z_E^{Ed}, \theta_{x_E}^{Ed}, \theta_{y_E}^{Ed}, \theta_{z_E}^{Ed}] \quad (26)$$

The control error is given as follows

$$\hat{W}_E^{Ed} = -\hat{W}_E^{Ed} \quad (27)$$

The position based task jacobian is used to convert the motion reference frame from the desired end effector frame to the base frame

$$\hat{W}_E^{Ed} = \begin{bmatrix} R_B^{Ed} & I_{3 \times 3} \\ I_{3 \times 3} & R_B^{Ed} \end{bmatrix} \begin{bmatrix} v_B^Ed \\ \omega_B^Ed \end{bmatrix} = J_{task} \begin{bmatrix} v_B^Ed \\ \omega_B^Ed \end{bmatrix} \quad (28)$$

By using the velocities of the end effector with respect to the base and the robot jacobian calculated in (4) one can get the robot joint velocities that will be sent to the robot controller to move the robot end-effector toward the desired pose.

Simulation the camera parameters:

Focal length=6 mm, inter pixel spacing in x axis, y axis of image plane $px = py = 10-5$, resolution the image = [1280 1024] and principal point of the image = [640 512]

The object 3d co-ordinates in the object frame measured in meter

X^o	-0.5	-0.5	.05	.05	-0.5	-0.5	.05	.05
Y^o	-0.5	.05	.05	-0.5	-0.5	-0.5	.05	-0.5
Z^o	-0.5	-0.5	-0.5	-0.5	.05	-0.5	.05	.05

Pose estimation simulation results

In this simulation, the perspective projection model was used as vision subsystem model that moved through specified path with respect to the object and the sequence of images was captured through this path. Posit algorithm and kalman filter were used to estimate the relative poses of the object with respect to the camera using this sequence of images, the following figures shows the specified relative pose of the object with respect to camera, the estimated relative pose using posit algorithm and the estimated relative pose using kalman filter. The values of kalman filter parameter used in the simulation are $r=1000$ and $q=.01$.

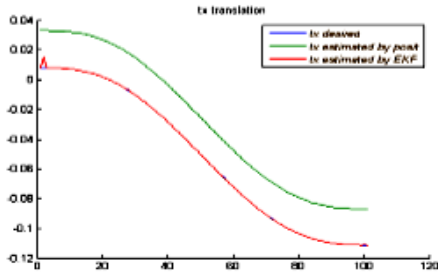


Figure 5: translation in x axis for the desired and estimated relative poses using kalman filter and posit algorithm

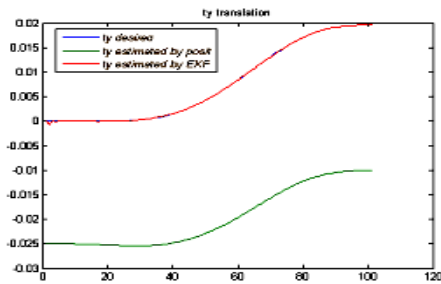


Figure 6: translation in y axis for the desired and estimated relative poses using kalman filter and posit algorithm

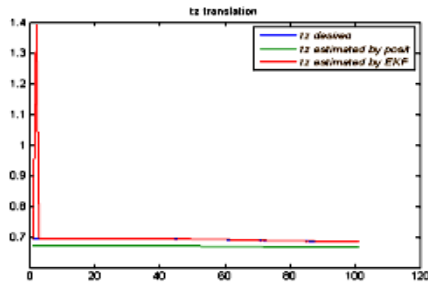


Figure 7: translation in z axis for the desired and estimated relative poses using kalman filter and posit algorithm

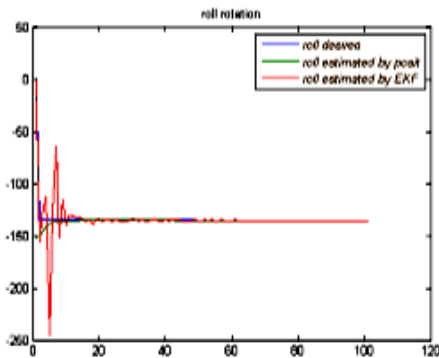


Figure 8: roll angle for the desired and estimated relative poses using kalman filter and posit algorithm

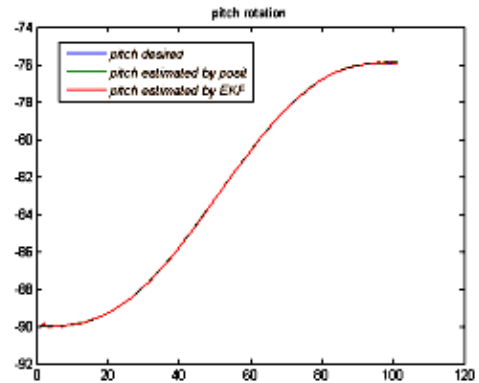


Figure 9: pitch angle for the desired and estimated relative poses using kalman filter and posit algorithm

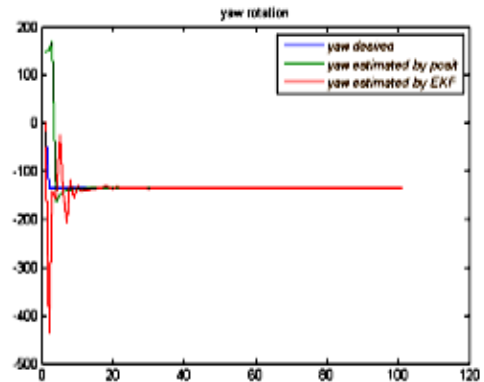


Figure 10: yaw angle for the desired and estimated relative poses using kalman filter and posit algorithm

PBVS simulation results

PBVS simulation is implemented using the CRS Catalyst-5t Robot kinematic model and the perspective projection model (as a vision sub system model) with the previously mentioned specifications. The vision sub system and the robot are combined in eye in hand style.

Posit technique is used as pose estimation module. Firstly the desired pose of the robot end effector is specified where the desired image is taken to specify the desired relative pose between the object and the camera, then the robot end effector is moved to any random pose through the robot work space where the object is maintained in the field of camera view. Camera starts to generate a live sequence to images through the robot motion, the images information used to estimate the current relative pose of the object with respect to camera. Control scheme calculates the error between the desired and the current relative pose and modify the robot motion. The calculated error of relative pose is converged to zero in the six relative pose parameters as shown in the following figures.

The pose estimation function in peter corke toolbox⁽⁸⁾ which based on effective perspective n points method is used instead of Posit method. The following figures shows the behavior of the control scheme using two pose estimation methods.

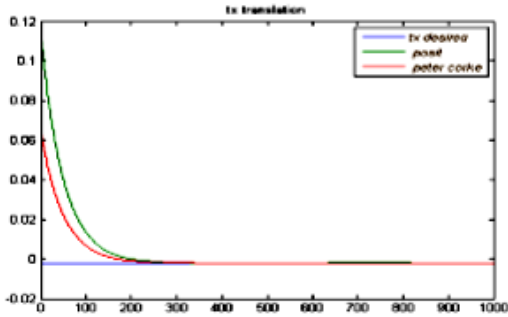


Figure 11: the desired and the current x axis component of the relative translation vector through PBVS

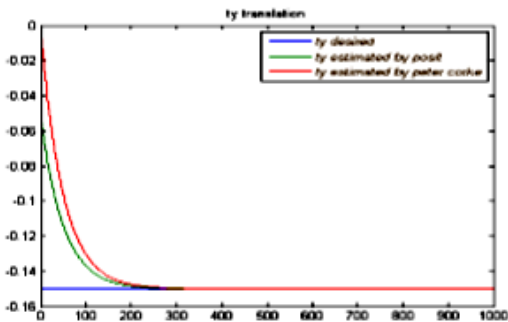


Figure 12: the desired and the current y axis component of the relative translation vector through PBVS

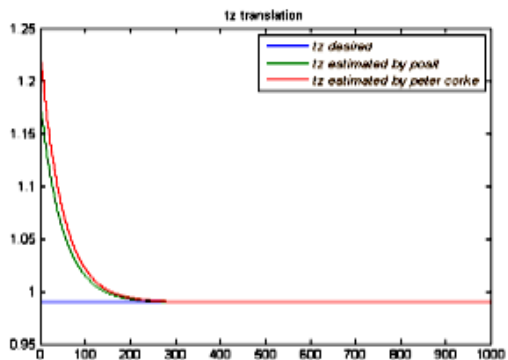


Figure 13: the desired and the current z axis component of the relative translation vector through PBVS

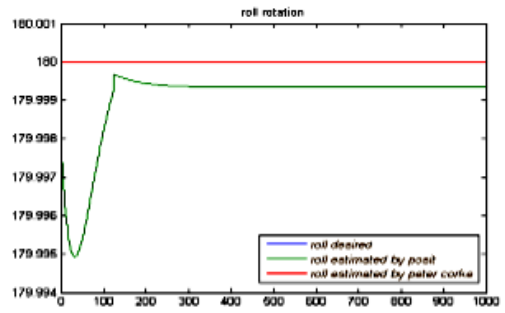


Figure 14: the desired and the current roll angle of the relative pose through PBVS

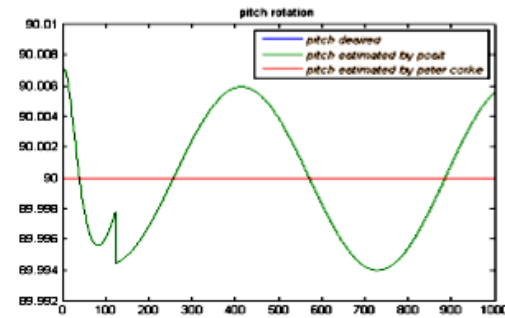


Figure 15: the desired and the current pitch angle of the relative pose through PBVS

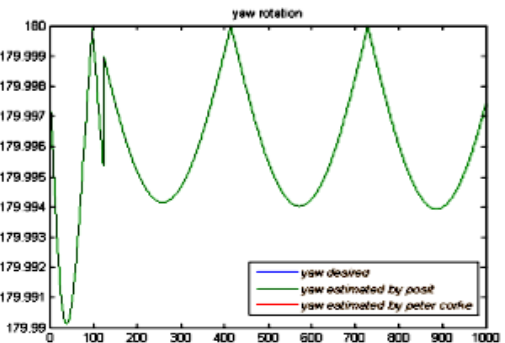


Figure 16: the desired and the current yaw angle of the relative pose through PBVS

Conclusion and Future works Simulation using the posit algorithm shows that this algorithm works perfectly in the estimation of the orientation parameters but it poorly estimates the translation parameters because it is based on the SOP model which is considered an approximation of perspective projection model. Pose estimation using kalman filter shows better results and its accuracy affected with the values of r and q parameters. The pose estimation is the main core of the PBVS system where the better pose estimation technique has the better response in PBVS system. The proposed

future work will be applying the pose estimation using kalman filter and efficient perspective n points techniques on the PBVS control scheme and studying their impact on the system response

References

1. Guiding a Robot by Visual Feedback in Assembling Tasks, Y. SHIRAI and H. INOUE, Electrotechnical Laboratory, Chiyoda-ku, Tokyo, Japan.
2. Opto-Mechatronic Systems Handbook, Ed. H. Cho, CRC Press, Boca Raton, FL, 2002, pp. 15-1—15-24. Visual Servoing: Theory and Applications, F. Janabi-Sharifi, Robotics and Manufacturing Automation Laboratory, Department of Mechanical, Aerospace, and Industrial Engineering, Ryerson University, Toronto, Ontario, M5B 2K3, CANADA.
3. Introduction to robotics, analysis, control, applications handbook, second edition, Saeed Binjamin Niku, mechanical engineering department, California polytechnic, state university, san-luis Obispo.
4. Relative End-Effector Control Using Cartesian position Based Visual Servoing, William J. Wilson, Member, IEEE, Carol C. Williams Hulls, Student Member, IEEE, and Graham S. Bell.
5. A tutorial on visual servo control, Sith Hutchinson, member IEEE, Gregory D. Hager, member IEEE and Peter I. Corke, member, IEEE.
6. Model Based Object Pose in 25 Lines of Code, Daniel F. DeMenthon and Larry S. Davis, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD20742
7. Comparison of Basic Visual Servoing Methods, Farrokh Janabi-Sharifi, Senior Member, IEEE, Lingfeng Deng, and William J. Wilson, Member, IEEE.
8. Robotics, Vision and Control Fundamental Algorithms in MATLAB springer handbook, Peter I. Corke, member, IEEE