M Saedan and M H Ang Jr, “3D Vision-Based Control of an Industrial Robot”, Proceedings of the IASTED International Conference on Robotics and Applications, Nov 19-22, 2001, Florida, USA, pp. 152-157.

3D Vision-Based Control On An Industrial Robot

Mana Saedan and Marcelo H. Ang Jr.*
Department of Mechanical Engineering
National University of Singapore, Singapore 119260
*mpeangh@nus.edu.sg

Abstract

This paper investigates the relative target-object (rigid body) pose estimation for vision-based control. A closed-form target pose estimation algorithm is developed and implemented. Moreover, PI-based visual control was designed and implemented in the camera (sensor) frame to minimize the effect of errors in the extrinsic parameters of the camera. The performance of the vision-based control algorithm has been veri£ed on a 7-DOF industrial robot.

1. Introduction

Industrial robots are designed for tasks such as pick and place, welding, and painting. The environment and the working conditions for those tasks are well set. If the working condition changed, those robots may not be able to work properly. Therefore, external sensors are necessary to enhance the robot’s capability to work in a dynamic environment. A vision sensor is an important sensor that can be used to extend the robot’s capabilities. The images of objects of interest can be extracted from their environment, then information from these images can be computed to control the robot. The control that uses the images as feed back signals is known as vision-based control. Recently, vision-based control has became a major research £eld in robotics.

Vision-based control[1] can be classi£ed into two main categories. The £rst approach, feature based visual control, uses image features of a target object from image (sensor) space to compute error signals directly. The error signals are then used to compute the required actuation signals for the robot.The control law is also expressed in the image space. Many researchers in this approach use a mapping function (called the image Jacobian) from the image space to the Cartesian space. The image Jacobian, generally, is a function of the focal length of the lens of the camera, depth (distance between camera (sensor) frame and target features), and the image features. In contrast, the position-based visual control constructs the spatial relationship, *target pose*[2], between the camera frame and the target object frame from target image features. Many construction algorithms have been proposed. Each algorithm has different assumptions and limitations.

[1]Some researchers use the term *visual servo control*.
[2]The position and orientation of target-object.

There are numbers of works on those two approaches. Feddema et al. [1], Hashimoto et al. [2] [3], and Papanikolopoulos et al. [4] are some of interesting works on the feature-based approach. In the position-based approach Chaumette et al. [5], Wilson and colleagues [6] [7] [8], and Martinet and Gallice [9] reported the works on position-based approaches that could achieve the same performance as feature-based approaches.

In this paper, a position-based approach is presented. The advantage of this approach is that the servo control structure is independent from the target pose reconstruction. Usually, the desired control values is speci£ed in the Cartesian space, so they are easy to visualize. One main issue on position-based approach is target pose reconstruction. To construct the pose of a target object from two-dimension image feature points, two cameras are needed. Image feature points in each of the two images have to be matched and 3-D information of the coordinates of the target object and its feature points can then be computed by triangulation. One-camera systems can, however, determine 3-D information if the geometry of the target object is known beforehand. The distance between the feature points in the target object, for example, can be used to help compute the 3-D position and orientation of the target with respect to the camera.

Several estimation methods were proposed using different techniques. Moving a camera to different positions (and orientations) can extract the depth information from target image without knowledge of the actual target object geometry (e.g., dimensions). This method, however, has a signi£cant depth estimation error. To reduce the error many different camera positions should be used. Thus, the method is not suitable for tracking a moving object. Another method, which was proposed by Wilson et al. [8], derives the relationship between target pose and image feature points in a recursive form; based on the assumption that the actual target object features, i.e. position of feature point with respect to the target frame, are known. The target pose can be estimated by using Kalman £lter. This method gives an accurate estimation when the vision system can be operated at high sampling rate, e.g. 61 Hz [8]. The main disadvantage of Wilson’s method is the plant error covariance is needed in the Kalman £lter and this is not easy to identify. In addition, the plant error covariance can only be estimated for some speci£c cases. Hashimoto et al. [2] used the closed-form pose estimation method to £nd the

depth information, which is needed for the image Jacobian. The estimation method is based on the assumption that the camera parameters and the position and orientation of the target feature(s) with respect to the target frame are known. Although the closed form estimation may be sensitive to noise, the robustness of this method can be improved by using redundant feature points.

The vision-based control in our work is implemented on the Mitsubishi PA-10 robot. The camera is mounted on the end-effector of the robot, i.e., an eye-in-hand con£guration. The closed-form target pose estimation is discussed and used in the position-based visual control. The control system consists of two control loops. The outer loop is the visual control, and the inner is the robot servo control. The visual control is expressed in the camera frame, therefore the system can tolerate errors in calibrations such as the eye-hand relationship (the relationship between the camera frame and the end-effector frame of a robot). This is very useful when the eye-hand relationship cannot be calibrated precisely. In the following sections, we discuss details of our implementation and present the experimental results.

2. Closed-Form Target Pose Estimation

The pinhole camera model (Fig. 1) is used for relating the object (spatial) space to the image space (2 dimensional space). Assuming all distortion effects, quantization effects and blurring effects are negligible, the transformation from the object space space to the image space can be written in a matrix form as

$$\begin{bmatrix} {}^{I}x_i w_i \\ {}^{I}y_i w_i \\ w_i \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & 0 & 0 \\ 0 & \alpha_y & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & f \\ 0 & 0 & -1 & f \end{bmatrix} \begin{bmatrix} {}^{C}X_i \\ {}^{C}Y_i \\ {}^{C}Z_i \\ 1 \end{bmatrix} \tag{1}$$

where $({}^{I}x_i, {}^{I}y_i)$ is the coordinate of the object point *i* in the image space, w_i is a scaling factor that can be cancelled when calculating ${}^{I}x_i$ and ${}^{I}y_i$, $({}^{C}X_i, {}^{C}Y_i, {}^{C}Z_i)$ is the coordinate of the object point *i* in the object space, α_x and α_y are the pitch length of the image pixel in x-axis and y-axis of a camera, respectively.

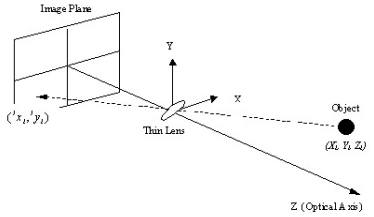

Figure 1. Pinhole camera model.

Equation 1 can be simpli£ed to:

$$\begin{aligned} {}^{I}x_i &= \alpha_x \left(\frac{f}{f - {}^{C}Z_i}\right) {}^{C}X_i \\ {}^{I}y_i &= \alpha_y \left(\frac{f}{f - {}^{C}Z_i}\right) {}^{C}Y_i. \end{aligned} \tag{2}$$

Thus, the inverse transformation from the image space to the object space can be written as

$$\begin{aligned} {}^{C}X_i &= {}^{I}x^*{}_i \left(\frac{f - {}^{C}Z_i}{f}\right) \\ {}^{C}Y_i &= {}^{I}y^*{}_i \left(\frac{f - {}^{C}Z_i}{f}\right) \end{aligned} \tag{3}$$

where ${}^{I}x^*{}_i = \frac{{}^{I}x_i}{\alpha_x}$ and ${}^{I}y^*{}_i = \frac{{}^{I}y_i}{\alpha_y}$.

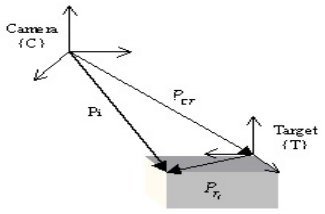

Figure 2. Target and camera coordinate frame.

The target pose estimation aims to reconstruct a spatial relationship between the target object and the camera. We refer to points in a target object that are used in the target pose estimation as *feature points*. The points in the object space are called *target feature points* and the points that are transformed to the image space are called *image feature points*.

Suppose that the target object is a rigid body with known geometry, i.e., its shape and size are known. A target feature point *i*, in Figure 2, can be transformed from the target frame ${}^{T}\mathbf{P}_i$ to the camera frame ${}^{C}\mathbf{P}_i$ as

$${}^{C}\mathbf{P}_i = {}^{C}\mathbf{R}_T {}^{T}\mathbf{P}_i + {}^{C}\mathbf{P}_T$$

$$\begin{bmatrix} {}^{C}X_i \\ {}^{C}Y_i \\ {}^{C}Z_i \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} {}^{T}X_i \\ {}^{T}Y_i \\ {}^{T}Z_i \end{bmatrix} + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{4}$$

In our work, we use a target with all target feature points on a plane. We de£ne the target plane such that all target feature points are in the XY-plane, thus ${}^{T}Z_i = 0$. Substituting Equation 3 into Equation 4, the simpli£ed equations are:

$$\begin{aligned} f\,{}^{T}X_i \frac{R_{11}}{f - Z} + f\,{}^{T}Y_i \frac{R_{12}}{f - Z} + {}^{i}x^*{}_i\,{}^{T}X_i \frac{R_{31}}{f - Z} \\ + {}^{i}x^*{}_i\,{}^{T}Y_i \frac{R_{32}}{f - Z} + f\frac{X}{f - Z} = {}^{i}x^*{}_i \\ f\,{}^{T}X_i \frac{R_{21}}{f - Z} + f\,{}^{T}Y_i \frac{R_{22}}{f - Z} + {}^{i}y^*{}_i\,{}^{T}X_i \frac{R_{31}}{f - Z} \\ + {}^{i}y^*{}_i\,{}^{T}Y_i \frac{R_{32}}{f - Z} + f\frac{Y}{f - Z} = {}^{i}y^*{}_i \end{aligned} \tag{5}$$

Equation 5 can be rearranged in a matrix form

$\mathbf{A}_i\bar{\mathbf{M}} = \mathbf{B}_i$, where

$$\mathbf{A}_i = \begin{bmatrix} {}^fX_i & {}^fY_i & 0 & 0 & {}^{i}x^*_i{}^TX_i & {}^{i}x^*_i{}^TY_i & f & 0 \\ 0 & 0 & {}^fX_i & {}^fY_i & {}^{i}y^*_i{}^TX_i & {}^{i}y^*_i{}^TY_i & 0 & f \end{bmatrix},$$

$$\bar{\mathbf{M}} = \begin{bmatrix} \frac{R_{11}}{f-Z} & \frac{R_{12}}{f-Z} & \frac{R_{21}}{f-Z} & \frac{R_{22}}{f-Z} & \frac{R_{31}}{f-Z} & \frac{R_{32}}{f-Z} & \frac{X}{f-Z} & \frac{Y}{f-Z} \end{bmatrix}^T,$$

and

$$\mathbf{B}_i = \begin{bmatrix} {}^Ix^*_i & {}^Iy^*_i \end{bmatrix}^T. \tag{6}$$

It can be seen from Equation 6 that at least four points (8 equations) are needed to solve for a solution $\bar{\mathbf{M}}$ (which has 8 unknowns/elements). By stacking each point together, the solution can be determined as

$$\bar{\mathbf{M}} = \mathbf{A}^{-1}\mathbf{B} \tag{7}$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix}^T$ and $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{B}_4 \end{bmatrix}^T$.

Although only four feature points can be used to solve for the solution, the result $\bar{\mathbf{M}}$ from Equation 7 may be inaccurate when the feature points data from the image are corrupted by noise (due to the image quantization errors for example). To overcome this problem, redundant feature points are introduced in order to reduce the pose estimation error. Data from the additional redundant feature points are stacked together and correspondingly appended to matrices $\mathbf{A}$ and $\mathbf{B}$ The solution is calculated by the least squares method:

$$\bar{\mathbf{M}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}. \tag{8}$$

This solution minimizes the squares of errors.

Once $\bar{\mathbf{M}}$ is computed, the orthogonal property of the rotation matrix is used to solve for the value of Z:

$$\begin{aligned} R_{11}^2 + R_{21}^2 + R_{31}^2 = 1 \text{ and} \\ R_{12}^2 + R_{22}^2 + R_{32}^2 = 1. \end{aligned} \tag{9}$$

The magnitude of Z must be much greater than the focal length of a lens (f) and Z must always be a positive value. Using Equation 9, therefore, *Z* can be computed as

$$\begin{aligned} Z = f + \frac{1}{\sqrt{\bar{M}_1^2 + \bar{M}_3^2 + \bar{M}_5^2}} \text{ or} \\ Z = f + \frac{1}{\sqrt{\bar{M}_2^2 + \bar{M}_4^2 + \bar{M}_6^2}} \end{aligned} \tag{10}$$

where $\bar{M}_i$ is the ith element of vector $\bar{\mathbf{M}}$ (Equation 6). The magnitude of *Z* from Equation 10 are not always equal, hence the average value of *Z* (from those two values above) can be used to determine any other values, i.e. X, Y, and rotational matrix $\mathbf{R}$.

3. Position-Based Visual Control Design

The frame assignment for vision-based control system is depicted in Figure 3 where $\{0\}$ represents the base frame,

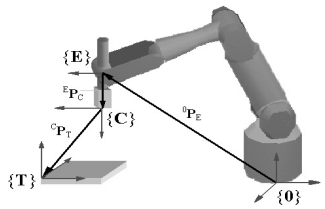

Figure 3. Frame assignment for vision-based control.

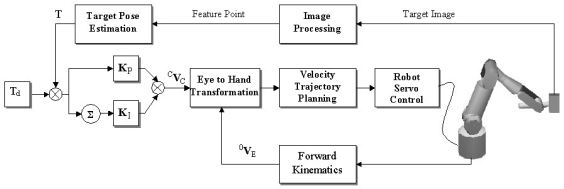

Figure 4. Visual control system structure.

$\{\mathrm{E}\}$ represents the current end-effector frame, $\{\mathrm{C}\}$ represents the current camera frame, $\{E_d\}$ represents the desired end effector frame, $\{C_d\}$ represents the desired camera frame, and $\{\mathrm{T}\}$ represents target frame. Since a camera is mounted on the robot end-effector, the homogenous transformation matrix from the camera frame to the end effector frame, ${}^ET_C$, is constant.

Although the desired target pose can be computed directly from the kinematic model, the performance of the vision system may not be stable when the eye-hand relationship is signi£cantly inaccurate. This paper, therefore, uses the PI control scheme in the visual control loop (as depicted in Figure 4) to stabilize the vision system. The PI controller generates velocity commands for the camera frame with respect to the camera frame itself. The required velocities are computed from the errors between the desired target pose and the estimated target pose. Advantages of PI control scheme are its simplicity for implementation and non-computational intensive nature. The use of this outer loop control makes the visual control robust to noise from image data.

The velocities of the camera frame ${}^C\mathbf{U}_C$ can be generated by a PI control law as

$${}^C\mathbf{U}_C = \mathbf{K}_P\mathbf{e}_t + \mathbf{K}_I\sum\mathbf{e}_t \tag{11}$$

where ${}^C\mathbf{U}_C$ represents the linear and an angular velocities of the camera frame with respect to the camera frame itself, $\mathbf{e}_t$ is a vector of position and orientation errors of the target frame with respect to the camera frame at each sampling period. $\mathbf{K}_P$ and $\mathbf{K}_I$ are proportional and integral gain matrices, respectively. The velocities generated represent the velocity commands to the robot.

The velocity commands that are generated from the visual controller in Equation 11 can be transformed to the

robot end-effector frame as

$$
\begin{aligned}
{}^0\boldsymbol{\omega}_E &= {}^0\mathbf{R}_E\,{}^E\mathbf{R}_C\,{}^C\boldsymbol{\omega}_C \\
{}^0\mathbf{V}_E &= {}^0\mathbf{R}_E\,{}^E\mathbf{R}_C\,{}^C\mathbf{V}_C - \mathbf{S}({}^0\boldsymbol{\omega}_E)\,{}^0\mathbf{R}_E\,{}^E\mathbf{P}_C
\end{aligned} \tag{12}
$$

where ${}^0\mathbf{R}_E$ is the orientation matrix of the robot end-effector frame with respect to the base frame, ${}^E\mathbf{R}_C$ and ${}^E\mathbf{P}_C$ are orientation and position of the camera frame with respect to the robot end-effector frame, $\mathbf{S}({}^0\boldsymbol{\omega}_E)$is an angular velocity screw matrix:

$$
\mathbf{S}({}^0\boldsymbol{\omega}_E) = \begin{bmatrix} 0 & -{}^0\omega_{E_Z} & {}^0\omega_{E_Y} \\ {}^0\omega_{E_Z} & 0 & -{}^0\omega_{E_X} \\ -{}^0\omega_{E_Y} & {}^0\omega_{E_X} & 0 \end{bmatrix}
$$

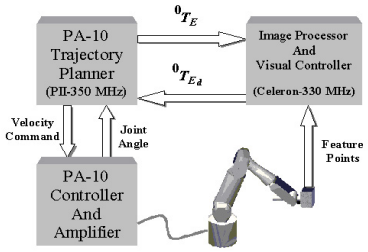

Figure 5. Hardware con£guration for vision-based control system.

4. Experiment Results

Two sets of personal computers were used in our vision-based control system, as shown in Figure 5. Both computers communicate though TCP/IP network system. The £rst computer, PII-300 MHz, was used for planning the trajectory sent to the PA-10 robot controller at a trajectory update period of 1.5 ms. The second computer, Celeron-300 MHz, was used for image processing and visual controller calculations.

The Mitsubishi PA-10 robot, which has seven rotating joints, was used in this experiment. The PA-10 robot servo control scheme was chosen to be velocity control. In this mode, the robot is moved by giving velocity commands to each joint. (The robot servo control scheme uses resolved motion rate control to compute the joint motion from the task-based velocity commands.) The camera was mounted on the end-effector of the PA-10 robot. Before conducting the experiment, the eye-hand calibration was carried out using the algorithm in [10].

The target object, in this work, consisted of nine black dots on white paper but only eight dots were used in the target pose estimation. The center of area of each dot was used as a feature point by processing the whole image inside a camera view. The resolution of the image was 300×220 pixels. The achievable frame rate (including image processing time) was about 20 frames per second.

The visual controller gains, $(K)_P$ and $(K)_I$, were manually tuned to optimize the response of the whole system. Keeping the target object £xed, the proportional and integral gains were tuned until the system can be driven to steady state as fast as possible, while minimizing the overshoot response.

Stationary Target Object

In this experiment, the robot was tracking a £xed target object. The response of the robot was observed while adding the error to the relationship between the camera frame and the robot end-effector. The results were compared with those results using pure kinematics calculations alone from the relative target-object pose estimation. The desired target pose was £xed at the position $(x, y, z) = (0, 0, 300)$ millimeter and the orientation $(Row, Pitch, Yaw) = (0, 0, 0)$. The tracking errors were computed as the difference between the desired target pose and the estimated target pose.

The results when using kinematic model to calculate the desired target pose are shown in Figures 6 and 7. It can be seen that the accuracy of the eye-hand relationship can affect the performance of the vision system. The visual control is implemented to control the position and orientation of the target in the camera frame. It has the same advantage as the feature-based approach that the tracking performance robust to the error of the eye-hand relationship. The results in Figures 8 and 9 show that the tracking performance is not affected even when the eye-hand relationship is not accurate.

Moving Target Object

The target was moved by human hands to verify the capability of the target pose estimation and the control algorithm. The robot hand is to maintain a constant position and of $(0, 0, 300)$ mm and orientation of (Row, Pitch, Yaw) = $(0, 0, 0)$ relative to the target.

The tracking errors, depicted in Figs. 10 and 11, show that the maximum position tracking error is 35.5 millimeter in X-direction of the camera frame and the maximum orientation error is 16.5 degree in pitch-angle of the camera frame. The video clip of this experiment can be downloaded from [11] .

5. Conclusions

The implementation of the closed-form target pose estimation has been presented. The capability of the target pose estimation method gives three-dimensional information on the relationship between the target and the camera with acceptable accuracy. Furthermore, the position-based visual control has been implemented. The control signals are expressed in the camera (sensor) frame. Consequently the control system is robust to errors in robot kinematics and eye-hand calibration errors. Moreover, the control design is expressed in the Cartesian/task space and this allows speci£cation of the desired target pose naturally without lost of visualization.

References

[1] J. T. Feddema, C. S. G. Lee, and O. R. Mittchell, "Weighted selection of image feature for resolve rate visual feedback control," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 1, pp. 2267–2272, 1991.

[2] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," *IEEE Intl. Conf. on Robotics and Automation*, pp. 2267–2272, 1991.

[3] K. Hashimoto, T. Ebine, and H. Kimura, "Visual servoing with hand-eye manipulator-optimal control approach," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 766–774, Oct. 1996.

[4] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moveing target by a camera mounted on a robot: A combination of control and vision," *IEEE Trans. on Robotics and Automation*, vol. 9, no. 1, pp. 14–35, 1993.

[5] F. Chaumette, P. Rives, and B. Espiau, "Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing," *IEEE Int. Conf. on Robotics and Automation*, pp. 2248–2253, 1991.

[6] D. B. Westmore and W. J. Wilson, "Direct dyanmic control of a robot using an end-point mounted camera and kalman £lter position estimation," *IEEE Intl. Conf. on Robotics and Automation*, pp. 2376–2384, 1991.

[7] J. Wang and W. J. Wilson, "3d relative position and orientation estimation using kalman £lter for robot control," *IEEE Intl. Conf. for Robotics and Automation*, pp. 2638–2644, 1992.

[8] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.

[9] P. Martinet and J. Gallice, "Position based visual servoing using a non-linear approach," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 531–535, 1999.

[10] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and ef£cient 3d robotics hand/eye calibration," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 3, pp. 345–357, 1989.

[11] M. Saedan and M. H. Ang Jr., "http://guppy.mpe.nus.edu.sg/∼mpeangh/vision," 2001.

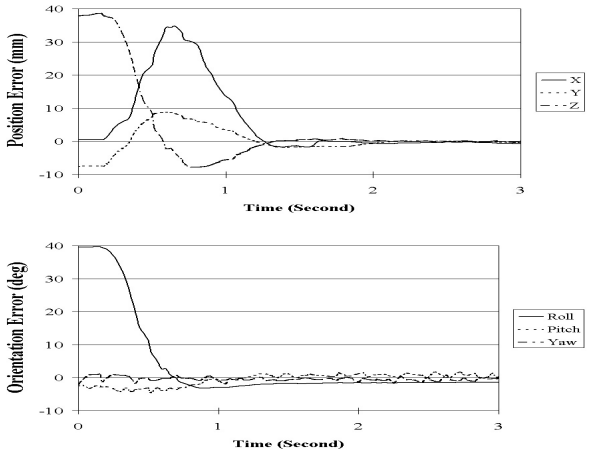

Figure 6. Relative target pose error when using kinematics calculation: accurate eye-hand relationship.

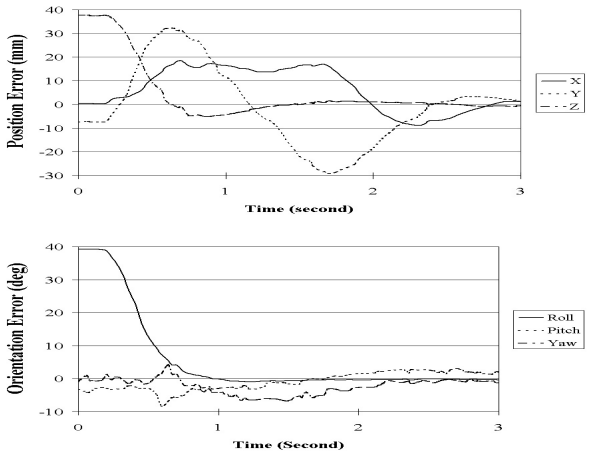

Figure 7. Relative target pose error when using kinematics calculation: inaccurate eye-hand relationship.

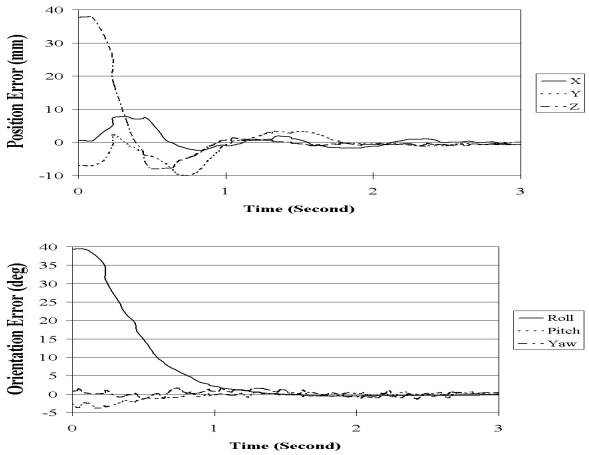

Figure 8. Relative target pose error when using PI control in visual control loop: accurate eye-hand relationship.

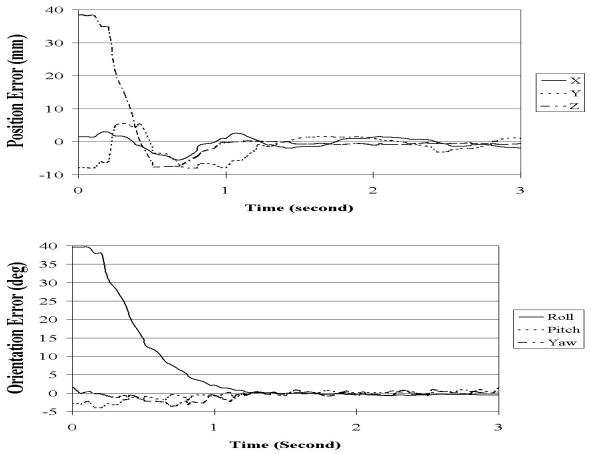

Figure 9. Relative target pose error when using PI control in visual control loop: inaccurate eye-hand relationship.

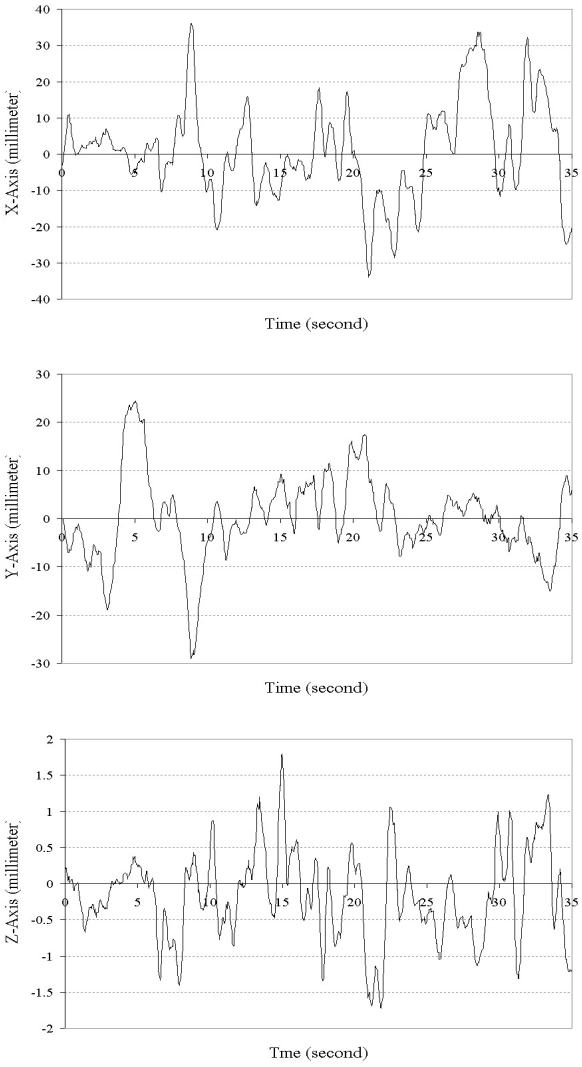

Figure 10. Target relative position error when moving the target.

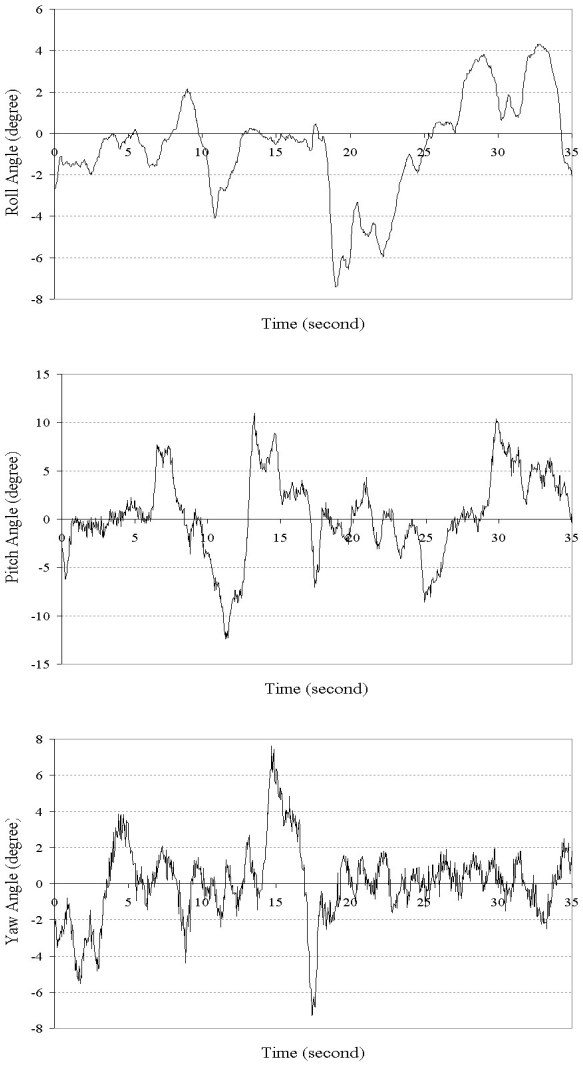

Figure 11. Target relative orientation error when moving the target.