14

Simulation of Visual Servoing Control and Performance Tests of 6R Robot Using Image-Based and Position-Based Approaches

M. H. Korayem and F. S. Heidari
Robotic Research Laboratory, College of Mechanical Engineering, Iran University of Science & Technology, Tehran
Iran

1. Introduction

Visual control of robots using vision system and cameras has appeared since 1980's. Visual (image based) features such as points, lines and regions can be used to, for example, enable the alignment of a manipulator / gripping mechanism with an object. Hence, vision is a part of a control system where it provides feedback about the state of the environment. In general, this method involves the vision system cameras snapping images of the target-object and the robotic end effector, analyzing and reporting a pose for the robot to achieve. Therefore, 'look and move' involves no real-time correction of robot path. This method is ideal for a wide array of applications that do not require real-time correction since it places much lighter demands on computational horsepower as well as communication bandwidth, thus having become feasible outside the laboratory. The obvious drawback is that if the part moves in between the look and move functions, the vision system will have no way of knowing this in reality this does not happen very often for fixture parts. Yet another drawback is lower accuracy; with the 'look and move' concept, the final accuracy of the calculated part pose is directly related to the accuracy of the 'hand-eye' calibration (offline calibration to relate camera space to robot space). If the calibration were erroneous so would be the calculation of the pose estimation part.

A closed–loop control of a robot system usually consists of two intertwined processes: tracking pictures and control the robot's end effector. Tracking pictures provides a continuous estimation and update of features during the robot or target-object motion. Based on this sensory input, a control sequence is generated.

Y. Shirai and H. Inoue first described a novel method for 'visual control' of a robotic manipulator using a vision feedback loop in their paper. Gilbert describes an automatic rocket-tracking camera that keeps the target centered in the camera's image plane by means of pan/tilt controls (Gilbert et al., 1983). Weiss proposed the use of adaptive control for the non-linear time varying relationship between robot pose and image features in image-based servoing. Detailed simulations of image-based visual servoing are described for a variety of manipulator structures of 3-DOF (Webber &.Hollis, 1988).

Source: Vision Systems: Applications, ISBN 978-3-902613-01-1
Edited by: Goro Obinata and Ashish Dutta, pp. 608, I-Tech, Vienna, Austria, June 2007

Mana Saedan and M. H. Ang worked on relative target-object (rigid body) pose estimation for vision-based control of industrial robots. They developed and implemented an estimation algorithm for closed form target pose (Saedan & Marcelo, 2001).
Image based visual controlling of robots have been considered by many researchers. They used a closed loop to control robot joints. Feddema uses an explicit feature-space trajectory generator and closed-loop joint control to overcome problems due to low visual sampling rate. Experimental work demonstrates image-based visual servoing for 4-DOF (Kelly & Shirkey, 2001). Rives et al. describe a similar approach using the task function method and show experimental results for robot positioning using a target with four circle features (Rives et al. 1991). Hashimoto et al. present simulations to compare position-based and image-based approaches (Hashimoto et al., 1991).
Korayem et al. designed and simulated vision based control and performance tests for a 3P robot by visual C++ software. They minimized error in positioning of end effector and they analyzed the error using ISO9283 and ANSI-RIAR15.05-2 standards and suggested methods to improve error (Korayem et al., 2005, 2006). A stationary camera was installed on the earth and the other one mounted on the end effector of robot to find a target. This vision system was designed using image-based-visual servoing. But the vision-based control in our work is implemented on 6R robot using both IBVS and PBVS methods. In case which cameras are mounted on the earth, i.e., the cameras observe the robot the system is called “out-hand" (the term “stand-alone" is generally used in the literature) and when one camera is installed on the end effector configuration is “in-hand”. The closed-form target pose estimation is discussed and used in the position-based visual control. The advantage of this approach is that the servo control structure is independent from the target pose coordinates and to construct the pose of a target-object from two-dimension image plane, two cameras are used. This method has the ability to deal with real-time changes in the relative position of the target-object with respect to robot as well as greater accuracy.
Collision detection along with the related problem of determining minimum distance has a long history. It has been considered in both static and dynamic (moving objects) versions. Cameron in his work mentioned three different approaches for dynamic collision detection (Cameron, 1985, 1986). Some algorithms such as Boyse's and then Canny's solve the problem for computer animation (Boyse, 1979) and (Canny, 1986); while others do not easily produce the exact collision points and contact normal direction for collision response (Lin, 1993). For curved objects, Herzen etc have described a general algorithm based on time dependent parametric surfaces (Herzen et al. 1990). Gilbert et al. computed the minimum distance between two convex objects with an expected linear time algorithm and used it for collision detection (Gilbert & Foo, 1990). Collision detection along with the related problem of determining minimum distance has a long history. It has been considered in both static and dynamic (moving objects) versions. Cameron in his work mentioned three different approaches for dynamic collision detection. He mentioned three different approaches for dynamic collision detection. One of them is to perform static collision detection repetitively at each discrete time steps (Cameran & Culley, 1986).
Using linear-time preprocessing, Dobkin and Kirkpatrick were able to solve the collision detection problem as well as compute the separation between two convex polytopes in O(log|A|.log|B|) where A and B are polyhedra and |.| denotes the total number of faces (Canny, 1986). This approach uses a hierarchical description of the convex objects and

extension of their previous work (Lin, 1993). This is one of the best-known theoretical bounds.

Some algorithms such as Boyse's and then Canny's solve the problem for computer animation (Gilbert & Foo, 1990); while others do not easily produce the exact collision points and contact normal direction for collision response (ANSI/RIA R15.05-2, 2002). For curved objects, Herzen et al. have described a general algorithm based on time dependent parametric surfaces (ISO9283). Gilbert et al. computed the minimum distance between two convex objects with an expected linear time algorithm and used it for collision detection (Ponmagi et al.).

The technique used in our work is an efficient simple algorithm for collision detection between links of 6R robot undergoing rigid motion ,determines whether or not two objects intersect and checks if their centers distance is equal to zero or not.

Due to undefined geometric shape of the end effector of the robot we have explained and used a color based object recognition algorithm in simulation software to specify and recognize the end effector and the target-object in image planes of the two cameras. In addition, capability and performance of this algorithm to recognize the end effector and the target-object and to provide 3D pose information about them are shown.

In this chapter the 6R robot that is designed and constructed in IUST robotic research Lab, is modeled and simulated. Then direct and inverse kinematics equations of the robot are derived and simulated. After discussing simulation software of 6R robot, we simulated control and performance tests of robot and at last, the results of tests according to ISO9283 and ANSI-RIAR15.05-2 standards and MATLAB are analyzed.

2. The 6R robot and simulator environment

This 6 DOFs robot, has 3 DOF at waist, shoulder and hand and also 3 DOF in it's wrist that can do roll, pitch and yaw rotations (Figure 1). First link rotates around vertical axis in horizontal plane; second link rotates in a vertical plane orthogonal to first link's rotation plane. The third link rotates in a plane parallel to second link's rotation plane.

The 6R robot and its environment have been simulated in simulator software, by mounting two cameras in fixed distance on earth observing the robot. These two cameras capture images from robot and it's surrounding, after image processing and recognition of target-object and end effector, positions of them are estimated in image plane coordinate, then visual system leads the end effector toward target. However, to have the end effector and target-object positions in global reference coordinate, the mapping of coordinates from image plan to the reference coordinates is needed. However, this method needs camera calibration that is non-linear and complicated. In this simulating program, we have used a neural network instead of mapping. Performance tests of robot are also simulated by using these two fixed cameras.

3. Simulator software of the 6R robot

In this section, the simulation environment for the 6R robot is introduced and its capability and advantages with respect to previous versions are outlined. This simulator software is designed to increase the efficiency and performance of the robot and predict its limitation and deficiencies before experiments in laboratory. In these packages by using a designed interface board, rotation signals for joints to control the robot are sent to it.

To simulate control and test of 6R robot, the object oriented software Visual C++6 was used. This programming language is used to accomplish this plan because of its rapidity and easily changed for real situation in experiments. In this software, the picture is taken in bitmap format through two stationary cameras, which are mounted on the earth in the capture frame module, and the image is returned in form of array of pixels. Both of the two cameras after switching the view will take picture. After image processing, objects in pictures are saved separately, features are extracted and target-object and end effector will be recognized among them according to their features and characteristics. Then 3D position coordinates of target-object and end effector are estimated. After each motion of joints, new picture is taken from end effector and this procedure is repeated until end effector reach to target-object.

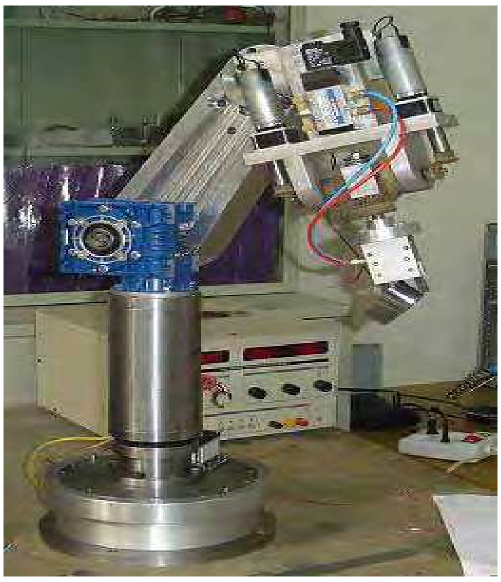


Figure 1. 6R robot configuration

With images from these two fixed cameras, positions of objects are estimated in image plane coordinate, usually, to transform from image plan coordinates to the reference coordinates system, mapping and calibrating will be used. In this program, using the mapping that is a non-linear formulation will be complicated and time consuming process so a neural network to transform these coordinates to global reference 3D coordinate has been designed and used. Mapping system needs extra work and is complicated compared to neural network. Neural networks are used as nonlinear estimating functions. To compute processing matrix, a set of points to train the neural system has been used. This collection of points are achieved by moving end effector of robot through different points which their coordinates in global reference system are known and their coordinates in image plane of the two cameras are computed in pixels by vision module in simulator software. The position of the end effector is recognized at any time by two cameras, which are stationary with a certain distance from each other. The camera No.1 determines the target coordinates in a 2-D image plan in pixels. The third coordinate of the object is also computed by the second camera.

A schematic view of simulator software and the 6R robot in its home position is depicted in Figure 2. In this figure, 6R robot is in homeposition and target-object is the red sphere. The aim of control process is guiding the end effector to reach the target-object within an acceptable accuracy.

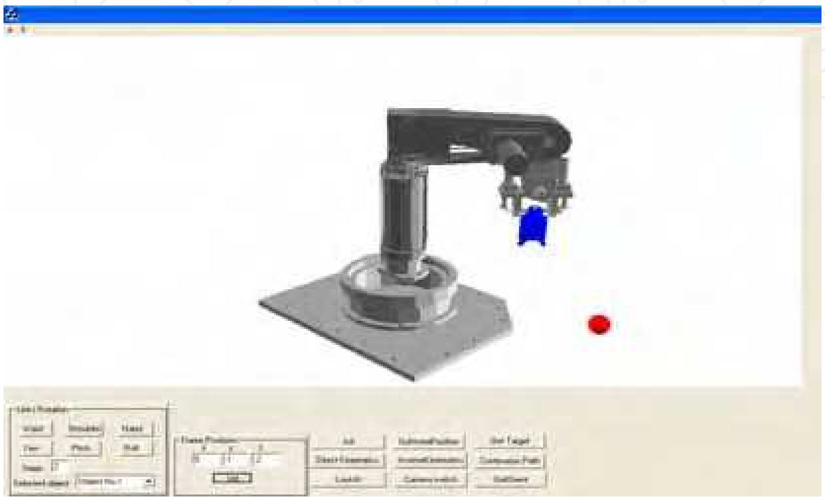
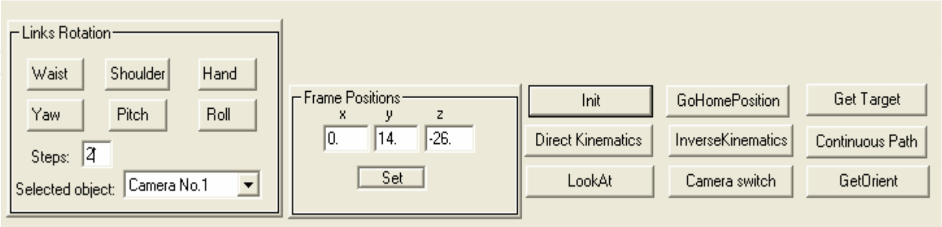


Figure 2. Schematic view of simulator software designed for 6R robot

In this software, not only controlling of the 6R robot is possible but also performance tests according to ISO and ANSI standards are accomplished and results could be depicted.

3.1 Capabilities of the simulator software

Different capabilities of simulator software are introduced. In Figure 3 push buttons in dialog box of simulator environment are shown. 'Link Rotation' static box (in left of Figure 3) is used for rotating each link of the 6R robot around its joint. Each of these rotations is performed in specified Steps; by adjusting amount of step, it is possible to place the end effector at desired pose in the robot's workspace. 'Frame positions' panel depicts 3D position of selected frame in 'Selected object' list box and also x, y, z coordinate of selected frame can be defined by user and placed in that coordinate by pushing 'Set' button.



Steps: 4

Selected object: Camera No.1

x: 0.
y: 14.
z: -26.

Figure 3. Control buttons in simulator software of the 6R robot

Init: At beginning of the program, this button is pushed to initialize variables in dialogue box.

GoHomePosition: Places frames and robot links in their homeposition and sets joint variables to their initial values.

Get Target: By pushing this button control process to guide end effector to reach the target is performed.
Direct Kinematics: Performance tests for direct kinematics are accomplished. Joint variables are determined in a text file by user.
Inverse kinematics: Inverse kinematics tests for the robot are done. Transformation matrix is defined by user in a text file. This file would be read and joint variables are determined to rotate joints to reach the end effector in desired pose.
Continuous Path: It guides the end effector during continuous paths such as circle, line or rectangle to simulate performance tests. Paths properties are defined in text file by user.
Look At: By pushing this button observer camera will look at robot at any pose.
Camera switch: change the view between two stationary camera's views. Switch from camera 1 to camera 2 or vice versa.
GetOrient: Changes the orientation of selected camera frame.

4. Visual servo control simulation

The goal of this section is to simulate:

- Position based visual servo control of the 6R robot
- Image based visual servo control of the 6R robot
- Compare these two visual servo control approaches

To attain these goals different theories of computer vision, image processing, feature extraction, robot kinematics, dynamics and control are used. By two stationary cameras observing the robot and workspace, images are taken, after image processing and feature extraction, target-object and the end effector are recognized, and their 3D pose coordinates are estimated by using a neural network. Then the end effector is controlled to reach the target. For simulating image based visual servo control of the 6R robot one of the cameras are mounted on the end effector of the robot and the other one is stationary on the earth.

4.1 Position based visual control simulation

In simulator software, function Capture Frame takes picture in bitmap format through two stationary cameras mounted on the earth and the images are returned in the form of array of pixels. Both of the two cameras after switching the view will take picture (to estimate 3D pose information of frames). After image processing, objects in pictures are saved separately, features are extracted and target-object and end effector will be recognized among them according to their features and characteristics. Then 3D position coordinates of target-object and end effector are estimated. After each motion of joints, new picture is taken from end effector and this procedure is repeated until end effector reach to target-object.
With images from these two fixed cameras, positions of objects are estimated in image plane coordinate, usually, to transform from image plan coordinates to the 3D reference coordinates system, mapping and calibrating will be used. In this program, a neural network has been used to transform these coordinates to global reference 3D coordinate. Mapping system needs extra work and is complicated compared to neural network. Neural networks are used as nonlinear estimating functions. A set of points has been used to train the neural system to compute processing matrix. Control procedure of robot to reach to target-object is briefly shown in Figs 4 and 5.

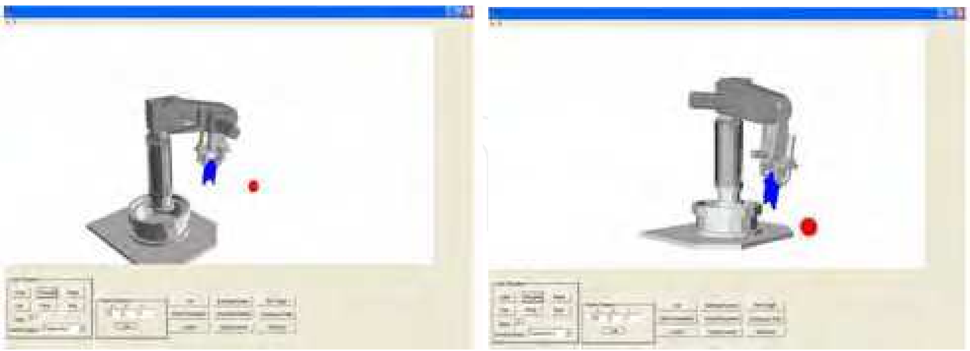


Figure 4. Robot at step 2 of control process in view of camera1 and camera2

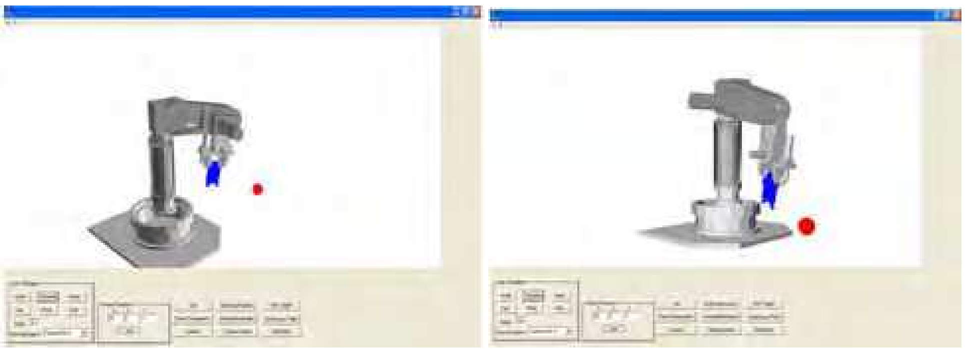


Figure 5. Robot at step 2 of control process in view of camera1 and camera2

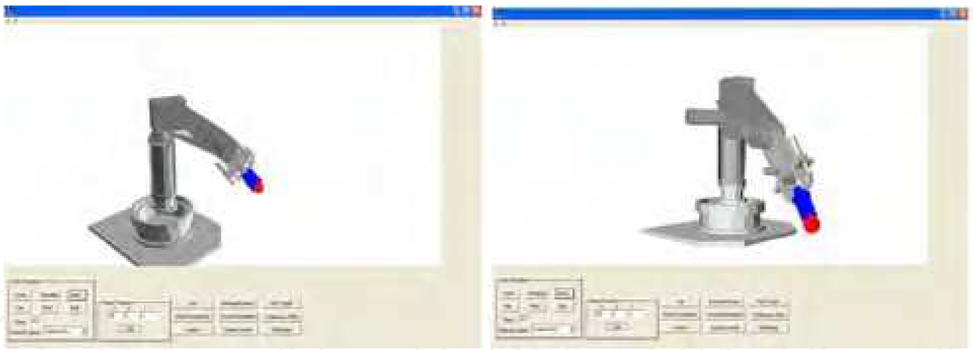


Figure 6. Robot at last step of control process reached to target-object in view of camera1 and camera2

Test steps:

1. Initialize the simulator environment by clicking Init button.
2. Select frame object No.1 from Selected object box.
3. Specify its 3D x, y, z position in Frame Position and click Set icon.
4. By Get Target icon, control process is accomplished.

Problems 1-2:

Set the target-object in four corners of a rectangle with coordinates as: (3,0,-2), (3,0, 2), (-3,0,-2), (-3,0, 2) and guide the end effector to attain the target-object.

For reaching end effector in (2,-1, 2) position, compute joint angles and compare them with actual joint angles at the end of the control process.

4.2 Mapping points in image plane to 3D system

As mentioned before a neural network has been used to transform 2D coordinates of image planes to global reference 3D coordinate. Collection of points to train the net are achieved by moving end effector of robot through different points that their coordinates in global reference system are known and their coordinates in image plane of the two cameras are computed in pixels by VisionAction module in simulator software. The position of the end effector is recognized at any time by two cameras, which are fixed with a certain distance from each other. The camera No.1 determines the target coordinates in a 2D image plan in pixels. The 3rd coordinate of the object is also computed by information from the second camera.

The used neural network is a back propagation perception kind network with 2 layers. In input layer (first layer) there are 4 node entrance including picture plan coordination pixels from two fixed cameras, to adapt a very fit nonlinear function 10 neurons in this layer with ‘tan sigmoid’ function have been used. In the second layer (output layer) there are 3 neurons with 30 input nodes and 3 output nodes which are 3D coordinates x, y and z of object in the earth reference system. The transfer function in this layer is linear.

This network can be used as a general function approximator. It can approximate 3D coordinates of any points in image plane of two cameras arbitrarily well, with given sufficient neurons in the hidden layer and tan sigmoid functions. As shown the training results in Figure 7 performance of trained net is 0.089374 in less than 40 iterations (epochs). This net approximates 3D coordinates of points well enough.

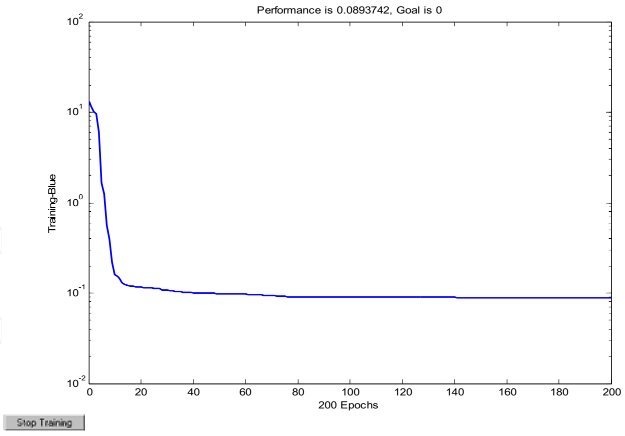
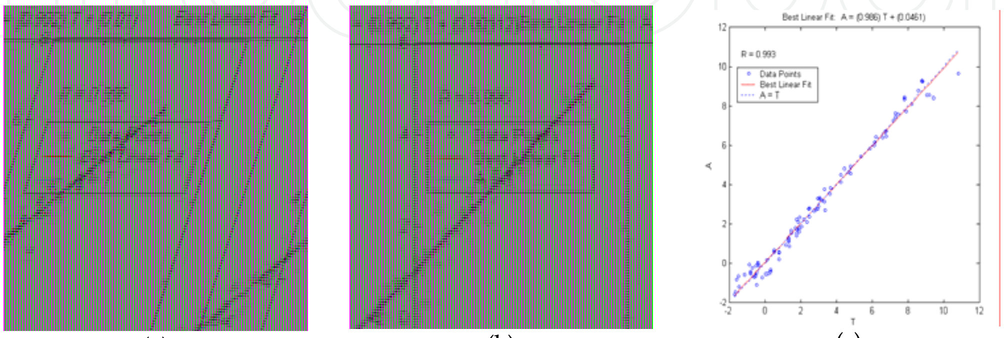


Figure 7. Training results of back propagation network

The performance of the trained network can be measured to some extent by the errors on the training, validation and test sets, but it is useful to investigate the network response in more detail. A regression analysis between the network response and the corresponding targets are performed. Network outputs are plotted versus the targets as open circles (Figure 8). The best linear fit is indicated by a dashed line. The perfect fit (output equal to targets) is indicated by the solid line. In this trained net, it is difficult to distinguish the best linear fit line from the perfect fit line, because the fit is so accurate. It is a measure of how well the variation in the output is explained by the targets and there is perfect correlation between targets and outputs. Results for x, y and z directions are shown in Figure 8.



(a) (b) (c)

Figure 8. Regression between the network outputs coordinates in a) x, b) y, c) z direction and the corresponding targets

4.3 Image based visual servo control simulation

Image-based visual servo control uses the location of features on the image plane directly for feedback i.e. by moving the robot the camera's view (mounted on the end effector) changes from initial to final view. The features of images comprise coordinates of vertices, areas of the faces or any parameter and feature of the target-object that change by moving the end effector and so camera installed on it.

For a robot with a camera mounted on its end effector the viewpoint and hence the features of images are functions of the relative pose of the camera and the target-object. In general, this function is non-linear and cross-coupled such that motion of the end effector will result in the complex motion of many features. For example, camera rotation can cause features to translate horizontally and vertically on the image plane. This relationship may be linearized about the operating point to become more simple and easy.

In this version of simulator software the end effector is guided to reach the target-object, using feature based visual servo approach. In this approach global 3D position of target and the end effector are not estimated but features and properties of the target images from two cameras are used to guide the robot.

For image based visual servo control simulation of the 6R robot, two cameras are used. One is mounted on the end effector (eye in hand) and the other one is fixed on the earth observing the robot within its workspace (eye to hand). Obviously eye in hand scheme has a partial but precise sight of the scene whereas the eye to hand camera has a less precise but global sight of it. In this version of simulator software, the advantages of both, stand-alone and robot-mounted cameras have been used to control robot’s motion precisely. Pictures are taken in bitmap format by both cameras through camera switch function then each image is returned in form of array of pixels.

System analysis is based on the stereovision theory and line-matching technology, using the two images captured by the two cameras. The vision procedure includes four stages, namely, calibration, sampling, image processing and calculating needed parameters.

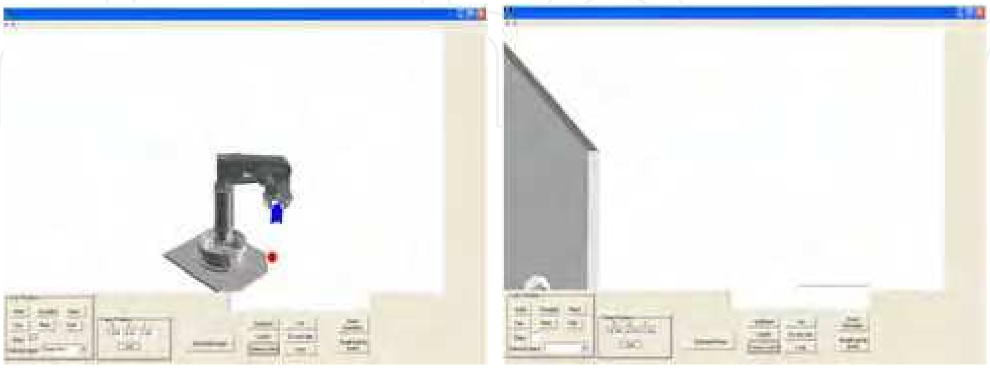


Figure 9. Robot in homeposition at beginning of control process in view of camera1 & camera2

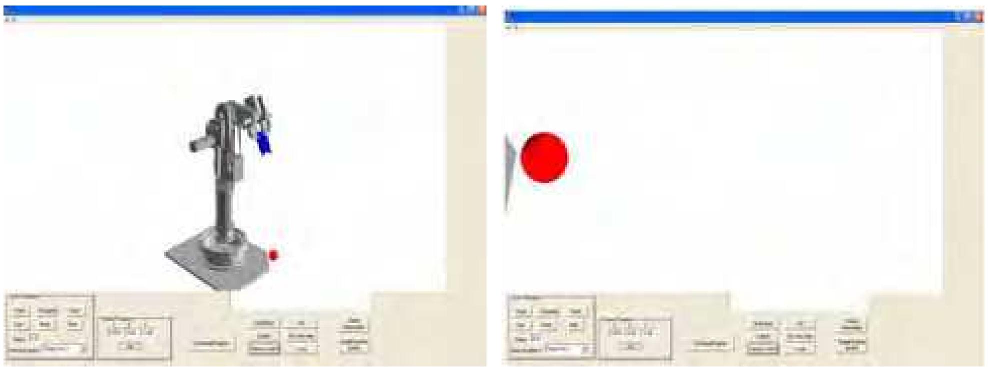


Figure 10. Robot at step 2 of control process in view of camera1 and camera2

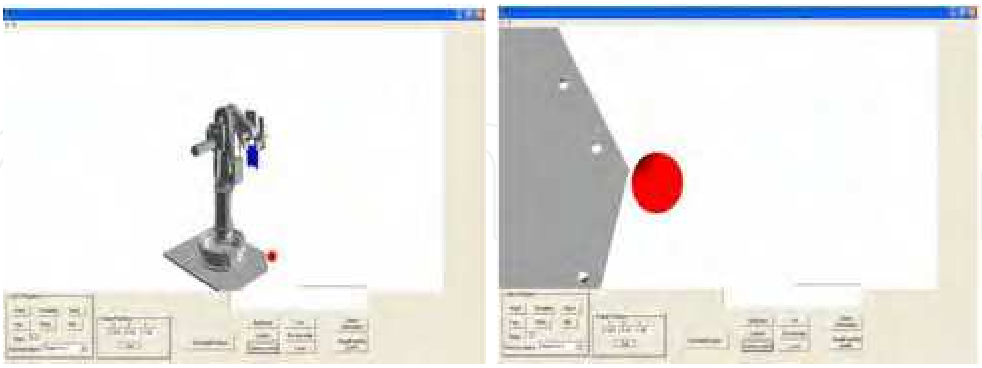


Figure 11. Robot at step 5 of control process in view of camera1 and camera2

First, the precision of this measuring system must be determined for simulator software. To maintain robot accuracy, calibration equipment is needed. In this simulator software, a selfcalibrating measuring system based on a camera in the robot hand and a known reference object in the robot workspace is used. A collection of images of the reference target-object is obtained. From these the positions and orientations of the camera and the end effector, using image processing, image recognition and photogram metric techniques are estimated. The essential geometrical and optical camera parameters are derived from the redundancy in the measurements. By camera calibration, we can obtain the world coordinates of the start points of robots motion and the relation between images of the target-object and its relative distance to the end effector. So amount and direction of the end effector's motion is estimated and feedback for visual servo system will be obtained.

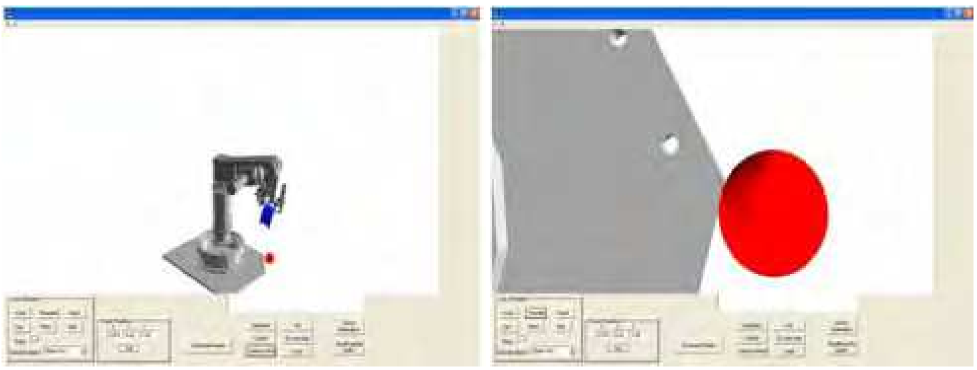


Figure 12. Robot at step 10 of control process in view of camera1 and camera2

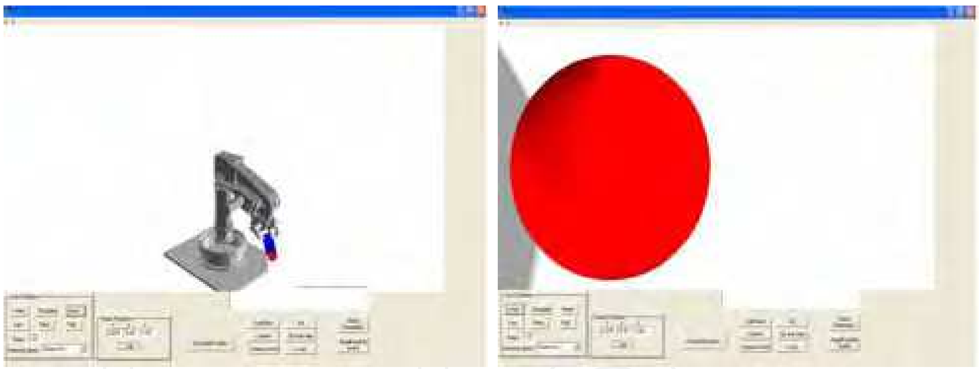


Figure 13. Robot at last step of control process reached to target-object in view of camera1 and camera2

At first control step as position of the target-object in 3D global reference system are not distinct, the end effector of the robot is moved to such a pose that target-object becomes visible in eye in hand camera view. It means that the end effector would find the target-object within robot's workspace. For this purpose, hand and wrist of the 6R robot rotate to reach the end effector to top point of workspace. By finding the target-object, the robot moves toward it to attain it. In each step, two cameras take picture from target and compare features in these images with reference image to assess required motion for each joints of the 6R robot. This procedure is repeated until the camera mounted on the end effector observes the target-object in middle of its image plane in desired size. Also in this algorithm, pictures taken by two cameras are saved in arrays of pixels and after threshold operations, segmentation, and labeling, the objects in the pictures will be extracted and each single frame is conserved separately with its number. Distance between end effector and target-object will be estimated, by using inverse kinematics equations of 6R robot, each joint angle will be computed then by revolution of joints end effector will approach to target. Control procedure of robot to reach to target-object is briefly shown in Figs 9 to 13.

4.4 Comparing IB and PB visual servoing approaches

Vision based control can be classified into two main categories. The first 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, between the camera frame and the target-object frame from target image features.

In this chapter, both position based and image based approaches were used to simulate control of the 6R robot. The advantage of position-based approach is that the servo control structure is independent from the target pose reconstruction. Usually, the desired control values are specified in the Cartesian space, so they are easy to visualize. In position-based approach, target pose will be estimated. But in image based approach 3D pose of the target-object and end effector is not estimated directly but from some structural features extracted from image (e.g., an edge or color of pixels) defined when the camera and end effector reach the target as reference image features, the robot is guided and camera calibrating for visual system is necessary.

To construct the 3D pose of a target object from 2D image feature points, two cameras are needed. Image feature points in each of the two images have to be matched and 3D information of the coordinates of the target object and its feature points can then be computed by triangulation. The distance between the feature points in the target object, for example, can be used to help compute the 3D position and orientation of the target with respect to the camera. However, in systems with high DOF using image based approach and camera calibrating to guide the robot will be complicated, rather than in position-based approach we have used a trained neural net to transform coordinates. The image based approach may reduce computational delay eliminate the necessity for image interpretation and eliminate errors in sensor modeling and camera calibration. However, it does present a significant challenge to controller design since the process is non-linear and highly coupled.

In addition, in image-based approach, guiding the end effector to reach target will be completed in some steps but in position-based, the end effector is guided directly toward the target-object. The main advantage of position-based approach is that it directly controls the camera trajectory in Cartesian space. However, since there is no control in the image, the image features used in the pose estimation may leave the image (especially if the robot or the camera are coarsely calibrated), which thus leads to servoing failure. Also if the camera is coarse calibrated, or if errors exist in the 3D model of the target, the current and desired camera pose will not be accurately estimated. Nevertheless, image based visual servoing is known to be robust not only with respect to camera but also to robot calibration errors. However, its convergence is theoretically ensured only in a region (quite difficult to determine analytically) around the desired position. Except in very simple cases, the analysis of the stability with respect to calibration errors seems to be impossible, since the system is coupled and non-linear.

In this simulator software control simulating of the 6R robot by using both position based and feature based approaches depicted that position based was faster but feature based more accurate. For industrial robots with high DOFs position based approach is used more, specially for performance testing of the robots we need to specify 3D pose of the end effector in each step so position based visual servo control is preferred.

Results for comparing two visual servo control process PBVS and IBVS are summarized in Table 1. These two approaches are used to guide the end effector of the robot to reach the target that is in a fixed distance from the end effector of the robot. Final pose of the wrist is determined and compared with target-object pose so the positioning error and accuracy is computed. However, the time duration for these processes is counted and control speed is compared in this way.

Visual Servoing Method	Control Accuracy (min error)	Performance Speed (process duration)	Computation delay	Controller design
PBVS Control	0.04 mm	20 sec	30 sec	simple
IBVS Control	0.01 mm	60 sec	10 sec	highly coupled

Table1. Results for comparing PBVS and IBVS approaches

5. The 6R robot performance tests simulation

In this version of software, performance tests of robot including direct kinematics, inverse kinematics and motion of end effector in continues paths like circle, rectangle and line is possible. In point to point moving of end effector, each joint angle is determined and robot will move with joints rotation. In inverse kinematics test, desired position and orientation of end effector is determined in transformation matrix T. amount of joint angles that satisfy inverse equations will be found and wrist will be in desired pose. Two observer cameras take pictures and pose of end effector will be estimated to determine positioning error of robot.

Then using ISO9283, ANSI-RIA standards, these errors will be analyzed and performance characters and accuracy of the robot will be determined. Results of these standard tests are used to compare different robots and their performance. In this chapter, we represent some of these tests by using camera and visual system according to the standards such as ISO-9283, and ANSI-RIA.

5.1 Performance test of 6R robot according to ISO9283 standard

a) Direct kinematics test of 6R robot (point-to-point motion)

In this part of test, position accuracy and repeatability of robot is determined. With rotation of joints, the end effector will move to desired pose. By taking pictures with two stationary cameras and trained neural network, we will have position of end effector in 3D global reference frame. To determine pose error these positions and ideal amounts will be compared. Positioning error in directions x, y, z for 10 series of direct kinematics tests is depicted in Figure 14. Amount of joint angles θi are defined by user in a .txt file this file is read by software and through RotateJonint function, each joint rotates to its desired value.

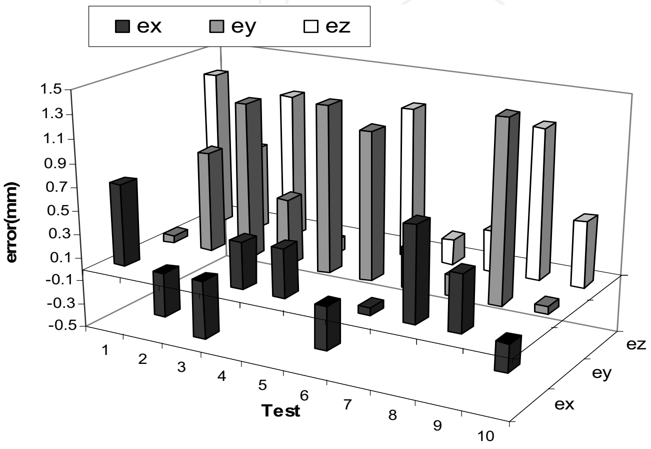


Figure 14. The error schematics in x, y, z directions for direct kinematics tests

b) Inverse kinematics test

In this stage, desired pose of the end effector is given to robot to go there. Transformation matrix containing position and orientation of the wrist frame is given by user in txt file. By computing joint angles from inverse kinematics equations and rotation of joints, end effector will go to desired pose. By taking pictures with two fixed cameras and trained neural network, we will have position coordinates of end effector in 3D global reference frame. By comparing the desired position and orientation of wrist frame with attained pose, the positioning error will be determined. Positioning error in directions x, y, z for 10 series of inverse kinematics tests is shown in Figure 15.

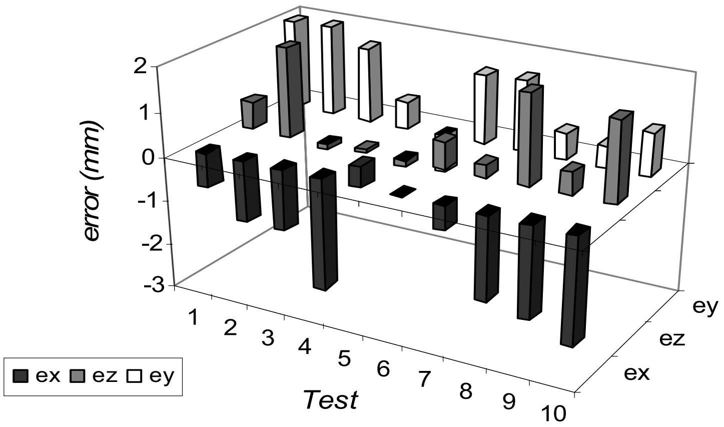
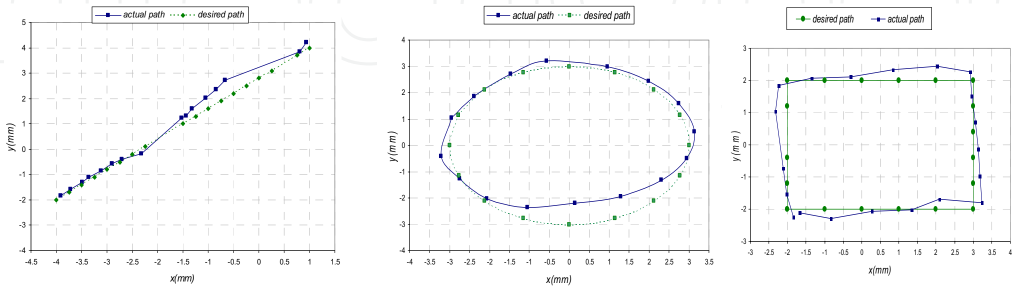


Figure 15. The error schematics in x, y, z directions for inverse kinematics tests

c) Continuous path test

To determine accuracy of robot in traversing continues paths wrist of robot is guided along different paths. In simulator software, three standard paths are tested (direct line, circular and rectangular paths). Results of moving the end effector along these continuous paths are depicted in Figure 16.



a): line paths b): circular path c): rectangular path

Figure 16. Continuous path test results for the 6R robot

5.2 Test steps

In this part, procedure to accomplish visual servo control and performance tests in simulator software for the 6R robot are prescribed. To control the robot by vision system, two camera frames have been installed on the earth watching the robot and its environment in front and right view. The camera one is lpCamera1 installed in point A(0,14,-26) and the other one is lpCamera2 installed in B(28,4,-1). Monitoring is possible through each of these cameras. Then images from these two cameras were saved in bmp format and used to train the neural network to find 3D positions of points in reference base coordinate. Position of two cameras can be changed through Frame position toolbox in simulator software. After image processing and recognition of the end effector, estimating its coordinate in image plane by neural network this coordinate are transformed to global reference coordinate. These steps are programed in simulator software and are done automatically. Performance tests of robot include direct kinematics, and motion of the end effector in continuous paths like circle, rectangle and line. In point to point moving of end effector, each joint angle is determined and robot will move with joints rotation. These joint angles are defined by user in a txt file. Two observer cameras take pictures and pose of end effector will be estimated to determine positioning error of robot. Standards such as ISO9283, ANSI-RIA are used to specify the robot error and path accuracy for continuous paths.

5.2.1 Direct kinematics test

Define amount of joint angles θi in angles.txt file in radian and save it. This file is read by software and through RotateJonint function, each joint rotates to its desired value.

When end effector stops two cameras take pictures from it and through VisionAction function and trained neural net x,y,z of center of the end effector in 3D Cartesian system is determined. It is saved in out_dirk1.txt file. Compute positioning error of robot during direct kinematics test.

Problem-3
In direct kinematics test, rotate joints in angles given in Table 2. Compute positioning error for each test in x, y, z directions and rotation of each joint and draw its graph.

test	1	2	3	4	5	6	7	8	9,10	
θ1	0.85	1.57	-3	-2	3.14	4.71	-1.2	0.8	1.8	-1
θ2	0	1	2	-1	-1	-0.5	1.57	0.75	-0.75	1.2
θ3	-0.25	2	2	2	3.14	1.57	-0.75	-0.5	1	-1.2
θ4	1	0	0.15	0.5	-0.5	0.75	0.5	-0.5	-0.5	0.5
θ5	0.5	0	0.5	-3	-1.57	-0.75	0.3	0.2	0.2	-1
θ6	1.1	0	-2	4	1	-1	0.5	-0.5	-0.5	1

Table 2. Joint angles for direct kinematics test

5.2.2 Inverse kinematics test

In this stage, desired pose of the end effector is given to robot to go there. Transformation matrix containing position and orientation of the wrist frame is given in txt file. Specify position and orientation of end effector in a transformation matrix of the wrist with respect to base T, in Tmatrix.txt file.

This matrix is used in InverseKinematics function to determine joint angles for desired pose of the end effector. By computing joint angles from inverse kinematics equations and rotation of joints, end effector will go to desired pose. By taking pictures with two fixed cameras and trained neural network, we will have position coordinates of end effector in 3D global reference frame.

Attained pose of end effector is saved in out_inv1.txt file. Positions and orientation error of this test are computed by data in this file.

By comparing the desired position and orientation of wrist frame with attained pose, the positioning error will be determined.

Problem-4

	desired position		
number	x	y	z
1	5.29	7.25	2.06
2	9.38	11.37	7.51
3	-12.08	-1.786	-0.22
4	-2.84	12.4	0.599
5	0.156	-13	1.09
6	10.5	1.53	0.12
7	3.34	3.57	-1.04
8	-1.165	6	1.707
9	-2.32	12	3.4
10	-3.48	8	5.1

Table 3. End effector positions for inverse kinematics test

In inverse kinematics test, define transformation matrix T with position of wrist according to Table 3. Orientation of the end effector can be defined by approach, normal and sliding vectors. Compute positioning error and accuracy of the robot in each test. Compare these errors with direct kinematics test results.

5.2.3 Continuous path test

To determine accuracy of robot in traversing continuous paths wrist of robot is guided along different paths. In simulator software, three standard paths are tested (direct line, circular and rectangular paths).

Specify type of path c for circular, r for rectangular and l for linear path and their specifications in path.txt file and save it. Each path must be entered in a distinguished line and its parameters in that line. For example:

Linear path: x1, y1, z1 is coordinates of start point of the linear path and x2, y2, z2 is for end point of the path.

l, x1, y1, z1, x2, y2, z2. (Fig. 17)

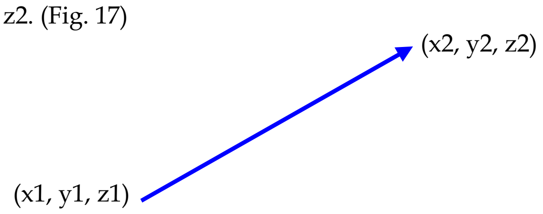


Figure 17. Coordinates specified for line path in performance tests of the robot

Rectangular path: x0, y0 determine coordinates of one corner of the rectangle and a, b are length and width of the rectangle.

r, x0, y0, a, b. (Fig. 18)

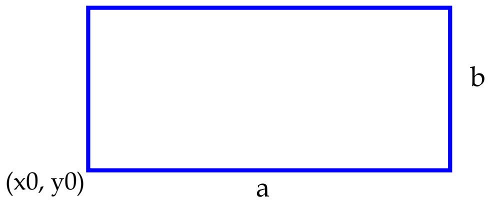


Figure 18. Coordinates specified for rectangular path in performance tests of the robot

For a circular path: x, y is coordinate of center of circular path and r is its radius.

c, x, y, r. (Figure 19-c)

Approach vector direction is normal to direction of paths i.e. wrist is always normal to its path. With pose of end effector and inverse kinematics equations of robot, joint angles will be computed. Joints rotate and end effector will be positioned along its path. Coordinates of end effector in global reference frame is determined by taking pictures with two fixed cameras and trained neural network.

Compute path accuracy and error of the robot by data saved in out_path.txt file.

Problem-5

Test motion of the end effector for given paths as following and draw the traversed paths by the end effector and desired path in one graph to compare them.

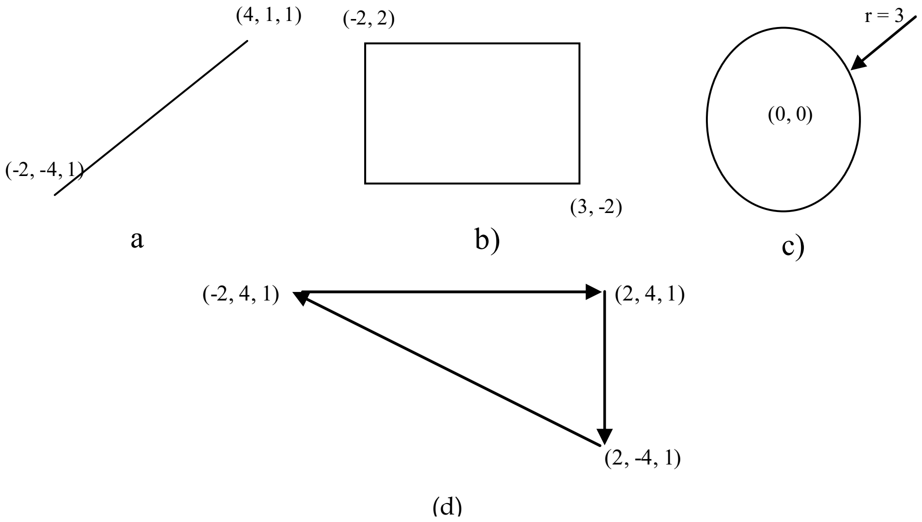


Figure 19. Continuous paths for performance tests of the robot

6. Error analysis of the 6R robot tests

Now we analyze results of previous tests according to different standards and we determine performance parameters and accuracy of 6R robot according to ISO9283 and ANSI/RIA.

6.1 Error analysis according to ISO9283

In this standard some performance parameters of robot to position and path traversing such as pose accuracy and distance accuracy are determined. For direct, inverse kinematics and continuous path tests of the 6R robot results are depicted in Figure 20.

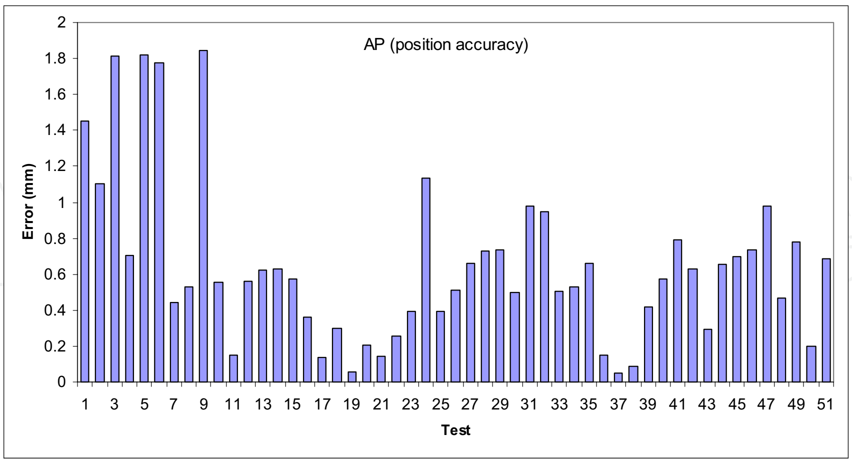


Figure 20. Position accuracy for the 6R robot in direct and inverse kinematics tests

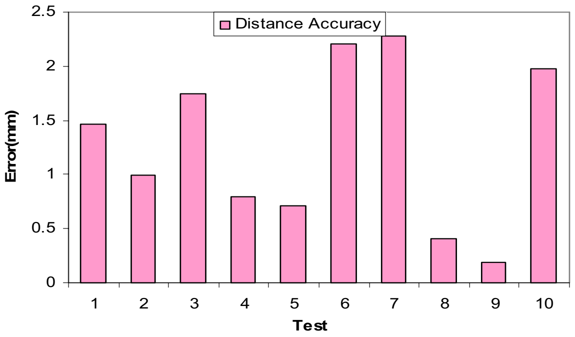


Figure 21. Distance accuracy investigated in positioning tests according to ISO9283

Traversing of the end effector in corners causes sharp changes in velocity so if these changes are high positioning and path accuracy of the robot must be controlled. For that, corners of paths are smoothed and curved to avoid sharp velocity changes. Error and accuracy of robot in traversing corners of paths are specified by cornering round off error (CR) and cornering overshoot parameters that are computed for the 6R robot during its motion in rectangular paths. These results are summarized in Figure 22.

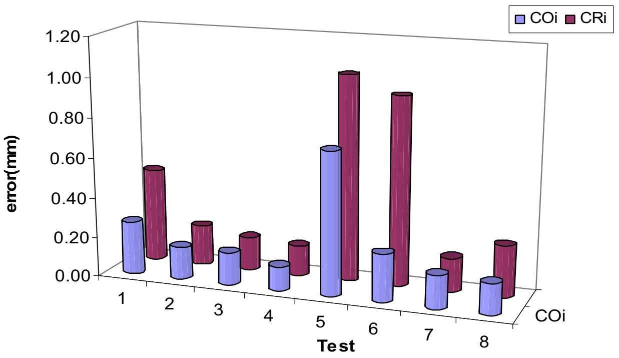


Figure 22. Cornering round off error and cornering overshoot in rectangular path tests according to ISO9283

6.2 Error analysis according to standard ANSI-RIA

Results of simulated tests in previous sections are analyzed with standard ANSI-RIA to compare with results of ISO9283.

TEST	AC	$\overline{AC}$	CR	CO
line	0.67	0.21	-	-
rectangle	0.47	0.09	0.47, 0.20, 0.17, 0.15	1.03, 0.95, 0.17, 0.26
circle	0.48	0.25	-	-

Table 4. Repeatability & cornering overshoot according to ANSI standard

Cornering round off error CR in this standard is defined as the minimum distance between the corner point and any point on the attained path. *Cornering overshoot* CO is defined as the largest deviation outside of the reference path after the robot has passed the corner. For rectangular path test of 6R robot the value of CR and CO are calculated. (Table 4) The tests were repeated 10 times (n = 10). Two cameras, observing the end effector at fixed distance in specified periods, take picture from end effector and its environment. Its coordinates are achieved from image plan with position based visual system.
To transform coordinates of wrist of robot to the reference frame as mentioned before, in this work we have used neural networks. Using neural networks we map coordinates from image plan into reference system, in order to have real distances. Maximum and mean path accuracy FOM and for rectangular path tests corner deviation error (CR) and cornering over shoot (CO) are listed in Table4.

7. Experimental results for performance tests of 6R robot

In this part, experimental results of the visual servo control and performance tests for the 6R robot are presented. To control the robot by vision system, two stationary webcams have been installed on the earth watching the robot and its environment in front and right view. Two webcams are installed in points A(0,-1,0) and B(1,0,0) as in Figure 23. Monitoring is possible through each of cameras. Then images from these two cameras were saved in bmp format and used to train the neural network to find 3D positions of points in reference base coordinate. After image processing and recognition of the end effector, estimating its coordinate in image plane by neural network this coordinate are transformed to global reference coordinate. These performance tests of robot include direct kinematics, and motion of the end effector in continuous paths like circle, rectangle and line. In point to point moving of end effector, each joint angle is determined and robot will move with joints rotation. Two observer cameras take pictures and pose of end effector will be estimated to determine positioning error of robot. Standards such as ISO9283, ANSI-RIA are used to specify the robot error and path accuracy for continuous paths.

7.1 Direct kinematics test of 6R robot (point-to-point motion)

In these tests, position accuracy and repeatability of robot is determined. Amount of rotation for each joint angle of the robot is specified in deg. With rotation of joints, the wrist will move to desired pose. By taking pictures with two stationary cameras and trained neural network, we will have position of end effector in 3D global reference frame. To determine pose error these positions and ideal amounts will be compared. Positioning error in directions x, y, z for 10 series of direct kinematics tests is depicted in Figure 24. Amount of joint angles θi (deg) are defined by user in running program of the robot written by Delphi software. In image processing and object recognition algorithm due to noises and ambient light, there were many noises and deviation from simulation results.

7.2 Continuous path test

Pictures taken by two cameras are saved in bmp format and they are processed through vision algorithm written in VC++. After image processing, objects in pictures are saved separately, features are extracted and target-object and end effector will be recognized among them according to their features and characteristics. Then 3D position coordinates of target-object and end effector are estimated. After each motion of joints, new picture is taken from end effector and this procedure is repeated until end of process. To determine accuracy of robot in traversing continuous paths wrist of robot is guided along different paths. In experimental tests, three standard paths are tested.

a) Direct line

To move end effector along a direct line its start and end must be determined. Approach vector direction is normal to direction of line path i.e. wrist is always normal to its path. With pose of end effector and inverse kinematics equations of robot, joint angles will be computed. Joints rotate and end effector will be positioned along its path. At each step, two stationary cameras take images from robot and its workspace. From these pictures and trained neural network coordinates of the wrist in global reference frame is determined. The positioning error is determined by comparing the ideal pose and actual one. Error of robot in traversing direct line path is shown in Figure 25-a.

b) Circular path

We investigate the accuracy, repeatability and error of robot on the circular continuous path traversing. Circle is in horizontal plane i.e. height of end effector is constant from earth level. Orientation of wrist is so that end effector is always in horizontal plane and normal to circular path and wrist slides along perimeter of circle. In this way sliding, approach and normal vectors are determined and inverse kinematics equations can be solved. During motion of wrist on the path, images have been taken from end effector using two webcams. In this way, end effector coordinates in image plan will be collected. Using neural network, image plan coordinates of points will be transformed to the reference frame. The desired path and actual path traversed by robot is shown in Figure 25-b.

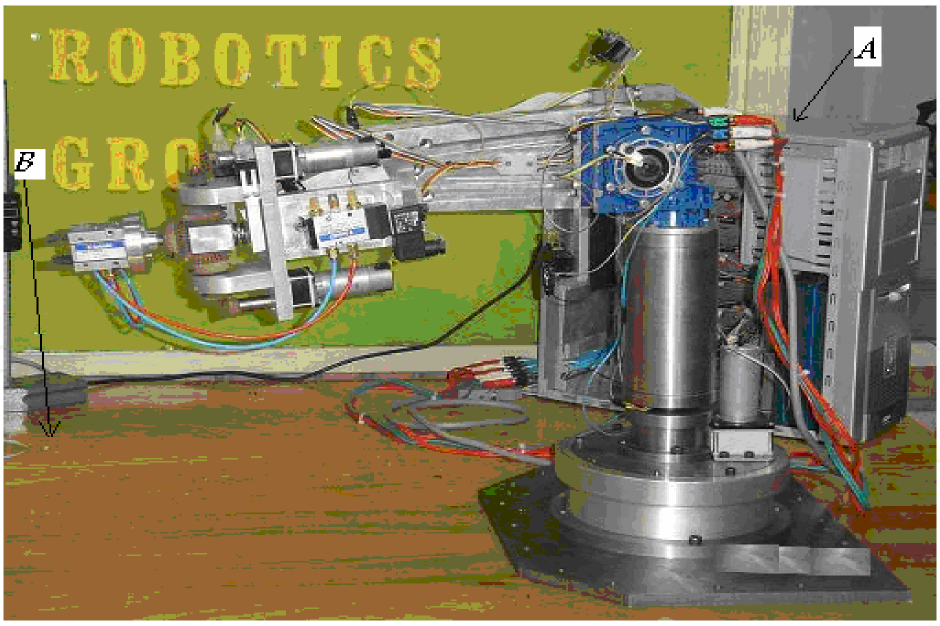


Figure 23. Webcams positions in experimental tests of robot (front and right cameras)

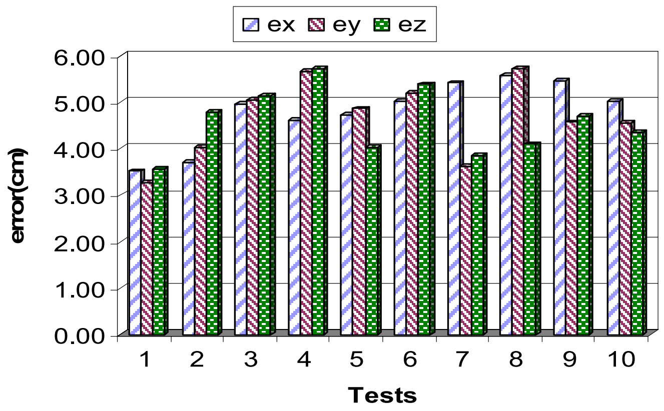
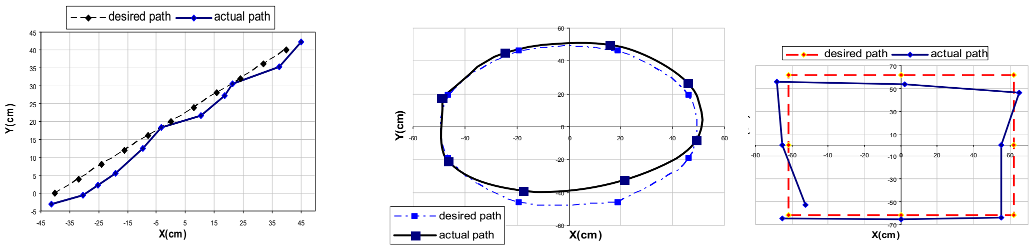


Figure 24. The error schematics in x, y, z directions for direct kinematics tests of the 6R robot



a): line paths b): circular path c): rectangular path

Figure 25. The error investigated in continuous path

c) Rectangular path

Path accuracy for movement of the end effector in rectangular path was also tested. Orientation of end effector is tangent to path. The desired path and actual path for rectangular path have been drawn in Figure 25-c.

8. Collision detection for the 6R robot using spheres

Collision detection or contact determination between two or more objects is important in robotics simulation and other computer simulated environments. Objects in simulated environments are stationary or dynamic. The previous works are mostly restricted to models in static environments. However, some of them concern the more sophisticated algorithms, such as BSP (one of the commonly used tree structure, binary space partitioning tree, to speed up intersection tests in CSG ,constructive solid geometry) (Lin, 1993) for dynamic simulation environments. We have used an efficient simple algorithm for collision detection and contact determination between links of 6R robot undergoing rigid motion. This technique however is a quite simple procedure but it is very useful also can be used for simulated environments with many dynamic objects moving with high speed. The main characteristic of this algorithm is its simplicity and efficiency. It has been implemented on simulation of control and performance tests of 6R robot to avoid contact of different parts of robot with each other and surrounding objects.

Main points in a simulation of collision among objects can be separated into three parts: collision detection, contact area determination, and collision response (Ponamgi et al). In this research, we have considered the first part to prevent penetration of links of the 6R robot in each other during their motion.

To determine whether or not two objects intersect, we must check if distance between their border edges is equal to zero or not. So lower bound for the distance between each pair of objects is equal to zero. In this paper the collision detection technique uses spheres attached to different parts of robot and moved as well as them. These spheres are arranged compactly enough to fit the robot shape so we have used a large number of spheres to do.

In an environment with D moving objects and S stationary objects, number of possible collision for each pair of the objects will be: $\binom{D}{2} + DS$ pairs at every time step. Which determining all of them would be time consuming as D and S are large. By considering the robot geometry and its joints rotations we can determine which pairs of spheres may contact and which pairs may not. So the total number of pairwise collisions will decrease and much time would be saved.

In Figure 26 schematic shape of 6R robot and bounding spheres on different parts of it are shown. Diameter of each sphere is determined according to size of object which is bounded by the sphere.

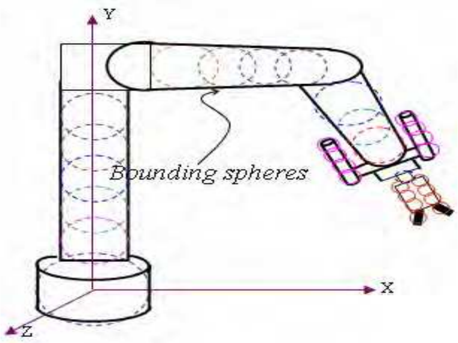


Figure 26. The 6R robot and bounding spheres

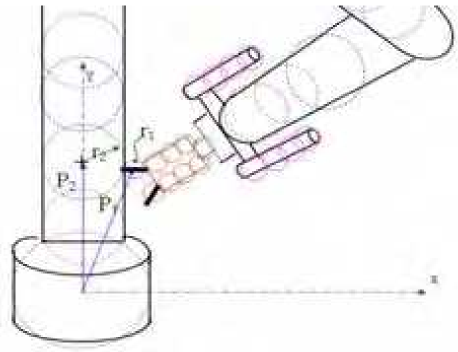


Figure 27. Collision between two spheres in the 6R robot

8.1 Colliding bounding spheres in the 6R robot

To avoid collision among different parts of the 6R robot, links and objects in simulated environment are bounded by small spheres (Figure 26). As joints of robot revolute, the links may collide and penetrate each other. We consider the situation when the tip of end effector collides to the waist of the robot (Figure 27) and find intersection point of two collided spheres. This procedure is the same for each pair of colliding spheres.
The simplest possible way to test collision between two bounding spheres is to measure the squared distance between their centers and to compare the result with the squared sum of their radii.

9. Object recognition algorithm

After taking pictures by two fixed cameras, these images must be processed to determine 3D information of the target-object and the end effector of robot and to estimate their pose in Cartesian global coordinate. So recognition of objects in the visual system is a key task. But the end effector of the 6R robot does not have any especial basic shape so we decided to use a definite color for it and it would be recognized according to its color. Upon this in simulation of the object recognition we used color based algorithm.

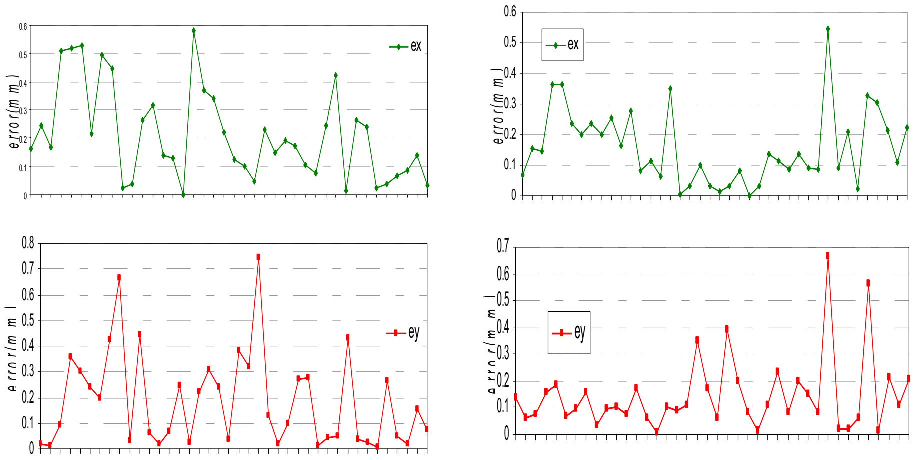


Figure 28. Performance of simulation color based object recognition algorithm to determine pose of (a) the end effector (b) target-object

Object recognition algorithm has two steps: first to assess objects of interest in pictures taken by two cameras and then to provide required information (e.g. pose) about these objects. To do the first step, the model or properties of objects of interest are provided for the vision system. As said before the end effector is not in basic geometric shape and also due to its roll and pitch rotations its dimensions and appearance are not the invariant in two cameras' view each time. So we can not use dimensions or distance set to recognize the end effector. We must identify the image features that are invariant with respect to image scaling, translation and rotation and partially invariant with respect to illumination changes. Also they are minimally affected by noise and small distortions. Lindeberg showed that under some rather general assumptions on scale invariance, the Gaussian kernel and its derivatives are the only possible smoothing kernels for scale space analysis (Low). To achieve rotation invariance and a high level of efficiency, we have defined two special RGB color for the target-object and the end effector of the 6R robot separately. By image processing RGB of each pixel in images are found and if they are the same as RGB of the object of interest, coordinate of those pixels will be saved and the center position of them in two image plane will be determined and then by using neural network we will have 3D coordinates of target-object and the end effector in global reference frame. The results obtained from simulation of color based object recognition algorithm for the end effector and target-object are presented in Figure 28. In these figures error of position estimation of the end effector and target-object in x, y and z directions are shown.

10. Conclusion

In this chapter, both position based and image based approaches were used to simulate control of the 6R robot. The IBVS control approach, 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 PBVS control constructs the spatial relationship, target pose, between the camera frame and the target-object frame from target image features. Many construction algorithms have been proposed. The advantage of position-based approach is that the servo control structure is independent from the target pose reconstruction. Usually, the desired control values are specified in the Cartesian space, so they are easy to visualize. In position-based approach, target pose will be estimated. But in image based approach 3D pose of the target-object and end effector is not estimated directly but from some structural features extracted from image (e.g., an edge or color of pixels) defined when the camera and end effector reach the target as reference image features, the robot is guided and camera calibrating for visual system is necessary. Test errors have been analyzed by using different standards and MATLAB to compute performance parameters of 6R robot such as accuracy, repeatability, and cornering overshoot. Performance parameters computed according to ANSI and ISO standards are fairly close to each other. Statistical quantities computed by MATLAB also certificate standards analysis. In simulator environment, we have determined collision between two parts of robot by using bounding-spheres algorithm. To improve the accuracy of the collision detection we have used very small bounding spheres, breaking links of robot into several parts and enclosing each of them in a bounding sphere of its own.

Finally simulation results of color based object recognition algorithm used to provide required information (e.g. pose) about target-object and the end effector were presented.

11. References

American National Standard for Industrial Robots and Robot Systems Path-Related and Dynamic Performance Characteristics Evaluation. *ANSI/RIA* R15.05-2. 2002.

Boyse, J. W. (1979) *Interference detection among solids and surfaces*. *ACM*, 22(1):3-9.

Cameron S.A. (1985) A study of the clash detection problem in robotics. *Proc. IEEE ICRA*, pages pp. 488-493.

Cameron, S.A. & Culley R. K. (1986) Determining the minimum translational distance between two convex polyhedra. *Proc. IEEE ICRA*, pages pp. 591-596.

Canny, J. (1986) Collision detection for moving polyhedra. *IEEE Trans. PAMI*, 8:pp. 200-209.

Gilbert E. & Foo C.P. (1990) Computing the distance between general convex objects in three dimensional space. *IEEE Trans. Robotics Aut.*, 6(1).

Gilbert, A. Giles, M. Flachs, G. Rogers, R. & Yee, H. (1983). A real time video tracking systems, *IEEE, Trans. Pattern Anal. Mech. Intell.* 2(1), pp. 47 – 56

Hashimoto, H. Kimoto, T. and Ebin, T. (1991). Manipulator control with image based visual servoing, *In Proc. IEEE, Conf. robotics and automation*, pp. 2267 – 2272.

Herzen, B. V. Barr A. H. & Zatz H. R. (1990) Geometric collisions for time dependent parametric surfaces. *ACM Computer Graphics*, 24(4), August. ISO9283, (1998) Manipulating industrial robots performance criteria & related test methods

Kelly, R. Shirkey, P. & Spong, M. (2001). *Fixed camera visual servo control for planar robots.*

Korayem, M. H. Khoshhal, K. and Aliakbarpour, H. (2005) Vision Based Robot Simulation and Experiment for Performance Tests of Robot", *International J. of AMT, Vol.25, No. 11-12,* pp. 1218-1231.

Korayem, M H. Shiehbeiki, N. & Khanali, T. (2006). Design, Manufacturing and Experimental Tests of Prismatic Robot for Assembly Line, *International J. of AMT, Vol.29, No. 3-4*, pp. 379-388.

Lin. M.C. (1993) Efficient Collision Detection for Animation and Robotics. *PhD thesis*, Department of Electrical Engineering and Computer Science, University of CB.

Ponamgi, M.K. Manocha D. and Lin. M.C. *Incremental algorithms for collision detection between solid models,* Department of Computer Science University of N. Carolina

Rives, P. Chaumette, F. & B. Espiau. (1991) Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *Proc. IEEE Int. Conf. Robotics and Automation,* pp 2248-2253.

Saedan M. & Ang M Jr. (2001) 3D Vision-Based Control of an Industrial Robot, *Proceedings of the IASTED Int. Conf. on Robotics and Applications*, Florida, USA, pp. 152-157.

Webber, T & Hollis, R. (1988) *A vision based correlation to activity damp vibrations of a coarse fine manipulator,* Watson research center.

Vision Systems:
Applications
INTECH