# A Model-Based Approach To Robot Hand-Eye Coordination System

K.N. Rao[1], P.S. Ramaiah[2]

ABSTRACT

This paper presents an implementation of PC-based Robot Hand-Eye Coordination System (RHECS) that positions a 4-Fingered Robot Hand using visual infor-mation from two cameras through a model-based approach. The RHECS tracks the robot hand's finger and visual information to define target posi-tions. An error based on the visual distance between the finger and the target is defined. To make this error to ZERO required control information is derived such that the robot hand goes to the required object. The control information is obtained by modeling the robot hand and the camera. This information is integrated into RHECS that performs tracking and stereo control with no special-purpose hardware requirements. Experiments with the RHECS have shown that the entire visual control system runs at a rate of 20 Hz. The control software for tracking is written in C++. Some experimental results of pick and place operation are discussed.

**Keywords:** Robot Hand Model, Camera Model, Stereo-Vision, Visual-Motor Coordination, Vision processing .

## 1. INTRODUCTION

In the field of robotics, robot hand-eye coordination, also referred to as Visual – motor coordination, is the process of using visual information to control a robot hand to reach a target point in its workspace. Robot hand-eye coordination has been an active area of research over the last 30 years [1, 2]. However the development of visual feedback mechanisms has lagged behind other areas of sensor feedback research. Vision systems are generally viewed as computationally expensive, error-prone, and difficult to calibrate. This is due to the use of vision to measure absolute position in the robot frame of reference. For accurate position estimates, the vision system must be extremely well calibrated at all times. Traditionally, this problem has been viewed as one of calibrating the camera and robot hand simultaneously. Many algorithms have been proposed for robustly calibrating this system, including adaptive calibration[3], where a model of the system is assumed to extract the parameters based on real-time data.

In the recent literature, many authors address the problem of servoing a camera to maintain a constant position relative to a known object with a single camera [4,5,6,7]. Alien et al. [8]describe a stereo motion based system for grasping a toy train using two stationary cameras. Rizzi and Koditschek [9] describe a stereo vision-based juggling system, and Anderson [10]describes a stereo vision-based ping-pong player.

None of these systems directly observe the robot end-effector; they only ob-serve and compute a goal position or goal configuration of the system. This relies on the hand-eye calibration because the movement to the computed position is open-loop with respect to the vision system. Maru et al. [11] describe algorithms for performing stereo vision-based positioning. The main

[12]Department of Computer Science and Systems Engineering, College of Engineering, Andhra University, Visakhapatnam 530003, INDIA.

differences between their approach and the results of this paper are that they use cameras mounted on the robot end-effector and their model of the camera system assumes that the cameras are aligned with one another and are perpendicular to the stereo baseline. In addition, the configura-tion of their system makes it impossible to observe the robot end-effector or objects held within it, so they cannot define visual errors by observing both the robot and a desired goal position. Hence, any application of their system will depend on knowing an exact hand-eye calibration. This paper describes an approach to stereo visual servoing where vision is used to measure error. Error is defined to be the visual distance between a robot manipulator and a pre-defined position in visual coordinates. Error is computed continuously using visual tracking.

In the next section, the major components of a generic stereo-based positioning architecture are discussed. The formalization of visual control problem with solutions is addressed in the subsequent section. Then experimental results from the implementation of visual control algorithms in the RHECS are presented.

## 2. Robot Hand-Eye Coordination System

Figure 1 shows a functional block diagram of Robot Hand-Eye Coordination System (RHECS) that provides a generic visual-motor coordination architecture. The major components are two video cameras on free-standing pan-tilt heads, a robot hand, controller and a PC that perform image processing, low-level control operations, and other interface-re-lated functions.

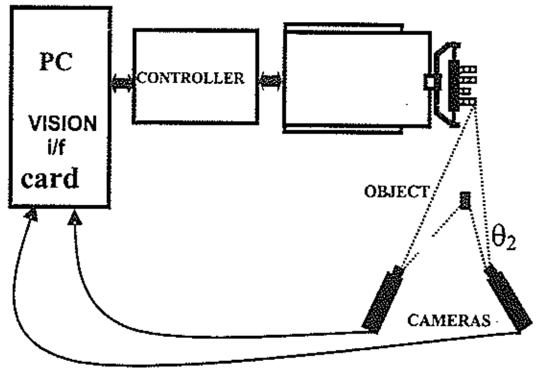



**Fig1. Block Diagram of Rhecs**

The traditional scheme to stereo-based posi-tioning assumes the cameras can be calibrated to the robot hand coordinate system. The observed features of the robot hand position and target position are then computed using stereo triangulation, and the robot is moved to the estimated position. The mapping from robot hand coordinates to images includes each camera's positional parameters, image center, image scale fac-tors, distortion coefficients, and focal length that are estimated using Tsai's algorithm[1]. It also includes parameters describing the kinematics of the pan-tilt heads and the kinematics of the robot hand itself. These parameters will often produce only an imprecise estimate of the hand-eye relationship due to modeling errors, mechanical backlash, and intensive calibration process.

System that uses fixed cameras or cameras mounted on the arm/hand itself can achieve higher accuracy. These configura-tions have the disadvantage that they limit the system to working configurations where the robot hand is both within the field of view and not obscured from the cameras. By using visual feedback it is possible to achieve extremely accurate positioning of a servo system relative to an observed target despite calibration error. Figure 1 explains the basic principle for translation positioning in which the cameras observe the image distances $\theta_1$ and $\theta_2$ between a point on the finger/gripper and the corner of the target box. These distances are called as the target-goal disparity that should become zero( $\theta_1 = \theta_2 = 0$) indicating that the robot hand has reached the target. The zero disparity between the robot hand and the goal/object in both images means that they are at the same point in space.

### 3. A Model-Based Approach To Rhecs.

The aim of RHECS is to position the robot hand at the target using the processed visual information from the pair of CCD cameras and controller interfaced to PC. Figure 2 shows the schematic diagram.

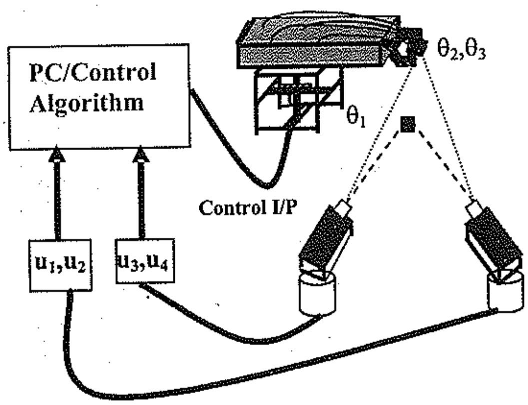



**Fig 2. Schematic Diagram Of Rhecs**

The visual processing phase returns the centroid coordinates of the specified object seen by the cameras as a pair of two-dimensional vectors ($u_1,u_2$) and ($u_3,u_4$), which are merged into a four-dimensional vector $u_{target} = (u_1,u_2,u_3,u_4)$. Only three degrees of freedom (DOF) of the robot hand arc considered. and the wrist is assumed to be rigid. The position of the robot hand is specified by a set of three joint angles $\theta = (\theta_1,\theta_2,\theta_3)$. To position the robot hand at the specified target point, the transformation $\Theta(u_{target})$ that maps the *input vector* $u_{target}$ to the corre-sponding joint angle *vector* $\theta$ is required. In a pick and place operation, the initial and final positions of the object are known in advance, hence their corresponding "image" coordinates become the reference signals for picking the object from the initial configuration and then placing it in the final configuration. Thus, the control problem is to find the joint angle coordinates corre-sponding to such reference signals that have to be actuated. This problem can be solved by a Model-based approach wherein the combined model of the robot hand and camera that can be used to com-pute the joint space coordinates from the camera coordinates. If the object position in Cartesian coordinates *(x, y ,z,)* is known, the inverse kinematics of the robot hand can be used to compute the joint angles required to reach that position. But since we only know the camera input coordinates ($u_1,u_2$) and ($u_3,u_4$), to complete the transformation, we need to derive a cam-era model that will obtain (x,y,z) from ($u_1,u_2$) and ($u_3,u_4$). Our approach is to derive a model that will transform the Carte-sian coordinates of the object to the camera input coordinates and to obtain the desired transformation by the inverse method.

### 4. The Robot Hand Model

The model of the 4-fingered robot hand developed in our laboratory has been derived from its link geometry. The link lengths were taken from its technical report. The kinematics were simplified by assuming the wrist joint to be rigid which is moved to main-tain its pose with respect to the base plane. This ensures that the hand behaves as a 3-DOF manipulator. The figure 3 shows the simplified structure of the robot hand. The manipulator model is a set of mathematical equations which map the space co-ordinates of the end effector to joint angles and vice-versa.

A robot arm is considered as an open linkage consisting of n rigid bodies interconnected by n one-degree of-freedom joints (Fig.3),and so the arm has n degrees of freedom (DOFs)and the system position is fully described by means of n joint coordinates. The numbers of DOFs required for executing a task are refereed to as dimension of operational frame or operational space . Let this frame be of dimension m .In this case one often says that the prescribed task requires m DOFs.

If m =n ,the robot is non redundant .If m<n one talks about a redundant robot. The arm has more DOFs than needed to execute the task. It means that the arm can move in different ways, still following the required trajectory of the end effector in the operational frame. This gives the possibility of additional maneuvering that can be used to get some benefits In some cases, this ability is used to avoid obstacles in the working space and some times for the minimization of a certain cost function. Is it necessary and possible to recognize some part of the robot as the redundancy? In many problems this is not needed and even not useful. If it is needed, then the general answer is that any m joints may be declared as the non redundant basic configuration ,while the others represent the redundancy. One possible choice is to consider the m joints with the highest inertias as the non redundant basic configuration, and the remaining n –m joints as the redundancy .Another choice would be to consider the joints that are rarely used (i.e. joints activated only in some specific situations, e.g. when avoiding obstacles)and declare them as redundancy.

The RHECS is a redundant robot as

(i) The wrist of the four fingered robot hand can be rotated in either directions through 360⁰ But this is not considered, to reduce the complexity of the manipulator model .

(ii) The intermediate joint is assumed to be fixed so that the proximal phalange and intermediate phalange constitutes a single joint angle $\theta_2$ as shown in figure 3

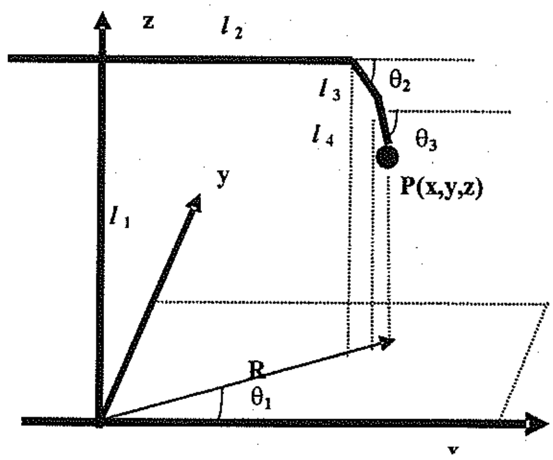



**Fig.3 Robot Hand Model**

The following equations model the geometry of the robot hand with respect to Figure 3.

The forward kinematics equations of the robot hand are $(\theta_1\theta_2\theta_3)$ ———> (x, y,z):

$$R = l_2 + l_3\cos\theta_2 + l_4\cos\theta_3$$

$$x = R\cos\theta_1$$

$$y = R\sin\theta_1$$

The inverse kinematics equations of the robot hand are (x, y,z) ———> $(\theta_1\theta_2\theta_3)$ :

$$\theta_1 = \tan^{-1}(y/x) \qquad (1)$$

$$\theta_2 = \cos^{-1}(s) \qquad (2)$$

where s is a root of $As^2 + Bs + C = 0$

$$\theta_3 = \sin^{-1}((b - l_4 \sin\theta 2)/l_3) \qquad (3)$$

$$A = 4a^2l_3^2 + 4b^2l_4^2 \quad B = -4al_3k$$

$$C = k^2 - 4b^2l_3^2, \quad k = a^2 + b^2 + l_3^2 - l_4^2$$

$$a = \mathrm{sqrt}(x^2+y^2) - l_2, \quad b = z - l_1$$

It is assumed that the wrist of the hand is rigid, assured by locking the joint for our purpose, and always holds the fingers parallel to the work table (the *XY* plane). The inverse kinematics of the manipulator are required for the calibration procedure to reach a point P(x, y,*z)*, given the joint angles($\theta_1$, $\theta_2$, $\theta_3$ ) as given in equations (1),(2),and (3).

The Camera Model

The camera model is a system of mathematical equations which gives the mapping between camera image co-ordinates obtained from the two images of CCD cameras $\bar{u}_{target} = (u_1, u_2, u_3, u_4)$ and the world co-ordinates of the end effector P $(x_w, y_w, z_w)$.

The process of mapping involves two steps :

1. Mapping the camera centered coordinates $(x_c, y_c, z_c)$ of the end effecter P to world coordinates $(x_w, y_w, z_w)$.
2. Mapping the camera image coordinates of end effector P obtained after processing the images obtained from two CCd cameras to camera centered coordinates $(x_c, y_c, z_c)$ using stereo correlation algorithm.

5.1 MAPPING FROM $(X_C, Y_C, Z_C)$ TO $(X_W, Y_W, Z_W)$:

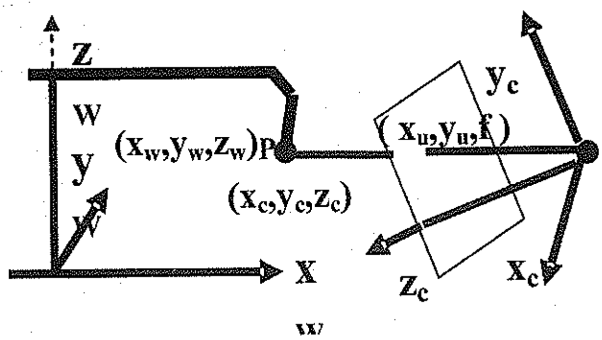



**Fig 4 The Geometry of A Camera**

From the figure 4, it can be stated that the origin of the camera-centered co-ordinate system$(x_c, y_c, z_c)$ coincides with the front nodal point of the camera and the $z_c$ axis coincides with the camera's optical axis. The image plane is assumed to be parallel to the $(x_c, y_c)$ plane at a distance f from the origin, where f is the effective fo-cal length of the camera. The relationship between the position of a point *P* in world coordinates, $(x_w, y_w, z_w)$ and the point's image in the camera's frame buffer, $(X_f, Y_f)$, is defined by a sequence of coordinate transformations. The first is a rigid-body transformation from the world coordinate system $(x_w, y_w, z_w)$ to the camera-centered coordinate system $(x_c, y_c, z_c)$. The transformation can be expressed as

$$[x_c\ y_c\ z_c] = [x_w\ y_w\ z_w]R + [T_x\ T_y\ T_z] \text{ - (4)}$$

R: 3 x 3 Rotational Matrix: Orientation of camera in world coordinate system.

T : Translational Position of Camera in world coordinate system.

<u>Calculation of Rotation matrix R:</u>

The calculation of Rotation matrix Ris done in two steps:

<u>Step1:</u> R is initially obtained by multiplying three matrices Rx,Ry,Rz

R=Rx . Ry . Rz

Where

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & COS\theta_X & SIN\theta_X \\ 0 & -SIN\theta_X & COS\theta_X \end{bmatrix}$$

$$R_Y = \begin{bmatrix} COS\theta_Y & 0 & -SIN\theta_Y \\ 0 & 1 & 0 \\ SIN\theta_Y & 0 & COS\theta_Y \end{bmatrix} \text{ and}$$

$$R_Z = \begin{bmatrix} COS\theta_Z & SIN\theta_Z & 0 \\ -SIN\theta_Z & COS\theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and where $\theta_x$ $\theta_y$ $\theta_z$ are the orientation of the camera coordinate system in x-axis,y-axis,z-axis respectively with respect to manipulator coordinate system.

<u>Step2:</u>

Whenever pan/ tilt is made to the camera coordinate system to get the manipulator and the object with its field the rotation matrix has to be changed.

When the camera is panned or tilted the change in orientation of the coordinate system attached to the camera can be quantified depending on the amount of angular rotation and depending on the geometry of the computer-controlled platform used. A number of different configurations of pan/tilt units are reported in the literature (see, for example,[13,14 ]).

If the initial pan and tilt are $\alpha_0$, $\beta_0$ and pan and tilt are $\alpha$ and $\beta$ then update rotation matrix as $R = P(\alpha,\beta) * R$

$$\begin{matrix} c\beta s(\alpha+\alpha_0)s\alpha_0 + c(\alpha+\alpha_0) & s(\alpha+\alpha_0)c\beta & c(\alpha+\alpha_0)s\alpha_0 - s(\alpha+\alpha_0)c\beta c\alpha_0 \\ s\beta s\alpha_0 & c\beta & s\beta c\alpha_0 \\ s(\alpha+\alpha_0)c\alpha_0 - c(\alpha+\alpha_0)c\beta s\alpha_0 & -c(\alpha+\alpha_0)s\beta & s(\alpha+\alpha_0)s\alpha_0 + c(\alpha+\alpha_0)c\beta c\alpha_0 \end{matrix}$$

**5. Stereo Correlation Algorithm:**

Mapping from $\overline{u}_{target} = (u_1, u_2, u_3, u_4)$ to $(x_c, y_c, z_c)$

From the camera image co-ordinates we can find the equation of the rays that formed the image of the target point on the camera focal planes.

According to the geometry of the image formation process outlined in figure 4, each ray passes through the points

$(x_d, y_d, f)$ where $(x_d, y_d)$ are the undistorted image co-ordinates obtained as using two transformations:

Using perspective projection of the point in camera coordinates to the position of its image in undistorted sensor plane coordinates, $(X_u, Y_u)$ as given below

$X_u = f*(x_c/z_c)$,

$Y_u = f*(y_c/z_c)$

f is effective focal length of camera

Using the transformation is from the undistorted posi-tion of the point' s image in the sensor plane to the true position of the point's image $(X_d, Y_d)$ that results from geometric lens distor-tion. $(X_d, Y_d)$ is described by

$$X_u = X_d(1+q\rho^2) \quad (5)$$

$$Y_u = Y_d(1+q\rho^2) \quad (6)$$

$$\rho = \mathrm{sqrt}\,(X_d^2 + Y_d^2) \quad (7)$$

where q is the coefficient of radial lens distortion

The $(0,0,0)$ - the optic center of the camera, in the corresponding camera co-ordinate system.

If the camera model was perfect these rays would intersect exactly at the target point. Due to errors in the estimation of camera parameters , these rays will not intersect and will be skew lines.

In this case the mid point of the line segment of the closest approach between the rays can be taken to the best estimate of the target point.

Let the parametric equations of the two rays (expressed in world co-ordinate system)

$$l_1 : \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + \begin{pmatrix} a \\ b \\ c \end{pmatrix} t$$

$$l_2 : \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} + \begin{pmatrix} p \\ q \\ r \end{pmatrix} t^1$$

The length of a line segment having one component on line $l_1$ and the other on line $l_2$ can be expressed as

$$\begin{aligned} f(t,t^1) = {} & (x_1 - x_2 + at - pt^1)^2 \\ & + (y_1 - y_2 + bt - qt^1)^2 \\ & + (z_1 - z_2 + ct - rt^1)^2 \quad \rightarrow (st1) \end{aligned}$$

The length of such a line segment is minimum when

$$\frac{\partial f}{\partial t} = 0 \ \ and \ \ \frac{\partial f}{\partial t^1} = 0 \quad \rightarrow (st\,2)$$

By solving $(st1)\ and (st\,2)$ we get t, $t^1$ and there by points $L_1 L_2$ and from that the mid point of $L_1$, $L_2$ Which is $P(x_c, y_c, z_c)$.

With the information of model of the system and estimated parameters, the required mapping between $u_{target}$ *to* $\theta$ can be obtained as follows:

From the camera image coordinates $u_{target}$, we can find the equation of the rays that formed the image of the target point on the camera focal planes. From Fig. 4, each ray passes through the points ($X_d$, $Y_d$, f), where ($X_d$,$Y_d$) are the undistorted image coor-dinates obtained from $u_{target}$ using equations (5)(6)(7), and (0,0,0) (the optic center of the camera) in the corresponding camera coordinate system. From the estimated camera parameters, we know the transformation from each of the camera coordinate systems to the world coordi-nate system. The transformation can be obtained by applying (4). By using the manipulator's inverse kinematics equations (1)-(3), the joint angles $\theta_1$,$\theta_2$,$\theta_3$ can be computed and the target position can be reached.

## 6. Image Processing Techniques Applicable In Getting $\bar{u}$ From The Images Obtained From Ccd Cameras

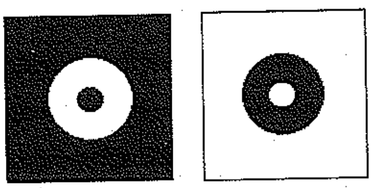
There are many image processing techniques to extract the required information from the image as specified in [15].

1. The image acquired by the frame grabber card and then it is processed by the PC .The region corresponding to the end effecter in the image is separated by the

(i) Thresholding

(ii) Filtering operations

Thresholding is done using the histogram of intensities of image where a priori information about the intensity of the end effecter in the image is used. The centroid of the region is taken to be the end effecter position seen by the camera. To keep the image segmentation problem simple a high contrast environment is used. The same technique is used to obtain the image coordinates.



2. Several artificial visual features consisting of black and white centered cues such as the following Were attached to the manipulable and non manipulable bodies. These types of features are easily detected using an algorithm that considers the change in the black and white pattern present in the images obtained with the ccd cameras.

## 7. Structural Details Of Robot Hand

The design of mechanical structures for GRH [Anu et. al. ] incorporates four digits: three fingers and one thumb, as shown in figure 5. Three digits are positioned on the base of an inverted triangle like structure and one digit is at the center of base of the triangle, since this geometry leads naturally to stable finger contact positions for an enclosing grasp [Mason, Salisbury]. Each finger consists of three rigid links (the proximal, intermediate and distal phalanges) constructed from cylindrical pipes. The phalanges are connected by three joints (the proximal, intermediate, and distal joints) which have parallel axes of rotation and are responsible for curling the finger tip toward the palm. The thumb is similarly configured except that it consists of only two rigid links (the proximal and distal) connected by joints (proximal and distal). The phalange lengths, $L_1$,$L_2$, and $L_3$ phalange width, w, and the distance between the finger and the thumb, $d_{ft}$, were selected as shown in Table 2.

The distance between the two fingers, $d_{ff}$, was computed to be the distance between the fourth and index finger to give the largest human finger displacement. The thickness of the supports, t, was chosen for availability and strength. To reduce construction costs, we configured the three fingers identically and the thumb to have identical link lengths as the proximal and intermediate links of the fingers. Each digit is controlled by three individual strings which are routed through phalanges as shown in figure 6.

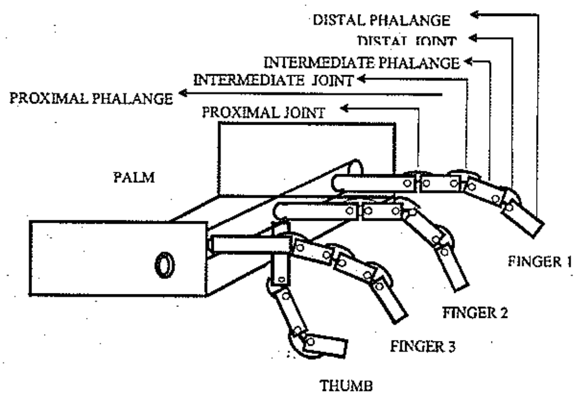



**Fig. 5. Grasper Robotic Hand**

**Table 2: GRH Parameters**

| **Description** | **Variable** | **Value (Cm)** |
|---|---|---|
| Distance from Palm to Proximal Joint | L0 | 4 |
| Distance from Proximal to Intermediate joint | L1 | 2.8 |
| Distance from Intermediate to Distal Joint | L2 | 2.8 |
| Distance from Distal Joint to Digit Tip | L3 | 2.8 |
| Distance from End to Start of Phalange | $l$ | 0.7 |
| Each Phalange Width | w | 1.2 |
| Distance between Fingers | $d_{fi}$ | 9 |
| Support Thickness | t | 0.1 |
| Each Double Revolute Joint Width | $R_j$ | 1.6 |

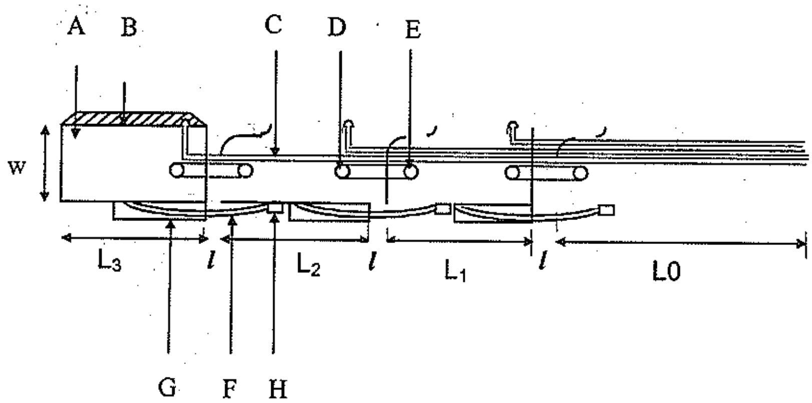
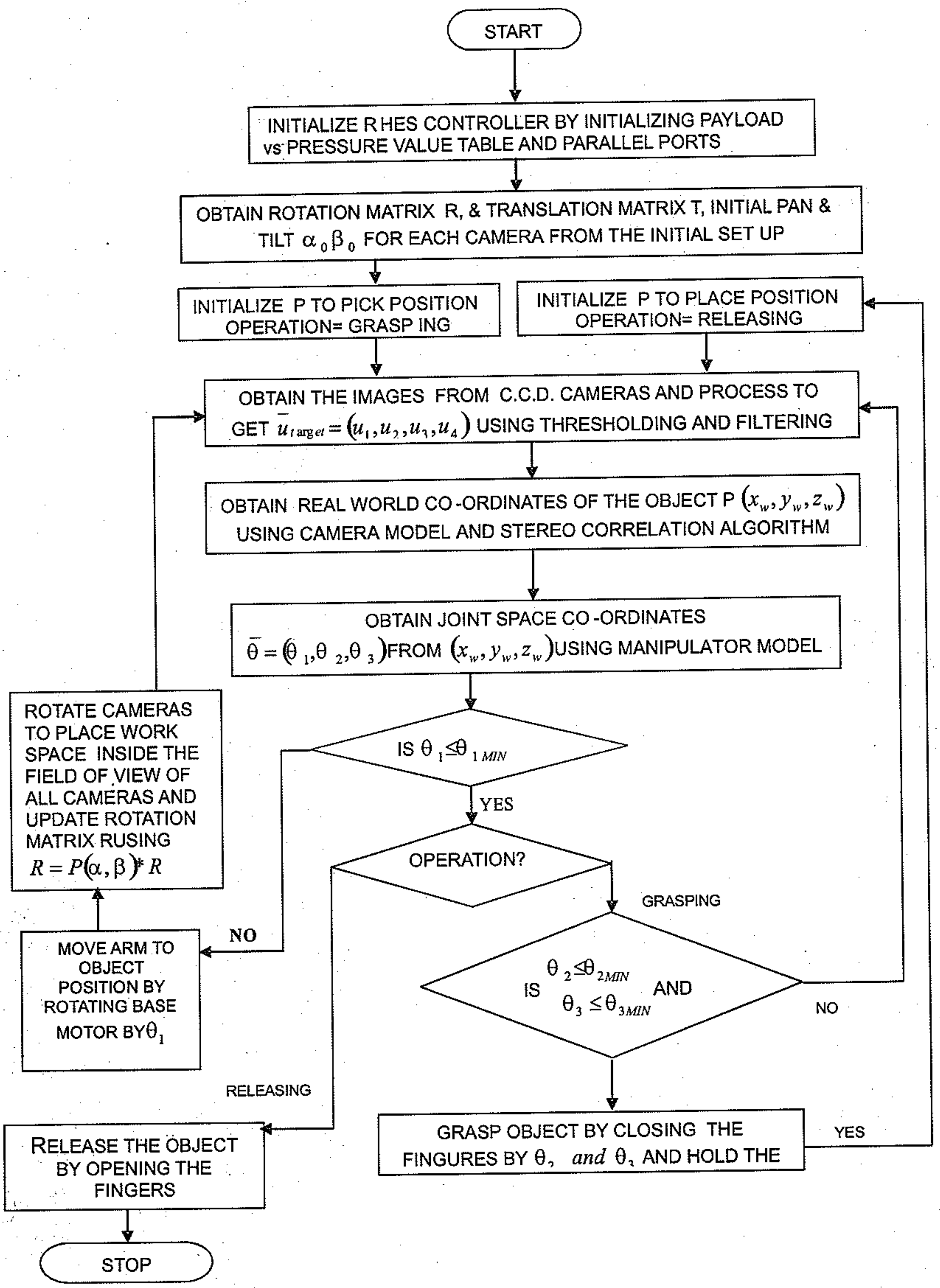



**Fig.6 Finger Structure**

A - PHALANGE ALUMINUM PIPE )
B - OUTER MASKS
C - CONNECTING STRING
D - FIXED RIVET JOINT
E - DOUBLE REVOLUTE JOINT
F - STEEL STRIP STRING
G - CLAMP ( THROUGH SPRING MOVING )
H - SPRING FIXED AREA
w - DIAMETER OF THE TUBE
l - DISTANCE BETWEEN PHALANGES

The double revolute joints through strings/tendons act as routers to create angular displacement of the links, but do not transfer any torque to the joints. Each phalange to phalange is connected by a double revolute joint. The spring plates brazed on top of each phalange control the digit in opposite direction when the string is pulled in opposite direction. One end of each tendon is attached to the lower end of each phalange and the other is wound about a pulley attached to the shaft with two rollers(pulley roller and brake roller) attached at its near ends. The pulley roller gets motion via an intermediate roller located in between pulley roller and main roller fixed on a shaft that is attached to a single, remotely-located DC stepper motor. The intermediate roller is attached through vertical plate to solenoid that controls motion transmission from main roller to pulley roller. Therefore, the finger joint moves when the pulley roller gets motion. The pulley roller moves when the intermediate roller upon solenoid's activation is intact with pulley roller and main roller which is driven by the stepper motor. The brake roller/drum is located between brake plates structure that is attached to solenoid. Applying brake plates to brake drum to *lock* finger joint motion is controlled by the solenoid. The strings in each finger are wound in both directions depending on the stepper motor direction. For practical reasons, aluminum is selected as the material for the pipes(fingers), arm, wrist and palm since it is strong, rigid, lightweight, relatively inexpensive, and easy to machine.


START
INITIALIZE R HES CONTROLLER BY INITIALIZING PAYLOAD vs PRESSURE VALUE TABLE AND PARALLEL PORTS
OBTAIN ROTATION MATRIX R, & TRANSLATION MATRIX T, INITIAL PAN & TILT $\alpha_0 \beta_0$ FOR EACH CAMERA FROM THE INITIAL SET UP
INITIALIZE P TO PICK POSITION OPERATION= GRASP ING
INITIALIZE P TO PLACE POSITION OPERATION= RELEASING
OBTAIN THE IMAGES FROM C.C.D. CAMERAS AND PROCESS TO GET $\bar{u}_{target} = (u_1, u_2, u_3, u_4)$ USING THRESHOLDING AND FILTERING
OBTAIN REAL WORLD CO-ORDINATES OF THE OBJECT P $(x_w, y_w, z_w)$ USING CAMERA MODEL AND STEREO CORRELATION ALGORITHM
OBTAIN JOINT SPACE CO-ORDINATES $\bar{\theta} = (\theta_1, \theta_2, \theta_3)$ FROM $(x_w, y_w, z_w)$ USING MANIPULATOR MODEL
IS $\theta_1 \leq \theta_{1MIN}$
YES
NO
OPERATION?
GRASPING
RELEASING
IS $\theta_2 \leq \theta_{2MIN}$ AND $\theta_3 \leq \theta_{3MIN}$
NO
YES
ROTATE CAMERAS TO PLACE WORK SPACE INSIDE THE FIELD OF VIEW OF ALL CAMERAS AND UPDATE ROTATION MATRIX RUSING $R = P(\alpha, \beta) * R$
MOVE ARM TO OBJECT POSITION BY ROTATING BASE MOTOR BY $\theta_1$
GRASP OBJECT BY CLOSING THE FINGURES BY $\theta_2$ and $\theta_3$ AND HOLD THE
RELEASE THE OBJECT BY OPENING THE FINGERS
STOP

## 9. Experiments

The RHECS consists of 4-fingered robot hand developed in our laboratory with PC controller, two pan-tilt heads, two WAT 201A color CCD cameras, Color Image Processing Frame Grabber card attached to PC via PCI bus. All image processing and vision control calculations are performed on the PC. Cartesian velocities are sent to the PC that converts them into coordinated joint motions at 140Hz. The cameras are placed approximately 90cm from the target along the x axis and 30cm apart along the y axis. They are oriented to point back along the x axis of the robot hand. Figure 7 shows the block diagram of the implemented system- Visual Control System and robot hand system controlled by robot controller directed by PC performing tracking and visual control.

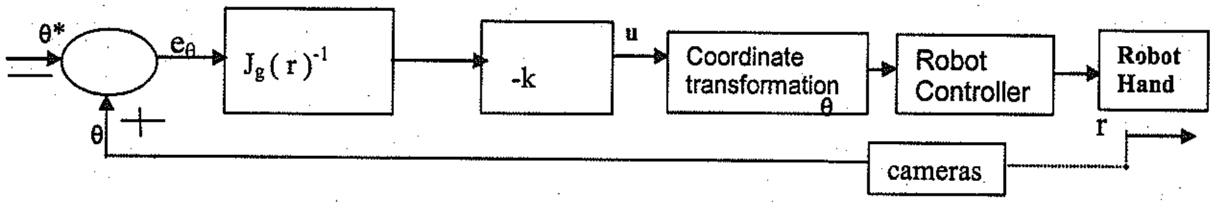



**Fig 7. Block Diagram Of Visual Control And Robot Hand System**

Let r be Ref. Point of Hand, r* be Cartesian Set Point of target and g be the Mapping -Cartesian Coordinates to Stereo Image Coordinates. θ = g(r) and θ * = g(r*), Error $e_\theta = \theta - \theta^*$ Error Integrating Subsystem: $z_\theta = e_\theta$ Control input $u = f(r, z_\theta, e_\theta)$ to stabilize the system,

$\dot{z}\theta = e\theta \qquad \dot{e}\theta = J_g(r)\, u \qquad \dot{r} = u$

$J_g(r)$ is the Jacobian of g at r

$z\theta = e\theta \qquad e\theta = J_g(r)\, u \qquad r = u$

$J_g(r)$ is the Jacobian of g at r.

System calibration was performed by tracking the robot through a series of known motions and performing optimization to determine the system calibration parameters relative to robot base coordinates[12].

Model IMAGE PRO-EXPRESS Windows based image processing software is used for obtaining visual data. The tracking control program is written in C++ that provides visual input for the coordinate transformation that in turn becomes the input for the controller to direct the robot hand for reaching the object under consideration. The system provides fast edge detection on a memory-mapped frame buffer. It also supports simultaneous tracking of multiple edge segments, and can also enforce constraints among segments. The experiments are based on tracking corners formed by the intersection of two line segments.

The hand successfully tracked and grasped a variety of objects, each having different shape, size, surface conditions, and hardness. Once the objects were crudely placed in the workspace of robot hand, the hand was issued a grasp command. The grasping is said to be successful if the hand correctly reaches the object through visual control loop and held the object autonomously.

## 10. Conclusions

A model-based approach to visual-motor coordination is implemented on a 4-fingered robot hand with a 7-degree - of - freedom. This approach performed well in the center of the workspace, giving an accuracy of within 4 mm , which is close to the maximum achievable with the hand. However, near the edges of the workspace, this model-based algorithm's accuracy decreases markedly. This is due to the fact that the camera model is also least accurate in this region with a tracking error of about 6mm.

The vision processing and control computation system does not use spe-cial hardware (other than a standard frame grabber) and can be run on common PC. At the current rate of progress, frame-rate (50/60 Hz) servoing will be feasible. Since the entire system, including image processing, runs in software, moving to a newer or more powerful system is a matter of recompiling. Furthermore, since the robot and camera interfaces are extremely generic, the system is also easily ported to other robot or imaging hardware. As reported, the current system can position the fingers to within a few millimeters relative to a target. This positioning accuracy could easily be improved by changing the camera configuration to a wider baseline, or by improving the image-processing to be more accurate.

## 11. References


[1] G. Hager and S, Hutchinson, editors, Proc. the Workshop on Visual Servoing, May 1994.

[2] K. Hashimoto. editor, Visual Servoing. World Scientific, 1994.

[3] P.Liang, Y.L.Chang, and S. Hackwood, *"Adaptive self-calibration of vision-based robot systems"*, in IEEE trans. on Systems, Man, Cybernetics, Vol.19, No.4,pp.811-824, 1989

[4] J. Feddema, C. Lee. and O.Mitchell, *"Weighted Selection of Image Features for Resolved Rate Visual Feedback Control,"* IEEE Transactions oк Robotics and Automation. 7(1):31-47, February 1991.

[5] K. Hashimoto. *"LQ Optimal and Nonlinear Approaches to Visual Servoing,"* in K. Hashimoto, editor. Visual Servoing, pp. 165-198. World Scientific, 1994.

[6] S. Hutchinson, *"Exploiting VisualConstraints in Robot Motion Planning."* in Proc. IEEE International Conference on Robotics and Automata pp. 1722-1727. IEEE Computer Society Press, 1991.

[7] P. Rives, F. Chaumette, and B. Espiau, *"Positioning of a Robot with Respect to an Object, Tracking It and Estimating Its Velocity by Visual Servoing,"* in Proc. IEEE International Conference on Robotics and Auto-mation, pp. 2248-2254. IEEE Computer Society Press, 1991.

[8] P. Alien, B. Yoshinu, and A. Timcenko, *"Hand-Eye Coordination for Robotic Tracking and Grasping."* in K, Hashimoto, editor, Visual Servoing, pp. 33-70. World Scientific, 1994.

[9] A. Rizzi and D.E, Koditschek, *"Further Progress in Robot Juggling: The Spatial Two-Juggle,"* in Proc. IEEE International Conference on Robotics and Automation, pp. 919-924. IEEE Computer Society Press, 1993.

[10] R.L. Anderson, *"Dynamic Sensing in a Ping-Pong Playing Robot,"* IEEE Transactions on Robotics and Automation, 5(6):723-739, 1989.

[11] N- Maru, H. Kase, A. Nishikawa, and K Miyazaki. *"Manipulator Control by Visual Servoing with the Stereo Vision,"* in Proc. IEEE International Conference on Intelligent Robots and Systems, pp. 1866-1870. IEEE Com-puter Society Press. 1993.

[12] C. Lu, E.J. MJolsness, and G.D. Hager, *"Online Computation of Exterior Orientation with Application to Hand-Eye Calibration,"* DCS RR1046, Yale University, New Haven, CT, August 1994.

[13] Tebo A.Active vision unlocks the door to robotics.OE Reports. Published by SPIE,The international Society for Optical Engineering,No.107,1992.

[14] Pelco Products,Bid ordering specification software (BOSS).CD ROM catalog.Pelco,300 West Pontiac Way,Clovis,CA 93612-5699,1998.

[15] J. craig *"Introduction to Robotics"*. Reading : Addison –Wesely,1986.

[16] *"PC Controlled Four Fingered Robotic Hand"* (International Seminar) Anu Gokhale, P. Seetharamaiah & KNageswara Rao, Proceedings of the 16 th International Conference on CAR & FOF, Advanced Mfg.& Engg. Center, Dept. of Mech. Engg., Univ. of West Indies, St. Augustien, Trinidad and Tobago, May 2000.

**Author's Biography**

**P. S. Ramaiah** received B.E. degree in Electronics and Communications Engineering in 1974 from Andhra University, Visakhapatnam, M.E. degree in Instrumentation and Control Engineering in 1977 from S.V. University, Tirupati and joined in Naval Science and technological Laboratory in 1977. In 1979, he joined as lecturer in the department of Electrical Engineering , Andhra University, Visakhapatnam, and received Ph.D. degree in Computer science and Engineering in 1990 from Andhra University, Visakhapatnam. He is presently working as Professor in the department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam since 1998. His present research interests are VLSI and Embedded Systems, Robotics, and Bionic implants.

**K. N. Rao** received his B.E ( Electronics & Communication Engg. ) from Karnatak University in 1977 and subsequently both M.Tech. ( Compuer Science & Technology ) and Ph.D. ( Computer Science & Systems Engg. ) from College of Engineering Andhra University. He joined Andhra University as a research scholar in December 1977 . He was on lien from Andhra University for some time and worked as Professor and Head of the Department for both Computer Science & Engineering and Information technology at M V G R college of Engineering, Vizianagaram. Apart from Robotics he is currently working in the area of Network Security.