Design and Development of 5-DOF Color Sorting Manipulator for Industrial Applications

Atef. A. Ata, Sohair F. Rezeka, Ahmed El-Shenawy, Mohammed Diab

Abstract—Image processing in today's world grabs massive attentions as it leads to possibilities of broaden application in many fields of high technology. The real challenge is how to improve existing sorting system applications which consists of two integrated stations of processing and handling with a new image processing feature. Existing color sorting techniques use a set of inductive, capacitive, and optical sensors to differentiate object color. This research presents a mechatronic color sorting system solution with the application of image processing. A 5-DOF robot arm is designed and developed with pick and place operation to act as the main part of the color sorting system. Image processing procedure senses the circular objects in an image captured in real time by a webcam fixed at the end-effector then extracts color and position information out of it. This information is passed as a sequence of sorting commands to the manipulator that has pick-and-place mechanism. Performance analysis proves that this color based object sorting system works accurately under ideal condition in term of adequate illumination, circular objects shape and color. The circular objects tested for sorting are red, green and blue. For non-ideal condition, such as unspecified color the accuracy reduces to 80%.

Keywords—Robotics manipulator, 5-DOF manipulator, image processing, Color sorting, Pick-and-place.

I. INTRODUCTION

INDUSTRIAL automation and robotics are at a high demand in the industry as both of them directly affect the growth of the development. Quality and flexibility of the product is the important criteria of the industry. Use of industrial robots is leading automation industry to another transition. In 1980's robots were used for general tasks such as machine tending, material transfer, painting, welding etc. Autonomous robots with sophisticated sensors are used in such industries with accurate and precise work output which directly results in the growth of the industry. In automation Industrial robots are programmed for a single task using the sensory information. A vision system is considered to be the most sophisticated sensor in the present automation industry.

The most common technology used at present in the industry is image processing. Due to the advent of powerful

Atef A. Ata is currently with the Department of Mechatronics Engineering, Mevlana University, Konya, Turkey (Corresponding Author; e-mail: atef@ mevlana.edu.tr).

Sohair F. Rezeka is currently a visiting professor at the Department of Mechatronics Engineering, Faculty of Engineering, Arab Academy of Science and Technology, Alexandria Egypt (e-mail: srezeka@yahoo. com co).

Ahmed El Shennawy is with the Electrical Engineering Department, Faculty of Engineering, Arab Academy for Science and Technology (e-mail: Ahmed.elshenawy@aast.edu).

Mohammed Diab is a Master Student with the Mechatronics Engineering Department, Faculty of Engineering, Arab Academy for Science and Technology (e-mail:mohammed-diab@live.com).

cameras, computers, controllers for controlling the machines and sophisticated tools image processing has become the most powerful emerging technology.

Image processing is basically improving and enhancing the images taken in daily life using cameras considered as vision sensors for various applications. In the last few decades different techniques have been developed for detecting the objects using vision systems. Color histograms were first proposed by Swain and Ballard [1] as a first approach for object recognition. Other features such as orientation, gradient magnitude were added to the histograms by Schiele and Crowley [2]. These inventions helped in changing the rotation, direction and deformation but did not help in object recognition. Schneiderman and Kanade [3] were the first to introduce categorization of objects in natural scenes using wavelet transform coefficients. Wavelet transforms decompose the image into a group of salient features wavelet transforms by reducing the redundancy of the pixels. When relationship between the parts of the object is considered rather than just the appearance, the most complex methods are used. Part based representation of objects detected in gray images was developed by Agarwal et al. [4].

In 2005, Ata et al. [5] presented an automated robotic cell designed specifically for color sorting in the visible range (400–780nm) to detect and grip different color specimens to perform necessary separation. In 2010, Khojastehnazhand et al. [6] developed and implemented an efficient algorithm for grading lemon fruits in visual basic environment. In 2010, Sato [7] presented an intelligent robot arm equipped with a C-MOS camera capable of autonomous grasping by using a new type of image processing. Senthilnathan et al. [8] presented an attempt to solve a typical problem that could be faced by a bearing manufacturing industry incorporating vision assisted robot based sorting of annular metallic ball bearings. In 2012, Vindhya et al. [9] presented the detection of different colors through image processing technique using MATLAB. In 2012, Kroeger and Padial suggested [10] a new visual servo control scheme that uses an online trajectory generator as an intermediate layer between image processing algorithms and robot motion controllers. In March 2013, Rokunuzzaman and Jayasuriya [11], presented their research about sorting of tomatoes keeping in view the techniques developed for object recognition. MATLAB has the most powerful toolbox for image improving, enhancing and categorizing different images using different features such as color, dimensions and texture of the object. In this paper the detection of different colors is done through image processing technique using MATLAB.

World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:7, No:12, 2013

The goal of the paper is to implement a color-based object sorting system using the designed and implemented five-degrees-of-freedom robot arm. This robot arm was commanded by a PC running MATLAB via a serial port. Small different-colored objects were placed inside the robot's workspace and imaged by a webcam also connected to the PC (high level control). Using MATLAB to process the image, the objects were detected, located, and ranked according to their color. The robot was then instructed to pick each object up and arrange them according to the visible color spectrum. The robot consists of nine servos DC motors for the base and links and for the gripper as well. The Arduino Uno R3 controller is used to control the motion of the robot.

II. MODELING AND MECHANICAL DESIGN

In many industrial applications of robots 5 degrees of freedom are sufficient to carry out the industrial tasks effectively. Many common industrial manipulators are, therefore, constructed with a 3-DOF arm and 2-DOF wrist with roll and pitch rotations. One such manipulator with an articulated arm is shown in Fig. 2. All the five axes of manipulator are revolute.
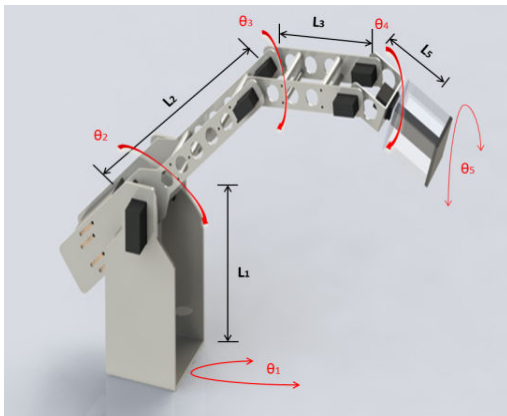

Fig. 1 The developed five-degree of freedom industrial manipulator

Note that the robot configurations and the assigned frames based on Denavit-Hartenberg notations are shown in Fig. 2. Frame {3} is the frame of arm end point as well as the frame wrist base [12]. The joint-link parameters for the manipulator based on the frame assignment are identified and tabulated in Table I.

TABLE I
D-H PARAMETERS OF THE ARTICULATED ARMS

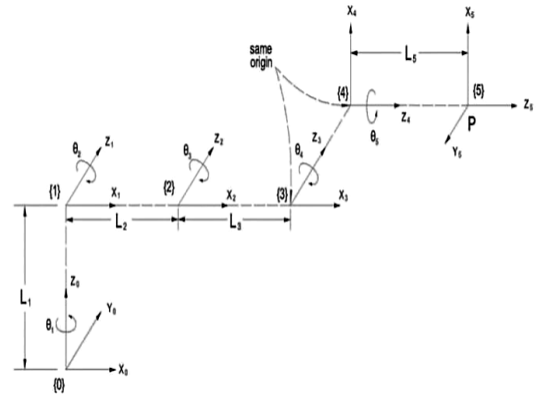| Link No.	θ	d	a	α
Link 1	θ_1	L_1	0	-90
Link 2	θ_2	0	L_2	0
Link 3	θ_3	0	L_3	0
Link 4	θ_4-90	0	0	-90
Link 5	θ_5	L_5	0	0

Fig. 2 Frame assignment for 5-DOF industrial manipulator [12]

and the transformation matrices are given by:

$$^0T_1 = \begin{pmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

$$^1T_2 = \begin{pmatrix} C_2 & -S_2 & 0 & L_2 \cdot C_2 \\ S_2 & C_2 & 0 & L_2 \cdot S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

$$^2T_3 = \begin{pmatrix} C_3 & -S_3 & 0 & L_3 \cdot C_3 \\ S_3 & C_3 & 0 & L_3 \cdot S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

$$^3T_4 = \begin{pmatrix} S_4 & 0 & C_4 & 0 \\ -C_4 & 0 & S_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$^4T_5 = \begin{pmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & L_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

The transformations matrix for the positioning mechanism is given by:

$$^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3 =$$
$$\begin{bmatrix} C_1 \cdot C_{23} & -C_1 \cdot S_{23} & -S_1 & C_1 \cdot (L_2 \cdot C_2 + L_3 \cdot C_{23}) \\ S_1 \cdot C_{23} & -S_1 \cdot S_{23} & C_1 & S_1 \cdot (L_2 \cdot C_2 + L_3 \cdot C_{23}) \\ -S_{23} & -C_{23} & 0 & L_1 - L_2 \cdot S_2 - L_3 \cdot S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:7, No:12, 2013

and the transformation matrix for the wrist orientation is given by

$$^3T_5 = {}^3T_4{}^4T_5 = \begin{pmatrix} S_4 \cdot C_5 & -S_4 \cdot S_5 & C_4 & L_5 \cdot C_4 \\ -C_4 \cdot C_5 & C_4 \cdot S_5 & S_4 & L_5 \cdot S_4 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

The overall transformation matrix for the manipulator is, therefore,

$$^0T_5 = {}^0T_3{}^3T_5 =$$
$$\begin{bmatrix} C_1 \cdot S_{234} \cdot C_5 + S_1 \cdot S_5 & -C_1 \cdot S_{234} \cdot S_5 + S_1 \cdot C_5 & C_1 \cdot C_{234} & C_1 \cdot (L_2 \cdot C_2 + L_3 \cdot C_{23} + L_5 \cdot C_{234}) \\ S_1 \cdot C_{234} \cdot C_5 - C_1 \cdot S_5 & -S_1 \cdot S_{234} \cdot S_5 - C_1 \cdot C_5 & S_1 \cdot C_{234} & S_1 \cdot (L_2 \cdot C_2 + L_3 \cdot C_{23} + L_5 \cdot C_{234}) \\ -C_{234} \cdot C_5 & C_{234} \cdot S_5 & -S_{234} & L_1 - L_2 \cdot S_2 - L_3 \cdot S_{23} - L_5 \cdot S_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} (8)$$

This represents the kinematic model of manipulator.

The general form of the overall transformation of the manipulator in symbolic form is given by:

$$^RT_H = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9)$$

The inverse kinematics analysis aims at finding the joint variables for a specific position and orientation of the end effector. To do this the decoupling techniques is used as follows [13].

$$^0T_1{}^{-1}T_E =$$
$$\begin{pmatrix} C_1 \cdot n_x + S_1 \cdot n_y & C_1 \cdot o_x + S_1 \cdot o_y & C_1 \cdot a_x + S_1 \cdot a_y & C_1 \cdot p_x + S_1 \cdot p_y \\ -n_z & -o_z & -a_z & -p_z + L_1 \\ -S_1 \cdot n_x + C_1 \cdot n_y & -S_1 \cdot o_x + C_1 \cdot o_y & -S_1 \cdot a_x + C_1 \cdot a_y & -S_1 \cdot p_x + C_1 \cdot p_y \\ 0 & 0 & 0 & 1 \end{pmatrix} (10)$$

And the right hand side of matrix equation is given by:

$$^1T_5 = {}^1T_2{}^2T_3{}^3T_4{}^4T_5 =$$
$$\begin{pmatrix} S_{234} \cdot C_5 & S_{234} \cdot S_5 & C_{234} & L_5 \cdot C_{234} + L_3 \cdot C_{23} + L_2 \cdot C_2 \\ -C_{234} \cdot C_5 & C_{234} \cdot S_5 & S_{234} & L_5 \cdot S_{234} + L_3 \cdot S_{23} + L_2 \cdot S_2 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (11)$$

or

$$\begin{pmatrix} C_1 \cdot n_x + S_1 \cdot n_y & C_1 \cdot o_x + S_1 \cdot o_y & C_1 \cdot a_x + S_1 \cdot a_y & C_1 \cdot p_x + S_1 \cdot p_y \\ -n_z & -o_z & -a_z & -p_z + L_1 \\ -S_1 \cdot n_x + C_1 \cdot n_y & -S_1 \cdot o_x + C_1 \cdot o_y & -S_1 \cdot a_x + C_1 \cdot a_y & -S_1 \cdot p_x + C_1 \cdot p_y \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} S_{234} \cdot C_5 & S_{234} \cdot S_5 & C_{234} & L_5 \cdot C_{234} + L_3 \cdot C_{23} + L_2 \cdot C_2 \\ -C_{234} \cdot C_5 & C_{234} \cdot S_5 & S_{234} & L_5 \cdot S_{234} + L_3 \cdot S_{23} + L_2 \cdot S_2 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (12)$$

Comparing the corresponding elements of equations yields:

$\theta_1 = \text{Atan2 } (p_y, p_x)$
$\theta_5 = \text{Atan2 } (-S_1 n_x + C_1 n_y, -S_1 o_x + C_1 o_y)$
$\theta_{234} = \theta_2 + \theta_3 + \theta_4 = \text{Atan2 } (-a_z, C_1 a_x + S_1 a_y)$
$\theta_2 = \text{Atan2 } [\tan \theta_5 (C_1 n_x + S_1 n_y) + C_1 o_x + S_1 o_y, n_z \tan \theta_5 + o_z]$
$\theta_3 = \text{Atan2}$
$$\begin{pmatrix} -C1S2px - S1S2py - C2(pz+L1) + L5(C1S2ax + S1S2ay + C2az), \\ C1C2px + S1C2py - S2(pz-L1) - L2 - L5(C1C2ax + S1C2ay - S2az) \end{pmatrix}$$
$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (13)$$

III. JOINT SPACE TRAJECTORY PLANNING

In this application, the initial location and orientation of the robot is known, and the inverse kinematic equations are used to find the final joint angles for the desired position and orientation. Now consider one of the joints, which at the beginning of the motion segment t_i is at θ_i and its final position θ_f is at time t_f [13]. One way to do this is to consider a polynomial to plan the trajectory, such that the initial and final boundary conditions match and that the velocities at the beginning and the end of the motion segment are zero (or known values). These sixth pieces of information allow solving for the six unknowns of the fifth degree polynomial in the form:

$$\theta (t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5 \quad (14)$$

The initial position $\theta (t_i) = c_0 = \theta_i$ and final position is 90°. For rest-to-rest maneuvering, $c_1 = 0$. For time interval t_f of 5 seconds, the trajectory coefficients are given by:

$\theta(t_i) = c_0 = \theta_i$
$c_1 = 0$ m/s
$c_2 = 5$ m/s
$t_f = 5$ sec. and acceleration $= \pm 0.2$ m/s^2

The variant velocities during certain time interval are shown in the following figures. Fig. 3 displays the velocity of the base $\dot\theta_1$ as time varies from 0 to 5 seconds and the angle of rotation θ_1 is changed from 0 to 180 degree. The maximum velocity of base is 65 RPM.

World Academy of Science, Engineering and Technology
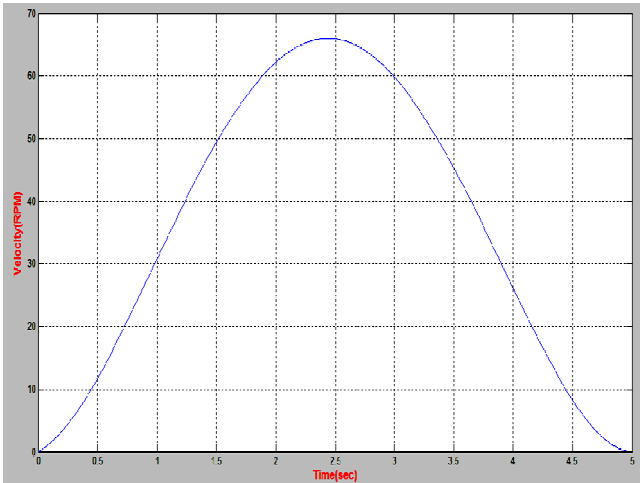International Journal of Mechanical and Mechatronics Engineering
Vol:7, No:12, 2013

Fig. 3 Velocity of both base motor and gripper

For link 2, the maximum velocity for the joint space trajectory planning is 32.5 RPM as shown in Fig. 4, when the angle θ_2 varies from 0 to 90 degrees at the time interval from 0 to 5 seconds.
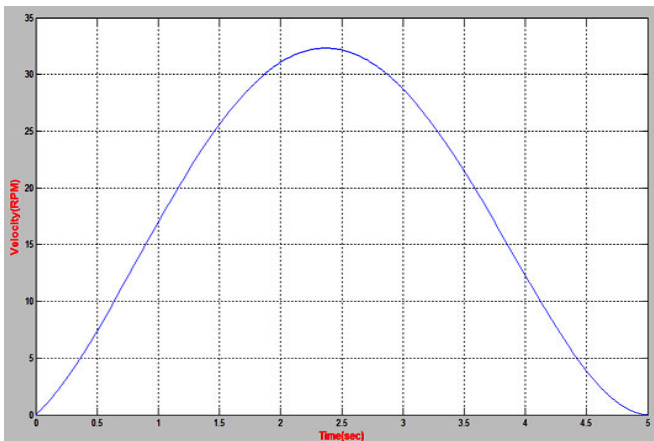

Fig. 4 Velocity of link 2 and link 4

For link 3, the maximum velocity of joint space trajectory planning is 46 RPM as shown in Fig. 5, when the angle θ_3 varies from 0 to 135 degrees at the same previous time interval. For link 4, the maximum velocity of joint is the same as that of link 2 because both have the same range of rotation (90°) as indicated in Table II. For the gripper; the range of rotation is the same as the base and the maximum velocity is the same of velocity of the base.
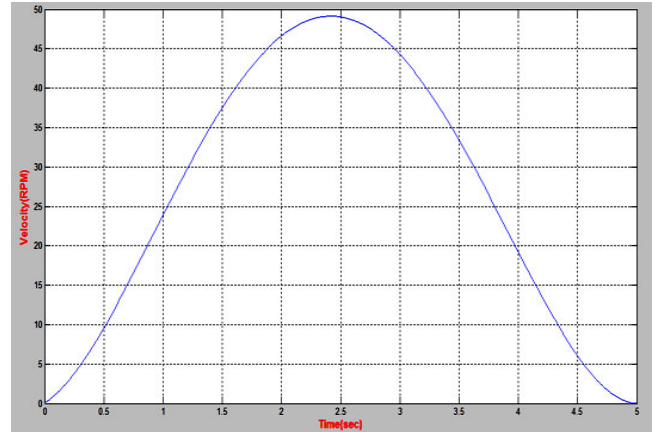

Fig. 5 Velocity of link 3

IV. TORQUE SIMULATION USING SOLIDWORKS

The design software SolidWorks is used to study the dynamic analysis of articulated robot for each joint. The input is the velocity of each link. The variant velocity for each joint is calculated using the joint space trajectory planning discussed in the previous section. The input average velocity can be considered as 62.5 RPM for each joint. The type of the actuator which is used in this analysis is DC Servomotor. The range of motion achieved by each link may be taken as shown in Table II.

TABLE II
LIMITS OF MOTION FOR DIFFERENT JOINTS

θ_1	0° to 180°
θ_2	0° to 90°
θ_3	0° to 135°
θ_4	0° to 90°
θ_5	0° to 180°

The output torque for the motors is illustrated in the following Figs. 6-9. Fig. 6 presents the torque of the link 2 where the maximum output motor torque is 6 Nm. Since two servo motors are attached at link 2, this means that the value of each servomotor = 3 Nm.
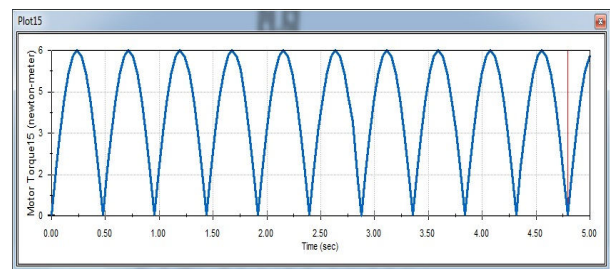

Fig. 6 Output motor torque of link 2

To reduce the torque of the link 2 as shown in Fig. 7, fixed balancing weight and springs may be used. The resulted maximum torque equals 0.15 Nm. Fig. 8 depicts the torque of link 3, and the maximum output torque is 2.7 N.m. Two servo motors will be attached to link 3. Fig. 9 shows the torque of link 4 and the base as well. The maximum output torque is 0.9

World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:7, No:12, 2013

Nm. Also two servo motors will be used. Table III shows the values of lengths and masses were measured using the SolidWorks model.
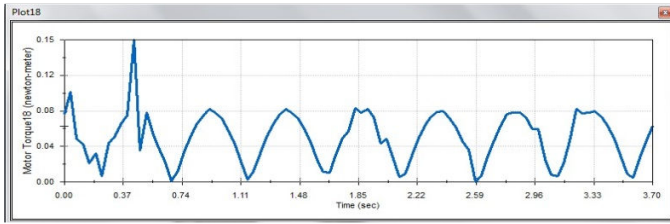

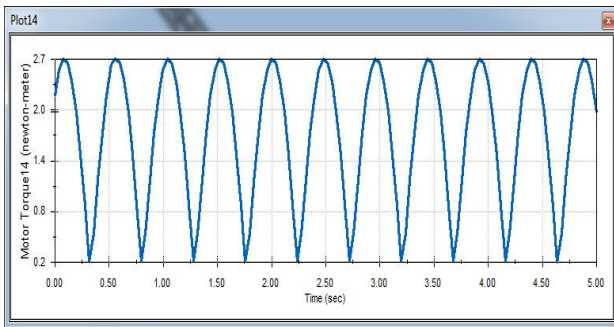Fig. 7 Torque of link 2 after adding a balancing weight
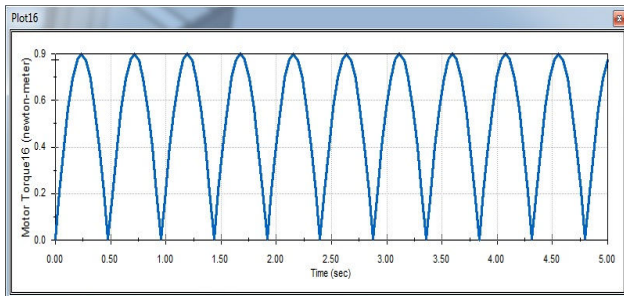

Fig. 8 Output torque of link 3

Fig. 9 Output torque of link 4 and the base

This simulated torques will be considered as a rough estimate for the torques of the servo motors. Radial bearing is used to overcome the vibration of manipulator during the motion and the thrust bearing is used to reduce the torque on the base motor and to provide smooth motion by reducing friction.

TABLE III
ROBOT PARAMETERS

Link	Mass(kg)	Dimensions(mm)
Base	1.32	706.85x140x6
Link 1	0.2	510x80x3
Link2	0.06	240x40x3
End effector	0.05	90

V. DEVELOPMENT OF IMAGE PROCESSING

The main aim of this paper is to present a method for visual servo control using only visual images from a webcam. Visual servo is the use of image data in closed loop control of a robot. The application of vision in robotic applications is rapidly increasing. This is due to the fact that vision based sensors such as webcams are falling in price more rapidly than any other sensor. It is also a richer sensor than traditional ranging devices, particularly since a camera captures much more data simultaneously.

The webcam captures in real-time the condition of work-pieces. This captured picture is processed to get data about position and color. Inverse kinematics analysis enables the processor to calculate the required joint angles. The data from this process will be transmitted as part of commands to the drive unit (Arduino board via Arduino USB program cable) to instruct the robot arm servo motors to move in certain pattern based on the color and position of the work-pieces under the sorting process. That sequence of commands will pick the work-pieces, one by one, from the stock and stack them in the storage area based on their color. The overall system flowchart is given in Fig. 10.

Based on the pervious flow chart, the robot imitates vision in 7 stages as follows:
1. Define of webcam.
2. Start the camera.
3. Get an image.
4. Take a capture every 5 seconds.
5. Save pictures as string in file.
6. Load pictures from file respectively.
7. Image processing program.

The manipulator will go to the home position where it can take a capture from it. After reaching the home position the webcam which it attached above the gripper will take a photo every five seconds then, the image processing program will start processing analysis to detect three things, center of product, boundary of product and the color.

After the detection the position and boundary and color, the gripper will pick the product and then, place it in its storage area. The image is recorded by a webcam which is installed above the robot. The image is then sent via a USB cable to a PC, to be processed by MATLAB using the algorithm developed the image is divided into segments, which become the export databases; usually they are the raw pixels data abstracted from the captured image.

The picture could be in JPEG or any photo format, in which case, every pixel point uses three numerical values, representing intensity levels of the primary colors: Red, Green and Blue (RGB) to depict its characteristics. Therefore, in such format, computing work-load to perform image processing would be very high. Hence, it would be desirable to convert the colored picture into a grey-scale image. Image processing methods are used first to identify the existence of product within the image frame. This is implemented in some steps using MATLAB program.

A. Object Detection

Object detection was accomplished using a generic USB webcam attached to the PC and interfaced with MATLAB. The webcam was used to take a picture of the workspace with objects, as seen in Fig. 11. The program calculates the centroids of each product as shown in the Fig. 12. Then the

World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:7, No:12, 2013

boundary of the farthest product from the origin point is declared as shown in Fig. 13.
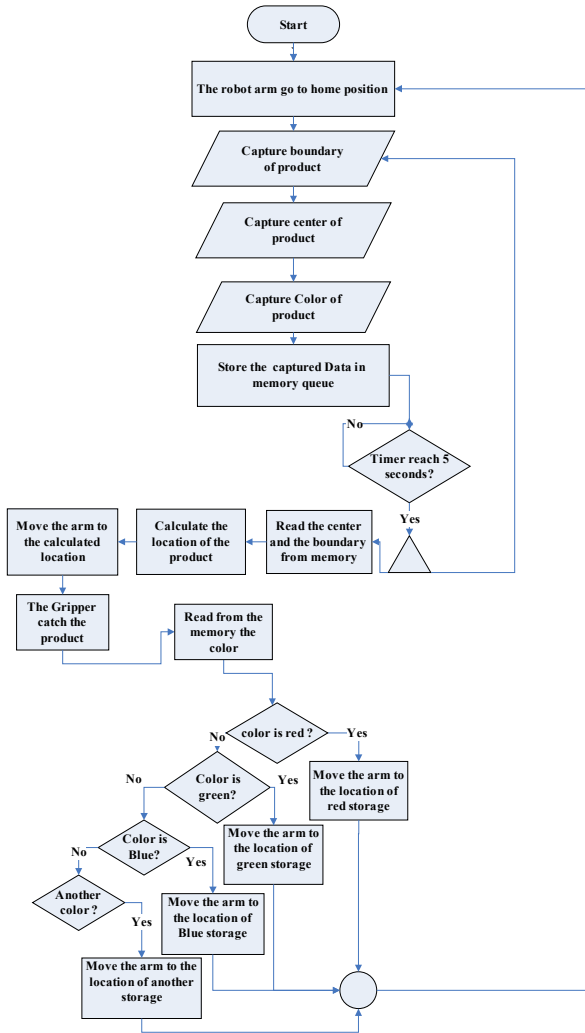

Fig. 10 Flowchart of the control strategy of manipulator
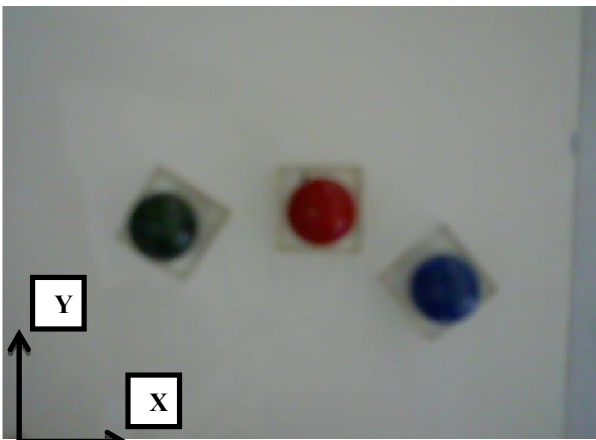

Fig. 11 Image of workspace with objects

Fig. 12 Centroids of each product

Fig. 13 Boundary of the farthest product

The output of MATLAB program (Fig. 14) is the location of the farthest product and its color.
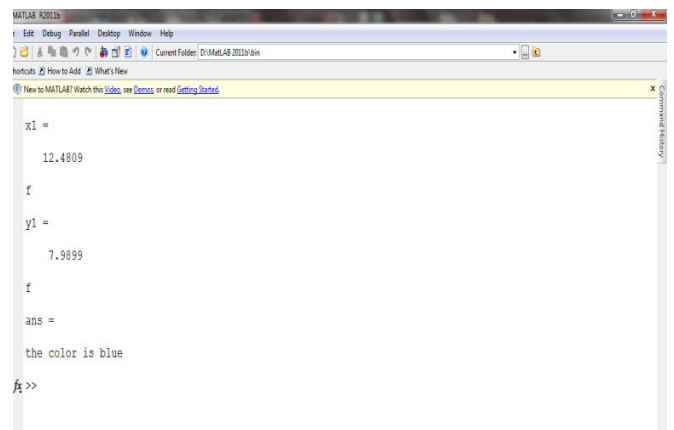

Fig. 14 Values of centroids and color

B. Object Location and Self-Calibration

Once the pixel centers of each object were determined, the actual physical coordinates could be calculated using a set of calibration parameters obtained through the system's self-calibration process.

World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:7, No:12, 2013

To self-calibrate, a calibration object was placed in the robot's gripper. The robot was then commanded to place the object at the edges of its usable workspace, which were at known Cartesian coordinates. Fig. 15 shows the calibration points in reference to the robot's usable workspace [14].

For each placement, an image was taken with the webcam and the pixel coordinates were obtained using the object detection method detailed above. Using the known Cartesian coordinates and the obtained pixel coordinates; mm-per-pixel factors were obtained for both the x and y directions using (15) and (16).
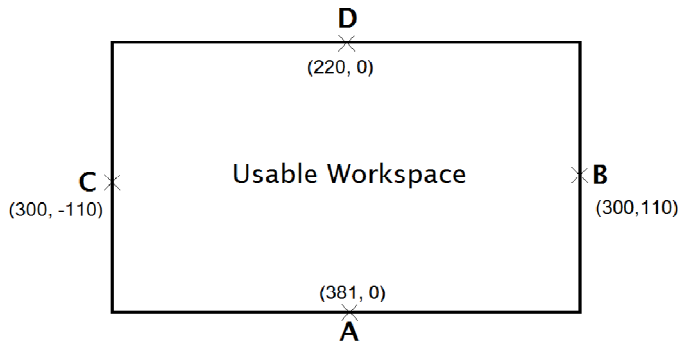

Fig. 15 Calibration points A, B, C and D with Cartesian coordinates (in mm) in relation to the usable workspace [14]

$$X_{factor} = \frac{220\ mm}{Bx\ pix - Cx\ pix} \quad (15)$$

$$Y_{factor} = \frac{161\ mm}{Ay\ pix - Dy\ pix} \quad (16)$$

The pixel offsets for both the x and y directions were obtained from the x value of point C and the y value of point A, respectively.

Once the calibration data was obtained, the Cartesian coordinates for each object could be calculated using (17) and (18) as follows:

$$X\ cart. = ((Ypix. - Y\ offset)*Y\ factor) - 110 \quad (17)$$

$$Y\ cart. = ((X\ pix. - X\ offset)*Xfactor) + 220 \quad (18)$$

Note that the x-direction Cartesian coordinates are obtained in terms of y-direction pixel coordinates and vice versa. This is because, in the final system setup, the webcam's x and y axes corresponded with the robot's y and x axes, respectively.

C. Color Detection and Sorting

After detecting the centroids of each product, the color detection and sorting according to the colors is performed. The proposed algorithm of MATLAB is designed to detect the ranges of red, green and blue as shown in the Table IV. The farthest product will be picked and placed at in the storage area of their color, and so on until to reach to the near point from the origin.

To form a color with RGB, three colored light beams (one red, one green, and one blue) must be superimposed. Each of the three beams is called a component of that color, and each

of them can have an arbitrary intensity, from fully off to fully on, in the mixture. In the first row, the red color is mixture between RGB and the ranges are shown in Table IV. This means that the percentage of red color used for detection is from $176 \le R \le 255$ for red and the green is $25 \le G \le 167$ and the blue is $30 \le B \le 205$. The mixture of three ranges of RGB gives the red color range used to detect the red product. For the second row, the green color used in detection is also mixture of RGB as the ranges $28 \le R \le 97$ for red and $142 \le G \le 229$ for green and $62 \le B \le 135$ for blue.

TABLE IV
RANGES OF COLOR

Ranges of color	Red (R)	Green (G)	Blue (B)
Red	$176 \le R \le 255$	$25 \le G \le 167$	$30 \le B \le 205$
Green	$28 \le R \le 97$	$142 \le G \le 229$	$62 \le B \le 135$
Blue	$0 \le R \le 166$	$0 \le G \le 166$	$40 \le B \le 255$

D. Arduino and MATLAB M File

Using PDE library of Arduino I/O file that is interfaced with MATLAB, after the processing of image program and detecting the color, the program sends certain number to Arduino, this number identifies the color of product, so the program of Arduino sends this data to another Arduino board to detect the angles to reach the position of product.

Using pins 7, 8, 12 and 13 to obtain the color of the product, the following Table V presents the function of each pin used in the interface between Arduino microcontroller and MATLAB. The interface between them will occur via serial port COM 4 using Arduino USB programming cable.

TABLE V
THE FUNCTIONS OF PINS INTERFACED BETWEEN ARDUINO AND MATLAB

Pin 7	To detect the red color
Pin 8	To detect the green color
Pin 12	To detect the blue color
Pin 13	To detect another color

Two Arduino microcontrollers will be used. The first controller is loaded with the PDE package program to read the output color from MATLAB as shown in Table V, then the inverse kinematics program calculates the angels of each link and the second controller will fetch the rotation of each motor according to the values of angels from inverse kinematics program to pick end effector the product then place in its storage area according to its color.

VI. RESULTS AND DISCUSSION

Fig. 16 shows the final system setup, including the robot arm on the right, the webcam attached to the top at gripper, and the workspace at the bottom with some sorted test objects. The test objects were Styrofoam cubes and squash balls wrapped in different colors. Fig. 17 shows the flowchart of MATLAB program including the calibration of image.

World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:7, No:12, 2013

Fig. 16 Final system setup

Firstly, the object pixel centroids are detected and the locations are specified. Secondly, RGB ranges are used to both detect and sort the object by colors according to the range of colors specified in Table IV. Performance analysis proves that this color based object sorting system works very accurate under ideal condition in term of adequate illumination, circular objects shape and color. Object location may be affected by the angle of the webcam, so self-calibration allows the accuracy of location to reach 100%.

For non-ideal condition, such as unspecified color or uncalibrated webcam angular position, the accuracy is reduced to 80%.
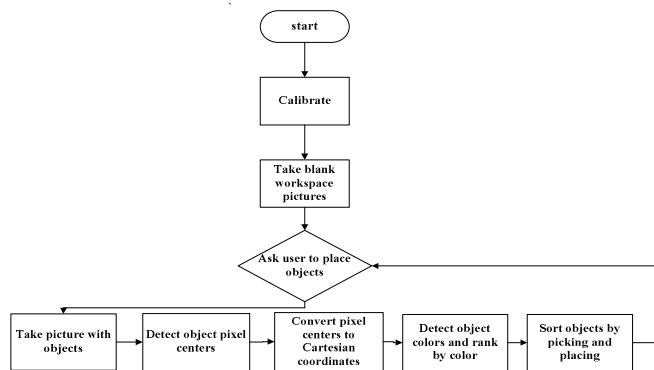

Fig. 17 Software flow diagram

In Table VI, the angles of robot arm shown according to the location of the product which is shown in Fig. 16.

TABLE VI
THE FINAL ANGLES OF THE FARTHEST PRODUCT

X	12.4809 cm
Y	7.9899 cm
Color	Blue color
Angles	θ_1= 32.6261^0, θ_2= 90^0,θ_3= -89,6299^0,θ_4=-0.3701^0 and θ_5= 44.866^0.

VII. CONCLUSIONS

The proposed color sorting system worked accurately as long as the objects are not placed too far away from the calibration lines, and the object detection and location system was spot-on. Color detection was more prone to the aforementioned errors, and the accuracy is reduced to 80%. The calibration routine was able to adequately account for slight translations of the camera mount when the system was taken down and set up again to ensure accurate object location. Though the test objects were only red, green and blue, the system was capable of sorting other objects as well, as long as their base colors were distinguishable.

REFERENCES

[1] M. Swain, and D. Ballard, Color indexing. International Journal of Computer Vision, 1 No. 7, 11-32, 1991.
[2] B.Schiele, and J. l. Crowley, Recognition without correspondence using multidimensional receptive field histograms. International Journal of Computer Vision 36, 31–50, 2000.
[3] H. Schneiderman, and T. A. Kanade, Statistical method for 3d object detection applied to faces and cars. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1, 1746, 2000.
[4] S. Agarwal, A. Awan, and D. Roth, Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 11, 1475 – 1490, 2004.
[5] A. A. Ata, A. Rafeek and H. Yusof, Sensory-Based color sorting automated robotic cell, Journal of Intelligent and Robotic Systems, 1, (2005), 99-110, 2005.
[6] M. Khojastehnazhand, M. Omid, and A. Tabatabaeeefar, Development of a lemon sorting system based on color and size, African Journal of Plant Science Vol. 4(4), pp.122-127, 2010.
[7] Y. Sato, Study of Intelligent Control of an Arm Robot Equipped With A C-MOS Camera Using a New Type of Image Programming, Yoshishige Sato, 2010.
[8] R. Senthilnathan, R. Sivaramakrishnan1, A. Jothilingam and C. Senthil Vel, An Instantaneous Kinematics Method for Robot Control in Vision Based Sorting Application, Journal of Engineering Research and Studies E-7916.
[9] V. Devalla, R. Singh, and A. K. Mondal, Design and Development of Object Recognition and Sorting Robot for Material Handling in Packaging and Logistic Industries, International Journal of Science and Advanced Technology Volume 2, No 9, 2012.
[10] T. Kroeger, and J. Padial, Simple and Robust Visual Servo Control of Robot Arms Using an On-Line Trajectory Generator, 2012 IEEE International Conference on Robotics and Automation River Centre, Saint Paul, Minnesota, USA May 14-18, 2012.
[11] M., Rokunuzzaman, and H. P. W. Jayasuriya, Development of a low cost machine vision system for sorting of tomatoes, Agric Eng Int: CIGR Journal Open access at http://www.cigrjournal.org Vol. 15, No.1 173, 2013.
[12] R. K., Mittal and I. J., Nagrath, Robotics and Control, Tata McGraw-Hill Publishing Company Limited, 2003
[13] S. B., Niku, Introduction to Robotics Analysis, Systems, Applications, Pearson Education, 2003.
[14] http://mablresearch.rit.edu/wiki/images/5/5d/Color-sorting-Picasso_Project_Techmemo.pdf.