Development of a Vision-Based Pick-and-Place Robot

Roneel V. Sharan and Godfrey C. Onwubolu
School of Engineering and Physics,
University of the South Pacific, Suva, Fiji Islands
sharan_r@usp.ac.fj

Abstract

Pick-and-place robots have their applications limited in a system where it is fed with all the information regarding its target. In other words, the robot is given a complete specification of each motion for a manipulation. This work considers the approach taken in automating the manipulation process of an in-house designed pick-and-place robot. It reports the studies of flexible manufacturing concepts using a combination of vision and motion. The vision system differentiates the work-pieces, placed on the work-plane of the robot, with respect to its shape and color and its position and orientation is also determined for manipulation. The pick-and-place robot used for manipulation has been designed as an educational robot in that it realizes pick-and-place operations with limited pay-load. A master control unit (MCU) manages the processing of the vision system and motion planning aspects for manipulation. In addition, a slave unit performs the actual manipulation based on instructions from the MCU.

Keywords: robotics, vision, image, colour, shape, automation

1 Introduction

Robotic manipulators are usually used for repetitive tasks whereby their operations are handled by simple position control strategies. To expand the applications of pick-and-place robots, it is vital to automate its manipulation process [1, 2]. Arguably, vision is thought to be the most valuable sense of automating a robotic system. Integration of a vision system in the workspace of the robot provides improved flexibility for adapting to various task requirements. It makes a robot manipulator more versatile by allowing it to deal with variations in work-piece position and orientation [3].

A vision system analyses and produces descriptions of what is imaged. In general, it takes in raw data and generates description or understanding as an output. The technique in robot vision is a combination of sensing and perception [4, 5]. Sensing is the process of acquiring the image into the computer memory and vision perception refers to the observation, collection, processing, and understanding of information from spatial measurements.

This work considers the approach taken to study flexible manufacturing concepts using a combination of vision and motion. It is carried out as part of development of a smart flexible manufacturing system (SFMS). The SFMS is to include an automatic guided vehicle (AGV) which will be transferring work-pieces, to be either milled or drilled, to the drilling and milling workstations. A pick-and-place robot is employed to pick and place work-pieces to/from the AGV from/to the workstations.

It involves manipulation of work-pieces with shapes ranging from rectangle, circle, and triangle and with possible colours of red, green, blue, yellow, and black using a five degree-of-freedom pick-and-place robot. Information obtained from an overhead mounted CCD camera is processed to differentiate the work-pieces with respect to its shape and colour. The extracted position and orientation information of the work-piece is then utilized to plan the motion of the robot such that the work-piece can be manipulated using its two-fingered gripper.

The work-pieces to be manipulated are placed at arbitrary positions and orientations on the two-dimensional work-plane of the robot covered by the field of view of the camera. On a user selection, using a graphical user interface (GUI), of the shape and colour of the work-piece to be manipulated, the robot, which has been appropriately programmed, then determines the angular rotation of its arms to pick the work-piece from the work-plane and place it at a predefined position. Figure 1 shows the architecture of the overall system.

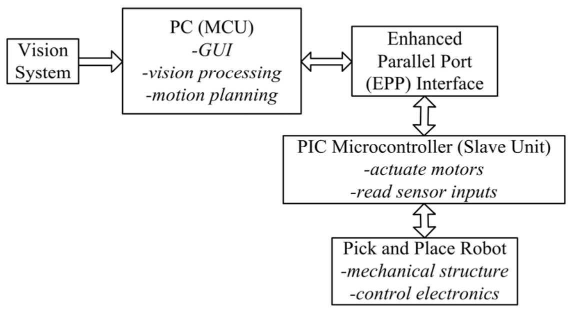


Figure 1: Architecture of the vision-based pick-and-place robot.

The vision system is equipped with an overhead CCD camera and a structured lighting system. It is linked to the MCU which handles the vision processing and motion planning of the robot. A GUI integrates the vision system with the pick-and-place robot and also provides a user interface to the integrated system.

The enhanced parallel port (EPP), through an interface card, is used as the communication medium between the MCU and the slave unit. A PIC microcontroller acts as a slave unit that carries out required motions of the robot based on commands from the MCU.

Moreover, the pick-and-place robot has been built in-house and has been named *RABBIT* (*r*obotic *a*rm for *b*roadly *b*ased *i*nter-station *t*ransfer). It is basically a five degree-of-freedom arm excluding the gripping/releasing movements. The joints of the arm are actuated using stepper motors and DC motors. Sensors in the form of a force sensor and a limit switch handle the gripping and releasing of the work-pieces respectively.

From hereon, the paper is divided into the following sections: (2) vision system; (3) pick-and-place robot; (4) experimentation and results, and (5) conclusion.

2 Vision System

2.1 Image Acquisition

The vision system includes a Sony digital video camera (model DCR HC42E) connected via the USB port to a PC running under the Windows XP platform. It is based under a structured lighting system such that specular reflections from the work-plane and the work-piece surface and shadows around the work-pieces could be minimized.

Image acquisition is performed in MATLAB using the image acquisition toolbox [6]. The rectangular view as seen by the camera is streamed into the MCU and a single snapshot image is taken for further processing. The captured image is represented using 320 x 240 pixels and is in the RGB colour format. That is, a single pixel is represented using the primary red, green, and blue colour values with 8-bit of information.

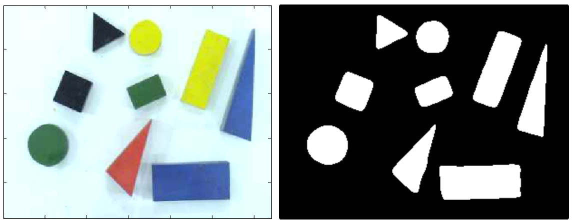
2.2 Image Processing

Feature extraction for shape recognition requires that the captured image be represented in binary format. That is, with pixel values of either 0 or 1 indicating either a black or a white pixel respectively. Conversion from the RGB colour format to the binary format follows spatial filtering, grayscale conversion, histogram equalisation, and finally binary conversion.

Spatial filtering performs neighbourhood operations on image pixels where the value of a filtered pixel depends on the value of its neighbouring pixels. This is achieved through correlation where the centre of the correlation filter, determined through trials with different size and coefficients, is superimposed on the pixel to be filtered. The coefficient of the correlation filter is then multiplied with the pixel value it is imposed on and the summation of the individual products estimates the value of the filtered pixel.

The filtered RGB image is then converted to a grayscale image, also known as an intensity image. It is in this case represented using 8-bit information where 0 represents black and 255 represents white. A grayscale image often has most of its intensity values concentrated within a particular range. Histogram equalization is performed to enhance the contrast of the grayscale image by transforming the intensity values of the grayscale image so that the histogram of the output image has the intensity values evenly distributed.

This enhanced image is finally converted to a binary image using a threshold where values greater than and equal to the threshold are converted to a pixel value of 1 (white) and values below the threshold are converted to a pixel value of 0 (black). A sample captured image with its corresponding binary image is shown in figure 2a and figure 2b respectively.



(a) (b)

Figure 2: (a) RGB format colour image and (b) its corresponding binary image.

The binary image is then segmented into different regions representing the number of objects, termed as image segmentation. This is done by determining the number of white pixels that are connected to each other which in turn denotes a connected component or object. The image processing steps are implemented using the image processing toolbox in MATLAB [7].

2.3 Feature Extraction

2.3.1 Shape Recognition

Based on the range on shapes to be differentiated, the feature of corner detection is utilized for shape recognition. This is a very realistic approach to the problem of shape recognition because it is usually the number of sides or corners that humans use for differentiating shapes. A corner, as defined by Jain [8], is a location on the boundary of an object where the curvature becomes unbounded and is defined as

$$|k(t)|^2 = \left(\frac{d^2 y}{dt^2}\right)^2 + \left(\frac{d^2 x}{dt^2}\right)^2 \quad (1)$$

where t represents the distance along the boundary of the segmented regions and a corner is declared whenever $|k(t)|$ assumes a large value.

Equation (1) can be modified for a digital binary image. If

$$\Delta x_z = x_{z+s} - x_z \tag{2}$$

$$\Delta y_z = y_{z+s} - y_z \tag{3}$$

where s is the sample length of the curvature in pixels, and with the approximation that

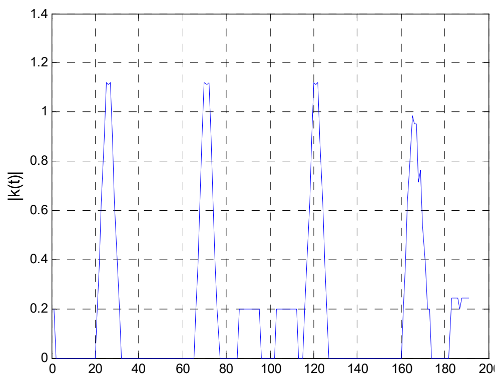
$$\Delta t_z = \sqrt{\Delta x_z{}^2 + \Delta y_z{}^2} \tag{4}$$

then the curvature of a digital binary image with boundary pixels z = 1, 2, 3, ..., Z, for a region, is given as

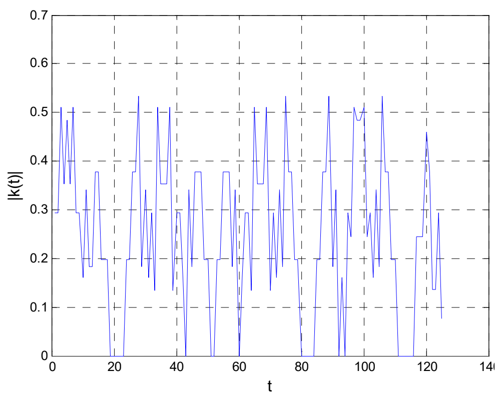
$$|k(t)|^2 = \left(\frac{\Delta y_{z+1}}{\Delta t_{z+1}} - \frac{\Delta y_z}{\Delta t_z}\right)^2 + \left(\frac{\Delta x_{z+1}}{\Delta t_{z+1}} - \frac{\Delta x_z}{\Delta t_z}\right)^2. \tag{5}$$

A sample length of five pixels was chosen, through trials with different sample lengths, for best results. The curvature for a sample rectangle, circle, and triangle sampled at five pixels is shown in figure 3a, figure 3b, and figure 3c respectively.

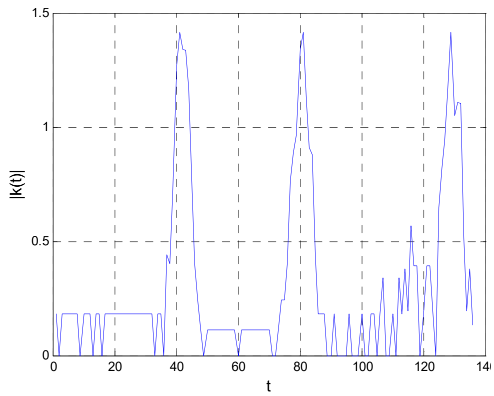
The four and three outstanding peaks in figure 3a and figure 3c correspond to the corners of a rectangle and a triangle respectively. For a circle, figure 3b, the curvature is nearly constant implying the non-presence of corners.



(a)



(b)

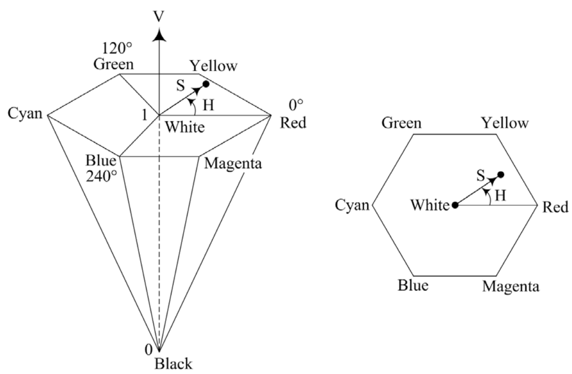


(c)

Figure 3: Curvature for sample of a (a) rectangle, (b) circle, and (c) triangle.

2.3.2 Colour Recognition

Various colour spaces are used for representing colour images for image processing. Colour spaces are where colours are specified by points in three-dimensional spaces. The colour spaces HSV (hue, saturation, and value) and HSI (hue, saturation, and intensity) are a more natural way to how humans perceive colour and is more often used for colour image processing than any other colour space. Due to its compatibility with the image processing toolbox, the HSV colour space, shown in figure 4, has been preferred.



(a) (b)

Figure 4: (a) The HSV colour model, and (b) hue and saturation in the HSV colour model.

The HSV colour model has colours defined inside a hexcone as shown in figure 4a with the positioning of the hue and saturation in the model, for an arbitrary point, shown in figure 4b. *Hue* describes the colour type given as an angle from 0° to 360°. Typically 0° is red, 60° is yellow, 120° is green, 180° is cyan, 240° is blue, and 300° is magenta. *Saturation*, the purity of a colour or the amount of white added to the colour, has values from 0 to 1 where $S = 1$ specifies a pure colour, that is, no white. *Value* is referred to as the brightness of a colour and ranges from 0 to 1, where 0 is black. The transformation from the RGB colour space to the HSV colour space is given by Rogers [9].

2.4 Feature Classification

Feature classification aims to categorize an object within an image to a particular feature class from a number of possible feature classes. A feature class is defined as a group of objects that share common feature or properties. The classification system is operated in two modes: training (learning) and classification (testing). In the training mode, a database of images is built that contains known work-pieces. The knowledge gained from the extracted features forms the database for known work-pieces with which the classifier is trained to distinguish the different feature classes in the feature space.

2.4.1 Training

Since the number of shapes and colours to be differentiated is three and five respectively, there are fifteen different possible feature classes. For shape recognition, the a priori knowledge is utilized that the number of corners for a rectangle, triangle, and circle are always four, three, and zero respectively. However, for colour recognition, the H, S, and V colour values are obtained through samples of forty-five work-pieces to represent a feature class.

The mean RGB and HSV colour values are given in table 1. Also, the ideal RGB values for the five colours are given.

Table 1: The ideal RGB colour values and the mean RGB and HSV colour values.

Colour	Ideal RGB colour values	Mean RGB colour values	Mean HSV colour values
Red	[255, 0, 0]	[173.8, 64.0, 47.6]	[0.0271, 0.7260, 0.6816]
Green	[0, 255, 0]	[44.2, 88.1, 57.9]	[0.3850, 0.4984, 0.3454]
Blue	[0, 0, 255]	[47.7, 75.2, 142.9]	[0.6184, 0.6661, 0.5603]
Yellow	[255, 255, 0]	[176.5, 192.8, 19.7]	[0.1824, 0.8967, 0.7561]
Black	[0, 0, 0]	[45.5, 49.3, 46.3]	[0.5354, 0.2304, 0.2245]

2.4.2 Testing

In the testing phase, an input feature vector is assigned to one of the fifteen feature classes. Due to the distinct nature of the feature classes, the simplest approach of minimum distance classification is utilized for feature classification. A minimum distance classifier assigns an unknown feature vector based on the minimum distance between the unknown feature vector and each of the feature classes. The distance here refers to the Euclidean distance given as

$$d_j(\mathbf{x}) = \left\| \mathbf{x} - \mathbf{m}_j \right\| \qquad j = 1, 2, \ldots, W \tag{6}$$

where $\|\mathbf{a}\| = (\mathbf{a}^T\mathbf{a})^{1/2}$ is the Euclidean norm, $\mathbf{x}$ denotes the unknown feature vector and $\mathbf{m}_j$ denotes the jth feature class [10]. Hence, a feature class $\mathbf{x}$ is assigned to class w_i if $d_i(\mathbf{x})$ is the smallest distance.

3 Pick-and-Place Robot

3.1 Mechanical Design

The pick-and-place robot has been designed and built in-house. It is designed for a payload of 500 *g* and is constructed of aluminium due to its light weight. The joints of the robot include the base, shoulder, elbow, wrist, and gripper (rotates about the wrist). The gripper is used as an end-of-arm-tool (EOAT) which grips and releases work-pieces. The operation of the gripper is based on the principle of lead screw. One finger of the gripper is fixed while the second finger moves along the lead screw as it is rotated.

The calculated values of the torque for all the joints are given in table 2.

Table 2: Torque considerations for each movement.

Joint	Length/radius of joint link (*mm*)	Torque (*Nm*)
Base	$R_B = 75$	7.79
Shoulder	$L_S = 181$	2.79
Elbow	$L_E = 185$	1.87
Wrist	$L_W = 150$	0.66
Gripper (rotate)	$R_G = 25$	0.66
Grip/release	-	0.013

Stepper motors, due to their good position integrity, are utilized for rotating the base, shoulder, elbow, and the wrist. All the stepper motors have been mounted on the base so that the torque on the joints is reduced. Transformation of the torque from the motors to the joints is performed using timing belts and timing pulleys.

However, the DC motors to rotate the gripper about the wrist and to open/close the gripper have been mounted on the gripper itself. The gripper is also equipped with a force sensor and a limit switch that govern the gripping and releasing of a work-piece respectively.

The robot operates on a work-plane with dimensions of 348.14 *mm* and 261.11 *mm*. This is determined through the maximum reach of the robot and the ratio of the x and y pixels (320:240 = 4:3) of the captured image.

3.2 Electronics Design

An EPP interface card [11] is utilized for communication between the MCU and the slave unit for this system. The EPP interface card offers two 8-bit ports of input and output. A PIC 16F877 microcontroller is used as a slave unit which operates based on instructions from the MCU. These instructions are in the form of the joint to be actuated, its direction of rotation, and the angle it has to be rotated.

The four-phase, 1.8°, hybrid stepper motors are driven using unipolar drive boards, commercially available driver boards for stepper motors. In addition, the light duty DC motors are controlled using the quadruple half-H driver (SN754410) IC.

Furthermore, the force sensor and the limit switch are mounted on the gripper. The output of the force sensor is resistance whereby applying a force to the active sensing area causes the resistance at the terminals of the force sensor to decrease. Using voltage division, this output resistance is then converted to a voltage between 0 V and 5 V which, based on calibration, depicts a force between the work-piece and gripping fingers.

3.3 Motion Planning

Motion planning of the robot is done based on the predefined home (initial) and place positions while the pick position is determined after analysis of the captured image. The centroid of the work-piece in pixels (x_p, y_p) is first transformed into the dimensions of the work-plane (x_w, y_w) and then to the C-space of the pick-and-place robot (x_r, y_r) as shown in figure 5. The z component is always constant as the gripper points vertically downwards when picking or placing work-pieces.

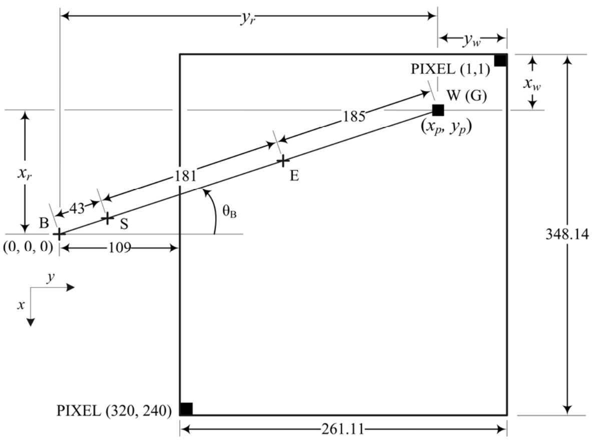


Figure 5: Representation of the pick position.

The angular rotation of the base is then defined as

$$\theta_B = \tan^{-1}\left(\frac{x_r}{y_r}\right). \tag{7}$$

Inverse kinematics is utilized to determine the angular positions of the shoulder, elbow, and wrist at the pick position as shown in figure 6.

The angle that the elbow link makes with the shoulder link, θ_E, is then given as

$$\theta_E = 180° - \beta \tag{8}$$

where

$$\beta = \cos^{-1}\left(\frac{H^2 - \left(L_S^2 + L_E^2\right)}{-2L_S L_E}\right) \tag{9}$$

and with reference to figure 5,

$$H = \sqrt{x_r^2 + y_r^2} - 43. \tag{10}$$

where 43 is the distance between the base joint and the shoulder joint.

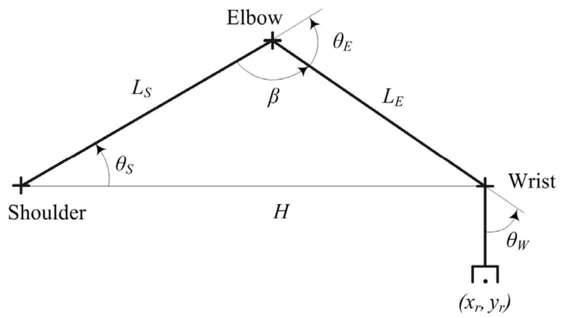


Figure 6: Angular position of the shoulder, elbow, and wrist at the pick position.

Using the sine formula, the angle that the shoulder link makes with the xy-plane is given as

$$\frac{\sin\theta_S}{L_E} = \frac{\sin\beta}{H}$$
$$\theta_S = \sin^{-1}\left(\frac{L_E \sin\beta}{H}\right). \tag{11}$$

The angle that the wrist link makes with the elbow link is now defined as

$$\theta_W = 90° - \left(\theta_E - \theta_S\right). \tag{12}$$

The flowchart for motion planning is illustrated in figure 7 where n and N denote the number of work-pieces manipulated and the number of work-pieces in the specified class.

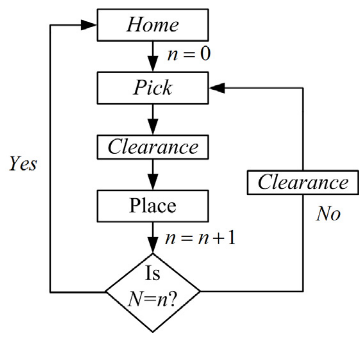
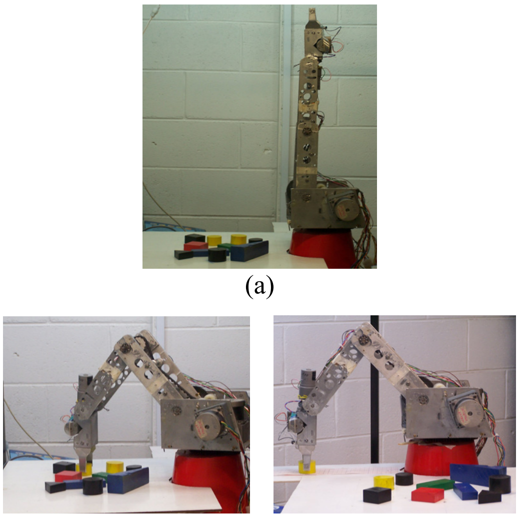


Figure 7: Motion planning of the robot.

From the home position, the robot picks the first work-piece, and moves through a clearance position to the place position. A clearance position is utilized since the pick and the place heights are same and it is not feasible to directly rotate the base, with the arm, to/from the pick and place positions. The pick-to-place and place-to-pick motions will continue until all the work-pieces of the selected class are manipulated, and then the robot returns to the home position. Figure 8 (a), (b), and (c) show the position of RABBIT at the home, pick, and the place positions respectively.



(a) (b) (c)

Figure 8: The robot at (a) home, (b) pick, and (c) place positions.

The robot operates in a calibrated environment. That is, the camera has been focused to the field of view of the rectangular work-plane. In addition, the joints, actuated using the stepper and DC motors, work based on calibration where a certain number of pulses imply a certain degree of movement of the arm link. However, calibration is not used for the gripper motor as it is controlled by the sensors mounted on it.

4 Experimentation and Results

The vision system gives 100% accuracy for both shape and colour recognition for the range of work-pieces considered for this work. For the pick-and-place robot, the accuracy of the joints is given in table 3. The elbow joint has two operation regions since it had different calibration relationship in its overall operation region.

Table 3: Results for the accuracy of the joints.

Movement/rotation		Error (°)	Repeatability (±3σ)
Base		$e_B = 1.18$	±0.27
Shoulder		$e_S = 0.59$	±0.30
Elbow	Region 1	$e_{E1} = 2.45$	±0.33
	Region 2	$e_{E2} = 1.58$	±0.27
Wrist		$e_W = 1.29$	±0.42

5 Conclusion

The automation of the manipulation process for a pick-and-place robot in its work-cell has been conveyed in this paper. Specially, the integration of vision in the robot work-cell and its implication in improving the flexibility of manipulation has been discussed. This flexibility can help eliminate the use of expensive and application specific manipulation in a variety of industrial applications. The work carried out has been very effective in studying flexible manipulation approaches using a combination of vision and motion.

6 Acknowledgements

This work was approved and funded by the Faculty of Science and Technology (FST) Research Committee and the University Research Committee (URC) as part of a Master of Science in engineering course at the University of the South Pacific (USP).

7 References

[1] T.L. Perez, J.L. Jones, E. Mazer and P.A. O'Donnell, "Task-level planning of pick-and-place robot motions", *IEEE Computer*, 22(3), pp 21-29 (1989).

[2] M.W. Spong, "Motion control of robot manipulators", *in Handbook of Control*, W. Levine, editor, CRC Press, pp 1339-1350 (1996).

[3] A. Nagchaudhuri, S. Kuruganty and A. Shakur, "Mechatronics education using an industrial SCARA robot", *Proceedings of the 7th. Mechatronics Forum International Conference and Mechatronics Education Workshop (Mechatronics 2000)*, Atlanta, Gerogia, (2000) CD ROM.

[4] G.W. Awcock and R. Thomas, *Applied Image Processing*, McGraw Hill, New York (1995).

[5] P.G. Ranky, "Advanced machine vision systems and application examples", *Sensor Review*, 23(3), pp 242-245 (2003).

[6] Mathworks, *Image Acquisition Toolbox for use with MATLAB: User's Guide*, The Mathworks Inc (2005).

[7] Mathworks, *Image Processing Toolbox for use with MATLAB: User's Guide*, The Mathworks Inc (2004).

[8] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, New Jersey (1989).

[9] D.F. Rogers, *Procedural Elements of Computer Graphics*, 2nd edition, McGraw Hill, New York (1997).

[10] R.C. Gonzalez, R.E. Woods and S.L. Eddins, *Digital Image Processing using MATLAB*, Pearson Prentice Hall, New Jersey (2004).

[11] G.C. Onwubolu, S. Aborhey, R. Singh, H. Reddy, M. Prasad, S. Kumar and S. Singh, "Development of a PC-based computer numerically controlled drilling machine", *Proc. Instn Mech. Engrs, Part B: J. Engineering Manufacture*, 216(B1), pp 1-7 (2002).