

University of Texas Rio Grande Valley

ScholarWorks @ UTRGV

Theses and Dissertations - UTB/UTPA

2002

Automated robotic inspection system for electronic manufacturing

Roberto Alejandro Balderas
University of Texas-Pan American

Follow this and additional works at: https://scholarworks.utrgv.edu/leg_etd



Part of the [Manufacturing Commons](#)

Recommended Citation

Balderas, Roberto Alejandro, "Automated robotic inspection system for electronic manufacturing" (2002).
Theses and Dissertations - UTB/UTPA. 533.
https://scholarworks.utrgv.edu/leg_etd/533

This Thesis is brought to you for free and open access by ScholarWorks @ UTRGV. It has been accepted for inclusion in Theses and Dissertations - UTB/UTPA by an authorized administrator of ScholarWorks @ UTRGV. For more information, please contact justin.white@utrgv.edu, william.flores01@utrgv.edu.

**AUTOMATED ROBOTIC INSPECTION
SYSTEM FOR ELECTRONIC MANUFACTURING**

A Thesis

by

ROBERTO A. BALDERAS

**Submitted to the Graduate School of the
University of Texas – Pan American
In partial fulfillment of the requirements for degree of
MASTER OF SCIENCE**

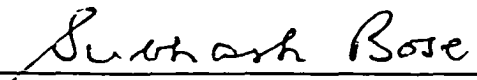
May 2002

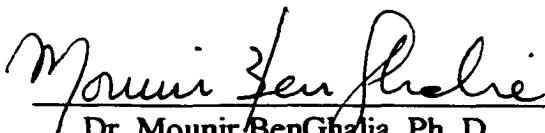
Major Subject: Manufacturing Engineering


**AUTOMATED ROBOTIC INSPECTION
SYSTEM FOR ELECTRONIC MANUFACTURING**

**A Thesis
by
ROBERTO A. BALDERAS**

Approved as to style and content by:


Dr. Subhash Bose, Ph. D., P.E.
Professor of Manufacturing Engineering
Chair of Committee


Dr. Mounir BenGhaila, Ph. D.
Assistant Professor of Electrical Engineering
Member of Committee


Dr. Immanuel Edinbarough, Ph. D.
Assistant Professor of Manufacturing Technology
University of Texas - Brownsville
Member of Committee

May 2002

ABSTRACT

Balderas, Roberto A., Automated Robotic Inspection System for Electronic Manufacturing. Master of Science (MS), May 2002, 66 pp.; 4 tables, 37 figures, references, 20 titles.

An automated robotic inspection system for electronic manufacturing has been developed to identify pin defects of IC packages mounted on printed circuit boards using surface mount technology. The automated robotic inspection system consists of two robots, a computer, a CCD camera with frame grabber for image acquisition, and a customized windows program using neural network for on-line defect identification. Gray scale images of the pins on IC packages are acquired using ambient light. The images are filtered and formatted to appropriate size, so that Matlab neural network tool could be used. The images are used to train neural networks using Matlab's Bayesian Regularization module. Optimal network was found to be a single-layer network with three outputs for each IC investigated. The weights and biases of each of the ICs investigated and the matrices of gray scale values for the IC images are saved as text files. A customized windows program uses these text files for on-line defect identification. The defect identification for the networks was found to be 100 percent for the two ICs investigated. The analysis and integration of an automated robotic inspection system for on-line monitoring of electronic manufacturing using neural networks is presented in this work.

ACKNOWLEDGEMENTS

I would like to thank my advisors, Dr. Subhash Bose, Dr. Mounir BenGhalia, and Dr. Immanuel Edinbarough for giving their support, sharing their knowledge and experience. I give special thanks to Dr. Subhash Bose for sharing his expert knowledge in robotic systems and automation equipment, and his invaluable help in keeping me on my toes to meet deadlines; Dr. Mounir BenGhalia for providing image acquisition and Mathlab software and Dr. Immanuel Edinbarough for his suggestions on image filtering. Finally, I would like to thank my mother Aracely Garza de Balderas for her moral and financial support.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. REVIEW OF LITERATURE	3
CHAPTER 3. ROBOTICS AND VISION SYSTEMS	10
3.1 Application of Robots for Automation	11
3.2 Description of Scrobot Model ER-V	14
3.3 Description of the Vision System	15
CHAPTER 4. NEURAL NETWORKS AND DEFECTS IN ELECTRONICS MANUFACTURING	18
4.1 Neural Network Architecture.....	19
4.2 Neural Network Topology	23
4.3 Graphical Device Interface (GDI) software.....	24
4.4 Defects in Electronics Manufacturing	26

CHAPTER 5. TRAINING ALGORITHM FOR INSPECTION	
AND CLASSIFICATION	30
5.1 Acquisition of Data	32
5.2 Network Architecture	34
CHAPTER 6. ON-LINE MONITORING SYSTEM	38
6.1 Graphical Features	38
6.2 Simulation Option	40
6.3 On-Line Option	47
6.4 Scorbtor ER-V Menu Items	50
6.5 Robot Program Menu Items	52
6.6 About Menu Item	53
CHAPTER 7. CONCLUSIONS	41
REFERENCES	57
APPENDIX A:	
DIRECT AND INVERSE KINEMATICS EQUATIONS FOR	
SCORBOT ER-V ROBOT.....	59
APPENDIX B:	
PERFORMANCE RESULTS OF IC TWO	63
VITA	65

LIST OF TABLES

	Page
Table 4.1 Commonly used transfer functions and Matlab equivalent	23
Table 4.2 Defect types and inspection capabilities of visual and x-ray systems	29
Table 5.1 Matlab algorithm speed comparisons.....	31
Table 5.2 Criteria used for defect generation for data generation	34

LIST OF FIGURES

	Page
Figure 3.1 (a) Cincinnati Milacron T ⁶ robot (b) PUMA 560 robot arm	12
Figure 3.2 Various robot arm configurations	13
Figure 3.3 Scorbot ER-V System	15
Figure 3.4 Automated Visual Inspection System	17
Figure 4.1 Multiple Input Neuron	20
Figure 4.2 A single layer of S neurons with R inputs	21
Figure 4.3 Three layer network with R inputs and S ³ outputs	22
Figure 4.4 Schematic of image acquisition of a six DOF robot	24
Figure 4.5 Information contained within PCB1data.txt file	25
Figure 4.6 File for ChipNumber1.txt of IC contained within PCB1data.txt file	26
Figure 5.1 (a) Real image with no defect (b) Simulated defect image	33
Figure 5.2 (a) Output Performance of First Neuron vs. Pin Number	37
Figure 5.2 (b) Output Performance of Second Neuron vs. Pin Number	37
Figure 5.2 (c.) Output Performance of Third Neuron vs. Pin Number	38
Figure 5.2 Classification performances for 288:3 neural network	37-38
Figure 6.1 Snapshot of the on-line monitoring system communications options	40
Figure 6.2 Snapshots of (a) Simulation selection (b) Number of PCBs to simulate	42
Figure 6.3 (a) Position 2, retracted, and facing the shipping station	44
Figure 6.3 (b) Position 3, retracted, and facing PCBs station	45

Figure 6.3 (c) Position 4, extended arm, and on PCBs station	45
Figure 6.3 (d) Position 5, retracted, and facing inspection station	46
Figure 6.3 (e) Position 6, extended arm, and on inspection station	46
Figure 6.3 (f) PCB Inspection station defect monitoring	47
Figure 6.3 (g) Position 7, retracted arm, facing rework station	47
Figure 6.3 (h) Position 8, extended arm, and on rework station	48
Figure 6.3 Snapshots of the Scorbobot ER-V inspection cycle	44-48
Figure 6.4 Snapshot of action menu	49
Figure 6.5 PCB board defect report	50
Figure 6.6 Scorbobot menu items	51
Figure 6.7 Robot coordinate selection pop up window	52
Figure 6.8 Robot coordinate animation result of coordinate selection	52
Figure 6.9 Robot program menu items	54
Figure 6.10 Developer information	54
Figure A-1 Gripper location definition in XYZ coordinate system	61
Figure A-2 Relation of robot joint coordinates to XYZ coordinates	62
Figure B-1 (a) Real image with no defects (b) Simulated defects image	64
Figure B-2 (a) Output Performance of First Neuron vs. Pin Number.....	64
Figure B-2 (b) Output Performance of Second Neuron vs. Pin Number	65
Figure B-2 (c) Output Performance of Third Neuron vs. Pin Number	65
Figure B-2 Classification performances for 480:3 neural network	64-65

CHAPTER 1

INTRODUCTION

Advancements in technology have led to miniaturization of components, leading to smaller and more powerful electronics products. Some examples of such products are laptop computers, palm pilots, cellular phones, and pagers. Although these products are highlights of electronics breakthroughs, manufacturing of these products has lead to problems, which affects production yields and thus profitability. The most obvious problem is assembling these miniaturized components that has more connections in less space, thus increasing defects probability. Because, the components are so small, it is very difficult, if not impossible for a human operator to assemble; it is slow and not efficient, therefore not profitable or competitive. Because of these reasons, electronics manufacturers have switched from manual operations performed traditionally by human operators to automated systems. For example, surface mount devices (SMDs) are assembled onto printed circuit boards (PCBs) with robotic assembling machine that uses visual imaging for accurate placement of the components onto PCB. Furthermore, this technology is being used in automated visual inspection systems for defect identification.

Chapter 2 describes several technologies that are being utilized by the electronics industry today. All of these technologies have something in common; they focus on conventional image processing and pattern recognition for classification of defects. The

procedures are intuitive but difficult to develop and are limited to particular application. Furthermore, these techniques need to be setup carefully and monitored by a skilled operator to ensure good results. Also, these techniques are computational expensive, slowing down the inspection process; and thus, the production line. This is where neural network approach offers several advantages for automated inspection. Neural networks have been used experimentally for decades. Neural networks are robust and can be implemented on-line, and are highly parallel which allows simultaneous computation using several processors to reduce computational time and thus reduce inspection time.

In this work the neural network approach and mechatronics are applied to create an automated visual inspection system capable of identifying pin defects on gull wing surface mount components. The outline of this thesis is as follows: in Chapter 2, the most relevant literature on manufacturing of electronics products, manufacturing task breakdown of typical defects found during visual inspection, automated visual inspection systems used in Japan, automated inspection of solder joint categories based on their operating principles, manufacturers experiences and solutions, and neural networks approach advantages over intuitive techniques. Chapter 3 gives a brief history on robotics, applications of robots for automation, description of the Scrobot ER-V specifications, and description of the vision system used in this work. Chapter 4 gives historical background on neural networks and applications, neural network architecture information, neural network topology used in this work, graphical device interface software developed, and defects in electronics manufacturing. Chapter 5 describes the training algorithm for inspection and classification used, how the data was obtained and generated, the network architecture chosen, and results of training neural network and its

performance. Chapter 6 describes the on-line monitoring system developed. Chapter 7 summarizes the results obtained in this study with suggestions for future analysis and improvements of the neural network-based automated inspection system.

CHAPTER 2

REVIEW OF LITERATURE

As stated in Chapter 1, the competitive nature of electronics industry and advancements in technology have led to miniaturization of components, leading to smaller and more powerful electronics products. Some examples of such products are laptop computers, palm pilots, cellular phones, and pagers. Although these products are highlights of electronics breakthroughs, manufacturing of these products has lead to problems, which affects production yields and thus profitability. The most obvious problem is assembling these miniaturized components that has more connections in less space, thus increasing defects probability. Because, the components are so small, it is very difficult, if not impossible for a human operator to assemble; it is slow and not efficient, therefore not profitable or competitive.

In a typical manufacturing plant, approximately 30% of all manufacturing tasks are related to inspection, of which 60% of inspection tasks are visual. The break down of typical defects found during visual inspection is approximately 30% part defects, 50% assembly defects (20% of which are incorrect parts or missing parts etc.), and 20% soldering defects (Hata 1990). It should be noted that, in order to maintain a certain level of quality in electronics manufacturing process, an increase in the number of solder joints by a factor of 10 requires that the number of defects be reduced by a factor of 10.

Therefore, the effectiveness of an inspection system used to control a process will have a direct impact on the quality of products shipped to customers. This places an extraordinary pressure on human inspectors who are trained to identify defective parts by visual examination. This is further compounded if the equipment is used by the military, which requires 100% inspection of solder joint as required by MIL-STD 2000 for electronics assemblies. A study conducted by AT&T showed that a decision repeatability of only 44% for the same inspector inspecting solder joints, and only 6% agreement between their judgments with four inspectors checking the same board. This shows that the descriptions of allowable variations are very qualitative and subject to interpretation by inspection experts and leads to increases in manual inspection costs which can be as high as 50% of assembly costs in some cases (Sankaran *et al.* 1995).

If we take a typical manufacturing process from start to finish of a typical device, a calculator, we can appreciate why there is a need for automation and inspection systems. The complexity, miniaturization, and mixed technologies, namely, Ball Grid Arrays (BGA), Pin Grid Arrays (PGA), Surface Mount Technology (SMT), Through Hole Technology (THT) and other technologies used in Printed Circuit Board (PCB) have lead to the creation of various commercial inspection and automated assembly systems in Japan. Some of these systems are Mask Pattern Inspection Machine, PCB Pattern Inspection Machine, Mounted Chip Visual Inspection Machine, Soldering Inspection Machine, and Assembled PCB Visual Inspection Machine (Hata 1990).

Competition, new technology, and miniaturization have led to automation using robotic equipment to assemble components, and automated visual inspection systems to reduce costs. Traditional methods once used to assemble, test, or repair are no longer

feasible and new methods are being developed. To achieve high yields, extensive inspection and rework activities must take place to remain competitive.

From the discussions, we can infer that there is no one unified system applicable to inspect or assemble parts needed in electronics assembly manufacturing. Depending on the stage of manufacturing process, specialized machines are used for inspection and assembly. Examples of the technologies that are being developed are described below.

Automated inspection of solder joints on PC boards can be categorized according to their operating principle as follows: digital image processing with CCD cameras, optical 3D sensor technology with laser scanners, X-ray techniques, and thermal laser-pulse infrared methods. All of these technologies have advantages and disadvantages. Digital image processing with CCD cameras are the fastest and can be implemented on-line. They are limited by the resolution of CCD camera and optics, and can only extract 2D data (picture) in either gray scale or color. Therefore, 2D data is useful where: orientation, alignment, and texture quality (surface defects) are needed from 2D images. The disadvantage is that the reliability of the results can be considerably impaired by contrasts found on PCB board. To increase the reliability of results, optical 3D sensor technology with laser scanners has been developed. The advantages of adding an extra dimension to the purely optical 2D, are the ability to calculate the maximum height gradient of a component, volume, shape, compute either a 2D or a 3D histogram, ability to identify 2D and 3D edges, identify 2D contours, and detection of 2D distortions. These extra features are important because it allows identification of tombstone (tilt) defect of components, which is not possible by pure 2D image. Its 3D operations are able to distinguish reliably the quality of a surface of a joint (insufficient solder, excess solder,

blow out etc...) and are applicable to both THT and SMT. It can be implemented on-line. The disadvantage is that it is limited to scanning speed and the resolution of laser, which is about 70 micrometers/pixel. Improvement in laser technology can lead to scan rates of up to 2MHz and 3D sensor resolution of about 10 micrometers/pixel but it is slower than pure optical CCD cameras. X-Ray techniques can perform all the operations of 3D sensor technology and identify voids in solder joints and its ability to inspect boards with components on both sides. The disadvantage is that it cannot be implemented on-line because the process is slow, thus it is used off-line with SPC charts to bring the process in control. Thermal Laser-Pulse (TLI) infrared method has been used for internal characterization of solder joints. The disadvantages of TLI method include: (i) not implementable for on-line operation; (ii) high dependency on surface impurities of solder joints, and on geometry and orientation of component leads. It can only be used as supplemental information to some of the other technologies (Mengel 1990).

Over the years, many inspection systems have been developed to solve manufacturing inspection difficulties by integrating various disciplines including robotics, optics, sensors, thermal, X-ray, neural networks, CAD/CAM, databases and other techniques. The main reason is to reduce inspection cost because conventional inspection is no longer acceptable; it is slow, high labor costs, expensive, and unreliable.

To solve the problem of inspecting Pin Grid Arrays (PGAs) solder joints, NEC corporation developed an automated system using a XYZ robot, a specially designed 0.9 mm Optical Fiber Scope (OFS) with light guides, a standard optical system (TV monitor, CCD camera, PC, Frame Grabber), and customized software. The automated system's objective was to identify defects on each joint on the PGA as good or bad (constricted or

not soldered). The developed automated system correctly identified defects 100% of the time, but with 0.8% false alarm rate. The author states that by using thinner OFS scope the false alarm rate can be reduced to 0.1% (Kashitani *et al.* 1993).

A manufacturer of Large Scale Integrated (LSI) packages has developed an integrated automation system for final visual inspection. The inspection was the only process that was not automated and lagged behind all other manufacturing processes. The decision was made because the packages were becoming larger and the lead pitches were finer. Therefore, to reduce inspection cost, a standardized automated inspection system was developed. The inspection items were separated into three general categories based on appearance of lead, mold, and mark. Inspection on leads were co-planarity, bend (pitch), evenness, marks, scratches, and foreign objects. Mold inspection criteria were voids, scratches, foreign object, and cracks. Mark inspection was based on off-center letters, broken characters, blurred characters, and overlapping characters. Based on the inspection criteria, six 2048-pixel CCD camera systems were developed for inspection of defects in parallel using six 2048-pixel DSP board and a computer with customized software. Parallel processing was used in order to be able to implement the system on-line, and be synchronized with other manufacturing processes (Okabe *et al.* 1993).

There have been attempts to use X-Ray Laminography for automated inspection of joints and application of neural networks for classification of defects. It was found that neural networks achieved poor performance because the image resolution of an X-Ray machine was poor and the imaging medium is still not well understood by the industry. Developing better imaging resolution would allow accurate three dimensional

measurement of solder joint structures and improved performance for neural networks (Sankaran *et al.* 1998).

All the above-mentioned literatures and techniques have something in common; they focus on conventional image processing and pattern recognition for classification of defects. The procedures used are intuitive but difficult to develop and are limited to particular application. Furthermore, these techniques need to be setup carefully and monitored by a skilled operator to ensure good results. Also, these techniques are computationally expensive, slowing down the inspection process; and thus, the production line. Any automated inspection system has to be able to implement an inspection system on-line, and if possible give feedback to other automated processes for continuous improvement. Neural network approach offers several advantages for automated inspection.

Neural networks have been used experimentally for decades. Neural networks are

- Adaptive: infer solutions often capturing subtle relationships.
- Able to generalize: can handle imperfect or incomplete data.
- Nonlinear: can classify defects that are not linearly separable.
- Highly parallel: large networks can be realized using parallel processors to achieve real-time speeds.
- Tolerant of unusual noise distributions.

Although neural networks offer these advantages, the network performance suffers if the resolution of the defect image is poor (blurry). Therefore, high-resolution images (clear) that accurately portray the defect being classified are preferred to yield higher detection performance. Higher resolution images require more memory and more processing time.

For a large network, this can be prohibitive. Therefore, data compression techniques have been developed to preprocess the data before it is fed to the neural network, thus reducing processing and training time. The idea is to achieve maximum compression without affecting the neural networks performance adversely. Two compression techniques that have been applied to BGA joint images with good results are Histogram based and Fourier transform based techniques (Sankaran *et al.* 1995).

In addition to data compression techniques, the raw data should dictate the form of the multilayer neural networks. To accomplish this, several factors must be considered: the network topology, the number of iterations during training, the learning rate, the training sample set, batch processing or continuous update of the network weights, and the network initial weights (Hagan *et al.* 1996).

In this thesis, a visual inspection system using neural networks is used to identify common defects in electronics manufacturing. A single layer network with multiple neurons is used to classify the defects.

CHAPTER 3

ROBOTICS AND VISION SYSTEMS

The quest for an “Artificial Man” is not a new concept. As early as the 1500’s, man has attempted to mimic (model) himself. This is evident from historical artifacts. In 1540 Giano della Tore created a figure of a girl playing a flute. In 1770 and 1773, Henry-Louis Jaquet-Droz created two mechanical dolls called a scribe and a musician playing a piano. In 1778 Wolfgang von Kempelen created a talking machine; the Philadelphia doll a writing mechanism created before the 1812 by Les Maillardet. The steam man created by George Moore in 1893; the automatic electromagnetic chess-playing machine created by Leonardo Torres Quevedo in 1912. In modern times, play writer Karel Capek coined the term robot from the Czech word robota meaning work. According to Capek, robots would be “mechanically perfect” and “highly intelligent.” Since the 1950’s, there have been three generations of robot design. The first generation had no computing or sensory capability. The second generation had limited computational power and feedback capabilities. The third generation design was considered as possessing intelligence and diverse sensing capabilities. Robotics has evolved to the point that different branches, such as industrial robots, mobile robots, legged transportation systems, medical prosthesis and orthosis, micro-robotics as well as autonomous (mobile) vehicles, have

reached a remarkable level of maturity as evidenced by immense results and a variety of applications (Fukuda *et al.* 2001).

3.1 Application of Robots for Automation

Robots come in many shape and sizes depending on the application. Usually, industrial robots are general-purpose machines that mimic human motion and are controlled by an intelligent system of algorithms utilized by a computer to control robot motion via a servomechanism. Mechanically, a robot mimics a human arm doing useful work by providing a rigid mainframe (or arm), a wrist subassembly plus a tool (Figure 3.1).

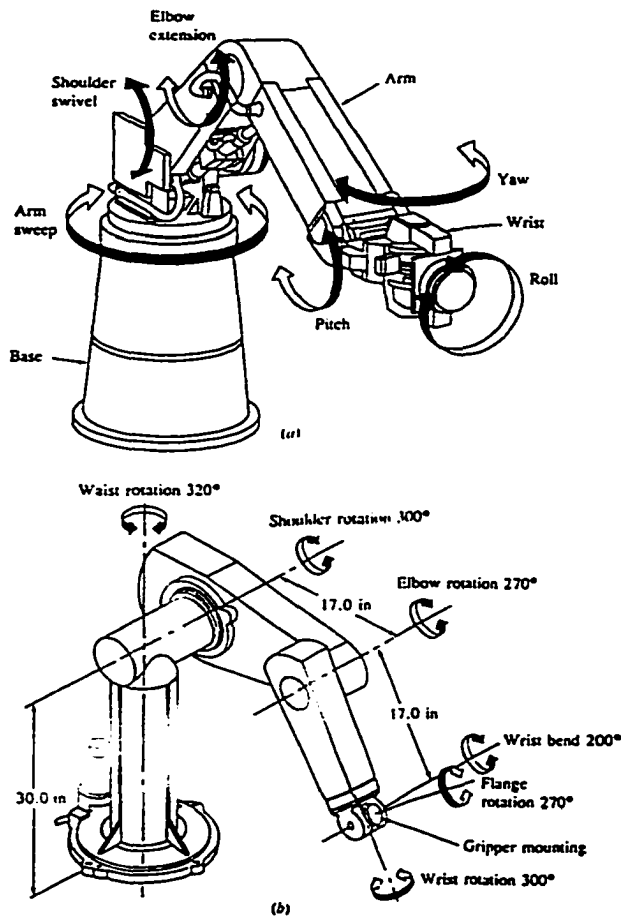


Figure 3.1 (a) Cincinnati Milacron T⁶ robot (b) PUMA 560 robot arm (Fu et al. 1987).

The robots shown in Figure 3.1 are considered six degrees of freedom (DOF) robots. The base, shoulder, and elbow joints provide 3 DOF and the wrist subassembly provides the other 3 DOF the pitch, the yaw, and the roll. Since each joint has a mechanical limit and the base is attached to a fixed frame, the robot arm is constrained to work within this spherical work volume. Normally, industrial robots are categorized based on their motion. The four basic motions can be accomplished by using Cartesian, Cylindrical, Spherical, or Revolute (articulated) coordinates (Figure 3.2).

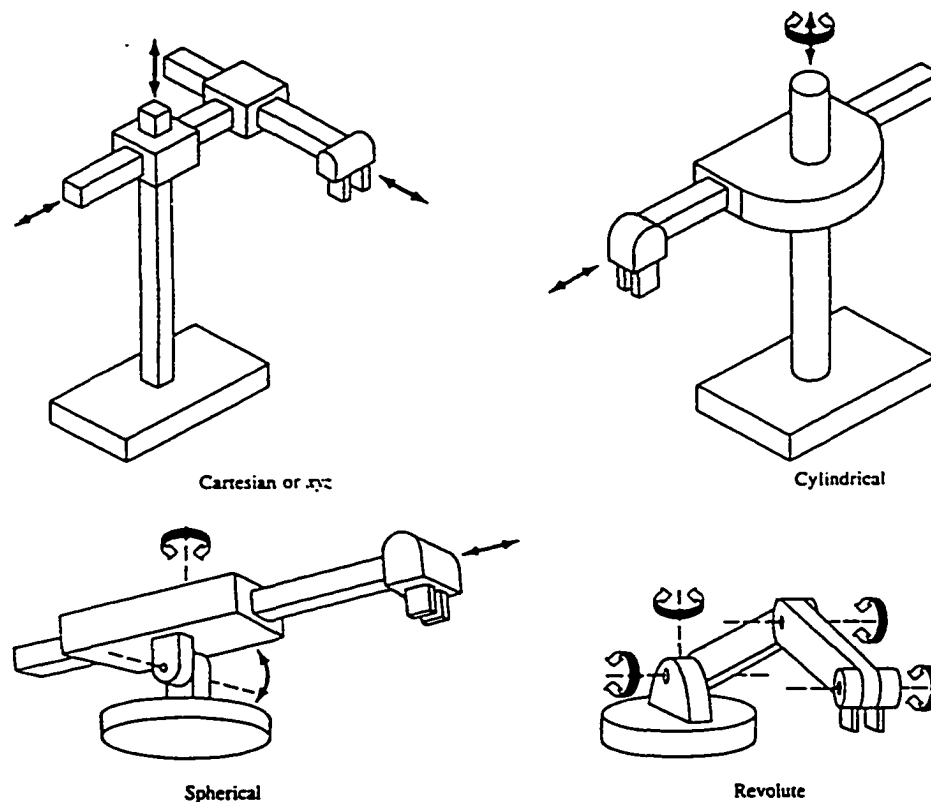


Figure 3.2 Various robot arm configurations (Fu et al. 1987).

A Cartesian robot has prismatic joints that can only move in the Cartesian x, y and z directions (e.g., IBM's RS-1 robot). A cylindrical robot has two linear and one rotary axis (e.g., Versatran 600 robot). A spherical robot has one linear and two rotary axes (e.g., Unimate 2000B). A revolute robot has at least three rotary axes (Figure 3.1 and Figure 3.2). Today most industrial robots are used for simple positioning tasks that have been preprogrammed by a user via a hand-held teach pendant. Thus, these robots are usually not equipped with sensors for obtaining information that is vital to its working environment. Therefore, robots are used mainly in relative simple, repetitive tasks that would bore or endanger a human operator (Fu *et al.* 1987).

In the electronics industry robotic assembly of PCBs reduce product cost and increase product quality. Cartesian and Scara robots are typically used for assembling large or irregularly shaped parts onto the PCBs (e.g., connectors, transformers, potentiometers, radial devices, crystals, light-emitting diodes, and large DIPs). To achieve accuracy and repeatability required in electronic assembly applications, mechatronics has become a necessity. Mechatronics is a term used to describe the combined technologies of electrical, mechanical, and computer engineering applied in design and development of high-precision machines, including assembly robots. Furthermore, machine vision technology is increasingly applied to assembly and inspection tasks traditionally performed by human operators. This is mainly because miniaturization of electronic components has increased circuit density and makes human assembly and inspection virtually impossible (Landers *et al.* 1994). Therefore in this thesis, the integration of several disciplines are exploited. A Scorbot ER-V robot and a vision system are used to develop an automated inspection system.

3.2 Description of Scorbot Model ER-V

The Scorbot ER-V is a light duty 5 DOF revolute robot (Figure 3.3).

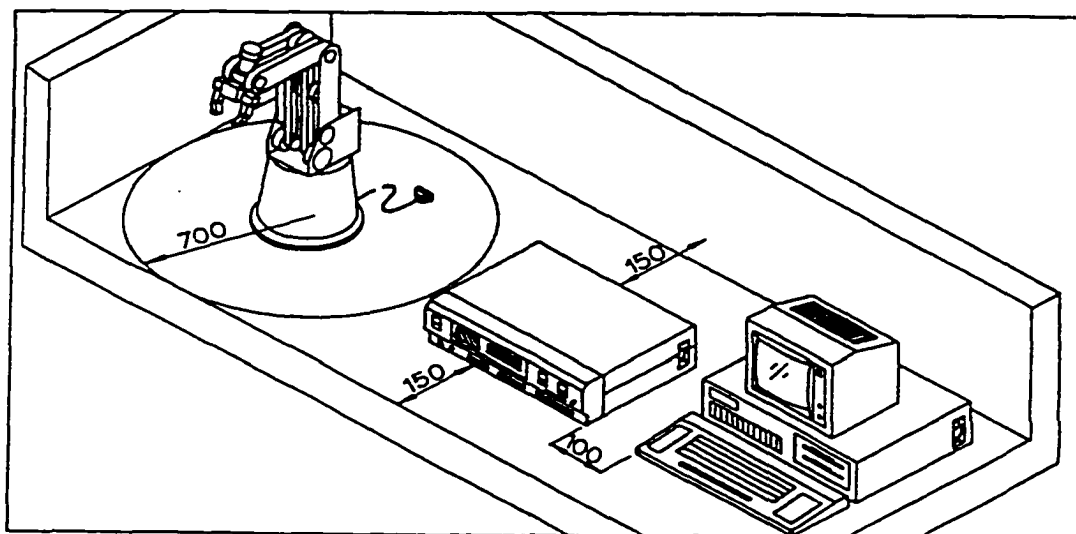


Figure 3.3 Scorbot ER-V System (Eched Robotec et al. 1992).

It has maximum carrying capacity of 1.0 kgf. The Scorbot robot motion positions can be programmed by an external teach pendant connected to the controller or via special software developed by a user or vendor. The user can develop the software using any high or low-level language (Basic, C++, Fortran, etc...). The programmed positions can be easily downloaded from the computer via the standard RS232C port. In order to position PCB boards accurately and consistently, the forward (Direct) and Inverse (Indirect) kinematics' equations of the Scorbot ER-V robot are used. The forward kinematics equation is used to determine the end effector position, given that the user knows the joint angles. More realistically, the user knows the position of an object and the trajectory an object is to follow. Thus, the user can define intermediate via points

along the trajectory to accomplish the task and use inverse kinematics equations of the robot to find the joint angles to control the robot. The forward and inverse kinematics equations are given in Appendix A. The Scrobot ER-V robot is used for loading and unloading the PCB board for automated visual inspection. The visual system then determines if there are any defects on the board. If there are defects, it will instruct the Scrobot ER-V robot to place the PCB board on a rework conveyor, otherwise, it will be sent to the next process.

3.3 Description of The Vision System

The vision system consists of a standard CCD camera, a PC computer with data acquisition board, a customized six DOF robot, a Scrobot ER-V robot, and a customized neural network software (Figure 3.4). The vision system uses multi-disciplines and standard manufacturing language to make the newly developed software easy to use in a manufacturing environment. A CCD camera is attached to the end effector of the robot. The programmer uses the PCB board's 2D CAD file to define the IC types and the centroid (x , y , z) of the ICs. It is assumed that the IC type has been previously defined and stored in the software database. If the IC type exists in the database, then the directions and positions of inspection are predefined, so that the software can compute and store the optimal focal distances needed to obtain the largest clear image of the IC under inspection. Once the optimal focal distances have been computed for all required orientations of the IC, the position of the camera can easily be found from the centroid (defined by the programmer), the shape and directions predefined for the IC, and the maximum magnification and resolution of the CCD camera. With this information, the

software uses the inverse kinematics equations of a six DOF robot to obtain the appropriate joint distances and angles. The software compensates for any delays and develops a sequence to achieve efficient movement and a clear image of the IC.

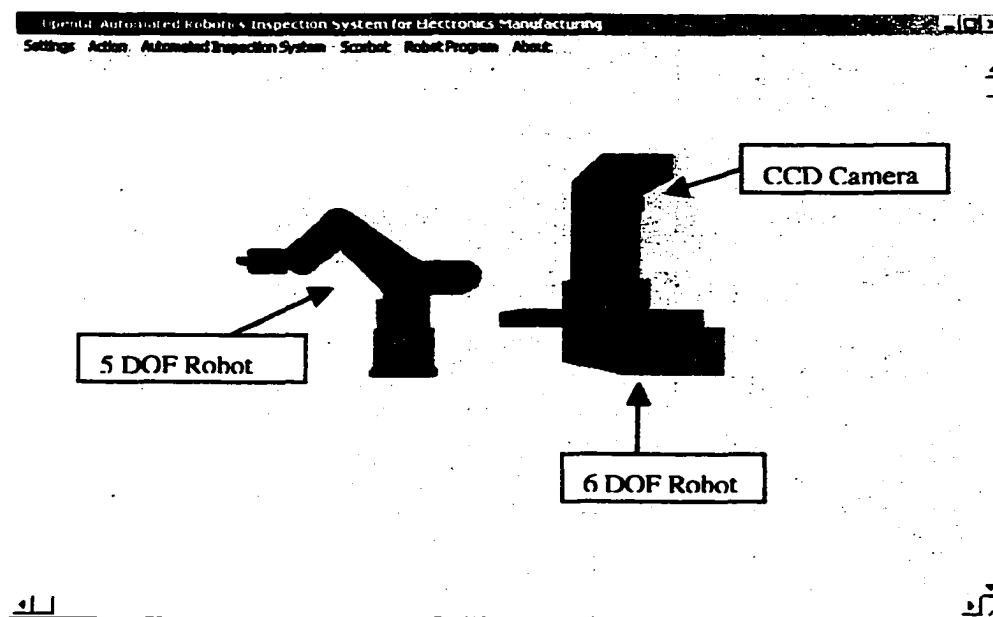


Figure 3.4 Automated Visual Inspection System

The information is then stored under the name of the PCB board inspected. When the program is executed, images are acquired from within a predefined window. All data outside this window will be discarded. The acquired image is saved and appended with additional information, namely, IC type, IC centroid position, direction of inspection, and number of pins. The previously trained and stored neural networks use this information to determine which IC identification network to use and how to preprocess the data, before it is fed to the IC network. The IC neural network will then output the pin number and the

defect type. With all input and output information now available, it can be stored and identified with the same scan code of the inspected PCB board. The program has to be executed once for each PCB board and all defects data are saved in a file containing the scan code for that PCB board. This file is then used to determine rework needed and to send signal to the Scorbot robot for appropriate placement of the PCB board on the rework conveyor or the next process conveyor. Files associated with rework PCB boards are arranged sequentially in a database with date and time inspected. Information in the files advice an automated rework station what needs to be done or a manual operator of how to fix the problem. In the next chapter, the neural networks topology used in this thesis is explained as well as the graphical device interface (GDI) software.

CHAPTER 4

NEURAL NETWORKS AND DEFECTS IN ELECTRONICS MANUFACTURING

The main inspiration of neural networks has been the biological neuron in our brain. It has approximately 10^{11} interconnected neurons (approximately 10^4 interconnections per neuron), which allow us to breath, read, move and think. The human brain is able to do all these complex tasks because of its parallel computational nature. Although circuits in computer systems operate 1 million times faster than biological neurons, they suffer because their architecture is sequential in nature and not highly parallel. Neural networks began to evolve in the late 19th and early 20th century by various scientists working in physics, psychology, and neurophysiology. This early work emphasized general theories of learning, vision, conditioning, etc. but did not include specific mathematical models of neuron operation. It was not until 1940's with the work of Warren McCulloch and Walter Pitts, who showed that networks of artificial neurons could in principle compute any arithmetic or logical function, interest in neural networks research grew. Their work is often described as the origins of modern neural networks. Work on neural networks continued by various researchers through the 1960s and was abandoned temporarily because of the work influence by Minsky and Papert and combined with the fact that there did not exist powerful digital computers on which to

experiment. It was not until the 1980s when personal computers became available to experiment, neural network research dramatically increased and new concepts were introduced (Hagan *et al.* 1996). Today neural networks have become a flexible and a successful tool. It has been applied to aerospace, automotive, banking, defense, electronics, entertainment, financial, insurance, manufacturing, medical, oil and gas, robotics, speech, securities, telecommunication, and transportation.

4.1 Neural Network Architecture

An artificial neural network mimics several characteristics of a biological neuron. Information processing occurs at many simple elements called neurons. Signals are passed between neurons over connection links. Each link has an associated weight that multiplies the signal transmitted and each neuron applies an activation function (usually nonlinear), also referred to as a transfer function, to its net input (sum of the weighted input signals of the links) to determine its output signal (Figure 4.1).

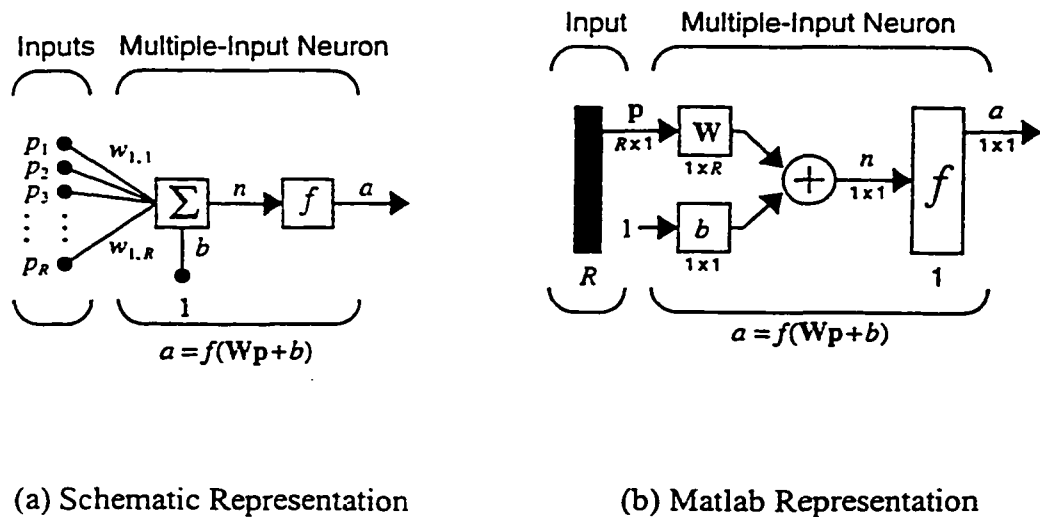


Figure 4.1 Multiple Input Neuron (Hagan *et al.* 1996).

Figure 4.1 shows a schematic and Matlab representations of multiple inputs. Bias being multiplied by corresponding input and bias weights are then summed up and results in a net value (n). The net value is then applied to a transfer function, which provides the neuron output (a). This is an example of a single neuron network. If we add S neurons in parallel each sharing the same input, the network is referred as a single-layer network with S neurons (Figure 4.2).

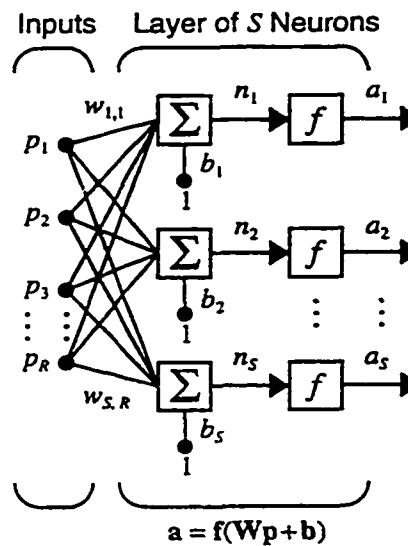


Figure 4.2 A single layer of S neurons with R inputs (Hagan et al. 1996).

Note that the network has S outputs. If we add another single-layer network with S^2 neurons sharing the output of the previous single-layer network as input, this network is referred as two layer network with S^1 neurons in the first layer and S^2 neurons in the second layer. If we keep adding layers with S^n neurons to the previous layer and using its output as current input, the network is referred to as a multi-layer network. Figure 4.3 shows a three layer network with R inputs and S^3 outputs

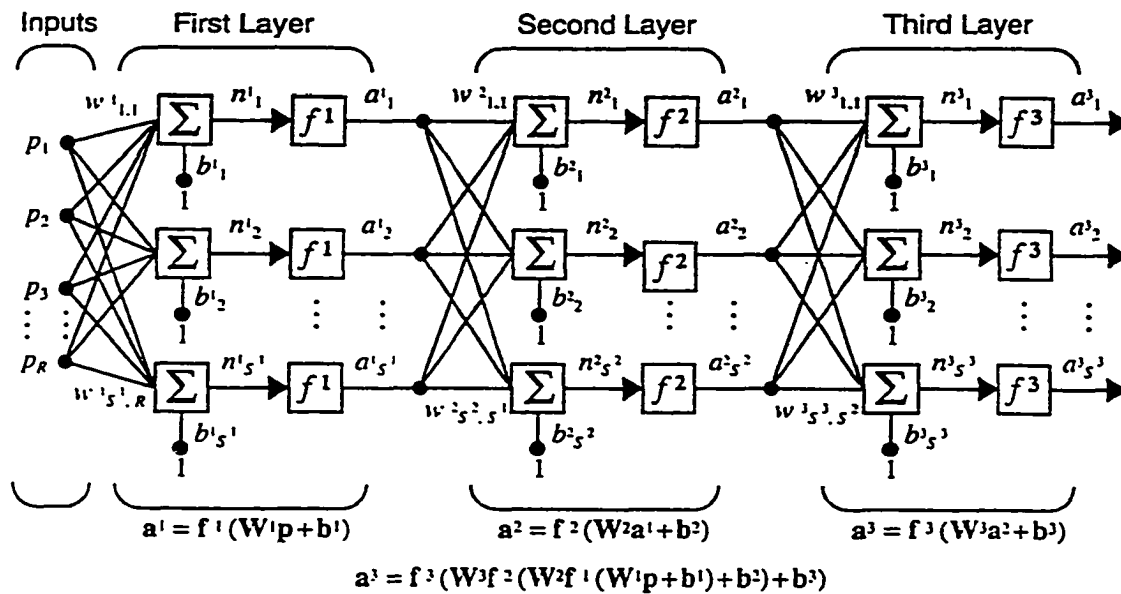











Figure 4.3 Three layer network with R inputs and S^3 outputs (Hagan et al. 1996).

A neural network is characterized by its architecture, training or learning algorithm, and its transfer function. The architecture is determined by the performance of the network. The goal is to find the minimum number of neurons in a layer necessary to accurately utilize the data to perform a particular function of interest, for example classifying the input data as belonging to one class or another. Determining how many layers and neurons to start with is usually determined by experience. Regardless, this is an iterative process. Start with a sufficiently large number of neurons and layers, train the network, and compare the performance. If the performance is good enough, try reducing the number of neurons or layers until the performance no longer meets the performance goal. At this point, the network should be optimal. The training or learning algorithm is

usually a function of speed of convergence. It is desired to implement the fastest algorithm available; which is especially true of large networks that may require months to train. The transfer function is usually chosen to improve convergence speed. The user can choose any function he/she believes would improve convergence. Table 4.1 lists the commonly used transfer functions.

Table 4.1 Commonly used transfer functions and Matlab equivalent (Hagan et al. 1996).

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1 \quad \text{neuron with max } n$ $a = 0 \quad \text{all other neurons}$		compet

4.2 Neural Network Topology

The automated inspection system uses a six DOF robot to position and orient the robot camera at a constant focal distance and angle to acquire IC's pin image. The positions of the ICs centroids and direction of acquisition have been previously defined by the programmer for a particular PCB. The IC's used for defect identification are gull-wing pin type. A schematic of this process is shown in Figure 4.4.

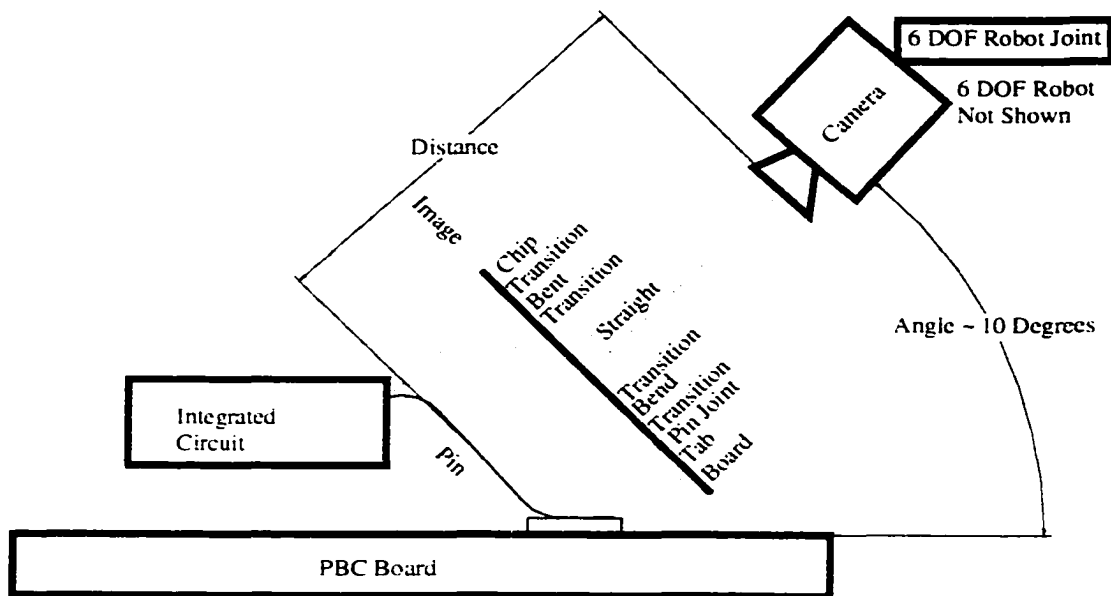


Figure 4.4 Schematic of image acquisition of a six DOF robot

The figure shows different regions of the image. The region of interest for defect identification is between the PCB board and the bottom bend of the gull wing pin. The defects that can be identified are the quality of solder joint, excessive pin bend, and lifted or missing pin. For solder quality, a resolution of at least 0.025 mm/pixel is needed but the camera has a resolution of only 0.1 mm/pixel. Therefore, the only defects that can be identified from the image are excessive bend and lifted pin defects.

The IC packages used for training, all had the same gull wing pin and same pin height, but the pitch varied. In real world applications the height and pitch for each different IC varies. Consequently, for the neural net topology, each IC that has different configuration will be trained individually and will have its own identification network. The Graphical Device Interface GDI software and the automated visual inspection system will select which network to be used depending on the IC being inspected.

4.3 Graphical Device Interface (GDI) software

The GDI software has several features. One of the features is to keep a database of all the ICs network weights and biases that are trained. The training is performed offline with acquired image data using Matlab neural network toolbox. The GDI software has the capability of adding new ICs network weights and biases. The GDI software will load a text program of a particular PCB that contains the number of IC to inspect and a list of text files containing the ICs inspection information (Figure 4.5).

```

3
ChipNumber1.txt
ChipNumber2.txt
ChipNumber3.txt

```

Figure 4.5 Information contained within PCB1data.txt file

The files are read sequentially to extract information on an IC. The information extracted contains the network type, inspection direction, number of images to acquire, and the robot execution procedure (Figure 4.6).

```

Chip Number      1
Chip Type        1
Number of Pins   40
Pin Height in Pixels  32
Pin Width in Pixels  3
Pin Gap in Pixels  3
Number of Images  4
Inspection Direction Sequence L B T R
Chip111.txt Chip112.txt Chip113.txt Chip114.txt
Chip Centroid (X,Y,Z) in mm  1      1      1
Mold Width in mm      120
Mold Height in mm     120
Mold Pin Separation in mm  1
Inspection Angle in (degrees) 10
Focal Length in mm    1000
Resolution in mm/pixel 0.1

```

Figure 4.6 File for the ChipNumber1.txt of IC contained within PCB1data.txt file

The six DOF robot uses this information to move the camera and acquire four images and saves the data in files Chip111.txt, Chip112.txt, Chip113.txt and Chip114.txt. The GDI software will then use the Chip Type to select the appropriate network and Chip data text files to identify defects. If defects are found, the GDI software reports the defect type and the pin number with additional information on the screen and to a report text file for the PCB being inspected. This file can be used by an automated rework station or a rework operator to correct the defect. The above procedure is repeated until all IC text files are read. At this point, the PCB is fully inspected and another PCB file can be loaded to continue inspection. This file could be different and corresponds to a different PCB

board. Thus the GDI and inspection system are flexible and allows inspection of different PCB boards sequentially. In addition to this flexibility, the GDI software shows a visual simulation of what is being inspected as the six DOF robot performs the task on-line. This would allow monitoring of inspection process at a remote location with feedback provided to the operator. The next section discusses the defects in electronics manufacturing.

4.4 Defects in Electronics Manufacturing

The competitive nature of the electronics industry and improvements in technology have led to the miniaturization of components, resulting in smaller and more powerful electronics products. Consequently miniaturization of components leads to some manufacturing difficulties. The most obvious difficulty is assembling these miniaturized components that possess more connections in less space, thereby increasing the defect probability. Because the components are so small, it is very difficult if not impossible for a human operator to assemble, it is slow and not efficient. Machine vision and automation solve these problems. Machine vision systems are used in the placement of surface-mount devices (SMDs) with lead pitches of 0.025 inch (0.6 mm) or less. In addition, vision systems are used to inspect solder joints before and after solder operations. Common inspection tasks on PCBs can be classified into three groups: solder paste deposition, component alignment, and solder joint integrity.

Solder paste deposition inspection tasks before component assembly on SMDs include; sufficient solder paste has been applied to solder paste screen and transferred to the PCB's component pads before a component is accurately placed by a pick and place

robot. Problems of insufficient solder on the pads or excessive solder paste on the pads causing shorts are common. The vision system inspects for these problems and automatically corrects solder quantity and squeegee pressure in the screening process. When the appropriate conditions are met, the component is placed on the PCB. Usually the sticky solder paste is sufficient to hold the component in place before it goes into the reflow process to bond the leads to the pads. Sometimes other factors cause the solder paste adhesion to be insufficient to maintain the component positioned correctly. Some of these factors are foreign matter, mechanical vibration during process transfers, incorrect paste chemistry, insufficient solder paste on pads, etc.... The vision system usually checks for any missing components or misaligned components. If components are misaligned, the PCB is placed on a rework conveyor or scrap area depending on the automated vision system used. Otherwise, the PCB is transferred to a reflow process to bond the components to the PCB.

Vibration, improper temperature, and excess solder or insufficient solder tend to cause the components to move and misalign slightly. Slight misalignment is normal but should not exceed a set amount. Machine vision systems use high-resolution solid-state cameras capable of a field of view of 0.5 inch (1.3 cm) that can detect spatial changes of 0.001 inch (0.025 mm) or less. This is sufficient resolution to measure leads and pads of 0.004-inch (0.1 mm) pitch. Excess solder tend to short pins or pads of components while insufficient solder degrades the integrity of solder joints. Solder joint problems include; bridging of solder between solder connections, lifted leads due to contamination, bent leads, and shortage/absence of solder in wetting angle (Landers *et al.* 1994).

In addition to machine vision, other technologies such as X-rays and laser scanning methods have been developed to inspect the integrity of solder joints, refer to Chapter 2. Table 4.2 shows the inspection capabilities of visual and X-ray systems (Sankaran *et al.* 1995).

Table 4.2 Defect types and inspection capabilities of visual and x-ray systems.
R= Reliable; N = Not Possible

Defect Types	Visual	X-Ray
Excess Solder	R	R
Insufficient Solder	R	R
No Solder	R	R
Dull Solder	R	N
Cold or Disturbed Joint	R	N
Porosity and voids	N	R
Bridging	R	R
Lifted Lead	R	R
Improper Wetting	R	N

It is obvious from Table 4.2 that a single sensing technique cannot determine all the possible defects that can occur during manufacturing. A practical solution would be to use a combination of technologies where appropriate to detect all possible manufacturing defects.

Vision systems have been developed for through-hole technology (THT). In many cases, the same SMT vision system can be used to inspect THT solder joints (Mengel 1990).

CHAPTER 5

TRAINING ALGORITHM FOR INSPECTION AND CLASSIFICATION

With personal computers (PC) speed reaching 2 GHz, and the availability of commercial neural network software packages, the training algorithm selected for defect identification is only a matter of speed of convergence and memory requirements. Table 5.1 shows the neural network algorithms that are available in Matlab and their speed of computation.

Table 5.1 Matlab algorithm speed comparisons (Hagan et al. 1998).

Function	Technique	Time	Epochs	Mflops
traingdx	Variable Learning Rate	57.71	980	2.50
trainrp	Rprop	12.95	185	0.56
trainscg	Scaled Conj. Grad.	16.06	106	0.70
traincgf	Fletcher-Powell CG	16.40	81	0.99
traincgp	Polak-Ribière CG	19.16	89	0.75
traincgb	Powell-Beale CG	15.03	74	0.59
trainoss	One-Step-Secant	18.46	101	0.75
trainbfg	BFGS quasi-Newton	10.86	44	1.02
trainlm	Levenberg-Marquardt	1.87	6	0.46

The above speed comparisons were performed on a 1:10:1 network on a data set with 41 input/output pairs until the mean square of error performance of 0.01 was obtained (Matlab Neural Net Toolbox). The Matlab users guide recommends that the Levenberg-Marquardt algorithm be attempted first. If this algorithm requires too much memory, then the BFGS algorithm (trainbfg) be tried, or one of the conjugate gradient methods, although the Rprop algorithm (trainrp) is also fast (Table 5.1).

One of the problems that occur during neural network training is called overfitting; that is, the error of the training set is driven to a very small value. When data is presented to the trained network, the error is large. Basically the network has memorized the training set, but it has not learned how to generalize to new situations. One method that can solve this problem is to use a network large enough to provide an adequate fit (not to over or under fit the training data), which raises the question of how large the network should be. This is a very difficult question to answer since it is difficult to know beforehand how large the network should be. There are also two other methods used in neural networks to improve generalizations. They are regularization and early stopping. Regularization is a technique used to modify the performance function of the algorithm, which is usually chosen to be the mean sum of squares of the network errors. Early stopping is what the name suggests; stop the network before it overlearns the data set. These two techniques require that the user be competent and skilled using neural networks and know how to adjust the performance function or how early to stop the network to prevent over or under fitting the data set. This is an iterative process that takes time, experience, and lot of patience. Fortunately, Matlab's toolbox provides a method for automatic regularization called Bayesian regularization to adjust the performance

function optimally and Levenberg-Marquardt training algorithm for speed of convergence. The Matlab function to accomplish this is `trainbr`. Furthermore, one feature of this algorithm is that it provides a measure of the number of network parameters (weights and biases) that are effectively utilized by the network. If we increase the network size, the number of parameters should remain the same. Thus, it takes the guesswork out in determining the optimum network size. Although slower than the pure Levenberg-Marquardt training algorithm, the user does not have to be an expert on neural networks. He/she only needs to acquire adequate network size, weights and biases for the neural network, which generalizes new data adequately. Matlab's `trainbr` function was used for training all the data sets in this work.

5.1 Acquisition of Data

Pin defects were created on ICs of PCB boards. Since there were insufficient boards available to create defects and obtain sufficient data, several images of ICs with no defects were obtained and new images were created that simulates defects. Table 5.2 shows the defect criteria used by Matlab defect generation program.

The Matlab defect generation program reads in a text file (containing one integer value for each pin defect wanted) and a real image containing no pin defects (Figure 5.1 (a)).

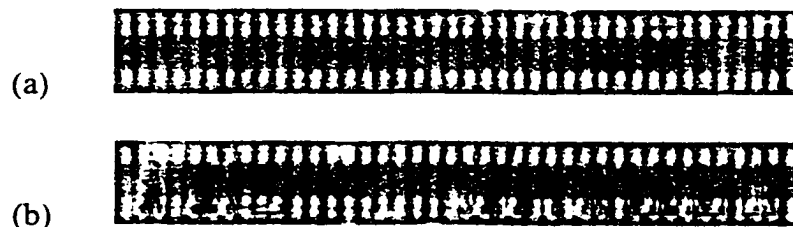


Figure 5.1 (a) Real image with no defect (b) Simulated defect image

The defect generation program outputs an image file containing the defects (Figure 5.1 (b)) and a text file containing columns of the neural network responses corresponding to each pin defect shown as positive and negative ones in a row (Table 5.2).

Table 5.2 Criteria used for defect generation for data generation

Integer	Defect Type	Neural Net Response
0	Straight Pin and Not Lifted	-1 -1 -1
1	Bent Pin 1 Pixel to the Left and not Lifted	-1 1 -1
2	Bent Pin 1 Pixel to the Right and not Lifted	-1 1 -1
3	Bent Pin 2 Pixel to the Left and not Lifted	1 -1 -1
4	Bent Pin 2 Pixel to the Right and not Lifted	1 -1 -1
5	Bent Pin 3 Pixel to the Left and not Lifted	1 1 -1
6	Bent Pin 3 Pixel to the Right and not Lifted	1 1 -1
7	Straight Pin and Lifted	-1 -1 1
8	Bent Pin 1 Pixel to the Left and Lifted	-1 1 1
9	Bent Pin 1 Pixel to the Right and Lifted	-1 1 1
10	Bent Pin 2 Pixel to the Left and Lifted	1 -1 1
11	Bent Pin 2 Pixel to the Right and Lifted	1 -1 1
12	Bent Pin 3 Pixel to the Left and Lifted	1 1 1
13	Bent Pin 3 Pixel to the Right and Lifted	1 1 1

Note the first two columns of the network output response shown in Table 5.2, corresponds to a binary number. For example, -1 -1 corresponds to a zero decimal value, which is used to classify a straight pin, not bent. A binary number -1 1 corresponds to a decimal value of one, bent pin 1 pixel to the right or left and so on. Also the last column of the network output response corresponds to a lifted or not lifted pin state, -1 for not

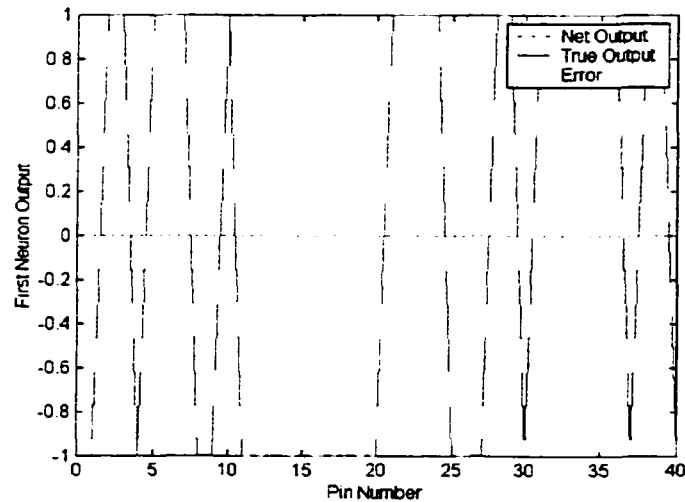
lifted and 1 for lifted respectively. The image of Figure 5.1 (a) was acquired from a quad package containing 40 gull wing pins on each side for a total of 160 pins. The pitch of this package is 0.3 mm. The image height is 32 pixels. The width of the pin is 3 pixels, and the gap between pins is 3 pixels. Thus the pitch is 3 pixels (0.3 mm), which corresponds to 0.1 mm/pixel CCD camera resolution. Matlab's imaging toolbox and functions were used to read the image file, convert the grayscale file into numerical values from 0-255 into an array of numbers which has a 1 to 1 correspondence with row and column of the pixel value in the image. Once this was accomplished and the locations of pin data (rows and columns) are known from the image, it is easy to generate bent pins to the left or to the right of its central location, and also lifted or broken pins from the input text file. Soldering defects (insufficient or excess solder) were not generated because the resolution of the CCD camera was 0.1mm/pixel, which is not sufficient to detect soldering defects. A resolution of at least 0.025mm/pixel is needed for solder defect identification. The newly generated numerical array of pins with defects is then transformed to an image with Matlab image function and saved as an image file along with its neural network response text file. The defect image and the output response text files were then used to train several networks and to find the optimum network size and the network architecture.

5.2 Network Architecture

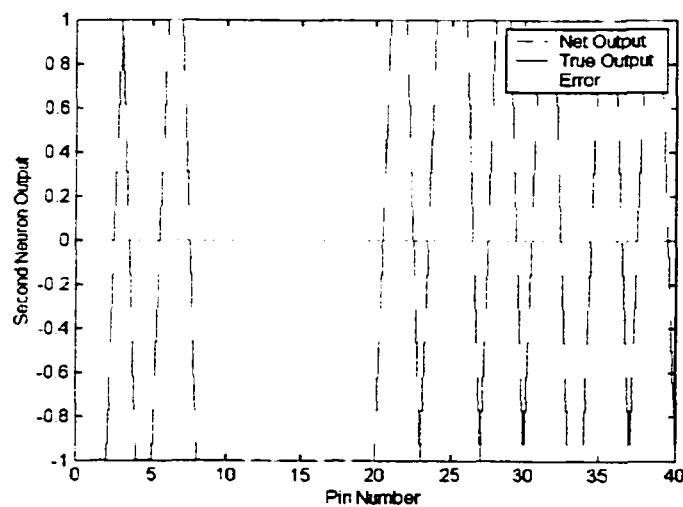
As stated earlier Bayesian regularization was used for training the neural networks with Matlab's `trainbr` function. It was also discussed that every IC will have its own neural network used for defect identification. The main reason is that the pitches of

IC packages vary. Also, pin height, pin configuration, and package technologies are different. It would be extremely difficult to find one neural network that can identify defects for all ICs with different package technologies, pin configuration, etc. Also due to limited resolution of the CCD camera, only the defects shown in Table 5.2 were considered to be possible in a manufacturing environment. Obviously the network used in this work has three outputs corresponding to the defects shown in Table 5.2. Generally a network is started with sufficient neurons and layers. It is common to start with a three layer neural network and then reduce or increase the size as necessary to find an optimum network that has good performance and is able to generalize. The following networks were tried for one IC type of the following sizes 288:6:3:3, 288:6:3, 288:3:3, 288:2:3 and 288:3. The value of 288 is the size of the input vector corresponding to one pin inspection area. The inspection area consists of a width of 9 pixels (3 pixel gap + 3 pixel pin + 3 pixel gap) multiplied by 32-pixel height for a pin totaling 288 pixel values or inputs. The input pixel values are each divided by 255 (max gray scale value) to keep the input normalized between 0 and 1. The numbers after each colon represent the number of neurons in its corresponding layer respectively. For example, a 256:6:3:3 neural network is composed of an input vector of 256 normalized pixel values with 6 neurons in the first layer 3 neurons in the second layer and 3 neurons in the output layer. All the networks above were successful in classifying the defects presented to the network, except 288:2:3 network which consistently failed to identify the defects in Table 5.2 of 3 pixels bent left or right and lifted. Therefore, the minimum network size was reduced to 288:3 neural network, 1 layer network with 3 neurons and 288 inputs and 3 outputs. The same 1 layer network with 3 neurons and different input vector size corresponding to a different IC

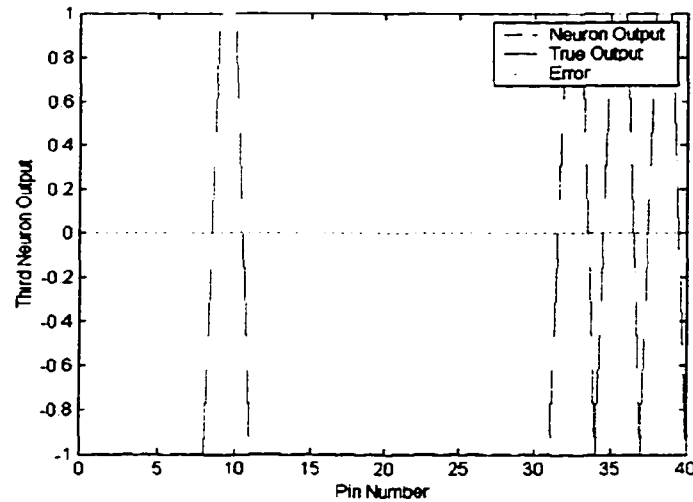
pitch and pin height was used to train similar ICs with the same defects criteria given in Table 5.2. Figure 5.1 (b) image and output response file generated were used to train the 288:3 neural network. Figure 5.2 (a), (b), and (c) show that each graph contains the corresponding outputs of each output neuron, the correct output response for that neuron and the error corresponding to the defect generated.



(a) Output Performance of First Neuron vs. Pin Number



(b) Output Performance of Second Neuron vs. Pin Number



(c.) Output Performance of Third Neuron vs. Pin Number

Figure 5.2 Classification performances for 288:3 neural network

Each graph has 3 lines - correct output response, network output response and the difference between the actual value and the network output for that neuron (error). It can be seen from the graphs, the error response is flat and very close to zero. This indicates perfect identification by the neural network. One other network was trained using the same procedure discussed previously on IC-two. Similar results were obtained (Appendix B). The training time for the networks was from 8 to 20 minutes on a 400MHz personal computer. The weights and biases for the ICs were saved as text files for use by an on-line monitoring system discussed in Chapter 6.

CHAPTER 6

ON-LINE MONITORING SYSTEM

The on-line monitoring system was developed using Pro-E, C++, and Scrobot ER-V language. Open Graphics Library (OpenGL) was used for high speed rendering of the Scrobot ER-V robot and for the animation of the inspection system. Microsoft Visual C++ (ver. 6.0) compiler was used to compile and debug the software. Microsoft WIN32 platform was used for developing high-speed rendering using OpenGL. Pro-Engineer (Pro-E) was used to create all solid models; for example, components of the robots, ICs, and PCB. The Pro-E models were exported as binary Stereo Lithography (STL) files and were used by the on-line monitoring system to create three-dimensional animations. Eye Image Calculator developers guide and software from IO industries was used to acquire gray scale images from the CCD camera and saved in TIF graphics format. The TIF files were used for training using Matlab and for defect identification by the on-line monitoring system. Matlab neural network modules were used to generate data for training the neural network, generate appropriate weights for the neural network, and export the weights, biases, and pixel data as ASCII files to the on-line monitoring system software. The Scrobot ER-V robot's direct and inverse kinematics equations were used by the on-line monitoring system to synchronize the robot motions with the animation. The following sections discuss the graphical features of the on-line monitoring system.

6.1 Graphical Features

The on-line monitoring system is a typical windows program with its set of menus, windows, radio buttons, check boxes, edit boxes, etc. (Figure 6.1).

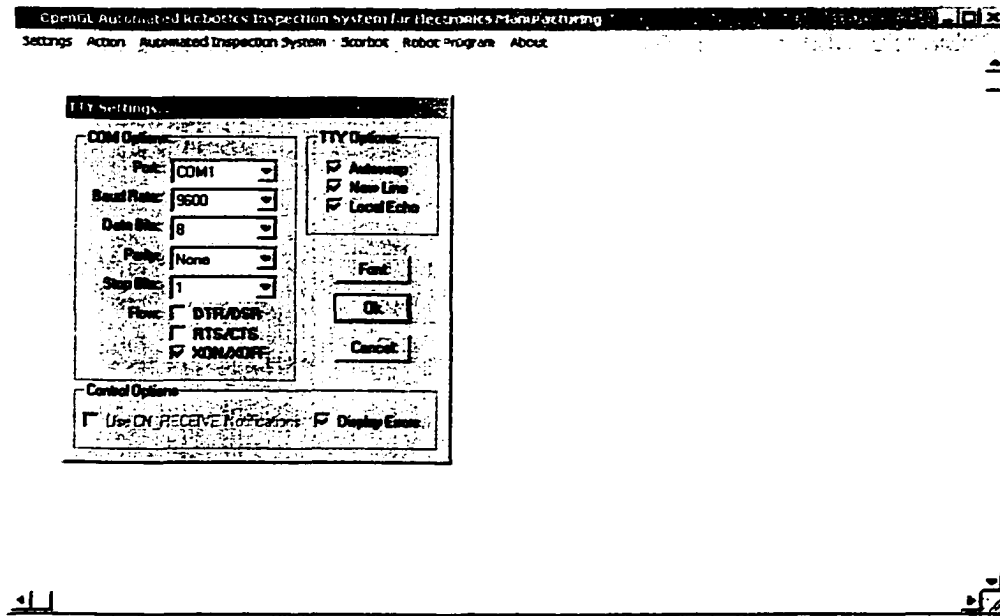


Figure 6.1 Snapshot of the on-line monitoring system communications options

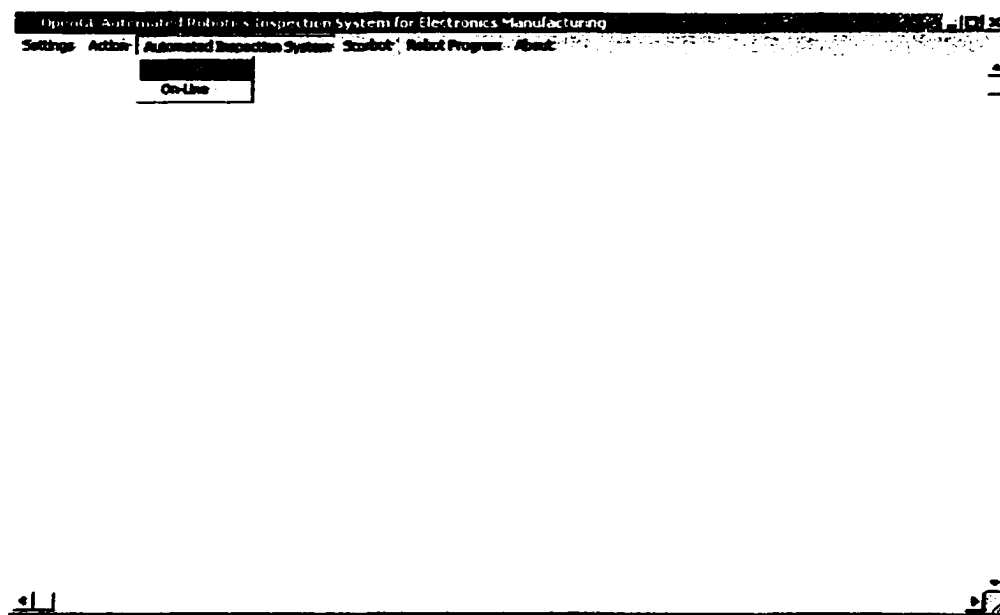
The program utilizes both Microsoft graphics libraries (slow) and OpenGL graphics (fast) and EYE IMAGE software libraries for image acquisition. The on-line monitoring system is capable of direct communication with the robot's controller using the RS232C communications port. The user selects the correct communications port, settings, text display format, font size, echo, etc. After this, the user simply selects connect. At this time the user can type any legal controller command, and the robot will execute the command. The user can also select the shift + ~ key, while connected, to shift from manual joint mode (press j key) to Cartesian mode (press c key). In joint mode the user

can press the 1/Q combination to move the robot's base to the left or the right respectively; 2/W combination to move the robot's upper arm up or down; 3/E combination to move the robot's forearm up or down; 4/R combination to move the robot's wrist pitch up or down; 5/T combination to move the robot's wrist roll clockwise or counter clockwise; and 6/Y to open and close the jaw. Similarly in Cartesian mode, the same keys are used to move the robot's hand in the +/- X, +/- Y, +/- Z, +/- pitch, and +/- roll. The user can press the shift + ~ key to exit manual mode and enter direct mode. In direct mode the user can define and save positions in the controller for future use by the on-line inspection system.

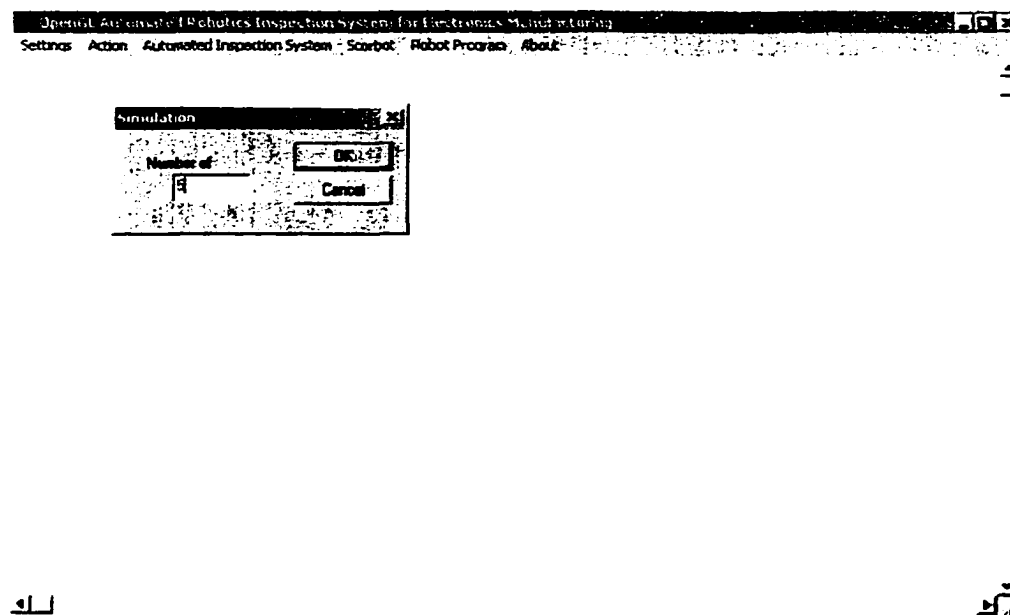
The on-line monitoring system keeps an internal database of all weights and biases used by the neural network for defect identification. The user can add or delete IC's weights and biases as needed. The on-line system has an option to simulate the inspection process or to start monitoring an on-line process. If on-line monitoring system option is selected, the computer directly controls the inspection process of image acquisition of the PCB with the Scorbot ER-V robot, the visual inspection system movements, image acquisition by the CCD camera, defect reporting, and good/rework bin placement using the Scorbot ER-V robot if any defects are found on the PCB. While the Scorbot ER-V robot is being commanded on-line, a synchronized animation of the process and the status report of defects classification are displayed on the computer screen for monitoring. The on-line monitoring system keeps track of the PCB and its defects for the rework process.

6.2 Simulation Option

The simulation options under the automated simulation menu (Figure 6.2 (a))



(a)



(b)

Figure 6.2 Snapshots of (a) Simulation selection (b) Number of PCBs to simulate

allow the user to animate the process of an actual system and estimate approximately the inspection time. When the user selects the simulation option, a pop up window appear and the user selects the number of PCB boards to simulate Figure 6.2 (b).

As can be seen from Figure 6.2 (b), the number of PCBs to simulate was chosen to be five. This means that the simulation will inspect five PCB boards. The inspection process involves four stations; PCBs station, inspection station, rework station, and shipping station. The Scorbot ER-V robot is in charge of transferring the PCBs to the appropriate stations. The Scorbot ER-V robot program needs only eight positions defined to make the appropriate movements; that is, two positions defined for each station. The station positions consist of a retract position and an extend position. The retract positions are chosen to avoid collisions with objects and the extend positions are chosen for picking and placing the PCB board at the correct locations on the appropriate stations. Snapshots of the animation of the inspection process are shown in Figure 6.3 (a), (b), (c), (d), (e), (f), (g), and (h) for one inspection cycle. The simulation starts at position 2 (retracted position), which faces the shipping station (Figure 6.3 (a)). The Scorbot ER-V robot rotates about its base joint -90 degrees (clockwise viewed from above) to position 3 (retracted position), which faces the PCBs station (Figure 6.3 (b)). The Scorbot ER-V robot extends its arm to position 4 with its jaw open and roll joint at 90 degrees (Figure 6.4 (c)). At position 4, the jaw closes gripping the PCB board. The Scorbot ER-V robot retracts back to position 3. The Scorbot ER-V robot then rotates about its base joint 180 degrees counter clockwise (CCW) to position 5 towards the inspection station (Figure 6.3 (d)). The Scorbot ER-V robot extends its arm to position 6 (Figure 6.3 (e)), opens its jaw, placing the PCB board on the inspection station, and retracts back to position 5. The

inspection station begins to inspect the ICs on the PCB board and reports any defects on the right half of the inspection screen (Figure 6.3 (f)). Note, the IC under inspection is shown in red. When the inspection of the PCB board is complete, the inspection system signals the Scorbot ER-V robot to remove the PCB board. The robot picks-up the PCB board and places it in the rework or shipping station. In this case, the PCB board has some defects; hence it will go to the rework station. The Scorbot ER-V extends its arm to position 6 with its jaw open, closes its jaw at position 6, gripping the PCB board, retracts to position 5, and then rotates to position 7 (Figure 6.3 (g)). Position 7 faces the rework station. The Scorbot ER-V robot extends its arm to position 8 (Figure 6.3 (f)), opens its jaw releasing the PCB board, retracts to position 7, and rotates to position 2 (Figure 6.3 (a)). As mentioned earlier, position 2 faces the shipping station, and at this point the robot waits for another fetch PCB board command from the monitoring computer. The monitoring computer would instruct the Scorbot ER-V to repeat the inspection cycle until all the PCB boards are inspected.

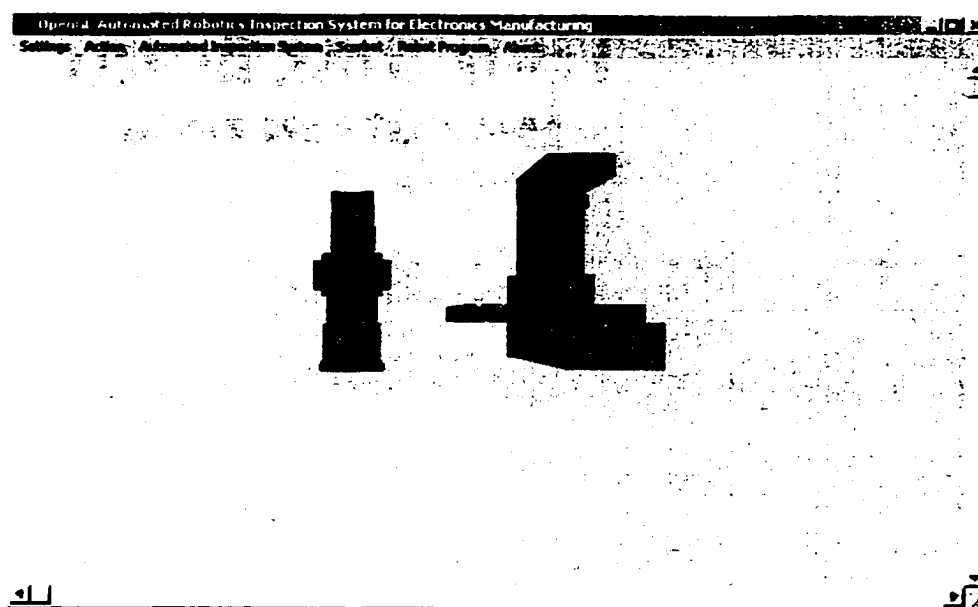


Figure 6.3 (a) Position 2, retracted, and facing the shipping station

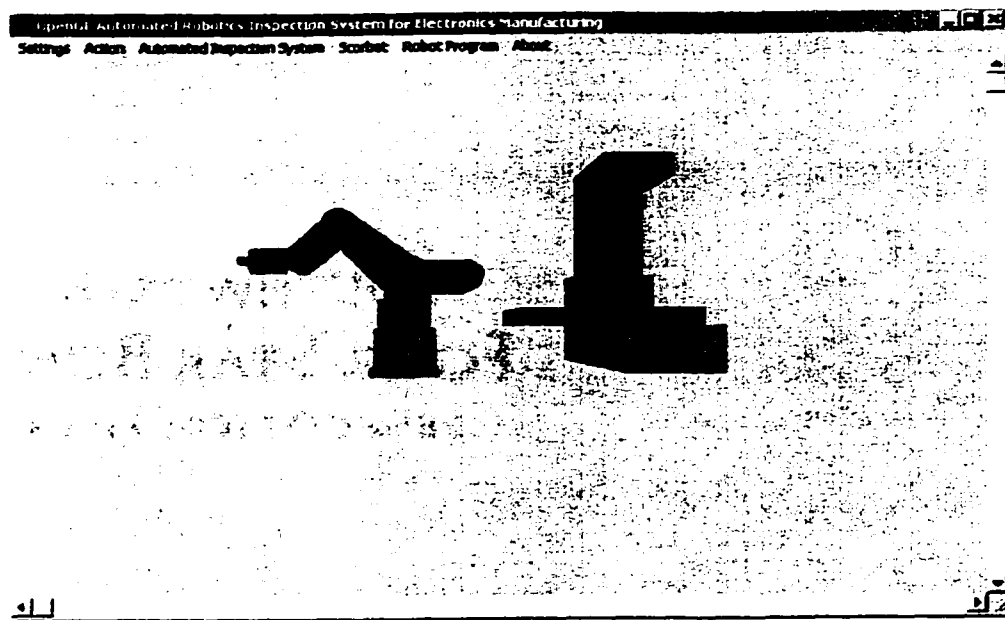


Figure 6.3 (b) Position 3, retracted, and facing PCBs station

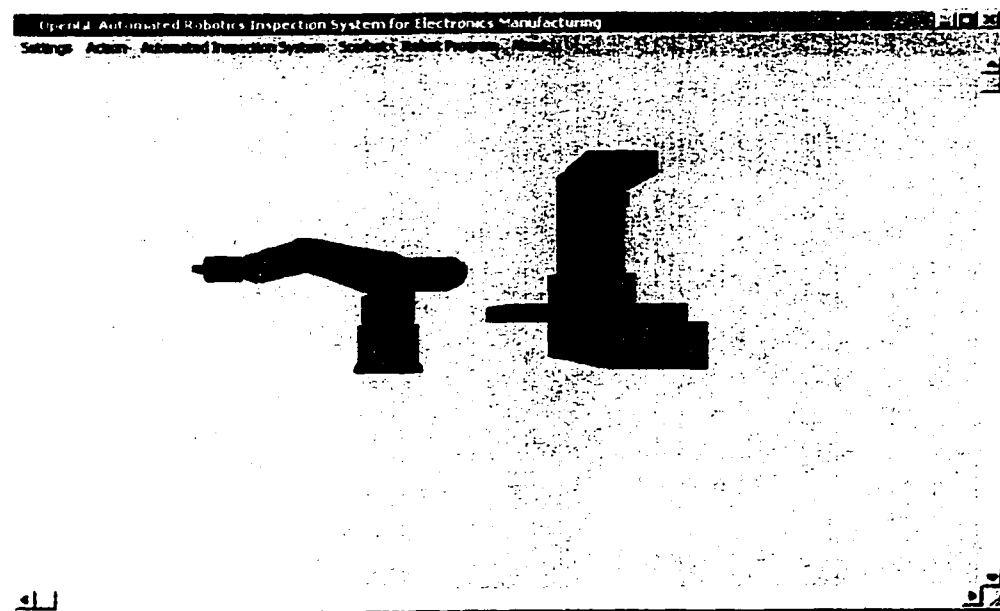


Figure 6.3 (c) Position 4, extended arm, and on PCBs station

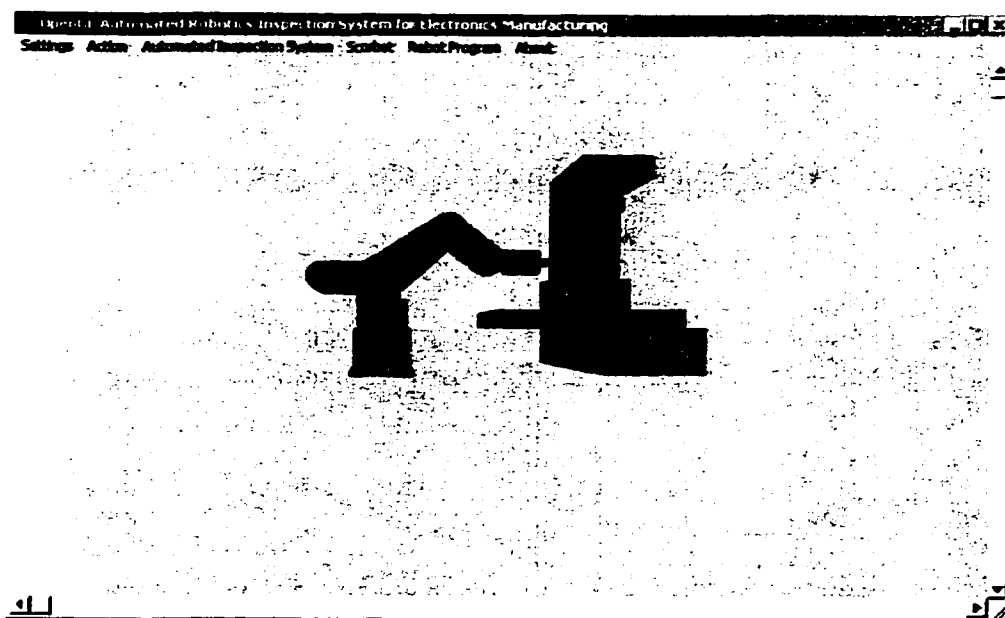


Figure 6.3 (d) Position 5, retracted, and facing inspection station

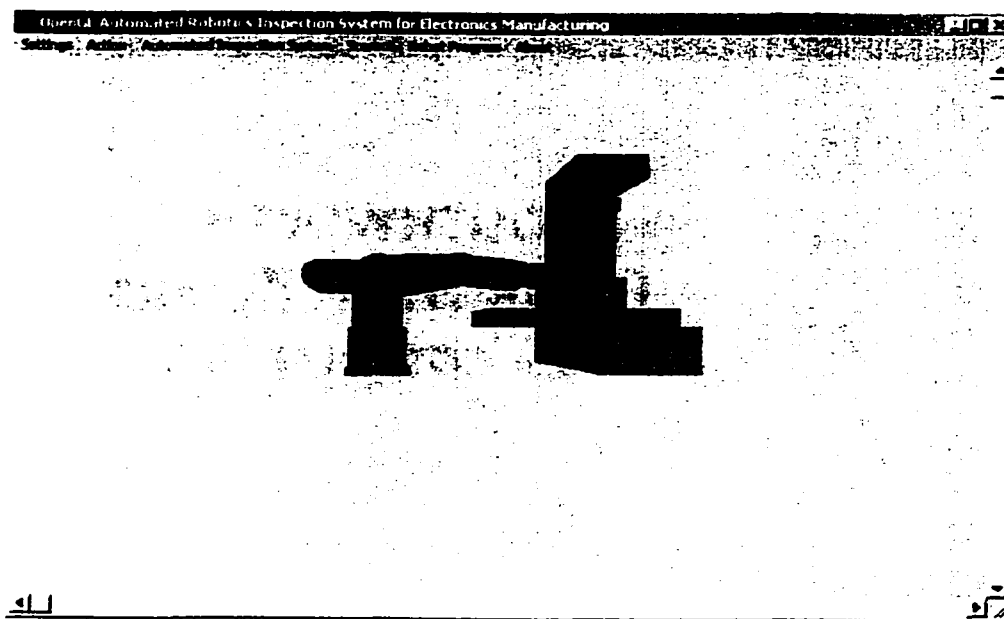


Figure 6.3 (e) Position 6, extended arm, and on inspection station

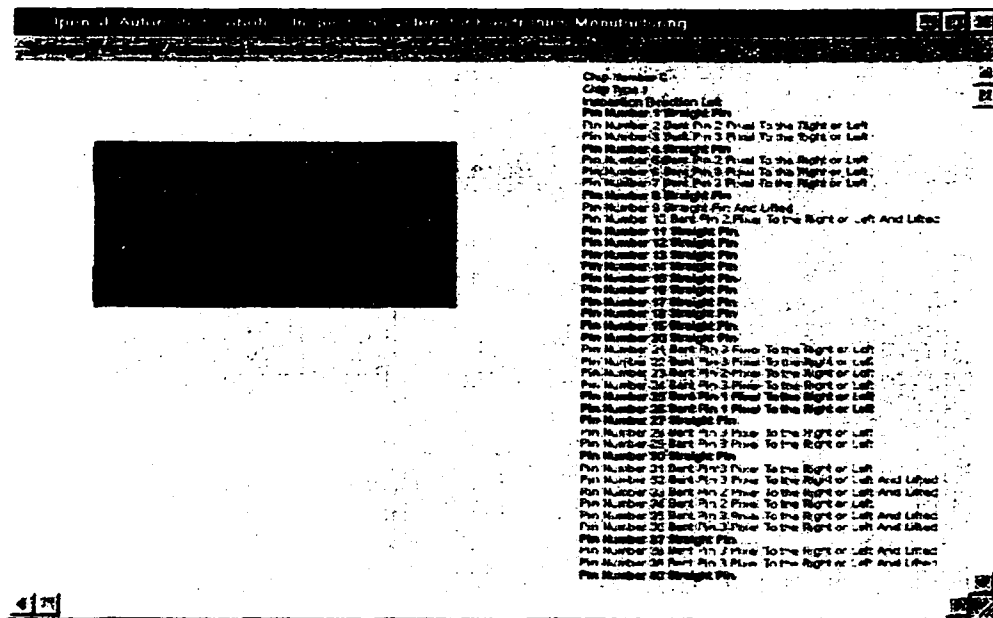


Figure 6.3 (f) PCB Inspection station defect monitoring

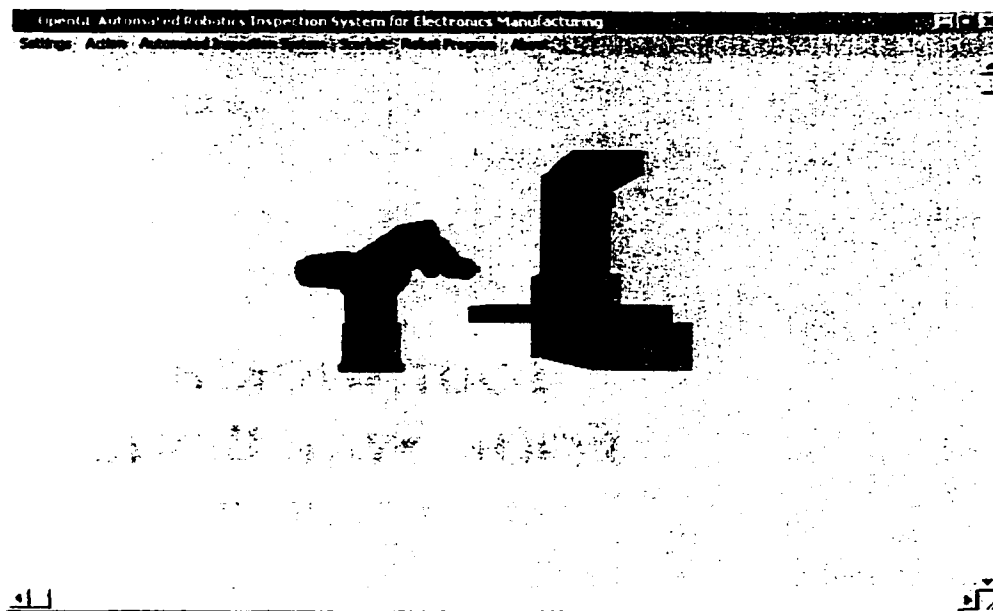


Figure 6.3 (g) Position 7, retracted arm, facing rework station

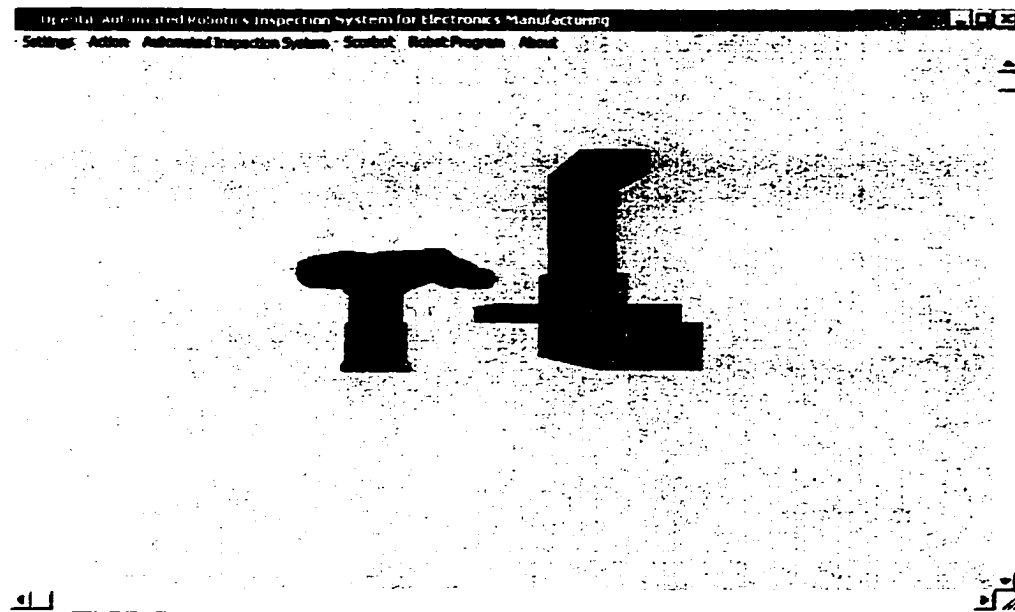


Figure 6.3 (h) Position 8, extended arm, and on rework station

Figure 6.3 Snapshots of the Scorbot ER-V inspection cycle

6.3 On-Line Option

To use the on-line option, the user has to select the communication settings correctly under the settings menu shown in Figure 6.1 and then select the connect option under the action menu (Figure 6.4). The settings shown in Figure 6.1 are the default settings. At this point the user can type any legal controller command, for example home. If the Scorbot ER-V robot homes successfully, communication is set correctly. If everything has been set correctly and the user can communicate with the Scorbot ER-V controller, on-line monitoring of inspection of PCB could begin.

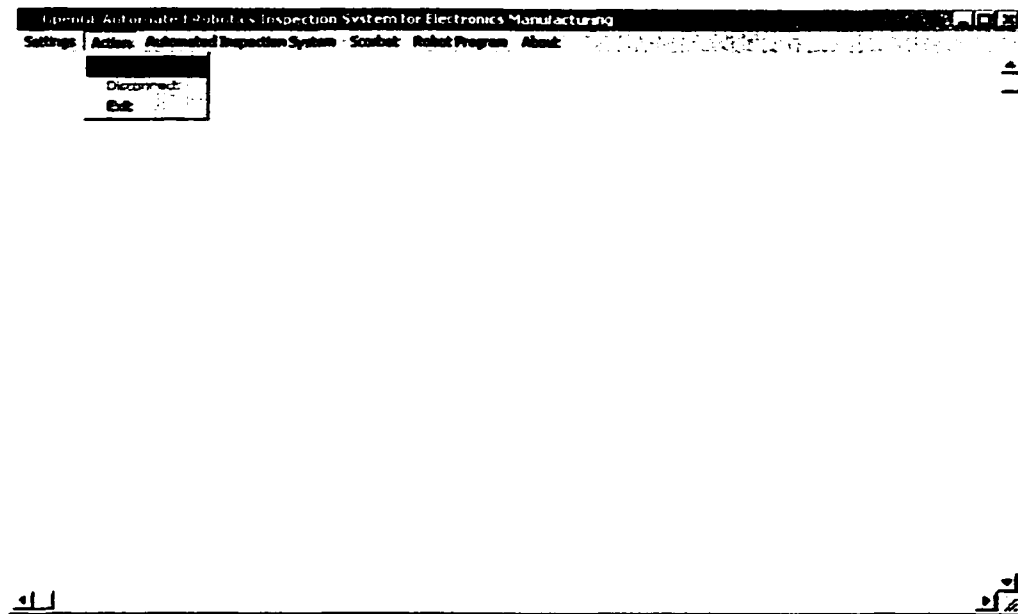


Figure 6.4 Snapshot of action menu

The on-line option under the automated inspection system menu (Figure 6.2 (a)) uses real data (previously acquired using the EYE image software) and monitors the inspection system on-line while an animation of the process is visualized on the screen (Figure 6.3 (a)-(h)). A report of the pin data is saved in a report text file for the PCB under inspection. Figure 6.5 shows a sample of the report file.

Chip Number 1
 Chip Type 1
 Inspection Direction Left

Pin Number 1 Straight Pin
 Pin Number 2 Bent Pin 2 Pixel To the Right or Left
 ...

Pin Number 37 Straight Pin
 Pin Number 38 Bent Pin 3 Pixel To the Right or Left And Lifted
 Pin Number 39 Bent Pin 3 Pixel To the Right or Left And Lifted
 Pin Number 40 Straight Pin

```

Chip Number 1
Chip Type 1
Inspection Direction Bottom

Pin Number 1 Straight Pin
Pin Number 2 Bent Pin 2 Pixel To the Right or Left
Pin Number 3 Bent Pin 3 Pixel To the Right or Left
...

Chip Number 1
Chip Type 1
Inspection Direction Left

Pin Number 1 Straight Pin
Pin Number 2 Bent Pin 2 Pixel To the Right or Left
Pin Number 3 Bent Pin 3 Pixel To the Right or Left
...

Chip Number 1
Chip Type 1
Inspection Direction Right

Pin Number 1 Straight Pin
Pin Number 2 Bent Pin 2 Pixel To the Right or Left
Pin Number 3 Bent Pin 3 Pixel To the Right or Left
...

Chip Number 2
Chip Type 2
Inspection Direction Right

Pin Number 1 Straight Pin
...
```

Figure 6.5 PCB board defect report

If the on-line system inspects five PCB boards, there will be five PCB board defect reports. Currently, these reports are saved as text files, but in the future these files would be saved as binary files and possibly compressed to minimize disk space.

6.4 Scorbot ER-V Menu Items

The Scorbot menu has two selection items; robot coordinates and clear screen (Figure 6.6). The robot coordinates selection allows the user to select the physical coordinates of the Scorbot ER-V gripper (Figure 6.7) and have an animation drawn at that position (Figure 6.8). This option is useful when the user is trying to decide where to define retract and extend positions described in section 6.2. Furthermore, as the user wants to try different positions, the user can connect to the controller and define the positions manually using the Scorbot ER-V language. This option has not been automated, but can be easily implemented in the future. The 'clear' screen option simply wipes any of the animations on the screen. The screen returns to its original white background.

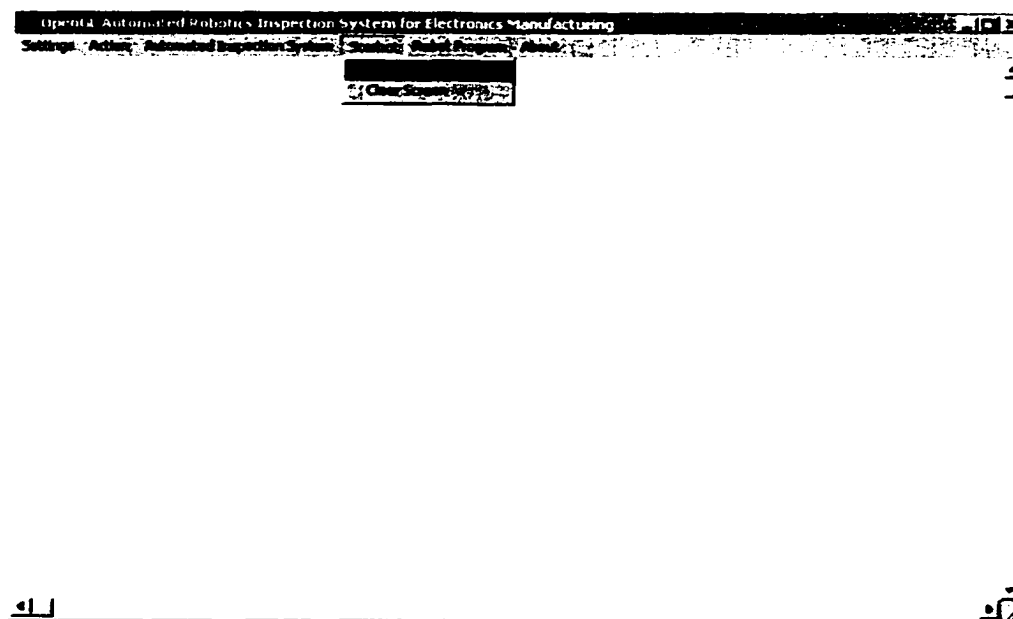


Figure 6.6 Scorbot menu items

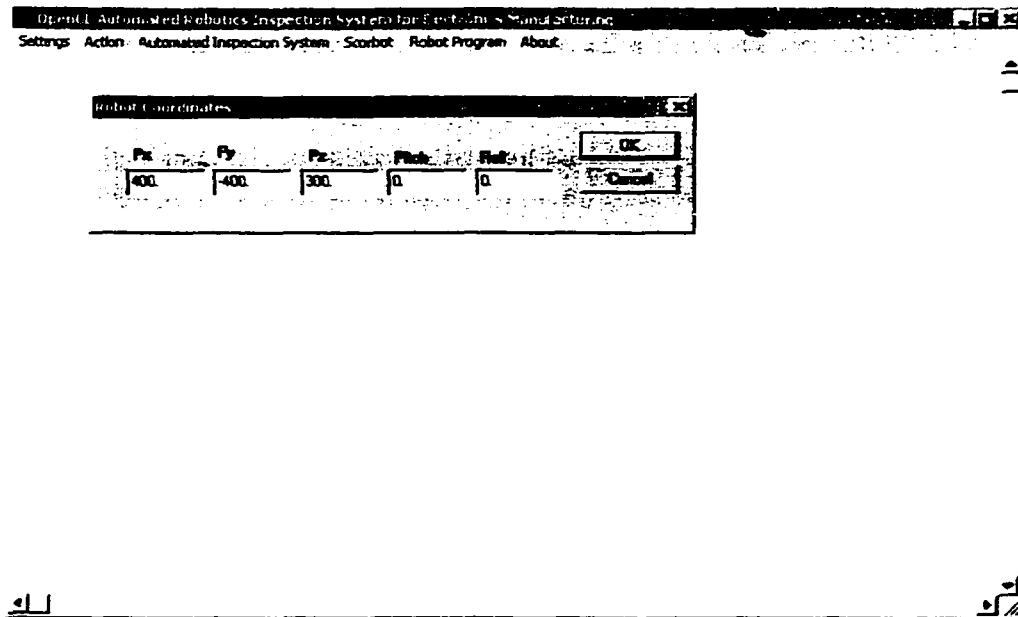


Figure 6.7 Robot coordinate selection pop up window

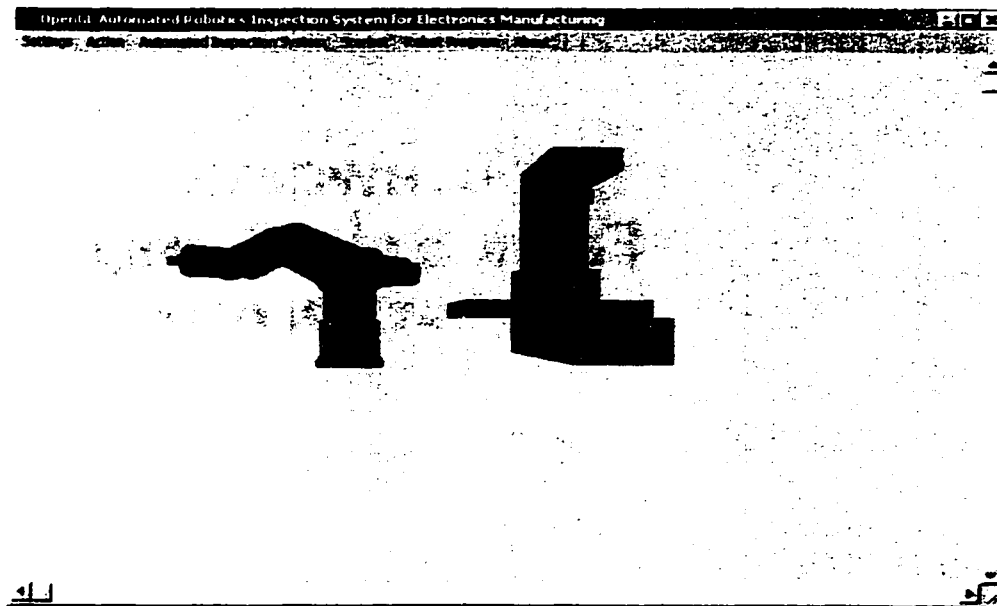


Figure 6.8 Robot coordinate animation result of coordinate selection

6.5 Robot Program Menu Items

The robot program menu has two options: program load and program run

(Figure 6.9). The program load option loads two text files; `program1.txt` and `program1pos.txt`. The `program1.txt` file contains the inspection systems instructions such as move 1, open 10, close 0, inspect 1, and rotate 4. The first instruction in the `program1.txt` file is the number of instructions that follow. The `program1pos.txt` file contains the position number, x, y, z, pitch, roll, and speed position information. As before, the first instruction is the number of positions that follow. These two files are loaded in memory, but no action takes place until the program run option is selected. Do not use the program run option if no program is loaded into memory; an error will occur. Always load a program before using the program run selection. When the program run selection is made and the communications have been activated, all position data in memory is defined and stored in the Scorbot ER-V controller. If all the data has been successfully transferred, the program instructions in memory are executed and an animation of the process is displayed as described previously in section 6.2. Note, the pop up window used for on-line simulation pops up (Figure 6.2 (b)) and ask for the number of PCBs to inspect. As before, choose the number of PCBs to inspect. The difference between the on-line option discussed in section 6.2 and the robot program option is that the positions are hard coded and never change; whereas, the robot program options are flexible and can change, based on the user needs. The robot program menu is not a comprehensive menu: other selection items as program save, position save, position load, and program database can be included. These options should be included for future work to facilitate writing and testing the Scorbot ER-V inspection system programs, and probably even include some animation before the program is implemented on-line.

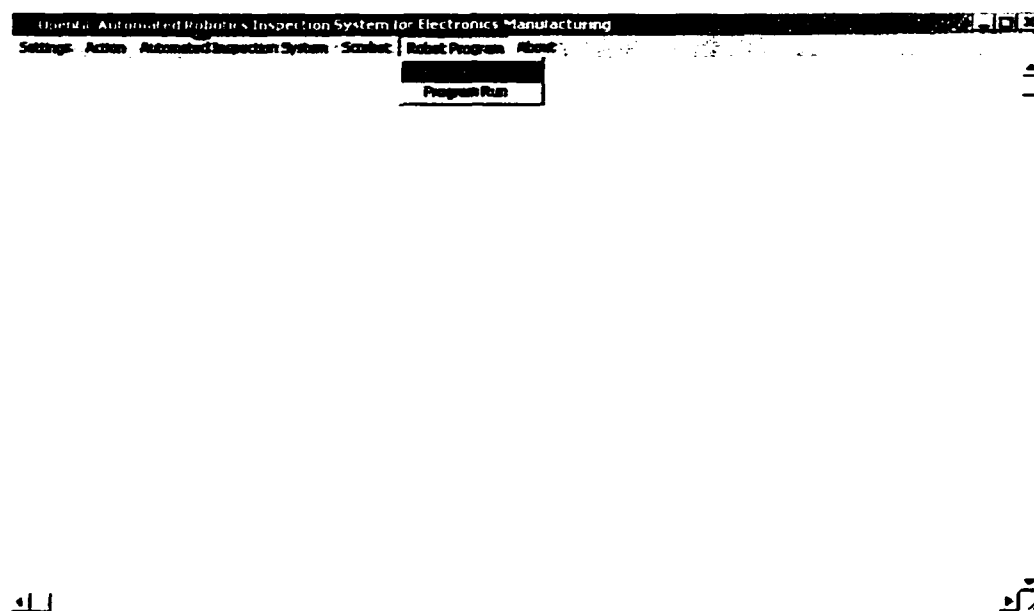


Figure 6.9 Robot program menu items

6.6 About Menu Item

The 'about' selection menu item creates a dialog window showing the developer information use as developer name and copyright information (Figure 6.10).

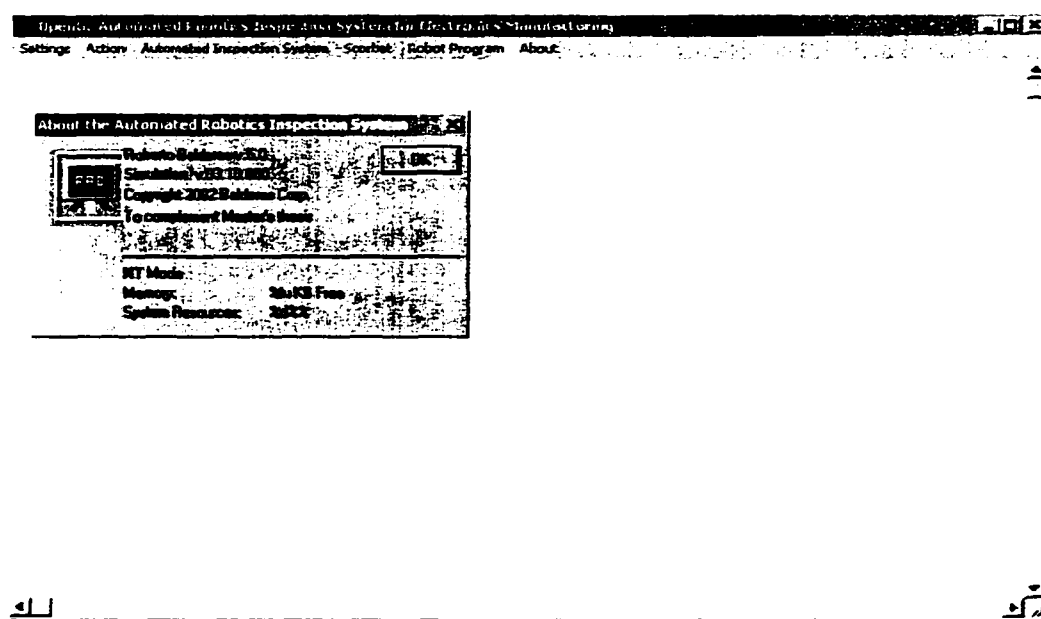


Figure 6.10 Developer information

CHAPTER 7

CONCLUSIONS

Advancements in technology have led to miniaturization of electronic components, leading to smaller and more powerful electronic products. Miniaturization of electronic products has lead to problems, which affects production yields and thus profitability. Because, the components are so small, it is very difficult, if not impossible for a human operator to assemble; it is slow and not efficient, therefore not profitable or competitive. This has led electronics manufacturers to switch from manual production operations performed traditionally by human operators to automated systems.

Chapter 2 reviewed several technologies being implemented in automated visual inspection systems and manufacturers approach to quality improvement of automated visual inspection systems. From the literature review, it was found that defect data in the form of images acquired with a CCD camera is the fastest and has been used successfully in the electronics industry.

In this thesis, neural network tools were used to classify defects in integrated circuits. Neural network was chosen because it has been successfully used in various applications for decades. Neural networks are robust and can be implemented on-line and are highly parallel, which allows simultaneous computation using several processors to reduce computational time and thus reduce inspection time.

Based on the defect criteria given in Table 5.2, it was determined that the optimum neural network size for the ICs being inspected was a single layer network with three outputs. Simulated and real defect data were used to test the robustness of the neural network. The performance of the network was found to be 100 percent accurate. The defect criteria was kept constant for each IC. The only thing that varied was the inspection area used for each IC due to the fact that the ICs have varying lead pitches, lead heights, and number of leads. Each of the ICs was trained separately, with each IC having a varying input vector, and thus a different network with its own weights and biases. The weights and biases for each of the networks were then saved as text files and used by an on-line monitoring system, which stores the information in an internal database.

The on-line monitoring system was developed using Pro-E, C++, and Scorbot ER-V language. Open Graphics Library (OpenGL) was used for high speed rendering of the Scorbot ER-V robot and inspection system animation. Microsoft Visual C++ (ver. 6.0) compiler was used to compile and debug the software. Microsoft WIN32 platform was used to develop high-speed rendering using OpenGL. Pro-Engineer (Pro-E) was used to create all solid models; for example, components of the robots, ICs, and PCB. The Pro-E models were exported as binary Stereo Lithography (STL) files and were used by the on-line monitoring system to create three-dimensional animations. Eye Image Calculator developers guide and software from IO industries was used to acquire gray scale images from the CCD camera and saved in TIF graphics format. The TIF files were used for training using Matlab and for defect identification by the on-line monitoring system. Matlab's neural network modules were used to generate data for training the neural

network, generate appropriate weights for the neural network, and export the weights, biases, and pixel data as ASCII files to the on-line monitoring system software. The Scorbots ER-V robot's direct and inverse kinematics equations were used by the on-line monitoring system to synchronize the robot motions with the robot animation. The on-line system simulation is by no means complete or perfect, but it lays the foundation for future development and work. Improvements in weight and bias database could be made. For example, instead of using text files binary files could be used and if necessary using data compression techniques. This will reduce disk space necessary to maintain a large database of weights and biases. Data management techniques could be explored to add and delete information from the database. The defect report database could be converted to binary files, eliminating unwanted information to reduce disk space. A CCD camera of at least 0.025 mm/pixel should be used if identification of the quality of solder joint leads is required.

REFERENCES

- Eched Robotec (1982) Ltd.,1992, *Laboratory Manual Scorbace Level 5*, Tel Aviv, Israel.
- Eched Robotec (1982) Ltd.,1992, *ACL Advanced Control Language for Scorbac-ER-V*, Tel Aviv, Israel.
- Eched Robotec (1982) Ltd.,1992, *Scorbac-ER-V User's Manual*, Tel Aviv, Israel.
- Fukuda T., Michelini R., Potkonjak V., Tzafestas S., Valavanis K., and Vukobratovic M., 2001, "How Far Away is Artificial Man," IEEE Robotics & Automation Magazine, pp. 66-73.
- Fu K.S., Gonzalez R.C., Lee C.S.G., 1987, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, Inc. New York.
- Hagan M.T., Demuth H.B. and Beale M., 1996, *Neural Network Design*, PWS Publishing Company, Boston, MA.
- Hagan M.T., Demuth H.B. and Beale M., 1998, *Neural Network Toolbox User's Guide*, The Math Works Inc., Natick, MA.
- Hata S.,1990, "Vision Systems for PCB Manufacturing in Japan," Industrial Electronics Society,1990, IECON '90., 16th Annual Conference of IEEE, 1990 , pp. 792-797, vol. 1.
- Hawkins K., Astle D., 2001, *OpenGL Game Programming*, Prima Publishing, Roseville, California.
- IO Industries Inc.. 1997, *Eye Image Calculator Developer's Guide*, Printed in Canada.
- Kashitani A., Takanashi N., and Tagawa N, 1993, "A Solder Joint Inspection System for Surface Mounted Pin Grid Arrays," Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of IECON '93., International Conference on, 1993, pp. 1865-1870, vol. 3.
- Landers T.L., Brown W.D., Fant E.W., Malstrom E.M., and Schmitt N.M., 1994, *Electronics Manufacturing Processes*, Prentice Hall, Upper Saddle River, NJ.

- Mengel P., 1990, "Automated Inspection of Solder Joints on PC Boards by Supplementary Processing of 3D and Gray-Level Images," Industrial Electronics Society, 1990. IECON '90., 16th Annual Conference of IEEE, 1990, pp. 786-791. vol. 1.
- Mitsubishi Robots, *Mitsubishi Industrial Micro-Robot System Model RV-M1 Instruction Manual*, Printed in Japan by Mitsubishi Electric Corporation.
- Moller T., Haines E., 1999, *Real-Time Rendering*, A K Peters, Ltd., Natick, MA.
- Okabe T., Akaiwa M., Shirakawa T., Yokouch T., and Sugimoto T., 1993, "Final Visual Inspection System for LSI Packages," Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of ICON '93., International Conference on, 1993, pp. 1877-1881, vol.3.
- Petzold C., 1988, *Programminmg Windows*, Microsoft Press, Redmond, Washington.
- Thompson C.M., Shure L., 1995, *Image Processing Toolbox User's Guide*, The Math Works Inc., Natick, MA.
- Sankaran, V., Chartrand, B., Millard, D.L., Embrechts, M.J., and Kraft, R.P., 1995, "Automated Inspection of Solder Joints-A Neural Network Approach," IEEE Int'l Electronics Manufacturing Technology Symposium, pp. 232-237.
- Sankaran, V., Kalaukin, A.R., and Kraft R.P., 1998, "Improvements to X-Ray Laminography for Automated Inspection of Solder Joints," IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part C, Vol. 21, No. 2, April 1998, pp. 148-154.

APPENDICES

APPENDIX A

DIRECT AND INVERSE KINEMATICS EQUATIONS FOR SCORBOT ER-V ROBOT

The following information was obtained from the Scorbot ER-V Laboratory manual sections 6-4 to 6-15 and the results are summarized below. Figure A-1 shows the gripper location definition in XYZ coordinate system, which is used to derive the direct and inverse kinematics equations for the Scorbot ER-V.

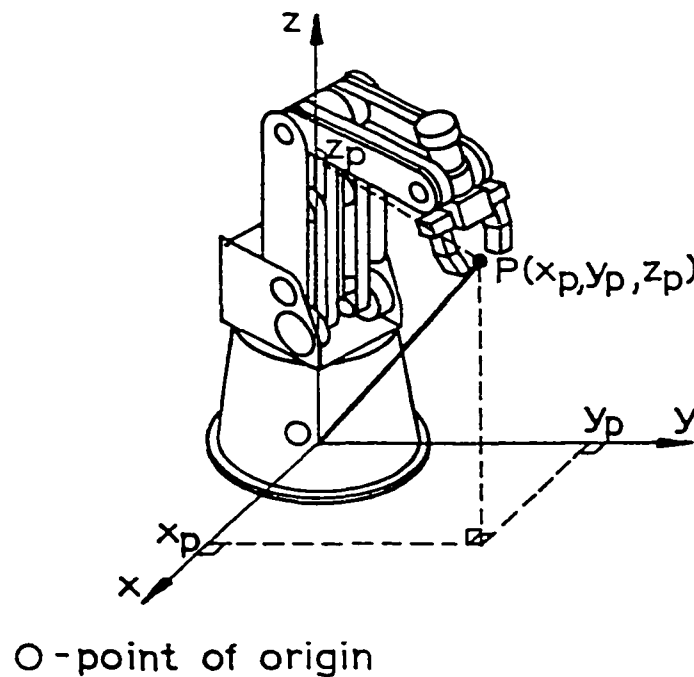


Figure A-1 Gripper location definition in XYZ coordinate system
(Eched Robotec et al. 1982)

The direct kinematics equations are shown below. Note that the joint angles are all measured from the horizontal plane, which is parallel to the floor.

$$\text{Radius} = L_1 + L_2 \cdot \cos(\sigma_2) + L_3 \cdot \cos(\sigma_3) + L_4 \cdot \cos(\sigma_4)$$

$$X = \text{Radius} \cdot \cos(\sigma_1)$$

$$Y = \text{Radius} \cdot \sin(\sigma_1)$$

$$Z = L_0 + L_2 \cdot \sin(\sigma_2) + L_3 \cdot \sin(\sigma_3) + L_4 \cdot \sin(\sigma_4)$$

$$\text{Pitch} = \sigma_4$$

$$\text{Roll} = \sigma_5$$

Where

L_0 = height from the robot base to axis 2 = 349 mm

L_1 = horizontal distance from base center to axis 2 = 16mm

L_2 = length of link2 (shoulder) = 221 mm

L_3 = length of link 3 (elbow) = 221 mm

L_4 = distance from pitch axis to end of closed gripper = 145 mm

Use Figure A-2 to reference the joint angles and link lengths.

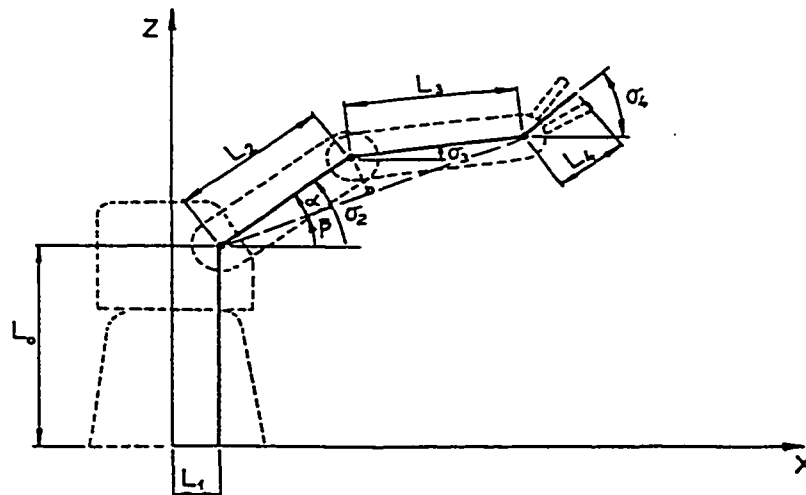


Figure A-2 Relation of robot joint coordinates to XYZ coordinates
(Eched Robotec et al. 1982)

The inverse kinematics equations for the Scorbob ER-V are,

$$\sigma_1 = \text{ArcTan}(Y/X)$$

$$\text{if } X < 0 \text{ and } Y < 0 \quad \sigma_1 = \sigma_1 - 180$$

$$\text{if } X < 0 \text{ and } Y > 0 \quad \sigma_1 = \sigma_1 + 180$$

$$\sigma_2 = \beta + \alpha$$

$$\sigma_3 = \beta - \alpha$$

σ_4 and σ_5 are specified in the XYZ position definition, they are the pitch and roll respectively.

Where

$$\beta = \text{ArcTan}(dz/dr)$$

$$\alpha = \text{ArcTan}(tb/ta)$$

$$.dr = \sqrt{(X^2 + Y^2)} - L_1 - L_4 * \text{Cos}(\sigma_4)$$

$$.dz = Z - L_0 - L_4 * \text{Sin}(\sigma_4)$$

$$.ta = \sqrt{(dr^2 + dz^2)}/2$$

$$tb = \sqrt{(L_2^2 + ta^2)}$$

APPENDIX B

PERFORMANCE RESULTS OF IC TWO

IC two has twenty pins, ten on each side. The pin width is three pixels. The pin gap is six pixels, and the pin height is thirty-two pixels. This gives an inspection area of $(2*6 + 3) * 32 = 480$ pixels. Figure B-1 (a) and (b) show the images of good and generated images used for training. Figure B-2 (a), (b) and (c) show the performance of 480:3 neural network.

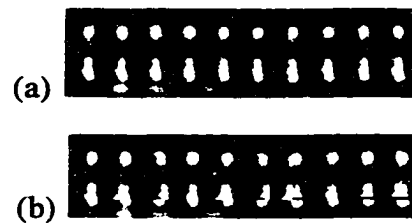
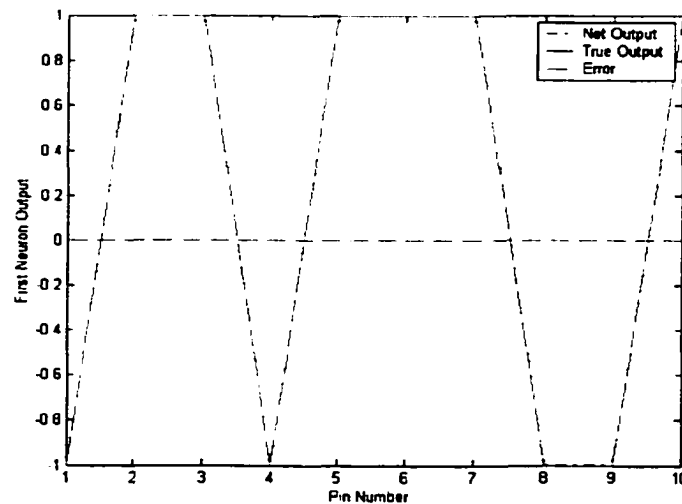
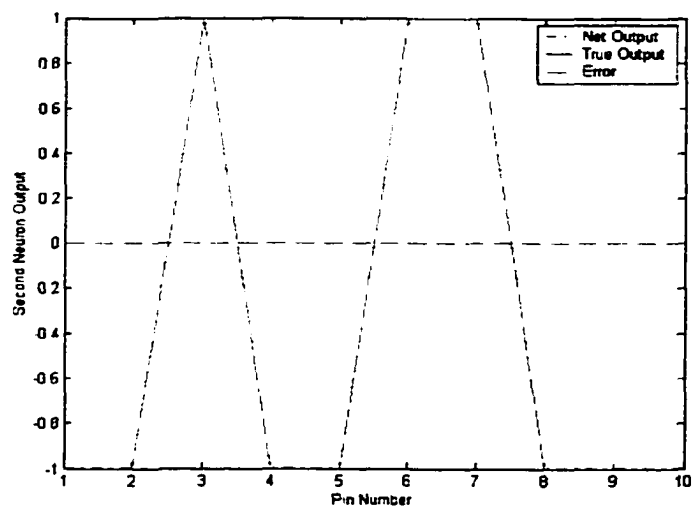


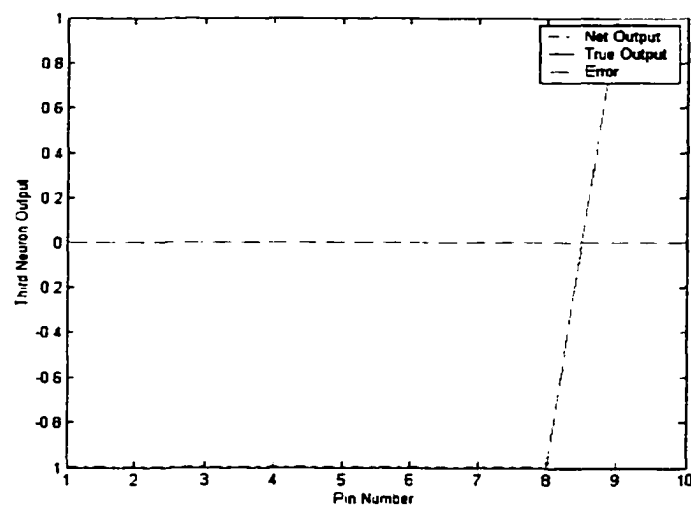
Figure B-1 (a) Real image with no defects (b) Simulated defects image



(a) Output Performance of First Neuron vs. Pin Number



(b) Output Performance of Second Neuron vs. Pin Number



(c) Output Performance of Third Neuron vs. Pin Number

Figure B-2 Classification performances for 480:3 neural network

VITA

Roberto A. Balderas obtained a Bachelor of Science degree in mechanical engineering from the University of Texas at Austin in May 1993. He received his second Bachelor of Science degree in manufacturing engineering from the University of Texas Pan-American in May 1995. He worked for several state institutions and private companies through out his career. He has taught 9th grade and high school students, algebra and computer aided drafting. He has also taught manual drafting at South Texas Community College (STCC). He worked for Baker Hughes oil and die casting industry as a manufacturing engineer, and as a production engineer for Allied Die Casting of Illinois. He has published a paper at the North American Manufacturing Research Conference held at Houghton, Michigan in 1995. His hobbies include developing computer games, motor sports, swimming, and marksmanship. He will by graduating with a Master of Science in Engineering - concentration in manufacturing from the Manufacturing Engineering Department at UT-Pan American in May 2002.

Permanent Address: 2833 North 28 ½ Street, McAllen, TX 78501.