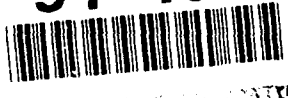
(2)

FINAL REPORT FOR NETROLOGIC'S PHASE I SBIR ON

ROBOTIC NON-DESTRUCTIVE INSPECTION OF AIRCRAFT

CONTRACT # N00014-91-C-0095

SUBMITTED: 5 SEP 1991



TO:

DR. JOEL DAVIS
OFFICE OF NAVAL RESEARCH
800 NORTH QUINCY STREET
ARLINGTON, VA 22217-5000

FROM:

NETROLOGIC, INC.
5080 SHOREHAM PLACE, SUITE 201
SAN DIEGO, CA 92122
(619) 587-0970 FAX: (619) 458-1624

Table of Contents

"Original contains color plates: All DTIC reproductions will be in black and white"

Statement A per Joel Davis
ONR/Code 1142
Arlington, VA 22217-5000

NWW 9/19/91

Abstract

NETROLOGIC, its subcontractor, the Southwest Research Institute, and its robotics consultant, Dr. Mahmoud Tarokh of San Diego State University investigated the application of robotics to the Non-destructive Inspection (NDI) of aircraft. It was found that it is feasible to transform the Navy's robotic deriveter to a neural network based NDI robot. A feedforward multilayer neural network was found to be very reliable at detecting cracks around rivets. In addition, an efficient new manipulator path planning method using neural networks was found to be useful for the robotic aircraft NDI.

Acknowledgement

The research team gratefully acknowledges the insights and information provided by Dr. Don Birx,of Systems Research Laboratories, Mr. Douglas Fromm, of McClelland AFB. and Mr. Don Hagemaier, of McDonnell Douglas.

Principal Investigator: Dan Greenwood

Project Support:

Dr. Ernest Franke, SWRI
Dr. Jay Fisher, SWRI
Mr. Dave Burkhardt, SWRI
Mr. Tom Baylor, NETROLOGIC
Mr. Spencer Menlove, NETROLOGIC
Mr. Richard Rea, NETROLOGIC
Dr. Mahmoud Tarokh, San Diego State University

1. Introduction

Non-Destructive Inspection (NDI) of aircraft is known to be a time consuming, boring, and an error prone task for human inspectors despite many advances in NDI sensor technology [Hagemaier], [Birx], [Johnson]. Clyde Kizer, of the Airline Transport Association, summarized the beliefs of many professionals involved in NDI that a method is needed for looking (inspecting) at an aircraft without reliability problems due to boredom while minimizing aircraft total inspection time during the life of the aircraft. In order to mitigate the above aircraft inspection problem NETROLOGIC set its primary Phase I research goal to be the specification of a mobile robotic NDI system which would be reliable, user friendly, adaptive and relatively inexpensive. An essential component of such a system would be a neural network based flaw detector. Our approach to specifying a robotic NDI system for Naval Aircraft maintenance was to exploit the previous Navy sponsored robotic deriveter developed by NETROLOGIC's subcontractor SWRI. This system would be enhanced by adding a completely automatic recognition system given that several networks could perform the recognition tasks without high reliability.

We decided to focus our investigation on eddy-current based sensors [Birx] during Phase I since this aircraft inspection method is universally accepted and the tediousness of the manual inspection method invites automation aids. However, in addition to eddy-current base inspection we gathered information on other inspection methods such as magnetic-optic sensors [Shih], Ultra-sonic sensors [Prabhu] and x-ray radiography [Berner]. In fact, neural networks were applied with great success by Prabhu and his colleagues at NASA Langley to aircraft lap-joint disbond ultra-sound based detection. In late 1990 Birx considered that state of the art in automatic NDI using technologies such as finite element analysis, inversion methods was still deficient in enabling reliable automatic inspection.

Birx reported in his PhD thesis (University of Dayton, 1990) significant progress in applying neural networks using complex (i.e. real and imaginary) neurons and weights. Eddy-current data is usually provided as two components, I and Q, which are In-Phase or Quadrature-Phase with respect to the excitation current. Should these numbers be treated using complex arithmetic, or just handled as two real components of the input to a generalized neural net? Philosophically, all inputs to a neural net are components of a single, multi-dimensional vector, even though there may be no simple geometric relationship between them as there is between I and Q. All components are treated as real, and the network weightings adjust themselves as required to make correct decisions about the data classification.

Birx provides simulation of a neural network which performs complex arithmetic on the I and Q components of EC data. He reports successful backpropagation training on fewer samples with complex arithmetic than were required for the same data treating both I and Q as real numbers. Savings in training cycles may not compensate for the

additional computer time needed for complex multiplication. One complex multiplication takes more than twice the computer time of two real multiplications. Birx showed that the real part of the error does not change the imaginary part of the weights, and vice versa. This suggests that there might be no advantage to doing complex arithmetic.

Time limitations during Phase I of the project have prevented NETROLOGIC from following up Birx's work in detail. There may be advantages to it which will become apparent after more study. We recommend to study this in more depth during Phase II. Our approach considers that total robotic inspection problem: automatic flaw detection and automatic measurement. Concerning robotic measurement, we investigated the modifications required to SWRI's deriveter and a new fast manipulator path planning technique [Tarokh, Appendix]. The latter methods will be valuable in applying robotic inspection to aircraft wings and tails and when scaffolding is in place around an aircraft.

2. Eddy Current and Robotic Considerations for Aircraft Inspection

2.1. Requirements for Automated EC Inspection

2.1.1 Background

Eddy current (EC) testing is based on the establishing electrical current flow in the part to be inspected and detecting defects based on associated changes in the current flow pattern. A probe using an excitation coil energized with alternating current flow is placed above the surface of the part, and the alternating magnetic field produced by the probe induces the flow of electrical eddy currents in the part. Sensing is performed either by monitoring the impedance of the coil, which is dependent on the eddy current flow characteristics, or by measuring the voltage induced in a separate, passive sensing coil placed near the excitation coil. Flaw detection using EC techniques generally requires that the probe be scanned over the surface of the part to be inspected. Certain characteristic changes in the eddy current response are then associated with the presence of a flaw.

Although EC testing is very sensitive to the presence of flaws, the response is also affected by other factors which can interfere with flaw detection. These factors include changes in the geometry of the part, changes in the electrical and magnetic characteristics of the part, and variations in liftoff (the spacing between the probe and the part).

Nondestructive inspection of aircraft skin for cracks (primarily those cracks originating from fastener holes) can involve the inspection of the first and/or second

layers of skin material. The most difficult is detection of cracks in the second layer, especially where the first layer is relatively thick or where there are variations in the thickness of the first or second layer. The first layer acts to significantly reduce the signal from the cracks and variations in thickness (e.g., from structural features such as ribs and spans) introduce artifacts in the data that must be distinguished from the signal features caused by cracks. For second layer inspection, the most likely required scan motion would be to center a probe over the fastener and rotate the probe so that the eddy current detector travels along the circumference of the rivet head. Automating this inspection process will require implementation of a procedure for accurately locating the center of the rivet head and then positioning a mechanism for circular scanning over the center. For both first and second layer inspection, the presence of a fastener can also produce undesired signals; if the probe moves from the skin to the fastener, a large signal is generated from the fastener/skin interface. Also, if the fastener is a different material from the skin (e.g. aluminum skin, steel fastener), an even larger response is obtained.

2.1.2 Scan Motion Considerations

The scanning requirements for EC inspection of aircraft skin are highly dependent on the layer to be inspected and the flaw size detection requirements. The greatest sensitivity to small flaws in both the first and second layers can be achieved by centering a probe with respect to the center of the fastener, and then rotating the probe around the fastener. This process greatly reduces the unwanted signal caused by the fastener/skin interface and by different fastener material since the distance from the probe to the fastener remains constant throughout the scan. The difficulty with this approach is that it can be time-consuming to precisely position the probe, and thus the inspection time for a large number of fasteners can be considerable.

An alternate approach is to scan the probe in a raster pattern over the fastener and adjacent area. This approach has the advantages of simplicity and significantly reduced inspection time because the precise positioning requirements are eliminated. The flaw detection sensitivity is reduced, but detection of both first and second layer flaws has been demonstrated with this approach. Figure 2.1 shows color images generated from raster scan data obtained from a wing section of an Air Force T-38 aircraft. The wing skin is aluminum alloy, the fastener is steel, the simulated cracks (0.375 and 0.2 inch long saw cuts) are radially oriented in the second layer, and the second layer has an edge oriented parallel to the row of fasteners. In both images, the presence of flaws are quite visible, even though the image also shows the presence of the second-layer edge.

The capability for detecting first layer flaws using a raster scan approach was demonstrated using a T-38 wing section similar to that described above. Figure 2.2 shows an image generated from a raster scan over a row of 8 fasteners. An indication

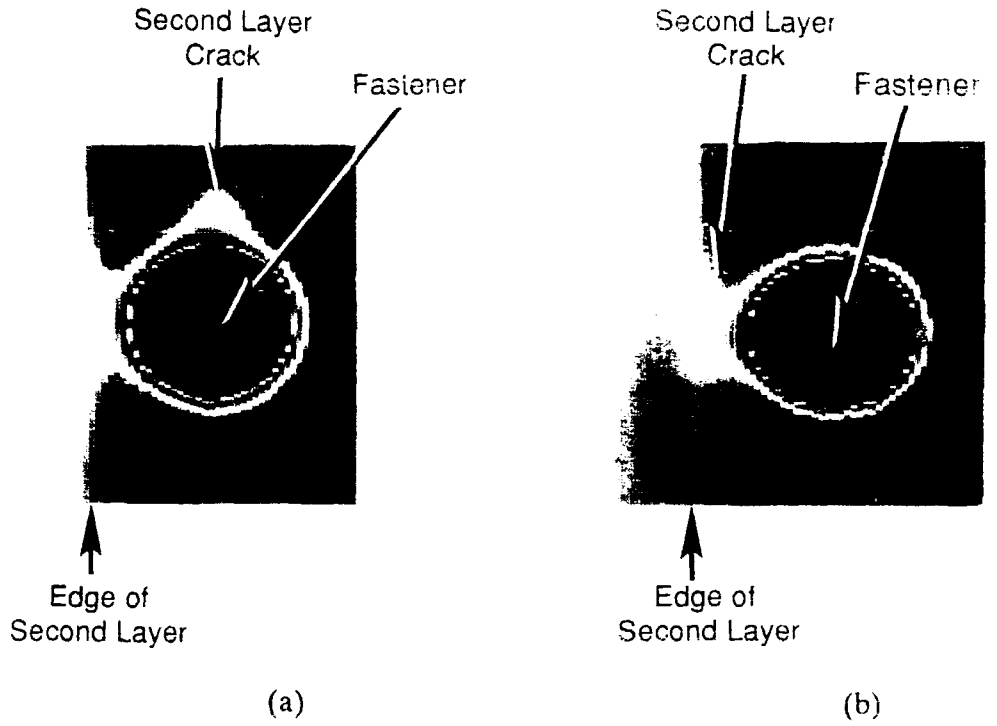


Figure 2.1 Eddy current images of second-layer flaws 9.53 mm (0.375 inch) long (a) and 5.08 mm (0.2 inch) long (b) in fastener holes. In both images, the fastener is clearly visible, and the second-layer edge shows up as a vertical band on the left side of the image. The two-dimensional pattern of the flaw responses shows up easily against the large response from the aircraft wing geometry.

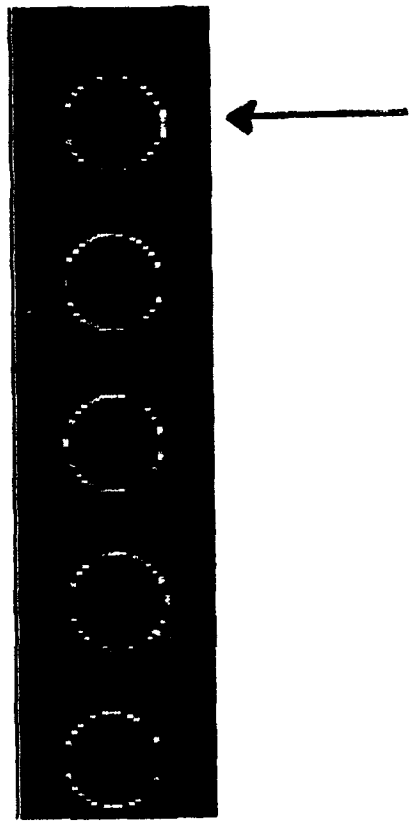


Figure 2.2 EDDY CURRENT IMAGE OF FIRST-LAYER FLAW (EXTENDING 0.05 INCH BEYOND FASTENER HEAD) IN FASTENER HOLE OF T-38 AIRCRAFT WING.

from a first layer simulated crack (saw cut) which extended 0.05 inch past the edge of the fastener head is visible in the image. These data were from a very limited investigation where little optimization of the EC probe and test parameters was possible. It should be possible to obtain improved results with additional investigation.

Based on the above results and the reduced requirements for a raster scan, it is recommended that a raster scan approach be considered as the primary choice for inspection of large areas of aircraft skin, especially if only a first layer inspection is to be performed.

2.1.3 Scanning Requirements

The limited first layer experiment described in the previous section was used to obtain preliminary data to determine raster scan requirements for inspection of aircraft skin. Based on this experiment, it was shown that flaws of a reasonable size could be detected using a raster scan having the following characteristics:

Scan track spacing	0.04 inch
Spacing between sampled data points	0.02 inch
Probe liftoff from skin surface	0.003 inch
Scan speed	0.5 inch/second

Any system for automated eddy current inspection of aircraft panels must provide a means for positioning and moving the EC probe over the area to be scanned. When the emphasis is the detection of cracks emanating from rivet holes in the skin, the system should concentrate scanning and analysis on the area immediately surrounding rivets and bypass other the remaining surface material. In order to do this it will be necessary to know the locations of the rivets relative to the coordinate system of the mechanism moving the eddy current probe.

Based on the prior experience of NETROLOGIC's subcontractor, Southwest Research Institute, (the Robotic Deriveter System), we do not believe that it is feasible to pre-program the locations of rivets for an aircraft part and expect this information to apply to all (or even most) copies of that part. Many aircraft (particularly those that are nearing the limits of their designed life) were assembled manually and rivet spacings and placement along spars and stringers were put in "by eye". Even where the original positioning on a part was precise, manufacturing tolerances for joining subassemblies and stresses induced during thousands of hours of flight have caused enough distortion in the airframe that preprogrammed fastener location data will not match the existing patterns.

SWRI's Robotic Deriveter System was developed for the Navy and uses a machine vision system (IRI P256) with a camera and illumination source mounted on the robot end effector to determine the locations of rivets and other fasteners. This technique used a small field-of-view (approximately 1" X 1") to provide accurate (.005") rivet

position measurement for drilling. In order to accomplish this the paint had to be removed from the aircraft surface and lighting conditions had to be carefully controlled. The aircraft surfaces are often marked by paint flecks, scratches, dents, and other features that degrade image quality. Because of this, the computer vision algorithms used were relatively slow, requiring several seconds to locate and identify each fastener. Computers for image processing operations have improved greatly since the time the deriveter system was designed and the location problem could be performed much more rapidly using currently available hardware.

For automated inspection, high accuracy is not required so long as the scan pattern of the probe covers the rivet locations. It is more important that the guidance system quickly determine the approximate locations of all rivets over a large area. Ideally this should be done without requiring the removal of paint or other surface coatings such as decals or stencils. In order to investigate this approach. several tests were performed using machine vision techniques to locate rivets in painted aircraft panels. In most cases the pattern of the rivet head can be seen through the layers of paint as a characteristic circular indention in the surface. When the rivet was installed flush to the surface with no discernable gap, it was very difficult to locate rivets covered by several coats of paint. Best results were obtained when the light source and the camera were both at moderate angles (30° to 45°) from the surface so that the contrast of surface features was emphasized.

Small variations in surface flatness can be emphasized by an optical technique employing retro-reflective illumination. This technique, developed and patented by Diffracto Ltd. under the name "D-sight", utilizes a camera and closely located light source directed toward a reflective surface at a shallow angle. Light from the source is reflected from the surface under examination to a sheet of retro-reflective material where is directed back along toward the surface and then to the camera. Best results are obtained with large, flat, polished, reflective surfaces. In some cases a film of oil or other material can be applied to the surface to increase the reflectance. When this technique is used on highly reflective surfaces, surface patterns on the order of 0.001 inch can be clearly seen.

The D-sight technique was discussed with Omer Hagemaier's of Diffracto and its use in locating rivets under several layers of paint was investigated in brief laboratory tests. A machine vision system was used to observe samples of riveted aircraft material with several layers of paint. Best results were obtained with the light source and camera oriented at approximately 45 degrees to the surface with a sheet of retro-reflective material on the placed opposite the area being imaged. The position and orientation of the retro-reflective sheet was not critical and wide variations in position and orientation did not cause any discernable changes in the images. The reflectance of the surface being examined is critical; it was necessary to spread a light coat of oil over the painted wing material in order for the technique to be effective. When this was done, rivets that were very difficult to see under normally lighting were readily visible using the

D-sight technique. Figure 2.3 shows the machine vision image of an area of a painted wing where the rivets on the right side of the figure are barely discernable. Figure 2.4 shows the same area under D-sight illumination clearly revealing the rivets and also showing evidence of slight surface deformations and skin buckling due to rivet installation.

2.2 The U.S. Navy Robotic Deriveting System

The Robotic Deriveter System (RDS) is a one-of-a-kind mobile robotic system for automated removal of rivets and eddy current inspection of the rivet holes. It was built between 1983 and 1986 by Southwest Research Institute for the U.S. Navy. The total cost of the system was $2,597,000. Photographs of the RDS in different tasks are included in the illustrations at the end of this report.

The RDS was delivered to North Island Naval Air Rework Facility (now Naval Air Depot) in the spring of 1986 and final acceptance tests were performed. The initial requirement for the RDS and much of the system specification was based on a program for modifications to F-4 aircraft but by the time the RDS was delivered, this program had been completed. Attempts were made to apply the RDS to several other workloads including engine cowlings and F-14 engine nacelles but the anticipated workload in these areas did not justify the retooling that would have been required to access these parts. For the next two years, the RDS was used only occasionally at North Island NAD. In 1987, Ogden Air Logistics Center at Hill AFB in Ogden, Utah, requested the use of the RDS for training technicians who would operate an automated deriveting system being constructed for the Air Force. This loan was approved and the system was shipped to OO-ALC and used briefly in 1988.

For the past three years the Robotic Deriveter System has remained at Hill AFB but is not presently being used for any productive work. The loan agreement has expired and OO-ALC is ready to return the RDS to the Navy but there is no available workload at NI-NAD. As a consequence, the mobile robotic system is not being utilized at this time. With proper justification, the system could be utilized for other automation or inspection projects in the U.S. Navy.

2.2.1 RDS Configuration

The Robotic Deriveter System was designed to be self-contained except for the requirement of 480 VAC, three-phase electrical power. In order to achieve this self-sufficiency, it includes a variety of subsystems.

The vehicle was designed specifically to allow transport of the robot, to hold the robot in a stable working position with a work envelope amenable to aircraft parts, and to provide a base for the auxiliary systems such as the robot controller, computer,

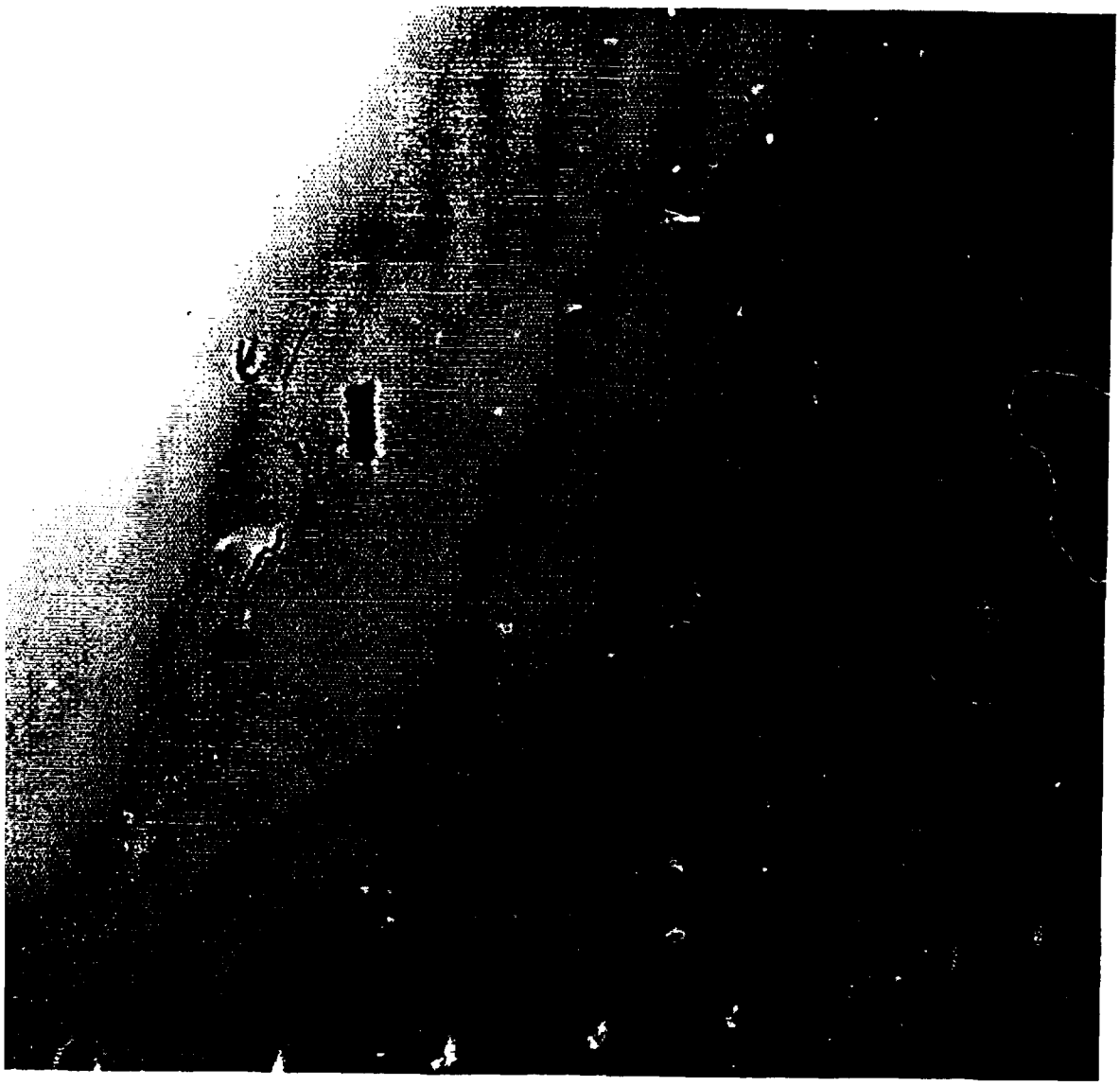


Figure 2.3 MACHINE VISION IMAGE OF PAINTED OVER RIVETS

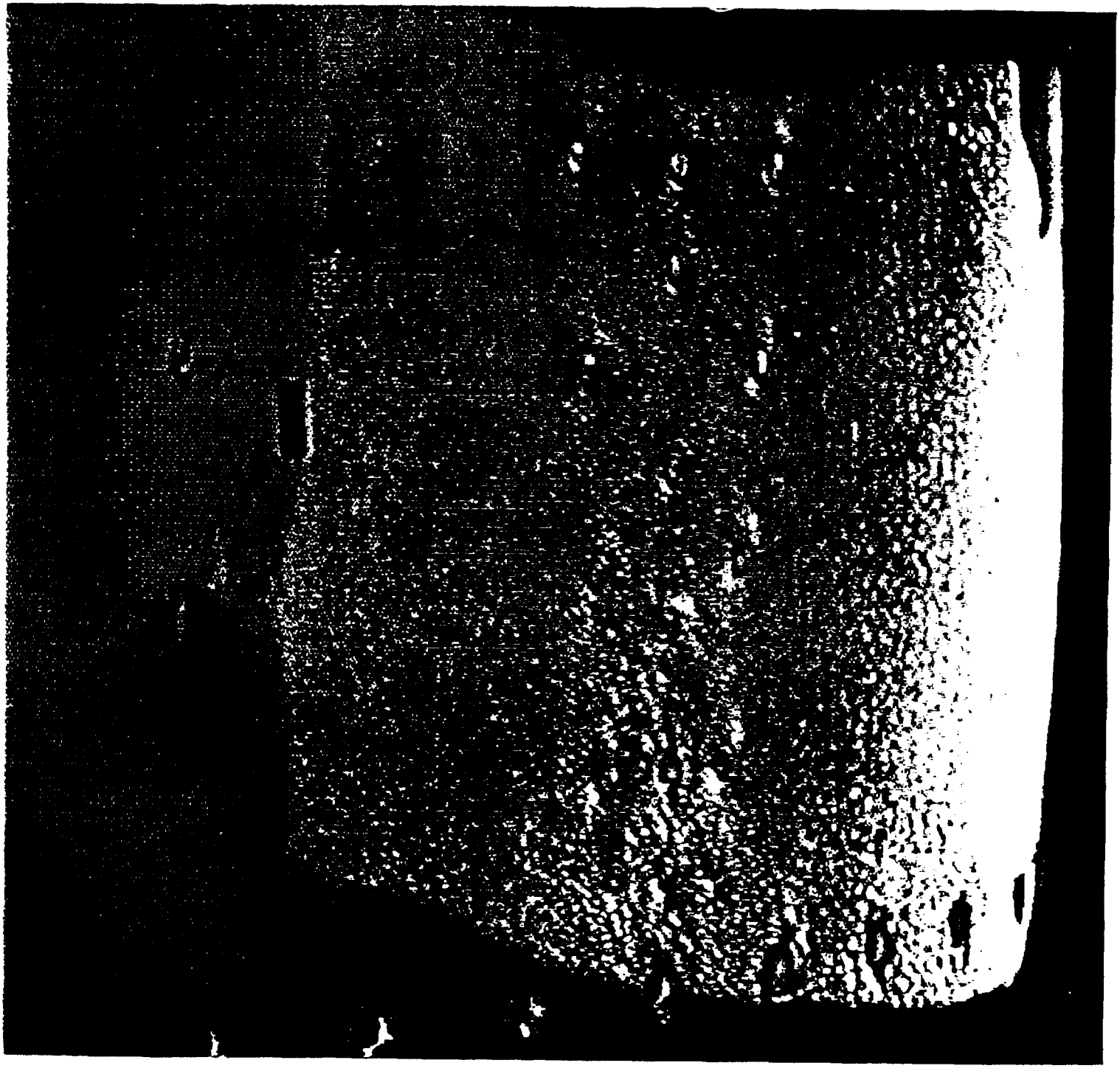


Figure 2.4 D-SIGHT IMAGE OF AIRCRAFT PANEL SHOWING RIVETS UNDER PAINT

machine vision system, and the eddy current subsystem. The vehicle uses rear-wheel (forklift style) steering to allow access into confined spaces and an air bag suspension system that allows the frame of the vehicle to be lowered to the shop floor to ensure stability during operation.

The robot is a Milacron T3-776 Robot with electric drives. It has 6 degrees of freedom and can carry a 150 lb payload. The stated repeatability of the robot is .02 inches but greater precision is possible if consistent paths and loads are used.

Tooling for rivet removal and inspection of rivet holes is included in an end-effector mounted to the robot wrist plate. The end-effector includes a turret which can be translated vertically or rotated by two stepper motors. The turret includes tools mounted so that appropriate rotation and translation can position tools including:

- a machine vision camera for locating and identifying fasteners.
- a drill for removing rivet heads
- a punch to drive the rivet shanks from the holes
- an eddy current probe to scan the sides of the rivet holes for cracks.

The only one of these tools that is of interest for the planned inspection project is the eddy current probe. This probe is a small (approx. 2 mm diameter) cylindrical device that can be inserted into rivet holes. Two stepping motors are used to rotate the probe about center and to drive the probe along a radius to vary the diameter of the scan pattern. In the proposed inspection application, a different probe and scan pattern will be used. The current EC scan technique is of interest primarily as an example of the ability of the robot to position the end-effector accurately and repeatability enough to allow use of eddy current scan techniques.

The robotic deriveter system also includes a Milicron Version 4 robot controller, a general purpose minicomputer, machine vision computer and an eddy current instrument. Supporting subsystems including an air compressor, hydraulic unit, and vacuum are mounted on the vehicle.

The system software is written in Fortran 77 and runs on a Hewlett-Packard A-700 minicomputer. A Winchester disk is used to store programs and rivet location databases for different aircraft parts. The HP computer system also includes an assortment of analog and digital I/O lines to monitor the status of the system and the process and to control the system.

Programs for teaching the robotic system the locations of rivets on parts to be deriveted are included in the RDS. Since the system is mobile and the robot coordinate reference position is not fixed, a touch-off method is used to define the location of a work-piece relative to the robot coordinate frame. Coordinates of the touch-off points used to establish the reference frame are stored with the database of rivet coordinates.

Rivet locations are then taught by using the teach pendent to move the robot end effector to the desired locations and storing the coordinates as determined from the robot controller. These locations are then refined by using the machine vision system to locate the precise center of each rivet on the part.

In the automatic derivet mode, the operator provides the name of the rivet location database to be used and manually (using the teach pendent) moves the robot to the part reference points. This provides coordinate data for transforming the stored rivet locations to robot coordinates for the actual part location. The HP computer then proceeds to retrieve the location of each rivet, move the robot to that location, correct for perpendicularity and offset, refine the rivet position and remove the rivet using by drilling and punching.If desired, the eddy current probe can be inserted into the hole to check the interior material for cracks.

The RDS used a contact method for determining offset and orientation of the work-piece. This was a consequence of the need to overcome robot deflections caused by the large forces encountered during drilling. In order to provide the stability needed for drilling, the RDS end effector included a ring that was preloaded against the aircraft surface prior to drilling. Sensors were included to measure the offset and orientation of the end effector relative to the aircraft surface from the position of the preload ring.

2.3 Adapting the Robotic Deriveting System for Aircraft Inspection

Based on the requirements for eddy current inspection described above, the Robotic Deriveter System was analyzed to determine how well each of the existing subassemblies met the needs of the automated EC inspection system. The subassemblies used for drilling and punching operations will not be needed. These include the hydraulic and vacuum pumps and reservoirs. Many of the modules of the end effector including the drill, punches, marker, and J-bolt installation tooling can be removed from the system thus lightening the end effector and improving both the speed and positioning accuracy of the robot.

The requirements for scan track spacing and probe liftoff are beyond the capability of large anthropometric robots designed in the early 1980's such as the Milacron 776 when used in a "free-hand" mode. In order to provide the accuracy needed for reliable crack detection, a mechanism providing either circular or raster motions and having sufficient compliance to follow surface contours is needed. Such a mechanism could replace one of the end effector modules that will be removed but a better alternative is to replace the entire existing end effector with a simpler, smaller, lighter end effector for manipulating the EC probe.

The simplest configuration for a scanning module will be to use roller contact with the surface to maintain both standoff and orientation. Three roller balls or three wheels

that can swivel to allow scanning in all directions can be used for surfaces with moderate curvature such as wings and some parts of the fuselage. Compliance could be provided by a flexible coupling and loading against the surface by either mechanical springs or air cylinders. This would be an inexpensive and reliable approach, well suited for initial investigations and tests but it would not be practical for scanning in confined spaces such as the wing root areas at the edges of engine nacelles.

For complete inspection of an aircraft, including these critical areas, a non-contact alignment system would be required. This can be accomplished by use of sensors to measure stand-off and an active control mechanism to adjust the position of the EC probe. Several different types of sensors could be used but optical triangulation sensors such as those manufactured by Keyence will be most suitable for the ranges and accuracy needed. Three sensors will be needed in order to measure both the standoff and orientation of the surface. The three sensors will need to be directed toward a small area so that the system can be used for inspection of rivets along the leading and trailing edges of wings. Older robot controllers such as the Milacron Version 4 do not provide the capability of adjusting the path dynamically in response to sensor inputs. In order to maintain probe standoff and orientation, the inspection module will include three independent axis of motion (z, pitch, and yaw) controlled in response to the three sensor signals. A dedicated microprocessor will calculate the axis motions from measurements made by the three position sensors. The required dynamic response of this control system will be relatively modest since robot speeds during EC scanning will be low (2 ips).

One obvious implication of the high resolution and accuracy required for scanning is that scanning the entire surface of an airplane will simply take too long to be practical. Any useful system must include a method for determining the location of rivets and other fasteners so that the EC probe can be positioned to scan the area immediately surrounding the fastener hole. The Robotic Deriveter System used a machine vision camera to obtain an image of the fastener and locate the center accurately (.005") enough for drilling but this approach required stripping the paint from the aircraft. For a practical inspection system it will be necessary to determine the approximate position of the rivets without removing the aircraft paint. A completely automated system could make use of the "D-sight" technique as shown by tests in SwRI laboratories but there are two major disadvantages to this approach:

1. D-sight requires shallow camera angles (approximately 45°) and placement of a large sheet of retro-reflective material opposite the camera. In order make all motions automatically, a very large end effector will be needed which will limit access to confined spaces.

2. The requirement for a highly reflective surface will probably mean that a light oil coat will have to be applied to all painted surfaces. If this is to be done automatically, additional complexity must be added to the end effector.

An alternative approach would to rely on the human operators to mark rivet rows using an infrared reflective paint or dye. One convenient technique will be to use a chalk line with IR reflective powdered material. Two operators could quickly stretch the line over rows of rivets and mark them so they would be clearly visible to a video camera with an IR filter. The end effector would be equipped with a camera and IR rich illumination system that could image the surface from a distance of several feet. A machine vision computer would calculate the coordinates of each of the marked lines and transmit this information to the robot controller for path scanning. Such a semi-automated procedure will be more cost effective during the early stages of system development than a totally automated system.

In the existing system, the rivet locations are taught by manually moving the robot end-effector using the teach pendent. These locations are then stored and later refined by the vision system to locate the center of each rivet accurately. This manual positioning typically takes hundreds of hours during system development and verification. The use of the vision system with marked lines described above coupled with a robot path planner can drastically reduce or eliminate the need for manual positioning. The path planner also ensures collision avoidance and prevents damage to the eddy current probe and to the airframe. The use of a robot path planner is particularly important for confined spaces such as wing root areas at the edges of engine nacelles.

The use of a fast robot path planner, development by NETROLOGIC, was investigated for robotic aircraft inspection. A PUMA 562 robot was employed for preliminary investigations. PUMA 562 is kinematically similar to Milacron T3-776 currently used in the Robotic Deriveter System. PUMA 562 is an articulate 6 degrees of freedom manipulator with 3 major joints at waist, shoulder and elbow and 3 minor joints for the wrist.

An efficient and fast path method was developed by decomposing the 3-dimensional space into a number of rectangloid slices. A method was devised to work directly with arm configurations instead of individual joint angles. These two aspects, namely decomposition and configuration control, greatly speeds up the path finding procedure and makes it possible to perform real-time planning in moderately cluttered environment that is encountered on the airframe. The path planner uses bitmap of the arm and obstacles which are read from a file. The file can be created using either prestored airframe shape or image obtained on-line from a vision system. A neural network was utilized to determine the optimal arm configuration for preventing unnecessary arm movements during path execution. The path planning algorithm consists of four basic steps: 1) map the workspace, 2) input the target location as determined by a vision system, 3) rotate the waist axis while using a neural network to change arm configuration to avoid obstacles, 4) when arm is in the plane closest to the target, use an inverse kinematic solution to move the end-effector to the exact target location. Several laboratory experiments were carried out using the PUMA robot to determine the effectiveness of the path planner. The obstacles consisted of boxes placed

on the floor and also several structures hanging from the ceiling. In all cases the path planner determined and executed a collision free path very fast. The path planner can easily be adapted to Cincinnati Milacron T3-776, GMF S-500 or similar robots.

2.4 Summary of Requirements

If EC inspection is done by raster scanning, the speed requirements of the robot positioner are not critical. Rows of rivets will be scanned at low speed, limited by the capabilities of the EC analysis equipment. Path planning can be used to move from one row of rivets to the next without having to cross large distances of unriveted surface. Only a small percentage of the time will be spent moving the robot from one row of rivets to another so the speed during this time will have little effect on the total task time.

An important requirement for the system is the development of a vision-based robot path planner. This would relieve the tedious and time consuming task of manual path teaching and positioning. This would require integration of a wide-angle camera vision system and a path planner. Our initial investigations and experimentation have indicated that it is possible to achieve a relatively fast and efficient path planning for robotic aircraft inspection.

Sensing requirements will include measurement of standoff distance with an accuracy of .005" and orientation with an accuracy of 2 °. Both standoff and orientation sensors need to be as compact as possible in order to allow operation in confined areas or on surfaces with significant curvature. Sensors for collision avoidance will also be required to prevent damage to the EC probe and to the airframe itself. This should be capable of detecting obstacles extending more than 0.25" from the surface for a distance of two inches from the edge of the scanner module. At a speed of two inches per second, this will provide one second in which to stop the robot if an obstacle is detected. Sensors for fastener location (for circular scanning) and path identification (for raster scanning) will also be needed. A video camera mounted on the end effector and a machine vision computer capable of 256 levels (monochrome) with resolution of at least 256 X 256 pixels will be needed.

2.5 RDS Modifications for Inspection

The requirements for an automated eddy current inspection system differ, to some extent, from the current configuration of the Robotic Deriveter System. Initial tests of the neural network crack detection technique could be done with only minor changes to the RDS in order to prove the feasibility of the concept. More extensive modifications could be done later in order to gain greater speed and automatic operation.

Use of the vehicle as a mount for the robot is a valuable asset for investigations, tests, and system development. Many of the questions concerning automation of EC inspection of complete aircraft can only be addressed by scanning of large parts and complete airframes. The existence of a mobile system will allow testing different methods and procedures with very little investment in hardware systems. The Cincinnati-Milicron T^3-776 robot has sufficient reach to cover typical aircraft parts without moving the vehicle and its payload capacity is more than adequate for the EC sensor and vision camera.

Of the auxiliary systems, the hydraulic system is used for power steering and should be retained on the vehicle. The compressed air and the vacuum systems will not be needed if a connection to shop air can be used to raise the vehicle for travel. The compressor and vacuum blower can be removed if desired or they can be left in place since they will not interfere with operation of the other components. The tool changer, consisting of a rotating turret just in front of the drivers station, can also be removed if desired.

The turret used to position the different modules required for rivet removal will not be used for EC inspection. It can be removed and stored for future use. In place of the turret, a single module containing an array of eddy current sensors will be used. This module will be include rollers to maintain reference standoff and orientation from the surface and will have a compliant coupling to the robot so that it can conform to variations in the position and orientation of the workpiece. The machine vision camera will be removed from the separate module of turret and attached to the new eddy current sensor module.

The robot controller will be reprogrammed for controlling the inspection system. The machine vision computer and Hewlett-Packard minicomputer currently installed in the Robotic Deriveter System use early 1980's technology and the software is written in Fortran 77 for rivet removal. These will be replaced by a faster and more powerful state-of-the-art computer and software. Sections of code for subsystem diagnostics and maintenance as well as the operator interface will be retained, but will be upgraded to the C code. Robot motion control program will be upgraded and a path planning program will be integrated. The image processing algorithms will be modified to speed up their operation by reducing the resolution and accuracy of the location determined. In addition an image processing algorithm will be added to identify and locate straight lines as might be drawn on the surface to indicate rows of rivets.

In addition to the modifications to the deriveter system it will be necessary to add some new capabilities. The most important will be the eddy current signal analysis electronics needed for each of the eddy current detectors. If circular scans with a single EC element are to be investigated it should be possible to use the Nortec instrument already present on the RDS vehicle. If a large probe using 10 to 20 EC sensors is used to simulate raster scanning, it will be necessary to obtain additional EC instruments or

develop a method for multiplexing the signals while traversing the surface at very low speeds.

2.6 Recommendations

The U.S. Navy Robotic Deriveting System, although not optimal for the NDE task, will provide the major components for a flexible and cost-effective test-bed for investigation and development. The greatest advantage can be obtained by use of a multistage development and test program. The first phase of testing would consist of design and fabrication of an EC probe optimized for detecting cracks under rivet heads. This would be either a module for circular scanning or a linear array of closely spaced sensors for one-pass linear inspection. The probe would initially be tested on test panels containing artificial cracks under rivet heads using laboratory instrumentation. After the first phase of laboratory testing is completed, the second phase would consist of scanning real aircraft for cracks. This would be done using the Deriveter System, modified as described above. The RDS would provide an economical framework for testing and verifying the operator interface, rivet location algorithms, robot speed control, as well as the fundamental issues of neural network analysis for crack identification.

Finally, after the procedures and algorithms are developed and thoroughly tested, it will be desirable to make additional modifications to the Deriveter System in order to improve its operational effectiveness. Items to be considered include upgrading to a faster, more powerful computer for robot control and image processing, programming more efficient methods for locating rivets to be inspected, adding a robot path planning algorithm, and upgrading to a newer robot. The Milicron 776 was designed for heavy loads as required in the deriveting scenario. The EC inspection task requires supporting only a probe module and would benefit from greater robot reach. It may be desirable to change the heavy duty 776 robot for a lighter robot with greater reach. One candidate is a GMFanuc S-500 robot designed for sealant application. This has a larger work envelope but only 20% the payload capacity of the 776. An additional advantage of the S-500 robot is that it lighter than the 776 and can be ceiling of wall mounted. This would allow placing the robot on a scissors-lift so that it can be raised to inspect parts of aircraft that might otherwise be inaccessible.

Data sheets showing the specifications of both the Milicron 776 and the GMFanuc S-500 are included in the illustrations at the end of this report.

3. Neural Network Based Flaw Detection

The network paradigm used in this study for automatic rivet inspection is a three layer feed forward network trained with back propagation of error. A network input layer that takes the incoming eddy-current representations as its activations, these activations are multiplied by a "weight" and sent through a squashing function then sent

to each hidden unit where it is summed up with the results of all the other input units. This sum determines the activation for the hidden unit and the whole process is repeated sending activations of the hidden units to the output units. The network is fully connected between successive layers, but has no connections which skip layers or are connected to the origin layer.

3.1 Data Representation Method

Present nondestructive metal failure analysis using eddy current signals is performed with an operator visually analyzing plots of the quadrature value of the sensor against the in phase value. Operators look at graphs of the data as both a two dimensional plot using both quadrature and in phase values, and as a one dimensional plot where the values are plotted against rotational position and analyzed separately (See Figure 3.1). Our first attempt at neural network fault classification used features derived from the two dimensional graph. The graph was treated like an image and features such as the percentage of black pixels is eight horizontal rectangles which sub-divide the image, were extracted from the graph. The resulting set of sixty two features was then fed as input to a feed forward network. The output was trained to 0.9 if the graph were from a rivet that contained a flaw and, the output teacher pattern was set to 0.1 if there were no flaw in the rivet being scanned. These features were then trained on both alone, and combined with the features from the graphs of the values scanned from the same rivet at different frequencies (for a total of 186 inputs). The networks trained on this data set were able to learn the training data very well but were unable to correctly classify unseen data above chance levels, classic symptoms of overtraining. Nets can be overtrained if the input set allows the net to key on idiosyncratic features of the input pattern rather than the features which characterize the domain the training data is intended to represent. This is a problem when the number of inputs is comparatively large compared to the total number of training patterns. In order to overcome this problem, a new training data set was constructed which keyed on the one dimensional plots in which each rotational location is treated separately. This had the effect of simultaneously greatly reducing the number of inputs while increasing the number of training patterns available.

3.2 Raw Data Set

The final neural network approach for rivet flaw recognition was performed on data from eddy current measurements taken from a probe at the low frequency of 1000 Hz, and at higher frequencies of 2000 and 3000 Hz. The experimental data were obtained by rotating the cross-axis probe 720 degrees around the target taking in phase and quadrature measurements at approximately every 0.7 degrees (See Figure 3.2). The targets were simple geometry specimens which have the same layer thickness as wing panels. Data was extracted to determine the effect of the second layer edge and the

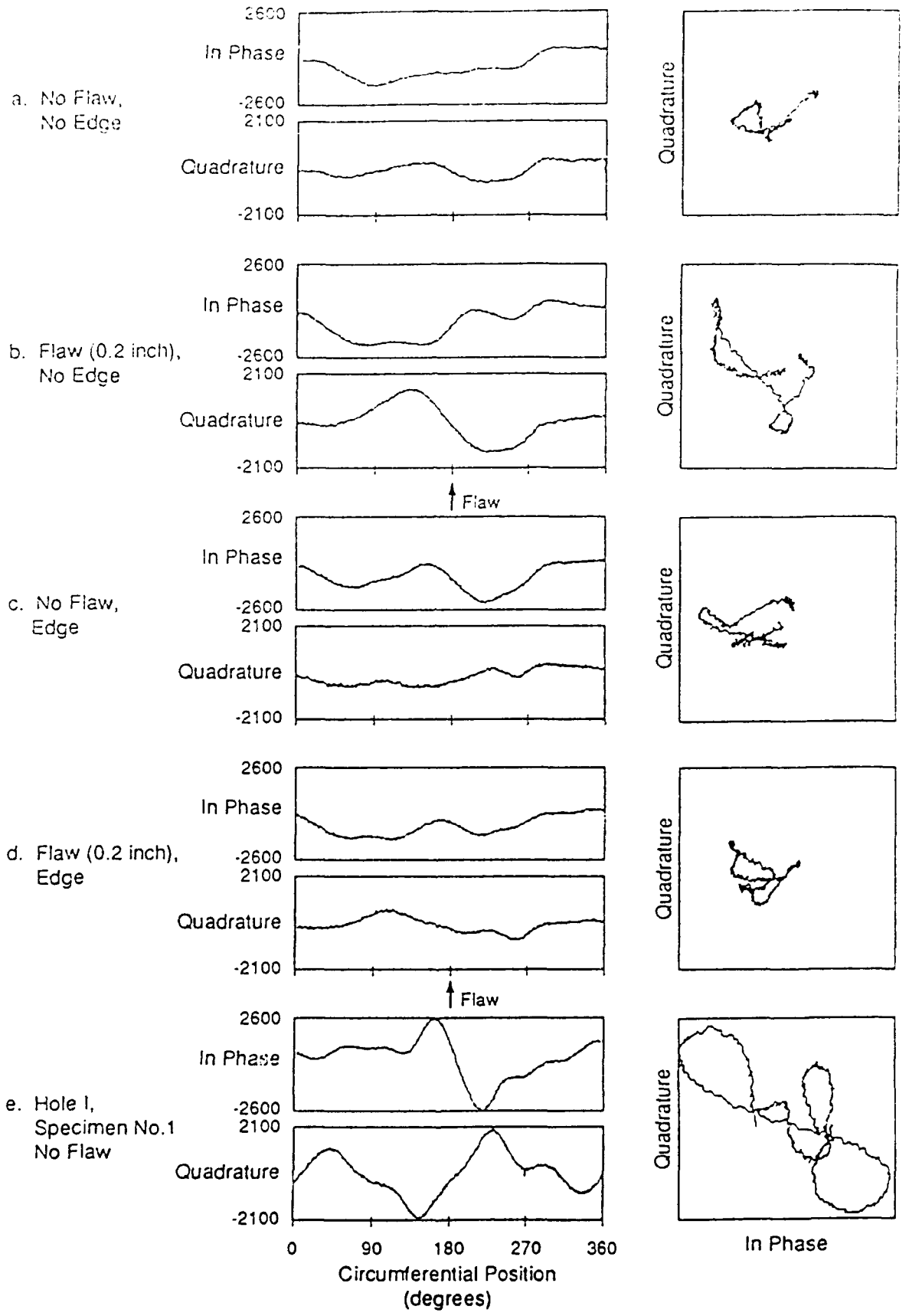


Figure 3.1 FLAW AND GEOMETRY SIGNALS FROM A SIMPLE GEOMETRY SPECIMEN (a-d) AND WING SPECIMEN 1 USING PROBE C. SIGNAL LEVELS ARE IN μ VOLTS.

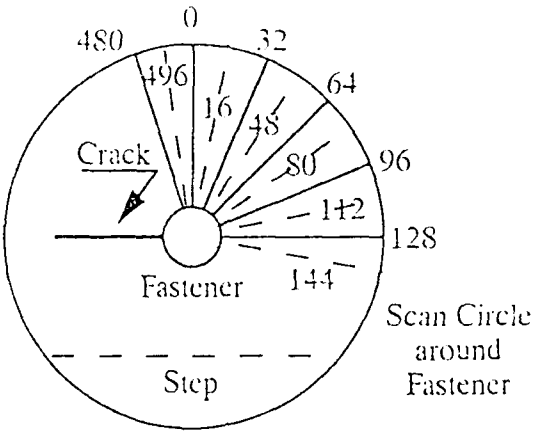


Figure 1. Scan Pattern Around Rivit

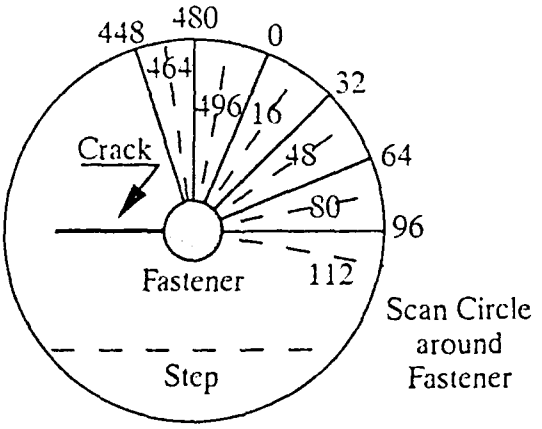


Figure 2. Offset data Reused for Training

Figure 3.2

change in thickness of the first layer across the scanning range. For application in the network data set, the in phase and quadrature measurements for all three frequencies were combined at each given angle measurement to form a single input set. The data was combined by taking the measurements at the different excitation frequency levels but at the same rotational position and putting them side by side in the same input pattern. The order of inputs is not important in neural network applications as each input is processed simultaneously. The frequency levels were combined in order to present the network with all three levels in each input. The purpose of multiple excitation frequencies is to obtain signals which differentiate flaw and geometry feedback: since, low frequency data contain both flaw and geometry information while high frequency data is predominantly geometry information. The reason the frequencies carry different information is that the higher frequencies do not penetrate as well to the second layer while lower frequency probes do penetrate well and return a signal more dependent on the internal layer. By presenting the information in tandem, the network has the information it needs to separate the flaw component from its geometric counterpart.

3.3 Network Data Set

Each case has slightly over two full rotations around the rivet being measured yielding approximately 1024 raw data points, each with an in phase and quadrature component. Six measurements correspond to each given rotational position, the two values at each of the three frequencies. These measurements occasionally had large quick value fluctuations (probably due to sensor anomalies). To counteract this, the raw values were smoothed with a low pass averaging function which spanned four degrees. Much of the information on the scanned rivet is carried in the change of sensor readings, so finding the change or first derivative of sensor values as a function of position is important in determining the presence of an irregularity. Instead of attempting to fit the signal to a function to enable derivation, the signal's rate of change or slope was calculated at each rotational position. A long range change value and a short range change value were added in with the smoothed raw value for a total of eighteen inputs to the network. Each of these inputs was normalized with its standard deviation to cover the range of approximately -1.5 to 1.5. The span of the slope features varied from 5 to 30 degrees. Optimal performance was found using a short range slope feature of 14 degrees and a long range slope feature of 28 degrees. The presence of both features was found necessary to prevent misfires while maintaining correct fires.

3.4 Network Configuration

The networks tested had from six to eighteen inputs, from five to ten hidden units, and one output unit, all were three layer feed forward networks trained with back propagation. Best generalization was obtained with eighteen inputs and as little as six

hidden units. The minimum number of hidden units was chosen to reduce training time and force a more highly distributed problem solution among fewer hidden nodes. The network output patterns consist of only one output. This output is trained to 0.9 to indicate a fault sample or to 0.1 to indicate a nominal sample. An input set was considered to represent a fault if it was sufficiently close to the fault location on the bolt. A fault region must be determined because back propagation learning requires a teacher pattern; so the network needs to be presented with patterns indicating are no fault, and patterns that represent a fault. Optimal generalization occurs when the training data corresponds to the natural transitions between decision regions.

In order for a network or any other method to differentiate between patterns, there must be some observable difference in the patterns. The dividing line for fault versus no fault training patterns should correspond to the actual change in input patterns. In order for a network to best identify the transition between fault and no fault data, points which characterize the central tendencies of a class of inputs must be presented along with points along the border of the decision regions. It is reasonable to assume that the location of the decision border varies with flaw size and orientation, but test runs indicate that the best range for declaring fault is those points within 20 degrees of the actual fault location. This resulted in the best network generalization. Generalization was determined by training the network on all of the data except for two cases which were excluded, then the network learning is turned off and the network is presented with the unseen pattern. The network's concentrations of fault outputs is then compared against the actual location of the rivet fault, if there is a fault.

3.5 Results

The feed forward network classified each sample on the scanner rotation as either fault or no fault, even rivets with flaws yielded no flaw samples because the data points beyond twenty degrees from the flaw location were trained as no fault. The network was trained on all available data, there were 18 cases with all three frequencies, and two cases that were held out to test the generalization of the network. The network was always able to learn the training data to within a maximum of 0.0113 RMS error. Different holdout cases were then tested to determine generalization to unseen data, there was always one holdout without a flaw and one holdout with a flaw to test any given set of weights. Within the 20 degree fault range around a flaw, the network always correctly classify at least some flaws. Each angle was presented separately, so for any given case there are many inputs and classifications. If each of these sets is viewed as an independent test, network performance varied across the holdout cases from 86% correct to 94% correct. This is the number of misfires plus the number of underfires divided by the total number of samples; a misfire occurs when the network classifies a fault on a sample that should be no fault, and an underfire occurs when the network classifies a fault input as no fault. A method for determining network performance that is more valid than viewing each input output grouping separately, is to find whole regions where

the net indicated fault presence, and then determine whether or not a fault actually exists there. For cases with flaws at least 0.187in deep, the number of correct fault classifications in a row was greater than the maximum number of incorrect firings in either the case of the faulty bolt away from the fault, or in the rivet that contained no flaw. The minimum number of contiguous correct firings was 37 while the maximum number of contiguous incorrect firings was 21. Even the holdout case with the smallest fault, 0.063in, fired 27 correct faults in a row although at one point it misfired 25 times in a row.

3.6 Recommendations

A simple three layered feed forward net was found to be effective in fault detection and performance should improve with fine tuning. This would include trying dynamic fault declare ranges to correspond with fault size, and different network outputs for different fault locations. The reason for the separate outputs is that the wing geometry may give faults at different rotational locations their own signature. Both of the alterations would be done in order to better arrange the training set to make fault classifications more natural.

References

[Birx, D.] The Design of a Neural Network that Performs a Complex Mapping for Phase-Sensitive Detection and Characterization of Eddy Current Impedance, PhD Dissertation, University of Dayton, December 1990

[Hagemaier, D.] NDT Development in the Aircraft Industry; McDonnell Douglas Paper MDC 91K0003, Huntington Beach, CA and Westec 1991 paper.

[Shih, W.] Magneto-Optic / Eddy Current Imager, PRI Instrumentation Company Literature, Torrance, CA.

[Berner, P.] "NDT's Search for Tomorrow", Aviation Equipment Maintenance, September, 1991.

APPENDIX

FAST PATH PLANNING FOR ROBOT MANIPULATORS*

M. Tarokh
Robotics and Intelligent Systems Laboratory
Department of Mathematical Sciences
San Diego State University
San Diego, CA 92182-0314

ABSTRACT

This report describes a path planning method suitable for real-time implementation on an industrial manipulator. Efficient and fast path planning is achieved by decomposing the 3-dimensional space into a number of 2-dimensional subspaces. A method is devised to work directly with arm pose (configuration) instead of dealing with individual joint positions. These two aspects, namely decomposition and pose control, greatly speed up the path finding procedure and make it possible to perform real-time planning in a moderately cluttered environment. The path planner uses bit maps of the arm and obstacles which are directly available from a vision system. A neural network is utilized to determine the optimal arm postures for preventing unnecessary large arm movements during path execution and for reducing torque requirements. The path planner has been successfully implemented and tested on an industrial manipulator in a moderately cluttered environment. Integration of a 3-D vision system with the planner and extension of the planning method to semi-structured and unstructured environments are planned.

1. INTRODUCTION

The general motion planning problem for a manipulator is to find a path from a given robot configuration to a goal position and orientation that avoids collision with a set of known obstacles. Several approaches to this problem have been proposed. Brooks [1983] has suggested a heuristic approach based on separate free space description for the upper arm and the payload. His method, although reasonably general, is fairly complex. The idea of using artificial potential field for robot manipulator and obstacles was introduced by Khatib [1985]. In this approach, the robot is pushed away by repulsive

* Most of this work was supported by NETROLOGIC Inc. through NASA Contract No. NAS7-1110.

field of the obstacles and is drawn towards the goal by an attractive field. Koditscheck [1987], addressed the potential field control from a topological viewpoint, showing that an almost global solution for the navigation problem can be developed using an appropriately specified potential field function.

Another approach proposed by Lozano-Perez [1987] is based on the use of configuration space also called C-space, which is a space of dimensions equal to the degree of freedom of the manipulator. The idea is a build a space free of obstacles for successively more links starting with the first two links. Thus for an n-link manipulator, a two dimensional C-space is built for the first two links, then a three-dimensional subset is constructed for the third link and this procedure is continued until an n-dimensional subset is built. This approach permits explicit characterization of the constraints on robot motion due to presence of obstacles. However, C-space obstacles tend to be complex especially when the dimensions are high. An improvement to this strategy is proposed by Gupta [1990], using a sequential search technique. This method results in one single dimensional plus $n-1$ two-dimensional path planning problems instead of one n-dimensional problem.

Another category of path planning approaches is hierarchical approximate cell decomposition introduced by Brooks and Lozano-Perez with subsequent contribution by other researchers [Faverjon 1986, Kambhampati and Davis 1986, Zhu and Latombe 1991). This approach consists of decomposing the configuration space of a manipulator into rectangloid cells at successive levels of approximation. Cells are classified as empty, full or mixed depending on whether they lie entirely outside the obstacles, entirely inside the obstacles or neither. The planner searches a graph for sequence of adjacent empty cells connecting the initial configuration to the goal configuration. If no such sequence is found, it decomposes the mixed cells into smaller cells and searches again until either a solution to the path planning problem is found or until mixed cells are smaller than some predefined size. The problem of decomposing a mixed cell is to maximize the volumes of the empty and full cells resulting from the decomposition so as to find a path, or to conclude the absence of a path, as quickly as possible.

In spite of the considerable efforts, almost all existing path planning methods lack efficiency, require considerable computational power and time, and face major difficulties in implementation on practical manipulators. We believe that there are two main factors contributing to these problems. First, the approaches tend to concentrate on a general but abstract manipulator-obstacles environment with little attention to common manipulator kinematic arrangement and geometry. Second, the path planning problem is generally set and solved at the joint level which requires finding a sequence for each manipulator

joint angle to move the arm from the initial configuration to the goal configuration. In this report we develop a path planner on different premises. We fully take the advantage of the manipulator kinematic arrangement to simplify path planning. We also treat path planning as the problem of finding a sequence of manipulator configurations rather than a sequence of joint angles. Although we do not cover every conceivable type of manipulator, the class of manipulators that can be treated using our method is sufficiently large and includes most available industrial manipulators.

2. DECOMPOSITION OF MANIPULATOR WORKSPACE

In this section we describe a method of decomposing the 3-dimensional workspace of a class of manipulators into a number of 2-dimensional workspaces. This will enable us to devise a very simple and efficient path planner.

The manipulators to be considered have n_1 major joints (arm) and n_2 minor joints (hand). Usually $n_1 = n_2 = 3$, however, we allow the possibility of any number of joints. We will assume that the major joint axis 2, 3, ...,n_1 are parallel to one another and perpendicular to the joint axis 1. This assumption does not restrict the application of the path planner to specialized manipulators. In fact many types of manipulators qualify the above description which include articulated (both elbow and parallelogram linkage types), spherical, cylindrical and Cartesian types. Figures 1a - 1d show examples of kinematic arrangements that can be used with our path planner. Examples of industrial manipulators of these types are PUMA 260, PUMA 560, PUMA 760, Cincinnati Milacron T^3 735 , Cincinnati Milacron T^3 886, GMF M-100 and Rhino XR-3, to mention a few. This kinematic arrangement enables us to develop a simple and efficient path planning algorithm, as will be seen. In order to simplify the presentation we will consider, from time to time, a robot with only three revolute major joints and refer to these joints as waist, shoulder and elbow joints. However, the method to be described is applicable to robots of the above type having any number of major revolute or prismatic joints. As with most other approaches, for the purpose of path planning, we use a simple conservative approximation for the minor joints and enclose the hand and any payload in a bounding box.

Consider a rectangloid of dimensions $L \times H \times D$ where L, H and D are length, height and thickness of the rectangloid. These dimensions are chosen such that for a specified joint 1 position θ_1 and for all joint positions $\theta_2, \theta_3, \ldots, \theta_{n_1}$, the arm from link 2 onwards lies entirely within the rectangloid, as shown in Figure 2. For reason that will become clear shortly, we must select L, H, D to be as small as possible. The height H and length L can be selected to be equal to twice the sum of lengths of links 2,..., n_1

and the bounding box containing the hand and payload. This will ensure that the arm stays within the rectangloid when it is fully extended up, down, forward or backward. The thickness D depend on the thicknesses of links 2,..., n_1 and any offset between these links. The thickness is small when there is no offset and the links are thin. For a PUMA 560, $L = H \approx 2000\ mm$ and $D \approx 200\ mm$, so that when θ_2 and θ_3 change through their entire ranges, links 2, 3 and the hand remain within a rectangloid of dimensions 2000×2000×200.

Consider now a plane that divides the above rectangloid along its thickness into two equal rectangloids of dimensions $L \times H \times \frac{D}{2}$ and denote by R the region of the plane that is common between the two smaller rectangloids. Let us place the horizontal axis u and the vertical axis v on the above plane such that v is along link 1, as shown in Figure 3. The axis u is now perpendicular to link 1 but is parallel to links 2,..., n_1. The origin of the $u-v$ axis is at the intersection of the plane with joint 2 axis. As the joint 1 position θ_1 changes (the waist rotates), the plane and the rectangloid will also change position (rotate) so as to keep the arm within the rectangloid. The position (angle of rotation) of the plane ϕ is related to θ_1 by $\phi = \theta_1 + \alpha$, where α is a constant. It is to be noted that for a particular ϕ, the planar region R is uniquely specified in the 3-dimensional space. We will denote by R_ϕ the planar region corresponding to position ϕ. As θ_1 changes from its minimum value to maximum value , ϕ alo changes through its range $(\phi_{max} - \phi_{min})$ and a volume that includes the robot work space is swept by the planar region R_ϕ. In doing so, R_ϕ also intersects obstacles that are in the robot work space.

Let the normal projection of links 2, ..., n_1 and the hand on the planar region R_ϕ be represented by the function $f_\phi(u,v)$ where for a point $u=u_1$, $v=v_1$ located on the projection we have

$$f_\phi(u_1, v_1) = a_1 \neq 0 \tag{1}$$

For the point (u_2, v_2) not on the projection, $f_\phi(u_2, v_2) = 0$. We generally set $a_1 = 1$, however, other values can also be used to assign weights to different parts of the arm. Note that when $f_\phi(u_1,v_1) \neq 0$, we assume that the arm occupies the whole thickness D at the point (u_1, v_1). This is conservative when links have different thicknesses or offsets. Since the arm is located in a rectangloid of thickness D rather than the planar region R_ϕ, we grow each obstacle by $\frac{D}{2}$ on each side, as shown in Figure 4. This will allow us to treat collision in terms of intersection of the planar region R_ϕ and obstacles instead of the more difficult problem of determining the intersection of the rectangloid with obstacles. Let the intersection of R_ϕ with the set of obstacles

$\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_\mu\}$, be defined by the function $g_\phi(u,v)$, where μ is the number of obstacles. The point (u_1, v_1) belonging to the region R_ϕ intersects an obstacle if

$$g_\phi(u_1, v_1) = b_1 \neq 0 \tag{2}$$

On the other hand if $g_\phi(u_2, v_2)=0$, the point (u_2,v_2) belongs to a free space. In (2) b_1 represents the weighting of the point on the obstacle and we generally set it set to 1. Collision of the manipulator and obstacles occurs when a point in the region R_ϕ satisfies both (1) and (2). The intensity of collision is given by

$$e_\phi = \int_{R_\phi} f_\phi(u, v)\, g_\phi(u, v)\, du\, dv \tag{3}$$

where the integral is evaluated over the region R_ϕ. It is more efficient and convenient to work with discrete intervals. Let the region R_ϕ be divided into a grid of $m_1 \times m_2$ squares of side equal to $s = \frac{L}{m_1} = \frac{H}{m_2}$. The arguments u and v in the functions $f_\phi(u,v)$ and $g_\phi(u,v)$ now takes only discrete values $u = 1,2,\ldots,m_1$ and $v = 1,2,\ldots,m_2$. If a's and b's in (1) and (2) can take only binary values 1 or 0, the functions $f_\phi(u,v)$ and $g_\phi(u,v)$ represent bit maps of the arm and obstacles respectively. The intensity of collision is now

$$e_\phi = \sum_{R_\phi} f_\phi(u,v)\, g_\phi(u,v) \tag{4}$$

where the sum is taken over all discrete values of u and v in the region. The accuracy of the representation can be increased by reducing the size of squares. Although any collision must be avoided, the notion of the intensity of collision is useful. Because of the conservative approximations, a low collision intensity value may in fact be a non-collision. Thus when e_ϕ is smaller that some predefined value, one may use a finer grid to determine more accurately the possibility of a collision. This is similar to the concept of cell decomposition []. However, we are now dealing with simple planar squares instead of cubes.

The position (angle) ϕ can also be discretized into N increments of

$$\delta\phi = \frac{\phi_{max} - \phi_{min}}{N} \tag{5}$$

One way of determining N is to find the maximum number of non-overlapping rectangloids in the range $(\phi_{max} - \phi_{min})$. For a prismatic joint 1, $\delta\phi = D$ and $N = \frac{\phi_{max} - \phi_{min}}{D}$. For a revolute joint 1, suppose that when the arm is fully extended

out its length is l. Let D the thickness of the rectangloid be the chord of a circle of radius l, as shown in Figure 5. Then, $\delta\phi$ can be obtained from

$$D = l\sqrt{2(1 - \cos\delta\phi)} \tag{6a}$$

or

$$\delta\phi = \cos^{-1}\left[1 - \frac{D^2}{2l^2}\right] \tag{6b}$$

The number N of planar regions R_ϕ is now obtained using (5) and (6b). The set of discrete positions of the planar region is now $\phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$. It is to be noted that the number of planar regions can be made as large as desired by considering overlapping rectangloids. However, little will be gained by having a large N.

Even though for a given ϕ_i, the planar region $R_i \equiv R_{\phi_i}$ is uniquely defined, the projection of arm, that is the bit map $f_i(u,v) = f_{\phi_i}(u,v)$ is not unique. This is due to the fact that for a given ϕ_i, the arm can have infinite number of different postures*, if the joint positions vary continuously. However, if we allow only discrete changes, the number of arm postures will be finite. In the case of a manipulator with three major revolute joints (waist, shoulder, elbow), the arm posture can change from arm fully extended up to arm fully extended down. For such an arm, intermediate postures correspond to different elbow up or elbow down arm positions. Figure 6 shows examples of arm configurations in the region R_i. The resolution of path planning algorithm depends on the number of arm postures ν that we allow. As ν is increased the resolution improves, however, more time will be needed to select a particular posture among the set $\mathbf{P} = \{P_1, P_2, \ldots, P_\nu\}$. Working with arm postures instead of joint angles is crucial in our path planning method.

PATH PLANNING METHOD

Consider a manipulator at some source (initial) configuration described by the planar region position and arm posture (ϕ_s, P_s). Suppose that the manipulator is required to move around the obstacles and place the hand at the goal position described by the Cartesian coordinates described by the vector n-dimensional vector x_g where $n \leq 6$. A set of manipulator joint positions corresponding to x_g can be found using an inverse kinematic method. Suppose that (ϕ_g, P_g) is the arm configuration corresponding

* **We use the term "arm posture" to denote the projection of links 2,..., n_1 on the region R_i. Arm configuration will be used to specify both the projection of links 2,..., n_1 on the planar region R_i and ϕ_i, the angle of the planar region.**

to the Cartesian goal coordinates x_g, where ϕ_g and P_g do not in general belong to the above defined sets $\boldsymbol{\phi} = \{\phi_1, \ldots, \phi_N\}$ and $\mathbf{P} = \{P_1, \ldots, P_v\}$. We can now select two arm configurations, say ϕ_i, P_j and ϕ_l, P_k that belong to the above set and are closest to (ϕ_s, P_s) and (ϕ_g, P_g), where we can use a conventional measure for the closeness to be discussed below. The path planning problem is now to move the robot without collision with obstacles from its source configuration (ϕ_s, P_s) to the configuration (ϕ_i, P_j), then to the configuration (ϕ_l, P_k) and finally to the goal coordinates in Cartesian x_g. The first and last motion can be performed using an inverse kinematic control method, assuming that the obstacles are not very close to source or goal configurations. The major part of path is, however, from (ϕ_i, P_j) to (ϕ_l, P_k), and we now describe how this path can be planned.

Consider a typical configuration (ϕ_i, P_j), $\phi_i \in \boldsymbol{\phi}$, $P_j \in \mathbf{P}$, a set of obstacles $\boldsymbol{\Omega} = \{\Omega_1, \Omega_2, \ldots, \Omega_\mu\}$ and the planar region $R_i \equiv R_{\phi_i}$. The next planar regions to be selected is either R_{i-1} or R_{i+1} depending which one is closer to the desired plane R_l. Let us assume, without loss of generality, that R_{i+1} is the closer plane. Let the bit map of arm posture and obstacles on this plane be $f_{i+1}(u,v)$ and $g_{i+1}(u,v)$, respectively and evaluate

$$e_{i+1} = \sum_{R_{i+1}} f_{i+1}(u,v)\, g_{i+1}(u,v)$$

Now if $e_{i+1} = 0$, there will be no collision and joint 1 position is changed (the waist is rotated), without changing the arm posture, so that the planar region is in the next location ϕ_{i+1}. However, if $e_{i+1} \neq 0$, we must select another arm posture among the set $\mathbf{P}$ and check for any collision. A physically meaningful criterion for posture selection is to pick one that is closest to the present posture P_j in the sense that a norm of joint position changes is minimum so as to reduce manipulator torque requirements. More precisely, let $\theta_j^T = (\theta_{j2}, \ldots, \theta_{n_1})^T$ be the vector of positions of joints 2,..., n_1 when the arm posture is P_j. Let the corresponding vector of arm posture P_k be θ_k. Then the next arm posture P_k is chosen according to

$$P_k : P_k \in \mathbf{P} \text{ and } \| \theta_k - \theta_j \| = \min \tag{10}$$

The arm posture P_k is now checked for collision using (4) and the procedure is repeated until either a suitable posture is found or the absence of a collision free arm posture is established. In the latter case, a higher resolution planner involving more planes and arm postures, i.e. higher N and v, can be attempted.

It must be emphasized that, the determination of the optimum arm posture according to criterion (10) can be performed off line and the results can be stored in a look up table. In this case, we can evaluate $|| \theta_k - \theta_j ||$ for $j,k = 1,2, ..., v$ and arrange $\{P_1, P_2, \ldots, P_v\}$ in increasing order of the norm. This reduces the on-line time for the selection of next configuration to extremely small values.

An alternative method of determining optimal arm postures is to use a neural network. Consider a backpropagation neural network with three layers, input, output and output. The input layer consists of 2 sets of neurons. The input to the first set is the v standard postures $\mathbf{P} = \{P_1, P_2, \ldots, P_v\}$, described above. At any given time only one of these inputs (postures) is present. The inputs to the second set of neurons is the set of unblocked arm postures. The output of the neural network is the arm posture P_k that is optimal in the sense of (10). The network is fully inter-connected so that each hidden neuron has connections to all input and output neurons. The network is trained by presenting it with different arm postures, arm bit maps, and the desired next arm posture determined from (10). The network weights are determined using a standard backpropagation neural network package so that the error between the desired network output and the actual network output is smaller than some specified value. Once the network is trained, it can be used to generate optimal arm postures extremely fast.

4. EXPERIMENTAL RESULTS

The path planning method described in this report was implemented using a PUMA 562 robot. The obstacles consisted of several boxes placed on the floor and hanging from the ceiling, some in stalagtite-stalagmite fashion. The obstacles can be created and stored in a file or read from a vision system. Several experiments were carried out to determine the effectiveness of the path planner. In all cases the path planner determined and executed a collision free path extremely fast. The software and various programs, their interactions, and modes of operations are described in a separate technical report.

5. CONCLUSIONS

A path planning method suitable for real-time implementation on a class of manipulators is developed and described in the report. The planner has a number of novel features. First, it decomposes the 3-dimensional search problem into a number of 2-dimensional problems. Second, to determine a collision free path, we have devised a method of working with arm postures, instead of dealing with individual joint positions. This simplifies path planning and greatly speeds up path finding procedure. As a result,

we are able to perform real-time path planning in a moderately cluttered environment. Another major advantage of the planner is that vision information on obstacles can be directly integrated into the path planner. This is due to the fact that the path planner uses bit maps of the type which is directly available from a vision system and no additional signal processing is needed to convert vision information into a suitable form for the path planner. A further feature of the approach is the use of neural a network to determine the optimal arm postures for reducing large arm movements during path execution and also reducing torque requirements.

Since we use conservative approximations, the path planner may not be able to find a path in a very cluttered environment, even when a collision free path does exist. However, the path planner has been successfully implemented and tested on an industrial manipulator in a moderately structured environment. At present we are working on the integration of a 3-D vision system with the planner and extension of the planning method to semi-structured and unstructured environments.

REFERENCES

1. R. Brooks, "Planning collision-free motions for pick and place operations," Int. J. Robotics Res., vol. 2, no. 4, pp. 19-44, 1983.

2. R. Brooks and T. Lozano-Perez, "A subdivision algorithm in configuration space for findpath with rotation," in Proc. 8th Int. Joint Conf. Artificial Intell., pp. 799-806, Karlsruhe, Germany, 1083.

3. B. Faverjon, "Object level programming of industrial robots", Proc. IEEE Int. Conf. Robotics Automat., pp. 1406-1412, San Francisco, CA, 1986.

4. K. K. Gupta, "Fast collision avoidance for manipulator arms: A sequential search strategy," IEEE Trans. Robotics Automat., vol. 6, no.5, pp. 522-532, 1991.

5. T. Lozano-Perez, "A simple motion planning algorithm for general robot manipulators," IEEE J. Robotics Automat., vol. 3, no. 3, pp. 224-238, 1987.

6. S. Kambhampati and L. S. Davis, "Multiresolution path planning for mobile robots", IEEE J. Robotics Automat., vol. RA-2, no. 3, pp. 135-145, 1986.

7. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", Int. J. Robotics Res., vol. 5, no.1, pp. 90-98, 1986.

8. D. E. Koditscheck, "Exact robot navigation by means of potential functions: Some topological considerations," Proc. IEEE Conf. Robotics and Automation, pp. 1-6, 1987.

9. D. Zhu and J. C. Latombe, "New heuristic algorithms for efficient hierarchical path planning," IEEE Trans. Robotics Automat., vol. 7, no. 1, pp. 9-20, 1991.

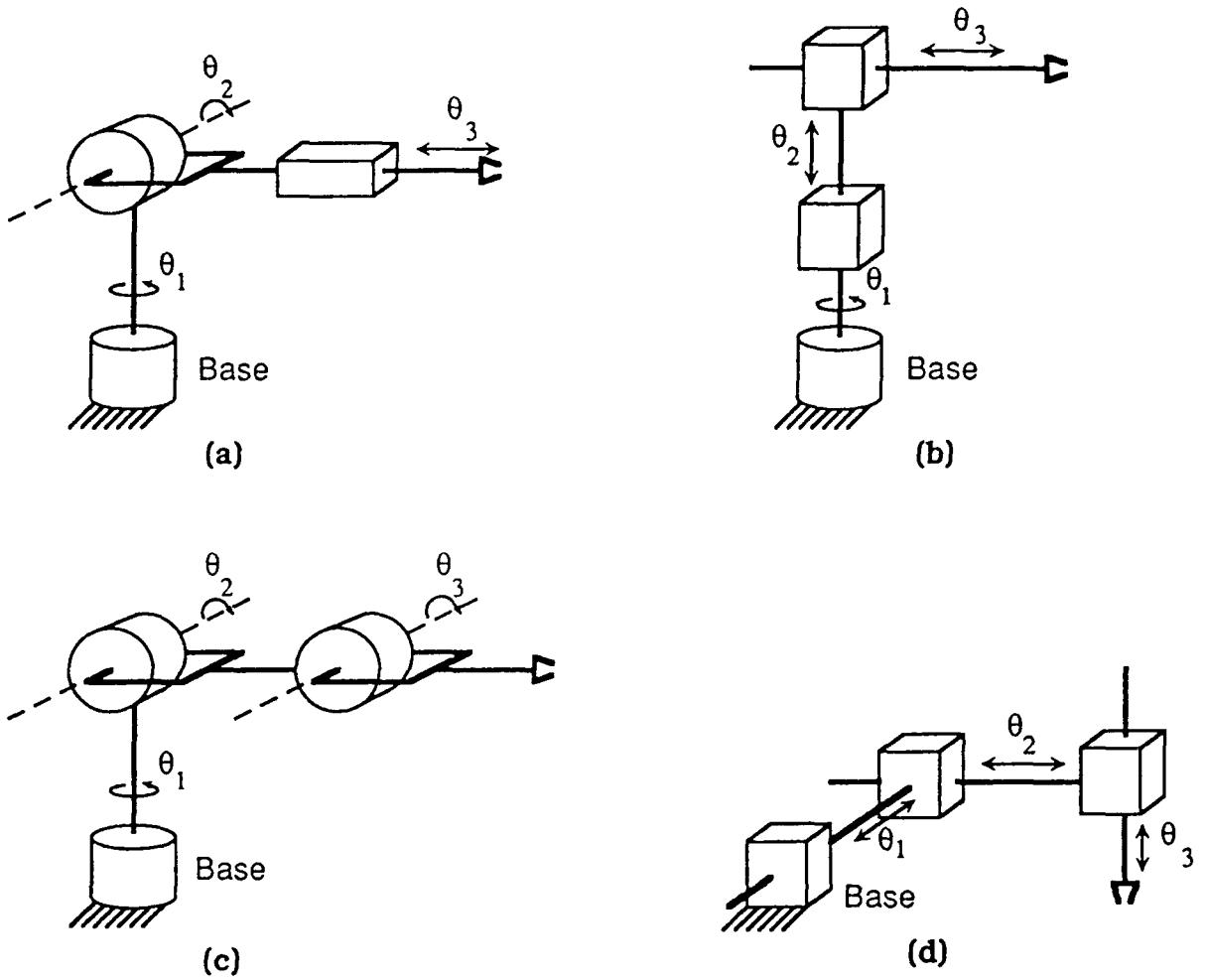


Figure 1
Various kinematic arrangements

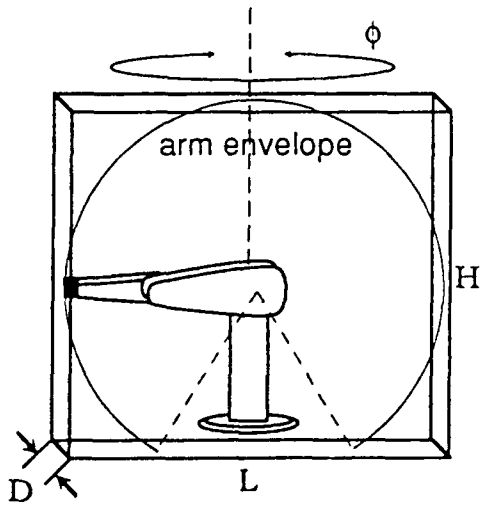


Figure 2
Rectangloid bounding the arm envelope

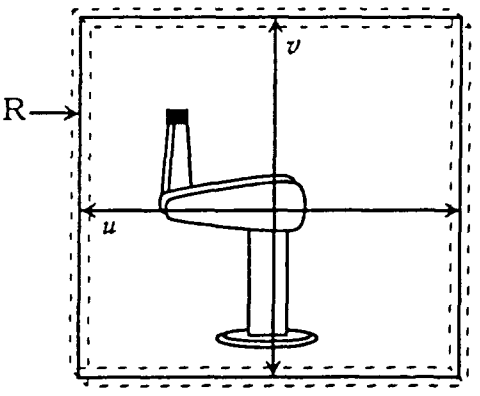


Figure 3
Plane R bisecting the rectangloid.

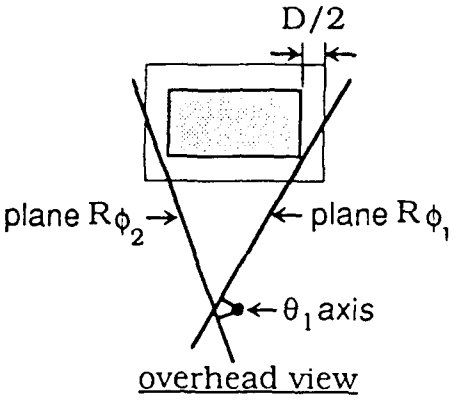


Figure 4
Intersection of two planes with an obstacle increased by a radius of D/2

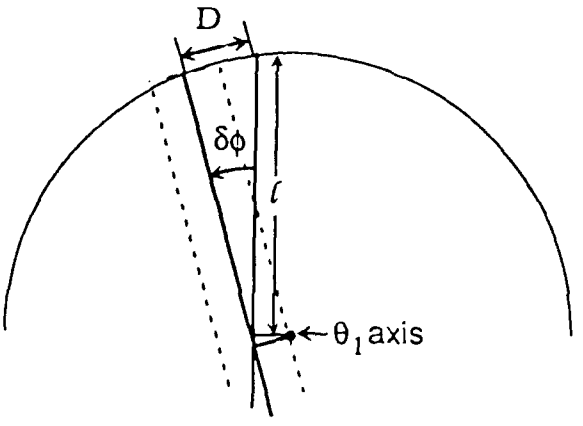


Figure 5
Incremental change in plane angle

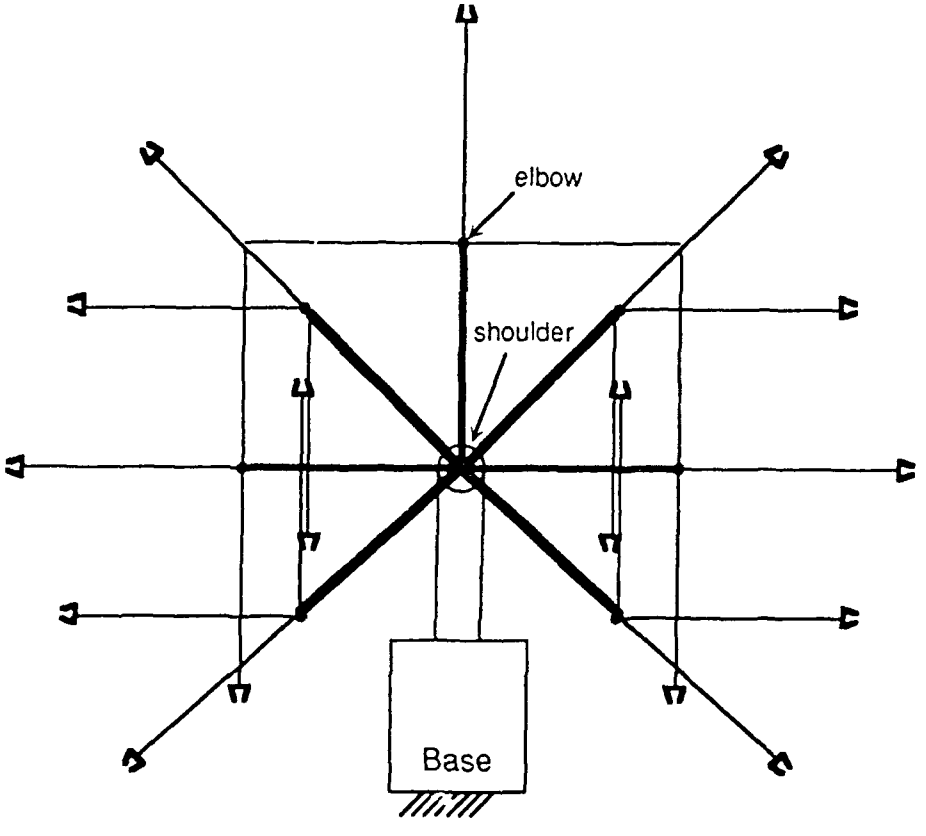
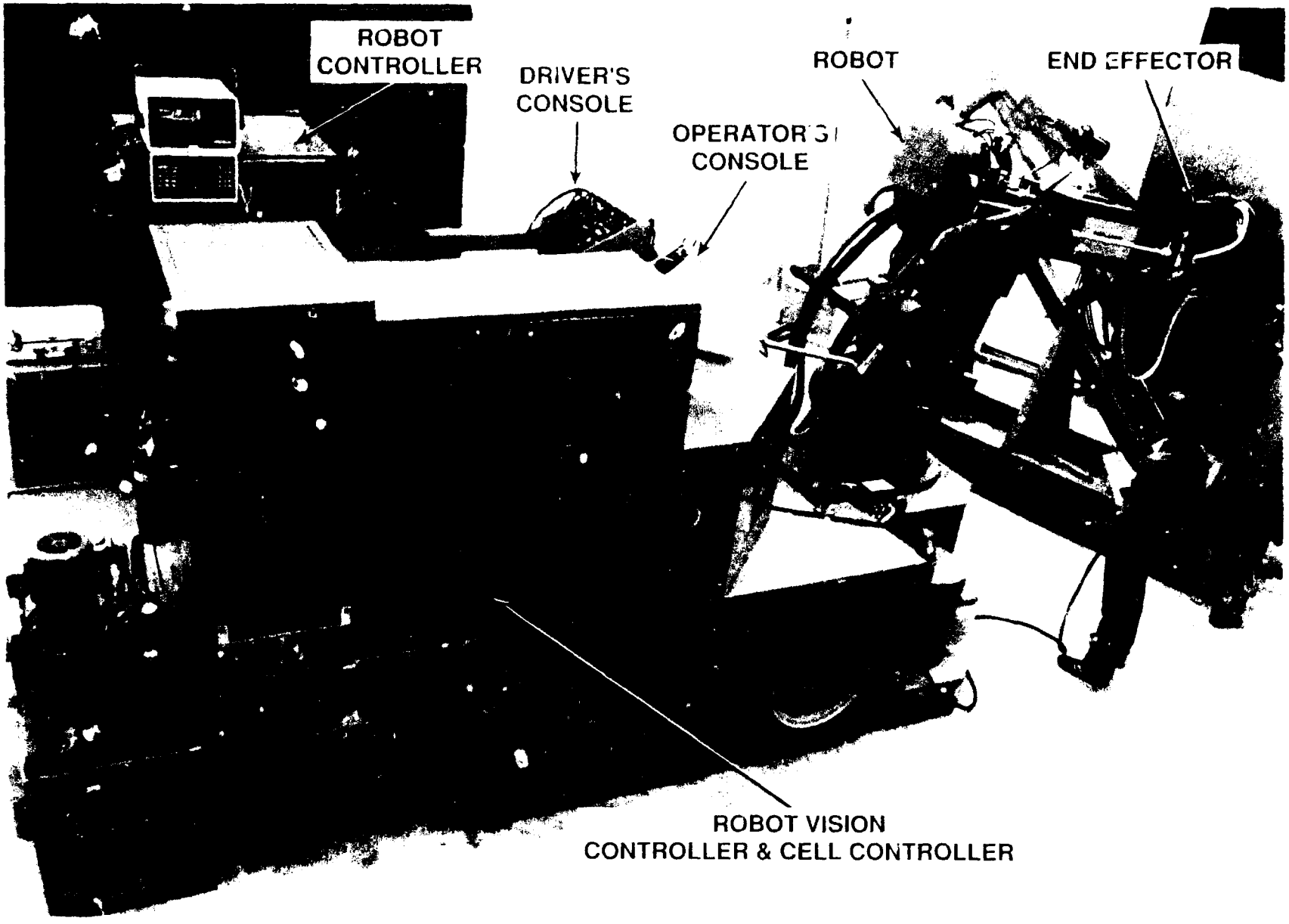


Figure 6
Arm postures for a plane R


ROBOT
CONTROLLER
DRIVER'S
CONSOLE
OPERATOR'S
CONSOLE
ROBOT
END EFFECTOR
ROBOT VISION
CONTROLLER & CELL CONTROLLER