INVESTIGATION OF ROBOTIC ASSEMBLY OF AN ELECTRIC MOTOR UTILIZING SINGLE ARM AND DUAL ARM ROBOT CONFIGURATIONS

By

JEFF REID

Bachelor of Science

Industrial Engineering and Management

Oklahoma State University

Stillwater, Oklahoma

1984

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
May, 1987

INVESTIGATION OF ROBOTIC ASSEMBLY OF AN ELECTRIC
MOTOR UTILIZING SINGLE ARM AND DUAL ARM
ROBOT CONFIGURATIONS

Thesis Approved:

John W Nazemetz
Thesis Advisor

K E Case

Joe H. Mize

Norman N. Durham
Dean of the Graduate College

PREFACE

An electric motor was assembled utilizing: (a) human labor only, (b) human labor combined with the assistance of a single robot arm, and (c) coordination between two robot arms. Assembly times were obtained for both human labor assembly of the motor and single-arm assembly of the motor; however, an unexpected equipment malfunction prevented the completion of the dual-arm assembly project. The data collected during the course of the study was analyzed to provide a basis for comparison between each method of assembly, as well as a comparison of each method with regard to its use in an actual manufacturing environment. Robotic applications in the manufacturing industry and the limitations of robotic equipment due to technological constraints were described. Hardware developments, which allow greater flexibility in robotics projects were illustrated, and their use in future projects is encouraged.

I wish to express my sincere gratitude to the individuals who assisted me in this project and during my coursework at Oklahoma State University. In particular, I wish to thank my major adviser, Dr. John W. Nazemetz, for his intelligent guidance, inspiration, and invaluable aid. I am also grateful to the other committee members, Dr. Joe H. Mize and Dr. Kenneth E. Case, for their advisement during the course of this work.

Special thanks are due to Oklahoma State University's Robotic Applications Laboratory which provided the robotic equipment and facilities for the study. I would also like to express my gratitude to Mr. Joe Reichert of Dayton Electric Manufacturing Company for providing information and illustrations on the electric motor used in the study.

The help of Unimation technical staff members, Rich Rothenhever, Pete Hellerman, Dave Lowe, John Silban, and Ken Yeomans, is also sincerely appreciated. A thank-you is also extended to Ray Carrier of Texas Instruments in providing technical assistance.

Special thanks are due to my sister, Anne, and Ms. Joy McIntosh for their participation in the study. My deepest appreciation is extended to my mother, who typed the manuscript, and to my father; both provided constant support, moral encouragement, and understanding.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF SYMBOLS

○ - Assembly operation (see Figures 18, 22, 36)

□ - Inspection operation (see Figures 18, 22, 36)

▽ - Move/Transport operation (see Figures 18, 22, 36)

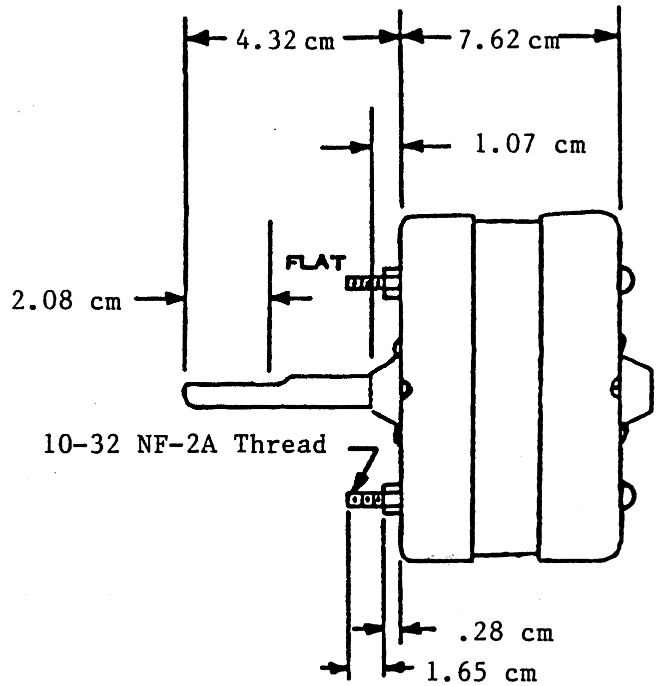
CHAPTER I

INTRODUCTION

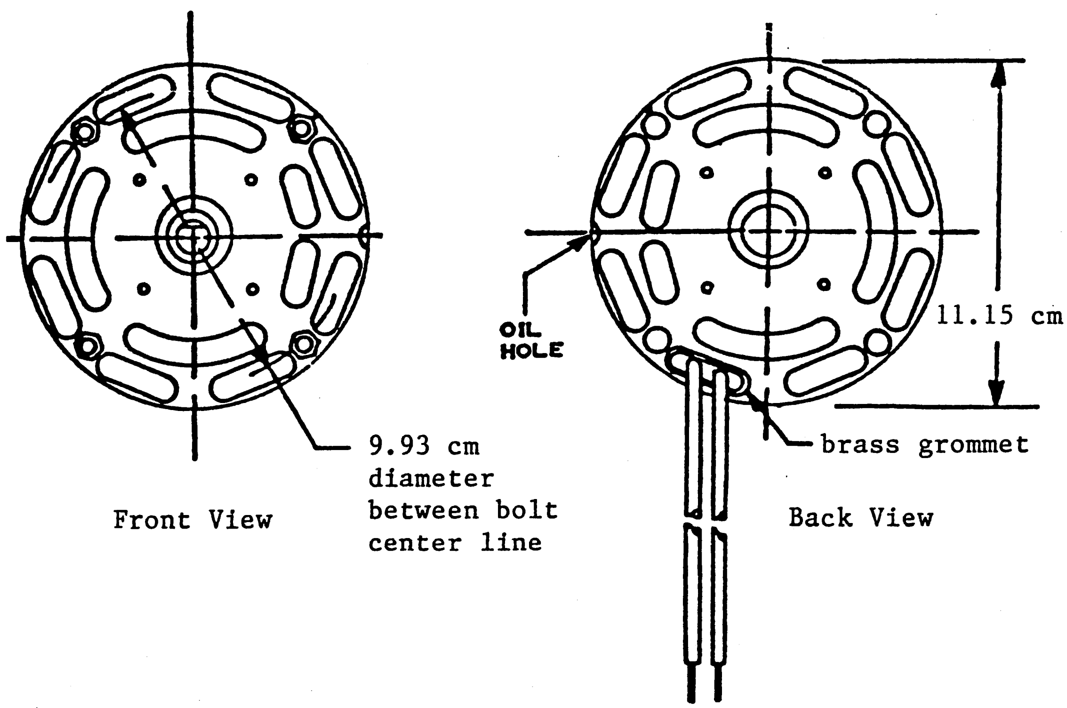
The automation of assembly tasks is one of the most formidable challenges in the manufacturing industry today and will continue to play an increasing role in the development of the fully automated "factory of the future" in the years ahead. The increasing utilization of robotics for assembly tasks has shown that a tremendous potential exists for robotic assembly of products, especially those which are produced in batch quantities.

The intent of this paper was to examine a product which would typically be produced in a batch quantity and perform a final assembly the product using (a) human labor only, (b) human labor combined with the assistance of a robot arm, and (c) coordination between two robot arms. These three methods of assembly could then be compared to determine various processing characteristics, such as: assembly time, fraction of parts to be reworked, production piece rates, etc. These processing characteristics determined by the assembly techniques can be compared to actual industry applications, and conclusions may be drawn with regard to the use of each method in actual manufacture of the product.

The product which was selected for the assembly experiment was a single-phase electric motor shown in Figure 1. A general-purpose motor of this type is typically utilized for powering fans, air conditioning



Side View



Front View

Back View

Figure 1. Electric Motor Finished Assembly

compressor drives, and many other appliances. It represents a prime candidate for which a batch-type automated assembly method could be used. This particular product contains four major parts which comprise the main assembly: the front cover plate (referred to as the "front bell", see Figure 2); the back cover plate (referred to as the "end bell", see Figure 3); the stator assembly (shown in Figure 4); and the shaft (shown in Figure 5). In addition, four bolts are used to secure the major components to form the final assembly. Figure 6 illustrates an "exploded" view of the motor showing the orientation required for the components.

The remainder of this chapter describes the objectives of the report in greater detail as well as the assumptions which were made during the study. Also included in the report in Chapter II is a Background of robotic applications in assembly, the limitations caused by undeveloped technology, and a discussion of the techniques typically considered in robotic assembly implementation. Chapter III contains a complete description of the experiment, including a description of the technique utilized, the equipment used, hardware and software development, and the actual manual, one-arm, and dual-arm assembly procedures. Chapter IV presents the results of the experiment for the three assembly methods, while Chapter V addresses an analysis of the results in relation to the use of these methods in an actual manufacturing environment. Finally, Chapter VI provides the conclusions of the study and suggestions for further research.

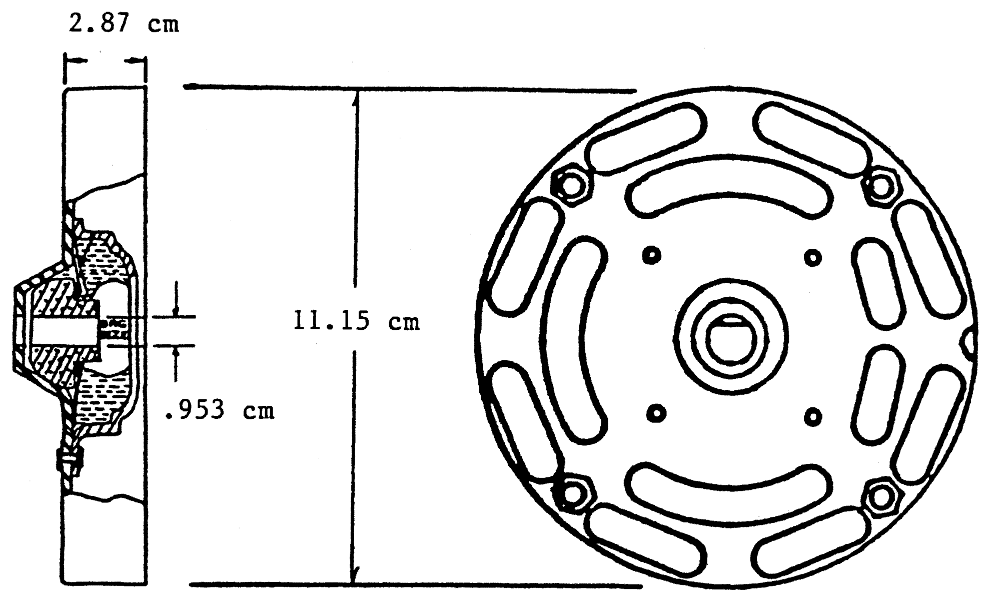


Figure 2. Electric Motor Front Bell

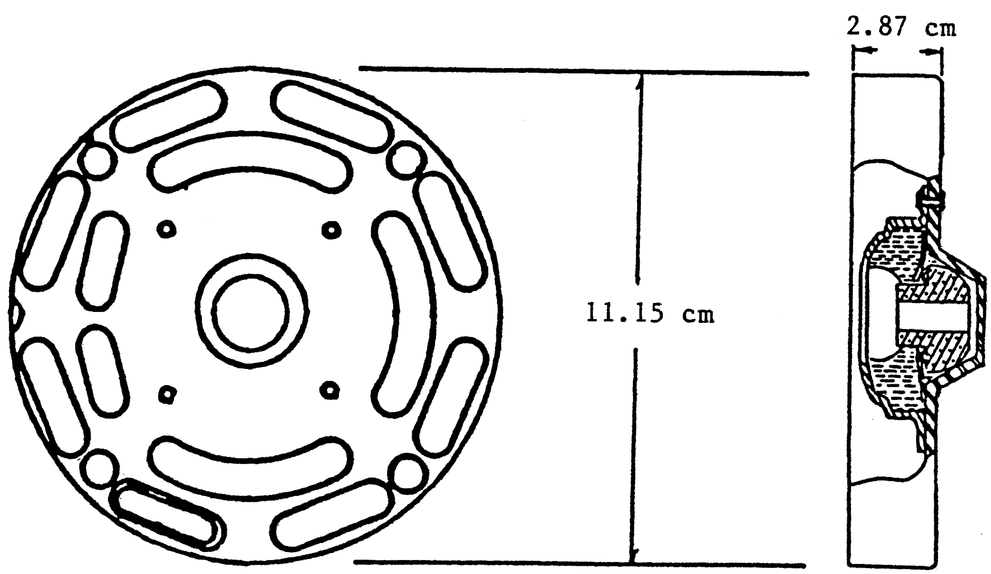


Figure 3. Electric Motor End Bell

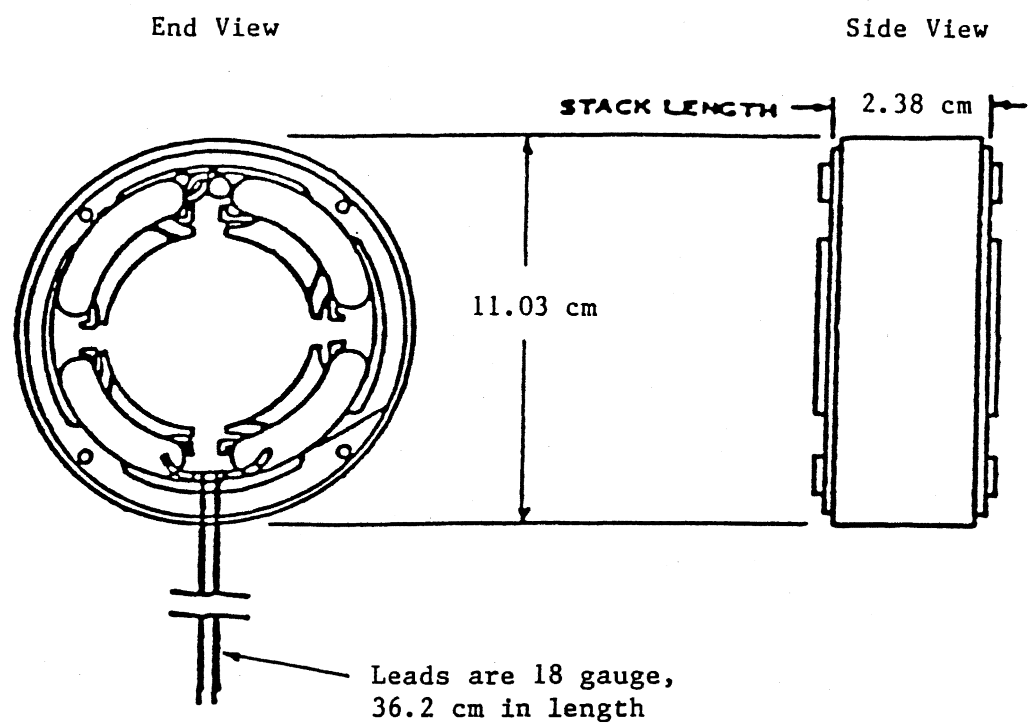


Figure 4. Electric Motor Stator Assembly

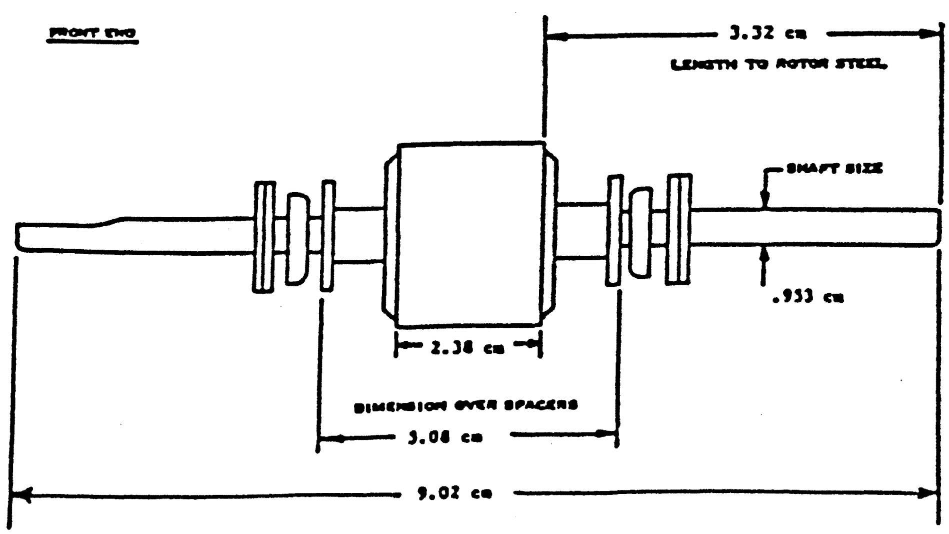


Figure 5. Electric Motor Shaft

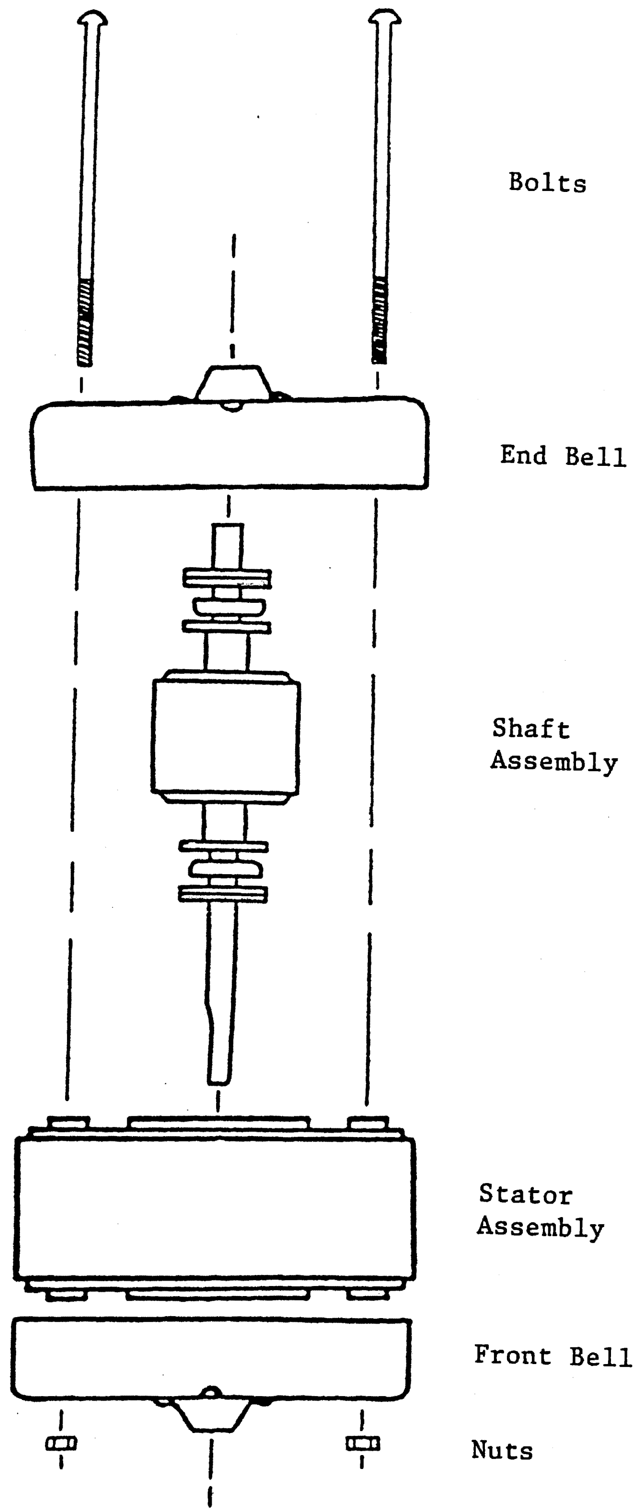


Figure 6. Exploded View of Motor

Objectives

As previously stated, the main objective of this thesis was to assemble a simple electric motor using first one robot arm, and then coordination between two robot arms to produce the assembled motor. Also, manual assembly of the motor was desired in order to compare the three methods. Conclusions could then be drawn as to the applicability of these methods in an industrial setting.

For each of the three assembly methods, certain variables could be quantitatively measured and statistically analyzed. For example, the total assembly times for each method were summed. This provided a basis for the calculation of a mean with a corresponding variance which established a statistical distribution for each method. Not only could a distribution of assembly times be achieved, but also a determination of the fraction rejected (rework) could be made. Based upon the mean time to assemble the motor, a production rate (pieces per hour, etc.) for each method was determined.

A secondary objective of the paper was to provide the reader with a brief overview of the current limitations facing the use of robotics in assembly applications. It is important to realize that although the equipment utilized in this study is highly sophisticated state-of-the-art industrial robotics, there exist at the present time many limitations on the use of such equipment for a given task. These limitations are described in the Background chapter.

The third and final objective of this report was an attempt to relate these quantifiable measures of performance for each method to the application of that method in an actual industrial setting. For each assembly technique, the advantages and disadvantages of the technique are discussed as well as how each method could be integrated into the overall manufacturing process.

This report has been written with the intent of satisfying these three objectives. Although these objectives were wide in scope, there were several limitations and general assumptions which restricted the overall scope of the assembly project. These limitations and several assumptions are discussed in the following section.

Limitations and Assumptions

Upon initiation of the study, it was apparent that certain assumptions would have to be made regarding the overall scope of the report. A major limitating factor was the time limit imposed on the project. The time constraint imposed a limit on the number of methods which could be used in assembly of the electric motor. Thus, the assembly procedures used in the study may not be optimal. Indeed, there exist many other assembly combinations which could decrease the assembly time from the time obtained using the current methods.

Perhaps the greatest limiting factor for a project of this nature is the availability of precision machining with which to fabricate fixtures, tooling, and other locator devices. In an actual manufacturing environment, all of the necessary jigs, fixtures, and special purpose tooling would be custom-built to close tolerances by a machine shop. The necessary equipment would then be securely installed by professional workers. Unfortunately, the precision machinery needed to produce the close tolerances was not available for the study. Because of this, the majority of fixtures and tooling produced for the experiment were composed of materials suitable for simple fabrication, such as wood and lightweight aluminum. Due to the smaller degree of strength and rigidness that wood and aluminum have when compared with steel, it is not surprising to discover that there will exist a corresponding lack of positioning accuracy when these types of fixtures are used. This was in fact a limitation in some aspects of the project, particularly with regard to positioning accuracy as related to robot arm speed. However, in several instances the flexibility provided by the wood construction prevented damage to the robot gripper when minor mistakes were made during initial arm positioning.

Another limitation to the assembly project was the limited choice of end effectors used on each robot arm. The only type available for use throughout the duration of the project were simple pneumatic "open and shut" pivot action grippers. Although this type of gripper provides sufficient holding force as well as a high degree of reliability, it is not well suited for precision assembly tasks involving complex motions and/or precision positioning of parts. In addition to the limited dexterity of the grippers, neither gripper was capable of providing sensory feedback to the robot controller. This force-sensing feedback or "touch" provides a way in which the robot can act upon information regarding the applied force acting upon the workpart at any given time. The ability of the robots to assemble the motor would have been enhanced by the utilization of force-sensing feedback, however, time limitations prevented development of such a feedback system.

In addition to the limitations imposed upon the study, several assumptions must also be made with regard to the applicability of the study in an actual manufacturing environment. First, the study assumes that only final assembly of the electric motor is to be considered and any intermediate sub-assemblies of which the motor is composed are neglected. These sub-assemblies are assumed to be completely assembled, transported to the final assembly area, and placed in the proper orientation prior to the final assembly procedure.

The second assumption involves the production quantities for the motor. Since robotic applications are especially suited for production of parts in batch quantities (typically from a few parts to several hundred parts), the production quantity for the electric motor was assumed to be 100 motors per "batch". This particular production quantity was selected mainly for ease of analysis in subsequent calculations of the performance of the assembly system.

The final assumption concerns both the physical ability of human "workers" involved in the study as well as the workers' experience in assembly of the motors. With regard to any manual operation performed in assembly of the motor, the assumption has been made that the worker maintained a normal work speed and that his ability or experience in no way placed a bias on the overall performance of the assembly system. The concept of "normal" work speed relates to a worker's "effort" rating. In the course of the study, all manual labor was assumed to be performed at an "effort" rating of 100%, indicating that the worker was neither excessively fast nor excessively slow in performance of the task. This effectively eliminated the possibility of an "above-average" or "below-average" worker distorting the variance of the assembly time distribution and helped to maintain a fair measurement of assembly times.

These are the majority of the assumptions and the limitations which have been incorporated into the study. Although restrictive in nature, they were necessary in view of the short time duration allotted for the study. They were also essential in that they provided a relationship between the results of the project and the application of the findings to an actual manufacturing environment.

CHAPTER II

BACKGROUND

The application of robots in today's manufacturing environments has become widespread and continues to promise a rapid growth as more companies realize the benefits of automated production. In the majority of applications, the typical single-arm robot is utilized to perform a wide variety of industrial tasks, including:

- material transfer
- machine loading
- welding
- spray coating
- processing operations
- assembly
- inspection (1)

With major research and development accomplishments in the near future, the application of robots will expand greatly to fully incorporate many other manufacturing applications, such as complex assembly and inspection tasks as well as more delicate machining tasks.

In order to realize manufacturing's goal of a truly automated factory, the "factory of the future", the formidable challenge of robotic assembly must be resolved (2). Considering the general tasks in the manufacturing environment, assembly presents the most difficult challenge for a robot. Even for the easiest of tasks which a human worker may perform, such as the attachment of screws into a small faceplate, the robot must receive, interpret, and react to an enormous amount of data about its environment in order to achieve this simple goal.

At present, the number of applications in robotic assembly is limited due mainly to undeveloped vision and sensor technology as well as a lack of available software with which to program assembly tasks. Also, end effector advancements have been slow to arrive on the market. Thus, only simple cases of robotic assembly are currently feasible. In the automotive industry, for example, robots insert small light bulbs into instrument panels. However, a slightly more complex task, such as the installation of a cover that must be screwed onto a frame, usually cannot be performed economically by today's robots. Using vision sensing the cover could be located properly, and various tactile force sensing located in the robot's gripper could be used to prevent excessive stress on the cover. However, most of the vision systems currently available are very expensive as well as the hardware required to interface the systems to the robot. Further, the force-sensing capability for small part manipulation has not been developed (2).

The majority of robotics experts agree that in order to be both practical and economical, a dexterous two-armed robot would be necessary in order to perform such operations (2; 3). It is the essence of this thesis to show that two-arm coordination in assembly can be accomplished and to show how the various methods utilized in the assembly of the electric motors can be compared to an actual industrial situation.

Robotic Applications in Assembly

One of the biggest areas in robot applications is that of assembly. Studies indicate that the use of robots in assembly applications will increase from a 10% market share in 1984 to as much as 25% by 1990 (3). As far as the traditional manufacturing environments (e.g., job shops, batch production, and automated production) are considered, batch-type assembly operations offer the most promise for using robots. The reason for this is twofold: first, products manufactured in batch quantities (ranging from a few dozen to several thousand units) are especially suited to the operational characteristics of the robot. In other words, most robots are too slow to meet most mass production requirements, yet are much faster than typical job shop needs. Second, in batch assembly there are variations in products which are significantly greater than in mass production. This results in a greater need for flexibility in line changeovers. Robots are ideally suited for this requirement due to their programmability.

In many companies utilizing robots in the assembly process, the robots are combined with human workers into what is termed as an "Adaptable Programmable Assembly System" (APAS) (1). The APAS system is typically composed of both conventional material handling devices (conveyors, part feeders, etc.) and robot arms, commonly arranged in an in-line fashion where the workpiece moves down a conveyor and is operated on by each successive robot. Assembly tasks requiring a special skill or judgment, of which the robot is not capable, are performed by human workers stationed along the line. As explained previously, only simple tasks may be performed by today's robots due to undeveloped sensing technology. The limitations due to these undeveloped capabilities are discussed in the following section.

Present Limitations Due to Undeveloped Technology

For many robot applications, especially with regard to assembly, the robot must incorporate "humanlike" capabilities such as vision (object recognition and hand/eye coordination) and tactile sensing (delicate part alignment and force measurement). Unfortunately, these capabilities have not yet been fully developed and many have not been incorporated into present-day robotic systems.

There are also a number of other areas in which significant improvements in robotics technology are required in order to provide robots which can perform a wide variety of common assembly tasks; some of these areas which are in need of development are:

° Low Cost, Effective Vision Sensing - One of the major limitations of robots in use today is the lack of a reasonably priced, effective sensing capability for determining the location, shape, and orientation of an object. Most of the systems available today cost from $20,000 to $30,000, and are economically prohibitive for most applications. Many vision systems currently use optical sensors (such as Charge-Coupled Device [CCD] cameras), although other types of sensors such as acoustic, electromagnetic, etc. are also employed. In addition to the hardware development, the software required for analyzing data received from sensors and converting it into a form usable by the robot is not well developed. Both the software interface capability and the sensing technology must be improved to enable robots to recognize patterns, determine location and orientation of objects, avoid collisions, and detect the presence of parts as well as flaws.

° Simple, Improved Gripper Dexterity - The basic open and shut operation of most currently available grippers is not adequate for some of the complex movements required in certain operations, especially those involving complex assembly and material handling operations. Today's typical grippers involve movement of a parallel-jaw with only one degree of freedom. In order to adequately encompass most of the assembly tasks to be performed by robots in the future, a gripper similar to a human hand would be required, with several fingers and at least three to four degrees of freedom. Although the mechanical design of such a device is currently under development by several research institutions, the main problem actually lies with the complex control algorithms needed for manipulation of parts, tools, and the like. To date, almost no control algorithms (even in their simplest form) exist for this type of dexterous gripper.

° Greater Flexibility - Most of today's robots, especially with regard to assembly, are not adequately flexible to enable them to perform a variety of different assembly tasks. This is of great concern in any manufacturing environment which contains a wide variety of assembly components.

° Improved Control Systems - Numerous areas of improvements are required in robot control systems. Controllers need to be much more sophisticated in their ability to interact between robots and sensors to cause changes in the movements of the robots based upon feedback received from sensors. The speed at which sensory data is received and translated into control instructions must be within just a few milliseconds. In addition, the ability of controllers must be improved to enable them to receive, and subsequently act upon, much more complex sensory data than presently possible. Control systems need more sophisticated database structure, including three-dimensional data bases similar to those found in Computer-Aided Design systems. Also, advances are desirable in development of hierarchal control structures and control logic systems, which would organize the various levels of control and use feedback logic to respond to events which occur in the robot's external environment.

° Low Cost, Effective Force Sensing - This is of particular concern for robots performing assembly operations. The robot must have a way to determine the position and orientation of an object through the measurement of contact forces. For example, a robot which is assembling a component using an automatic screw driver must be able to sense when the tightening process is complete in order to avoid stripping or breaking the screw. This implies some form of torque sensing capability to provide feedback information to the robot. Some of the major areas in which this type of improvement is needed are: texture recognition, thermal conductivity, and sensing large areas using compliant arrays of sensors.

° Lighter, Smaller Robots - The majority of robots in use are typically very large and heavy, and are able to lift (at best) weights equal to only about 10% of their own weight. The need exists for robots which have greater relative load capacities as well as smaller robots to perform assembly operations with delicate or intricate parts. This goal will involve combining advanced servo capabilities with developments in lightweight composites.

° Speed Increases - Although some robots are relatively fast, with end-of-arm speeds up to 60 inches per second, robots are generally unable to complete most manufacturing cycles at rates faster than humans. In some operations, this is not a problem, but in others, such as in assembly and certain material handling operations, the cycle time can be limited by the speed of the robot rather than the dynamics of the operation. In order to speed up the robot's movement to match or exceed that of a human assembler, servo systems must be improved to better accommodate the rapid changes in inertial characteristics of the manipulator as velocities and accelerations change during the cycle.

° Improved Positioning Accuracy and Repeatability - Many robots operating today can achieve positioning accuracies as close as ±.010 inches. However, many assembly and machinery operations require accuracies of at least ±.005 inches. This type of tolerance generally cannot be achieved using off-line programming; it must be manually "taught". This, however, incurs a high programming cost and makes small batch quantities or job shop assembly impractical. Also of concern in assembly operations is the issue of repeatability or the ability of the robot to return to the same position each cycle. Improvements must be done through better servo feedback and controller optimization algorithms as well as improved mechanical arm and manipulator drive systems.

° Improved Interfacing Capabilities with Existing Equipment - Many companies have experienced difficulty in attempting to integrate robots with machine tools, computers, sensors, and other manufacturing equipment. With the increasing use of computer-integrated manufacturing systems, there is a need for standardized interfaces and programming packages to enable all components of the system to communicate with each other. A majority of robots manufactured today have only limited communication ability, typically consisting of an on/off sensing capability. This communication ability needs to be expanded, both in hardware as well as in software development so that many robots, machine tools, sensors, material handling equipment, and large mainframe computers can be connected together to form integrated systems.

Until these areas have been researched and the resulting improvements have been incorporated into robots, the integration of robots into the manufacturing process will continue to remain a challenge, involving a greater degree of custom end effectors, fixtures, tooling, programming, and setup time.

Robotic Assembly Techniques

The techniques utilized in robotic assembly are numerous and are as varied as the products of the manufacturers themselves. Since most of today's applications are specialized, the appropriate technique of assembly is also somewhat specialized. As mentioned previously, some manufacturers will incorporate robots in an APAS fashion (utilizing the robots directly on the assembly line), while other manufacturers may opt to incorporate robots into work "cells" (1; 4; 5; 6; 7). The robots are enclosed in a cellular manner, where parts may arrive and depart in bins, and the robots process the parts in a so-called "island of automation" away from the main assembly line. Some companies may even utilize both of these methods in their factories.

Along with the decision of the correct placement of the robot, there also arises the decision of which type of robot to utilize for the assembly process. The choice of cartesian, polar, cylindrical, or jointed-arm robot configuration largely rests with the work envelope, motion characteristics, and degree of positioning flexibility required for the assembly task. Besides the type of robot selected for the task, there is also a decision to be made on the number of robots to accomplish the task. To a large degree this is dependent upon the configuration and complexity of the workpiece. In most current assembly applications, one robot arm (or at most, two robot arms) are utilized for the task.

End effectors and various types of grippers play an important role in robotic assembly. Due to the tremendous variation in parts and components which are candidates for robotic assembly, most end effectors are "custom fit" for the task. Since an effective "general purpose" gripper has not yet been produced for today's industrial robots, the full range of assembly operations required to make a finished component generally cannot be accomplished; however, in some cases this has led to the development of "quick-change" end effectors which allow multiple tools to be accessed by the robot in order to accomplish the entire assembly procedure. In addition to end effector variations or combinations, an even wider variation exists for the fixtures needed for correct part orientation. Not only must the workpiece be properly located in the assembly fixture, but all other related assembly components also must be correctly positioned for grasping and the pickup and release point(s) must be located within the robot's work envelope.

In many assembly applications, the need for additional feedback to the robot (other than stepper motor encoders, etc.) via external sensors is important for the accomplishment of the task. Many different techniques are utilized to provide this additional feedback, including the use of machine vision systems (both CCD camera gray scale imaging and parallel or stereo projection optics using conventional black and white television cameras), various tactile sensors (piezoelectric, for example), and auditory sensors (including ultrasonic, voice-activated, etc.) (8; 9; 10). These devices can be incorporated into the assembly process singly or in combination to enhance the efficiency and safety of the operation.

In addition to the many techniques and considerations listed, there remains yet another choice in the application of a robot in the assembly procedure: the amount of direct computer control to the robot during the process. The amount of hierarchal control is to some extent dependent upon the overall manufacturing process control (i.e., the robot needs continuous monitoring, or the robot can remain autonomous for considerable periods of time). Again, many different levels of hierarchal computer control can exist for any number of different assembly operations. The technique selected remains application dependent.

No attempt has been made in this section to describe in detail any one specific assembly technique simply because each application of robots to an assembly task is so dependent upon the product to be assembled, the robot's characteristics, and the manufacturer's process requirements. Until a truly effective "general purpose" assembly robot is developed, these applications and techniques will continue to be dependent upon the many factors outlined in this section.

CHAPTER III

DESCRIPTION OF EXPERIMENT

This chapter describes the experiment in detail, including the technique and equipment which was utilized, the hardware and software developed for use with the robots, and the procedures used in each method to assemble the motor. Essentially, the experiment can be divided into two parts: (1) the manual assembly of the motor with a human worker situated at a workstation using only his hands and the required hand tools for the task, and (2) the robotic assembly of the motor using (a) one-arm coordination with a human laborer to complete any operation which the robot could not accomplish, and (b) two-arm coordination in which the motor is assembled without any assistance from a human laborer. Both the manual assembly as well as robotic assembly of the motor were timed in order to obtain data which would be useful for comparison purposes. The results of the experiment and an analysis of the results are described in following chapters.

Techniques Utilized in Assembly of the Motor

The techniques which were utilized for each of the three assembly procedures in the study were developed from both the actual geometry of the motor as well as the required final assembly sequence. In order to complete the motor in the final assembly stage, the motor's components must be installed in a sequential order. As the reader can observe in

Figure 6 (refer to page 6), certain components such as the stator and shaft assemblies must be inserted into the main motor assembly prior to the end bell placement. After the front bell, stator, shaft, and end bell have been attached to form the body of the motor, the four bolts and corresponding nuts must be placed and tightened to secure the assembly. Therefore, the technique developed for each method was dependent upon the final assembly sequence of components.

The geometry of the motor and its components also played a part in the development of the techniques, although more so in the robotic methods than in the manual method. In the manual method, the part geometry was not a critical factor simply because of the tremendous adaptability of the human hand to handle any of the motor components quite easily. In the robotic methods of assembly, however, the gripper attached to each robot arm was not as dexterous. Careful attention had to be given to the orientation and overall geometry of each component in order to ensure the correct grasp was achieved by each robot arm.

The techniques obtained for use in the manual assembly of the motor essentially involved the determination of (a) the correct assembly sequence, (b) the correct placement of parts at the workstation in order to minimize arm reach distances and the coordination of arm motions for efficient assembly of the motor, and (c) the necessary tools or fixtures required to complete the assembly. A complete description of this procedure is provided in the section on manual assembly of the motor later in the chapter.

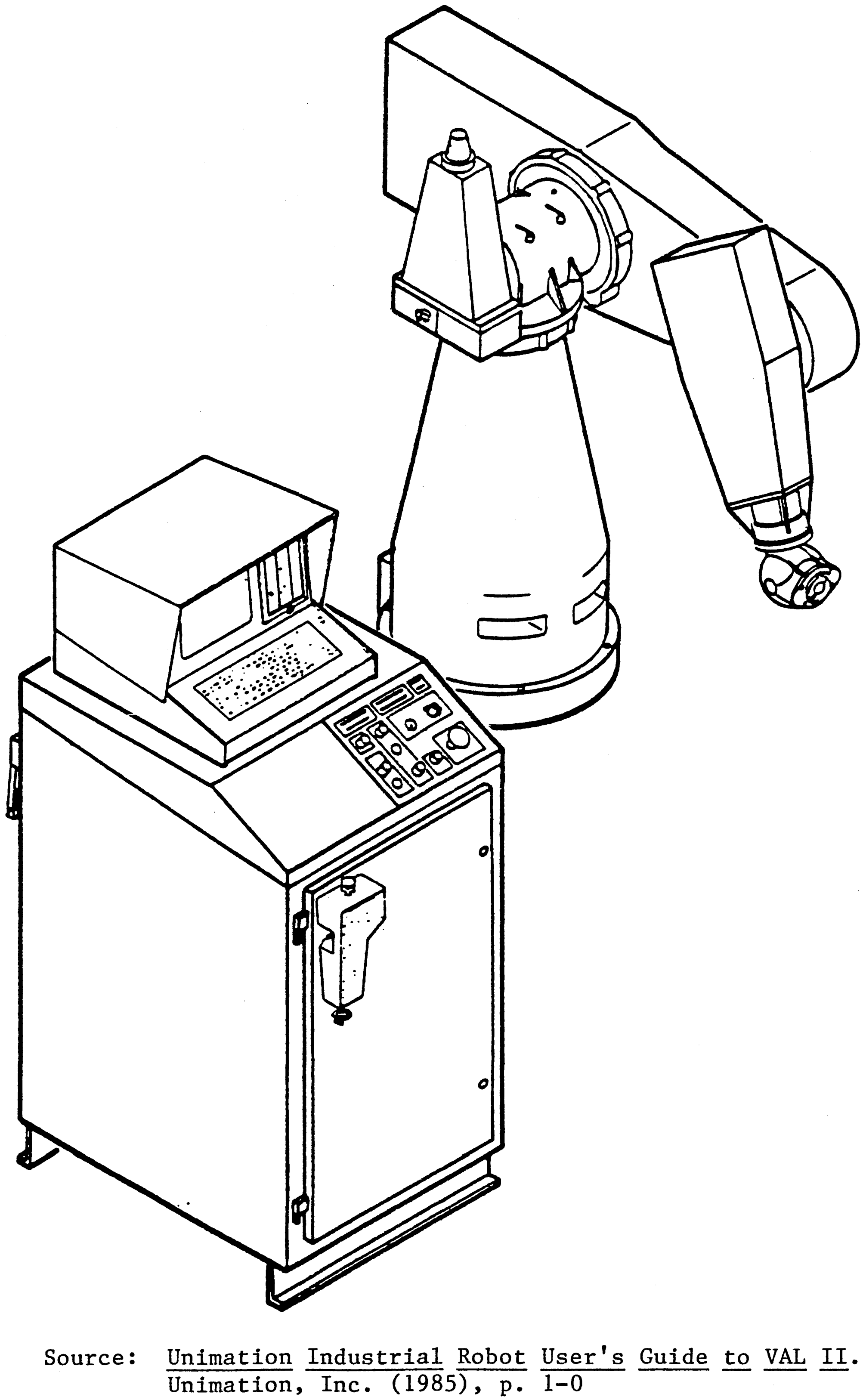
The technique utilized in the single-arm and dual-arm assembly was developed from (a) the analysis of the appropriate assembly sequence, (b) the determination of the correct placement and orientation of motor parts presented to each robot, (c) the determination of all necessary fixtures and tooling required to complete the assembly task, and (d) an analysis of the human/robot interface (one-arm procedure) and robot/robot interaction (two-arm procedure) in the assembly process.

The motor assembly sequence for both the one-arm and two-arm procedures was identical. The same assembly fixture was also utilized for both single-arm and dual-arm procedures. Additional tooling was required for dual-arm assembly as well as additional communications interface hardware. These features are discussed further in later sections. The major difference between the single-arm and dual-arm techniques was the interaction of a human worker with the single robot arm versus almost no human interaction with the dual robot configuration.

In the single-arm experiment, the robot's task was to stack all large diameter motor components while the human laborer performed smaller component assembly tasks. This required the human to interact with the robot control program in a manner which would not pose a danger to the worker while attempting to perform an assembly task. This type of interaction with the robot control program was not required for the dual-arm procedure however, since the entire assembly process was performed by the robots. The specific procedures for manual, one-arm, and two-arm assembly of the motor are detailed later in this chapter.

Equipment Utilized

The equipment which was used for the study represents the current state-of-the-art in robotics development. Two UNIMATE "Puma" Model 762 Series robots were utilized, each with a respective controller and



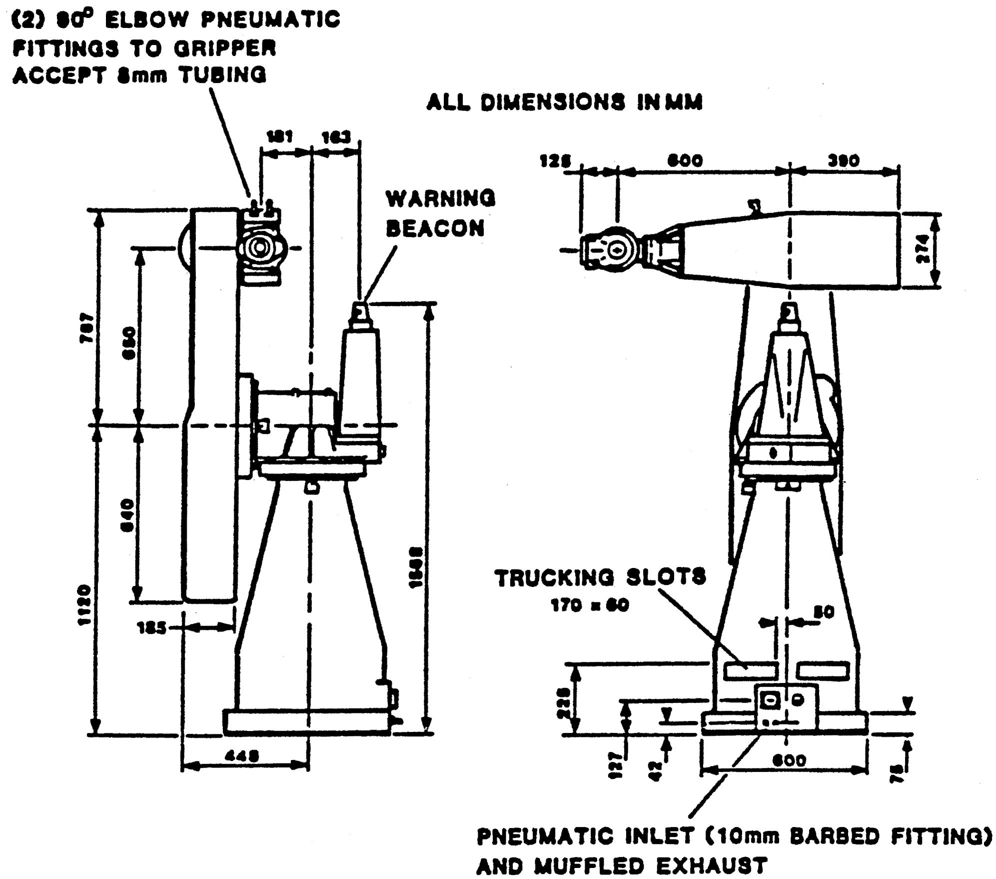
Source: Unimation Industrial Robot User's Guide to VAL II. Unimation, Inc. (1985), p. 1-0

Figure 7. Unimate PUMA 700 Series, Mark III-VAL II System

visual display monitor/disk drive unit (refer to Figure 7). The robot configuration was of "jointed-arm" type with six degrees of freedom, which utilized electric direct-current servomotors to drive each joint of the arm. The gripper on each robot was pneumatically activated, providing simple "open and shut" operation. Each robot controller also included a teach pendant which was used to position the robot without operator interaction through the terminal. This provided assistance when critical positioning of the robot was needed near the operation to be performed.

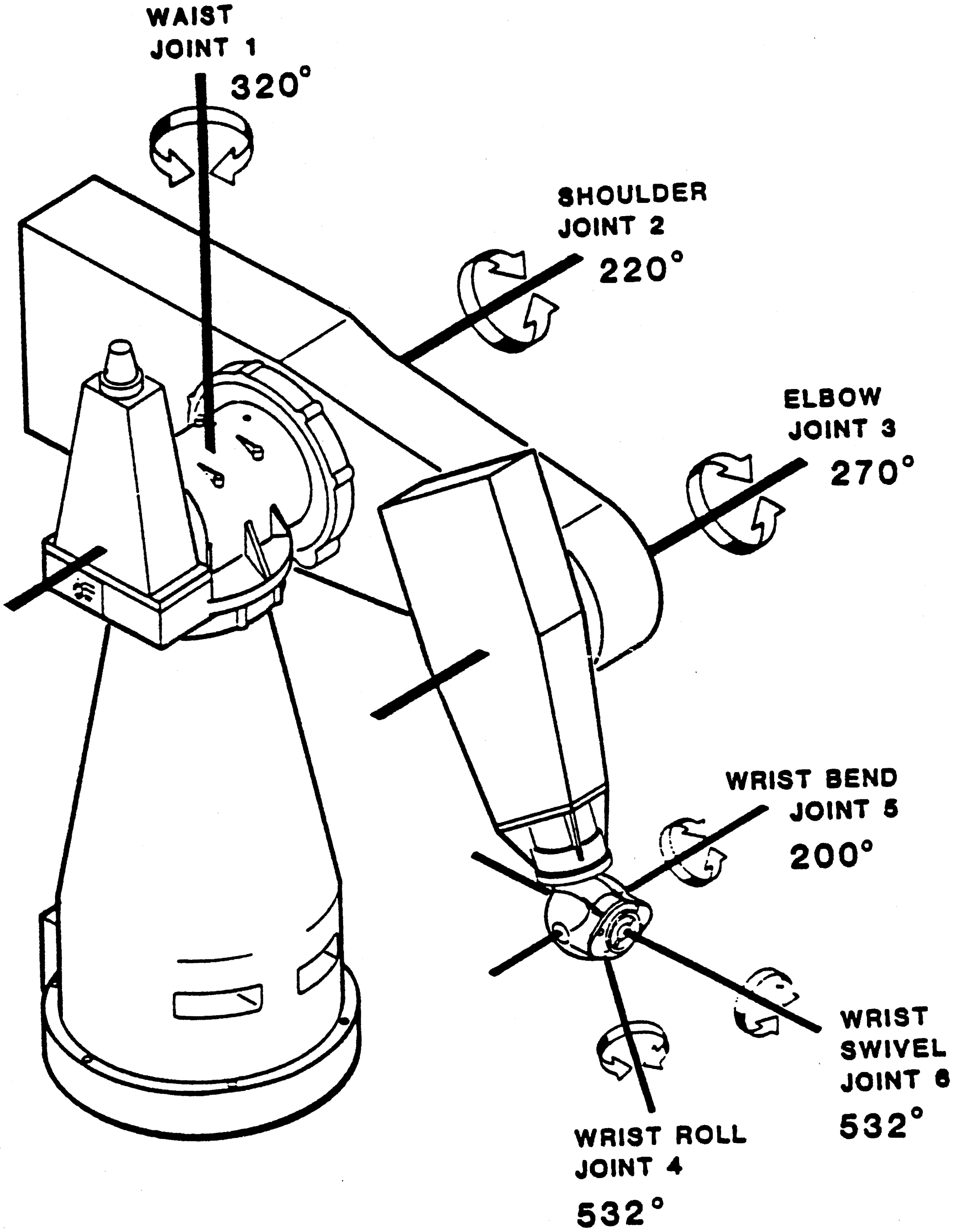
Figure 8 illustrates the dimensions of the UNIMATE PUMA 762 Series robot. The robot arm payload capacity of 44 pounds (including the weight of the end effector) was more than adequate for the experiment considering each motor weighed approximately five pounds. The PUMA robot arm joint angles and ranges of joint rotation can be seen in Figure 9. Extreme amounts of rotation for particular axes (joint 6, for example) result in twisting of the pneumatic hoses around the forearm and wrist of the robot arm. In order to prevent fouling from this occurrence, the air hoses were bundled (tie-wrapped).

In addition to the robot arms and their associated control equipment, a small thermal printer manufactured by Texas Instruments (Model 710 Portable) was used to obtain hardcopy printouts of programs, disk directory listings, point location files, and other desired information from the robot controller. Other equipment utilized during the study included small electric hand tools (drill, sabre saw, etc.). These tools were used mainly to produce the fixture devices necessary for the experiments. A 12-Volt direct current power supply was also used to supply power to switching relays located inside each robot controller.



Source: Unimation Industrial Robot User's Guide to VAL II. Unimation, Inc. (1985), p. 1-1

Figure 8. Dimensions of Robot Arm



Source: Unimation Industrial Robot User's Guide to VAL II. Unimation, Inc. (1985), p. 1-24

Figure 9. Robot Arm Joint Axes and Ranges of Rotation

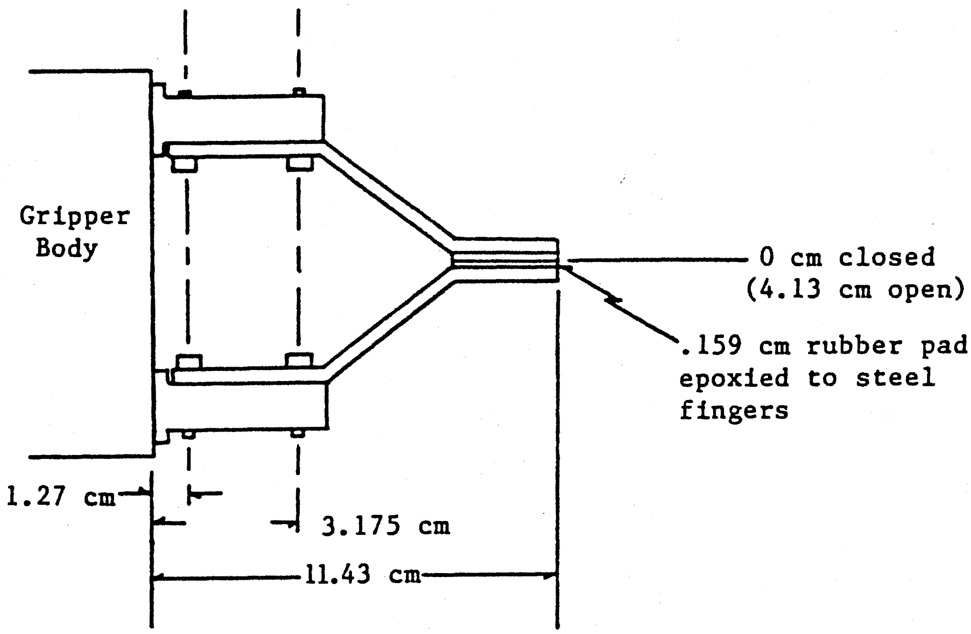
The relays provided each controller with the ability to send and receive external signals in order to provide coordination between each robot arm.

Hardware Development

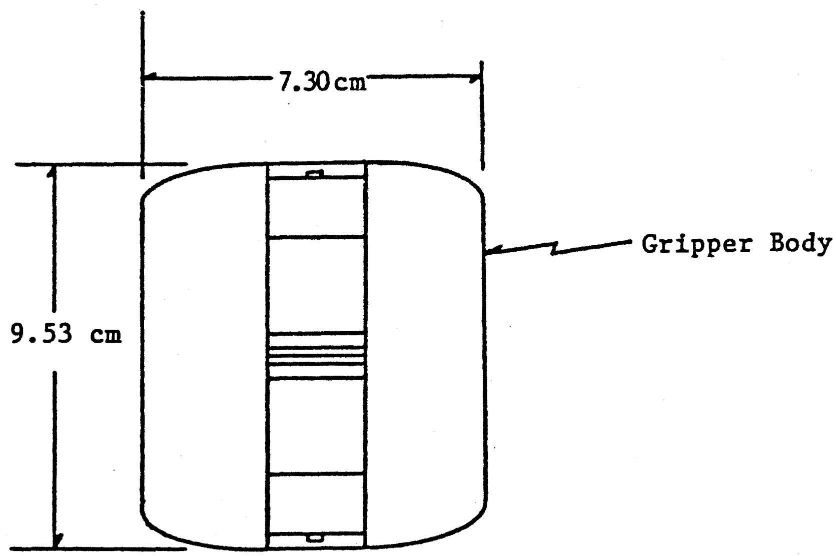
The hardware which was developed for the study included: (a) an end effector designed for use with small motor components, (b) an end effector designed for use with large motor components, (c) a cable harness specifically designed to carry external input/output signals between robot controllers, (d) an assembly fixture which was designed for placement of the motor components prior to assembly as well as placement of the motor during the final assembly process, and (e) various special fixture devices utilized for particular assembly operations. Each of these developments involved fabrication using one or more materials such as metal, wood, and rubber.

Both grippers were fabricated using 1.90 cm. wide x .635 cm. thick steel bar stock first cut to length and then bent into the desired shape. Next, two .635 cm. holes were drilled into each gripper side in order to mount the "finger" to the pneumatic actuator. The steel stock was then bent to the desired shape using a press brake. After the appropriate angle had been set, a .159 cm. rubber pad was cut to fit and epoxied into place at the end of the fingers to provide an increased friction factor when gripping an object. Figure 10 illustrates the end effector for small motor components. The fingers open slightly over 4 cm. to accept the shaft assembly, the bolts, and additional small tooling.

The end effector constructed to handle larger motor components



Side View

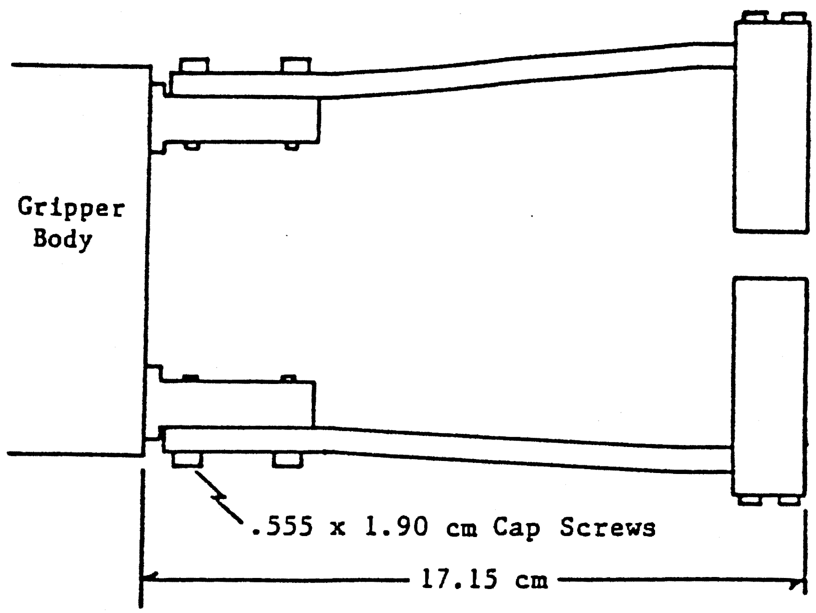


End View

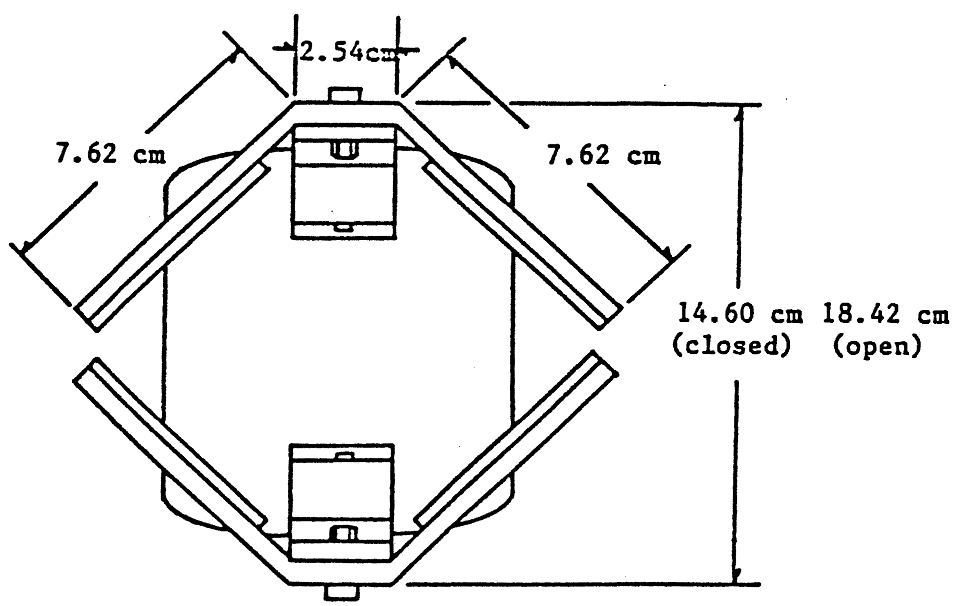
Figure 10. End Effector for Small Motor Components

(front and end bells, stator assembly) involved a slightly more complex design. Due to the round part geometry of the end bells and stator, it was necessary for the gripper to be "self-centering". In other words, when the gripper closed around the part, it would center the part with respect to the center lines of the gripper. This action ensured correct alignment each time the part was gripped. This self-centering action was achieved by the attachment of a 90°-angled steel extension attached to each gripper finger. Figure 11 illustrates the gripper design clearly showing the angled extension on each finger upon which the thin rubber pads were epoxied into location.

In addition to the gripper development, a cable harness designed to carry external input and output signals between robot controllers was fabricated. Special signal connectors supplied with the robot system were connected by standard 3/4" (19 cm.) diameter electrical metal conduit. Four 15-foot lengths (4.57 m.) of 22-gauge four-conductor cable were threaded through the conduit, and individual conductors were soldered to the appropriate input/output pin according to instructions provided by the robot manufacturer's equipment manual. Figure 12 illustrates the particular pin designations for the external signal connector number J147. Of the sixteen individual conductors utilized in the harness, seven wires were dedicated to input and seven wires were dedicated to output for each controller. The two remaining wires were utilized to conduct current between the 12-Volt power supply and each controller. A separate power source was required due to the fact that the controllers did not have an internal power supply to activate the signal relays. After completion and installation of the external input/output harness, both of the robot controllers were provided with



Side View



End View

Figure 11. End Effector for Large Motor Components

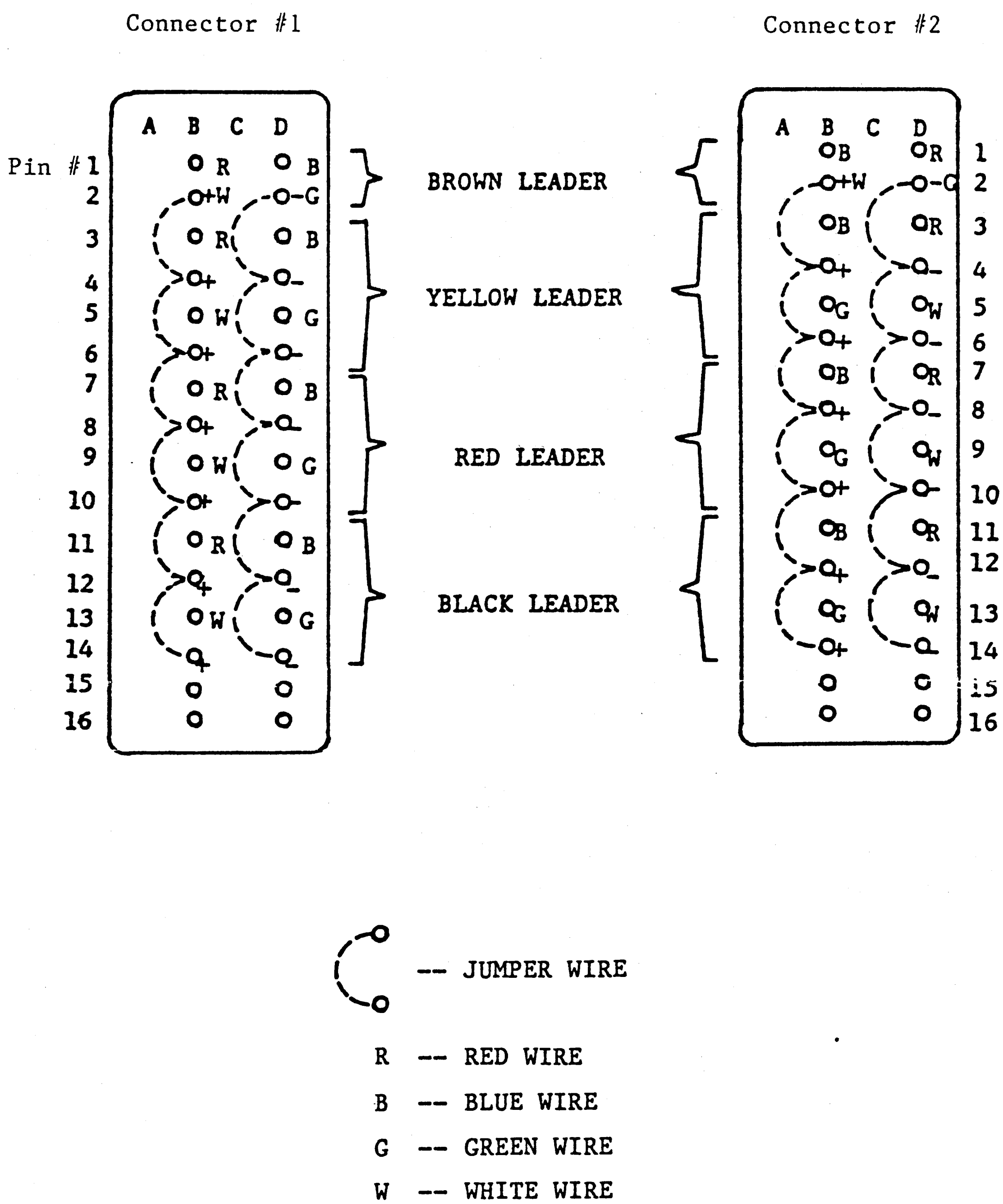


Figure 12. Pin Designations for External Signal Connector J147

seven separate channels with which to send communication signals. With this capability, both robots could effectively communicate their respective positions and thus prevent interference or collisions between the arms.

The final hardware component constructed was the part location fixture used for both the one-arm and dual-arm assembly procedures. The fixture was intended to provide a specific location and correct orientation for each motor component prior to assembly as well as provide a specific location and orientation for assembly of the motor. Constructed from soft pine and plywood, the overall box shape with removable top surface provided modifications to be made quickly and easily. Figures 13 and 14 show the assembly fixture without the motor components in place. As shown in these figures, the individual motor components were located around the periphery of the fixture, with the front bell, stator, and end bell correctly oriented using wood dowel locator pins which prevented rotation when these parts were grasped by the robot. This was very critical to ensure final alignment of the bolt holes, both in the end bells and stator, such that the bolts would be correctly aligned through the motor. The remainder of the motor components did not require such critical orientation; however, their position on the fixture still required careful attention. The shaft assembly as well as the four bolts which secured the entire final assembly were positioned using appropriate diameter holes drilled into the fixture. The four nuts used to secure the bolts were each positioned on a small length of brass rod which was supported by a small wood block. Each nut rested against a spring secured to the wood block, preventing the spring from slipping off the brass rod. This configuration allowed each nut to be

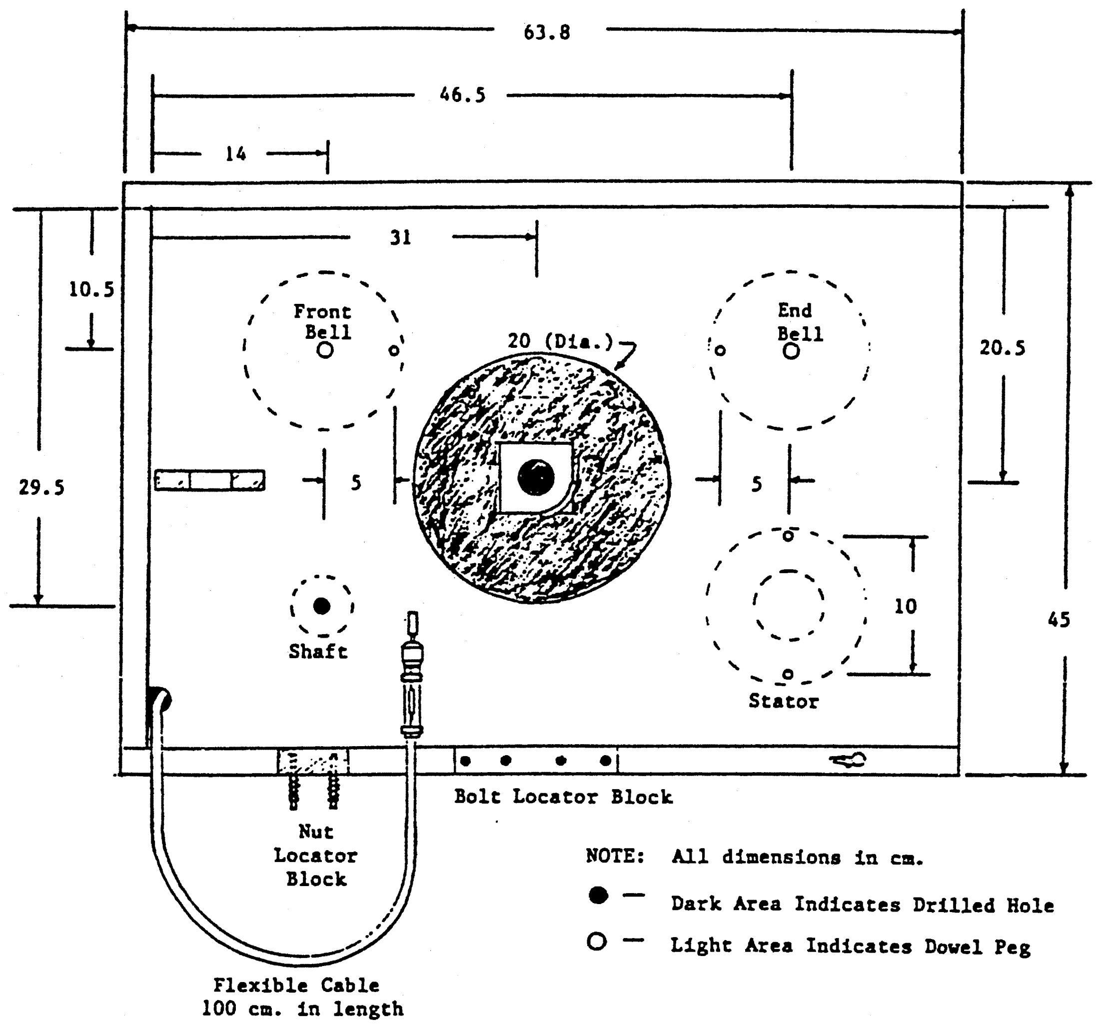


Figure 13. Assembly Fixture Top View

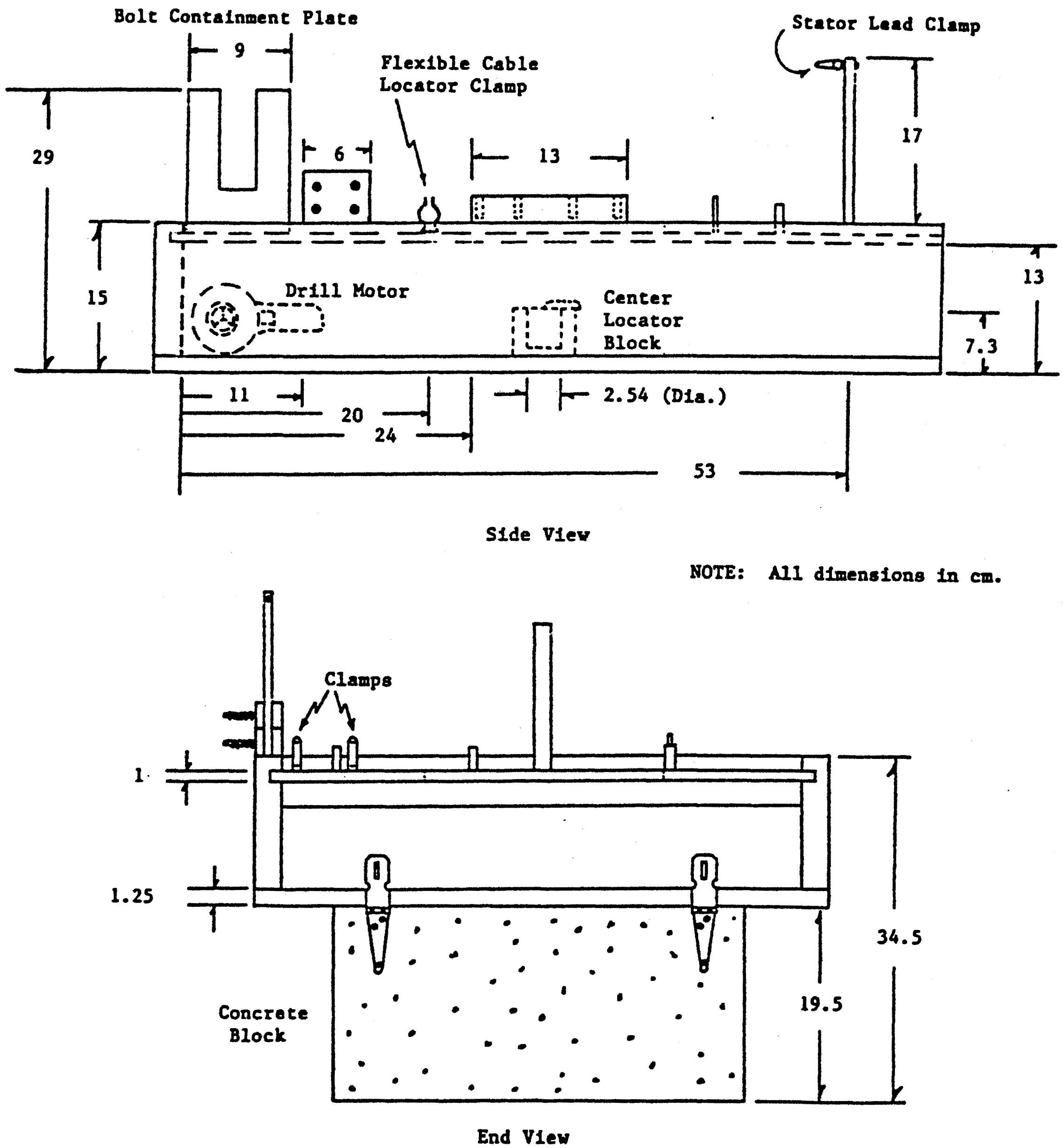


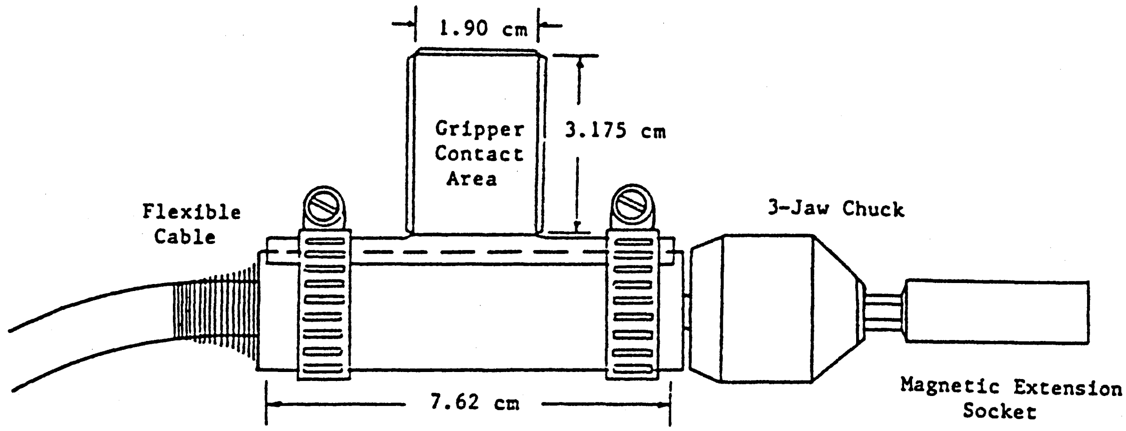
Figure 14. Assembly Fixture Side and End View

pressed into a special magnetized nut driver and subsequently withdrawn to be driven onto the bolt.

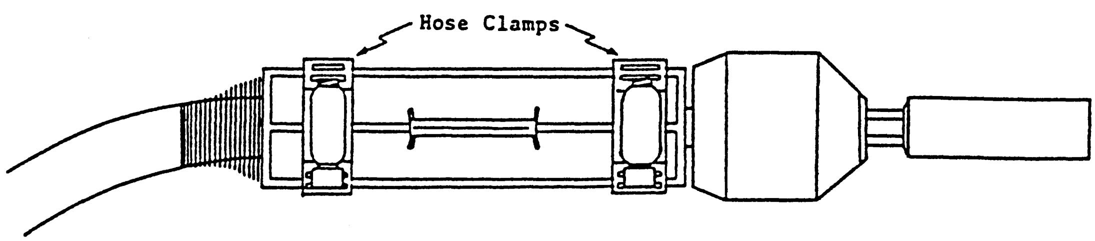
The center of the assembly fixture located the components as they were stacked together to form the final product. A hole in the top surface of the fixture was cut out large enough for placement and removal of the large diameter parts by the robot. A small wood block with a specially shaped locator pin was positioned at the bottom of the fixture to locate the front bell (for the beginning of the assembly sequence).

There are three additional features on the assembly fixture which do not involve location of the major components, but instead are necessary for performance of the assembly operation. First, a clamping device was necessary to secure the electrical leads attached to the stator assembly to prevent the leads becoming tangled when the stator was clamped by the gripper. The clamp device consisted of a small alligator clip with the serrated jaws filed smooth and mounted upon an 18 cm high wood dowel. The dowel was then mounted upon the side of the fixture (refer to Figure 14).

Second, a special fixture fabricated from aluminum was attached to the end of a universal flexible cable to serve as a grasp location for the robot gripper, since the robot was unable to adequately grip the cable itself. This special fixture was attached to the flex cable with two hose clamps which enabled the robot to grip the flex cable for the required manipulation (for an explanation of the required operation, refer to the section concerning "Assembly Procedure Utilizing Two Robot Arms"). Two small spring steel clamps were installed on the top surface of the assembly fixture to secure the flex cable when not in use.



Side View



Top View

Figure 15. Special Fixture Providing Grasp Area on Flexible Cable

Figure 15 shows the fixture attached to the end of the flex cable along with the custom-fabricated magnetic extension socket used in threading nuts onto the motor bolts. The socket was constructed from a standard 5/16" (.794 cm.) magnetic nut driver welded with a 5/16" (.794 cm.) deep well socket cut to the required length. The resulting extension socket was held in place on the flex cable via a standard 3-jaw chuck at the end of the cable. The cable shaft was rotated from the opposite end by a standard variable speed hand drill mounted inside the assembly fixture. Operation of the drill/cable system was controlled by a simple on/off toggle switch placed in line between the 110-Volt power source and the drill motor.

Third, a special "U" shaped plate was cut from plywood and mounted on the top of the assembly fixture to facilitate threading nuts onto the motor bolts without the bolts slipping out of the motor in the process. The completed body of the motor, with bolts inserted, was backed against the U-shaped plate; thus, the plate prevented the bolts from slipping out of the motor when pressure was applied from the threading operation. In order to more fully understand this particular aspect of the fixture, the reader should refer to the section on "Assembly Procedure Utilizing Two Robot Arms".

Figures 16 and 17 illustrate the assembly fixture with the addition of the motor components shown in their respective initial positions prior to the start of the assembly sequence. It can be seen from the figure that all of the large components were symmetrically placed around the assembly location, with the smaller components located closest to the robot to which the small component gripper was mounted. It can also be seen from the figure the detail of the electrical lead clamp device

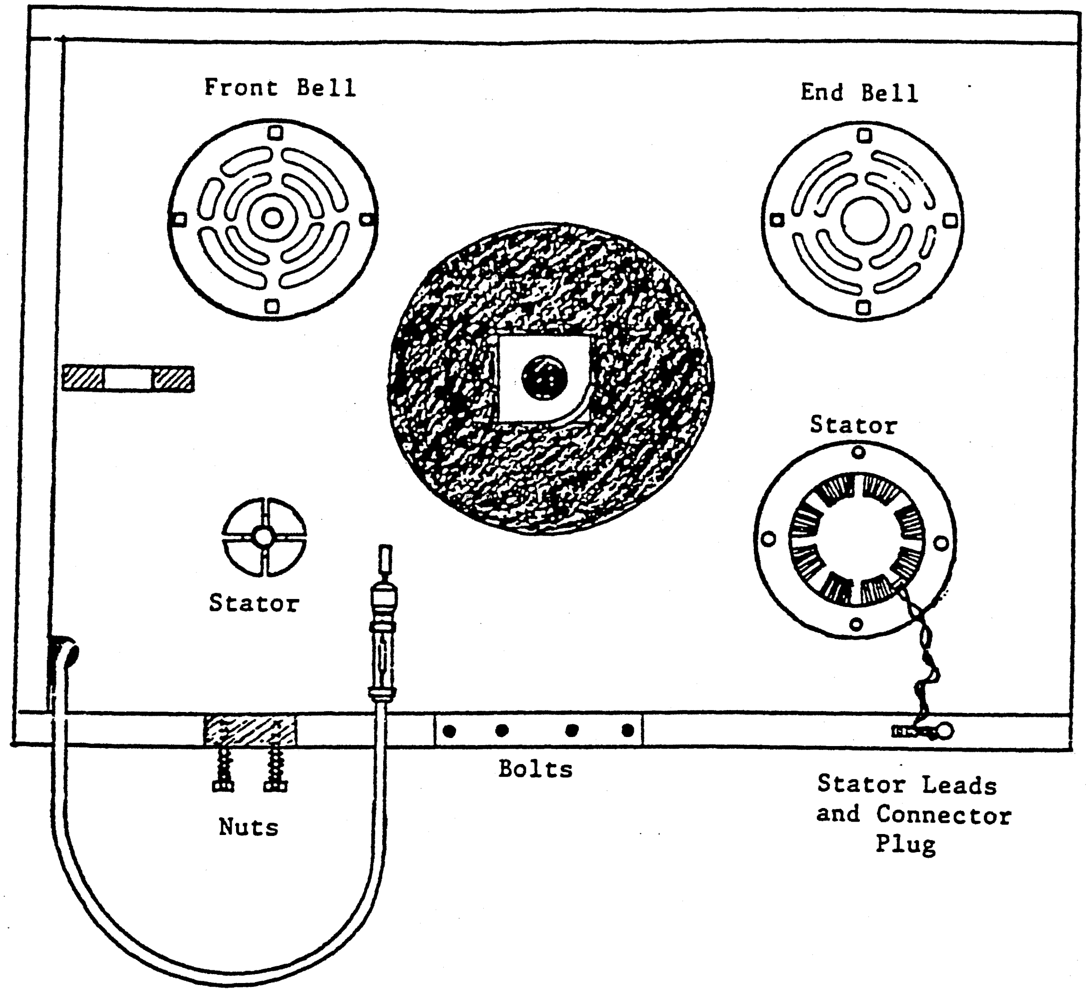
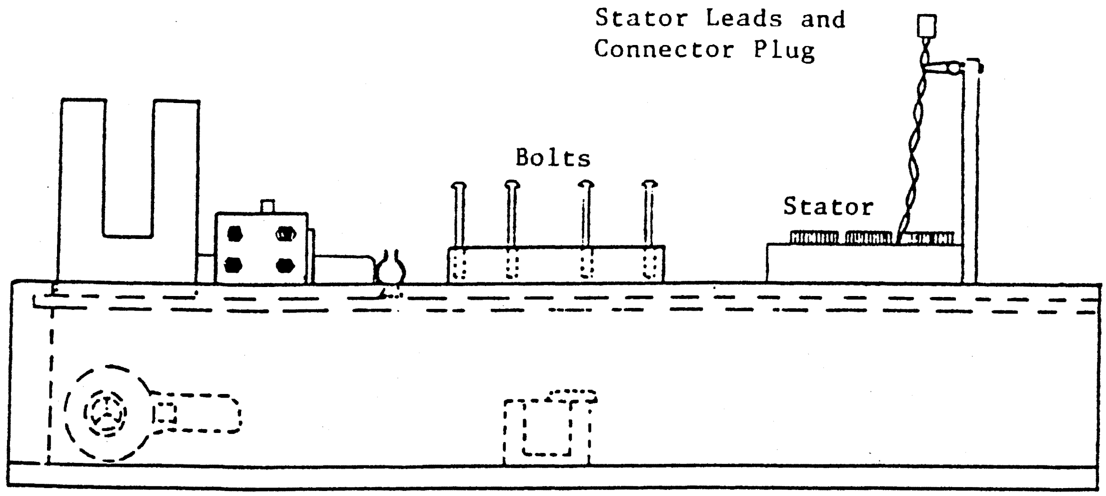
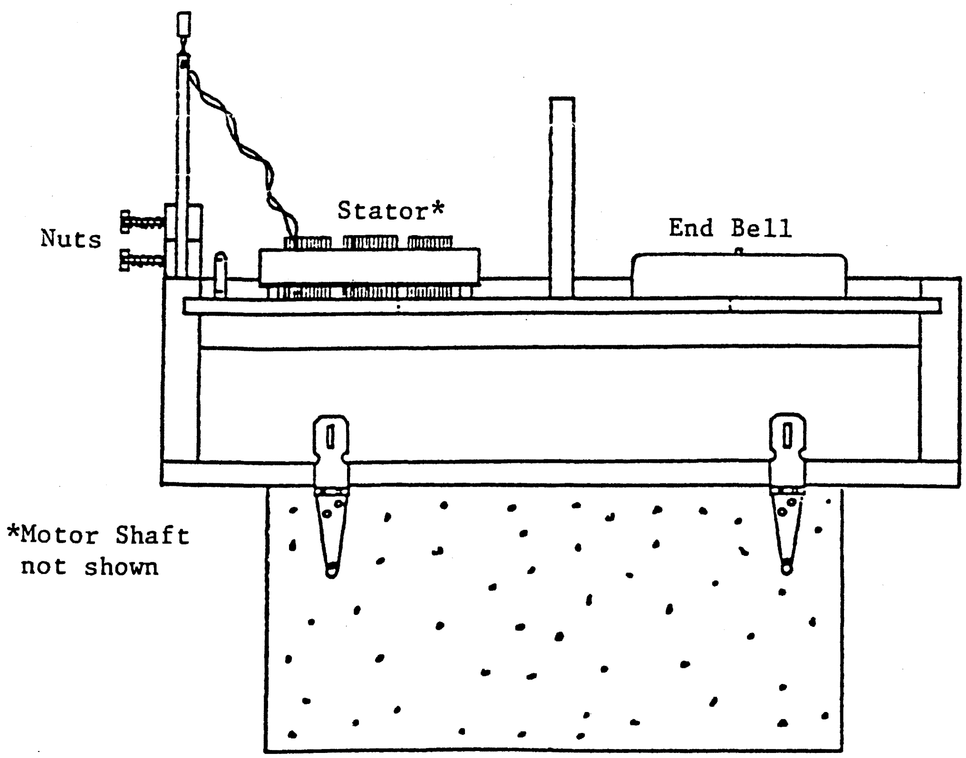


Figure 16. Assembly Fixture with Placement of Motor Components (Top View)



Side View

NOTE: Flexible cable removed for clarity



End View

Figure 17. Assembly Fixture with Placement of Motor Components (Side and End Views)

as well as the positioning of the motor bolts and nuts. In addition, the fixture was mounted upon three 20 cm. x 20 cm. x 40 cm. concrete blocks using metal hasps (see Figure 14). This was done for two reasons: (1) to elevate the fixture into a larger area in the robots' workspace, thereby providing greater ease of extensions by each robot arm, and (2) to securely anchor the assembly fixture, thereby minimizing the possibility of small displacements in the location of the fixture resulting from inadvertent forceful contact with the fixture. The hasps attached to the concrete blocks and the wooden assembly fixture enabled the fixture to be removed from the blocks if necessary for ease of transport.

Software Development

The software developed for the experiment consisted of computer programs generated for both the one-arm and two-arm assembly procedures. The programming language utilized for the assembly routines was VAL II version 1.4B, furnished by Unimation for use with the PUMA 762 Series robots (11). The VAL II robot control language was designed specifically for use with Unimation industrial robots and incorporates high level English-type commands to direct robot motion.

In order to arrive at a complete and comprehensive coding of the assembly programs, knowledge of the entire assembly procedure for each method was required. Therefore, each procedure was fully developed before addressing the problem of program operation. The first step in the development of each program was to compose a flowchart which provided a logical directive of program execution. The flowcharts provided a framework from which coding of the programs was subsequently accomplished. After the coding process was completed, the code was entered into the robot controller memory through the system terminal.

The next step in the process was to debug the programs to ensure correct operation with regard to program procedures, functions, control structure, and format of desired output parameters. At this stage of program development, only the essential operating structure of the robot control program and its associated real variables were completed; the location variables or "points" had yet to be "taught". Only when the necessary tooling and fixtures had been fabricated and had satisfactorily passed preliminary testing were the location variables entered into the programs' location file.

In order to enter the required location variables into the location file which, in turn, would enable the robot to move to these locations (or points), a feature of the controller known as a "teach pendant" was utilized. For example, the desired point in the assembly sequence of the program was entered into the location file by first manually moving the robot arm to the desire position and then pressing the "record" button on the teach pendant. This action stored the desired points sequentially in the file so that the points would comprise the destinations of the robot arm for assembly of the motor.

After the location file for all points had been entered into the robot controller, the final task was to debug any remaining flaws in each program and then test each program for correct operation in the assembly operation. For the one-arm assembly procedure, only one program was required for the process. In the two-arm procedure, however, two separate programs were required; one program acted as a "master" (primary), while the other was delegated as a "slave" (secondary). The primary program contained the majority of user messages and prompts in order to spare the robot operator unnecessary movements between controllers while responding to program messages. Both programs in the dual-arm routine were coordinated by use of the external input/output signal channels described previously.

A program description which details the operation for the single-arm assembly procedure can be found in Appendix C. The flowchart for the routine is shown in Appendix D, and the program listing for the one-arm procedure is given in Appendix E. A program description for the dual-arm primary and secondary routines can be found in Appendix F. A flowchart for each of the routines is given in Appendix G, and the programs are listed in Appendix H. A complete description of the method by which all these programs were integrated into the overall assembly process is provided in the last two sections of this chapter.

Manual Assembly of the Motor

Assembly of the electric motor using only human labor was approached by utilizing the principles of classical time and motion studies (12). First, the assembly function was analyzed and divided into task elements. Breaking the entire assembly process into elements provided a detailed analysis of the assembly motions which could then be compared directly with results obtained using robotic assembly. The task elements selected were based upon the logical order of assembly (refer to Figure 6 on page 6). An operations process diagram which illustrates the order of the assembly tasks is shown in Figure 18.

After the assembly tasks were identified, the next step was to design a workstation which would provide a suitable location for the

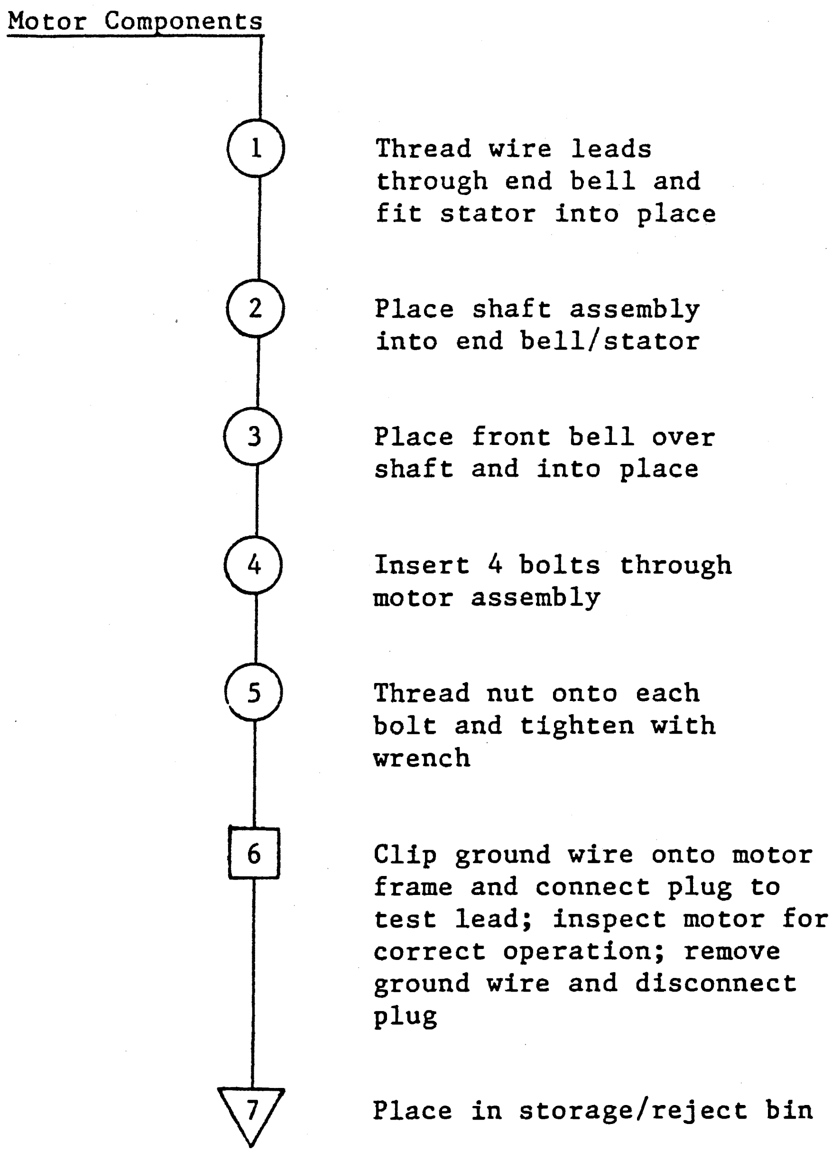


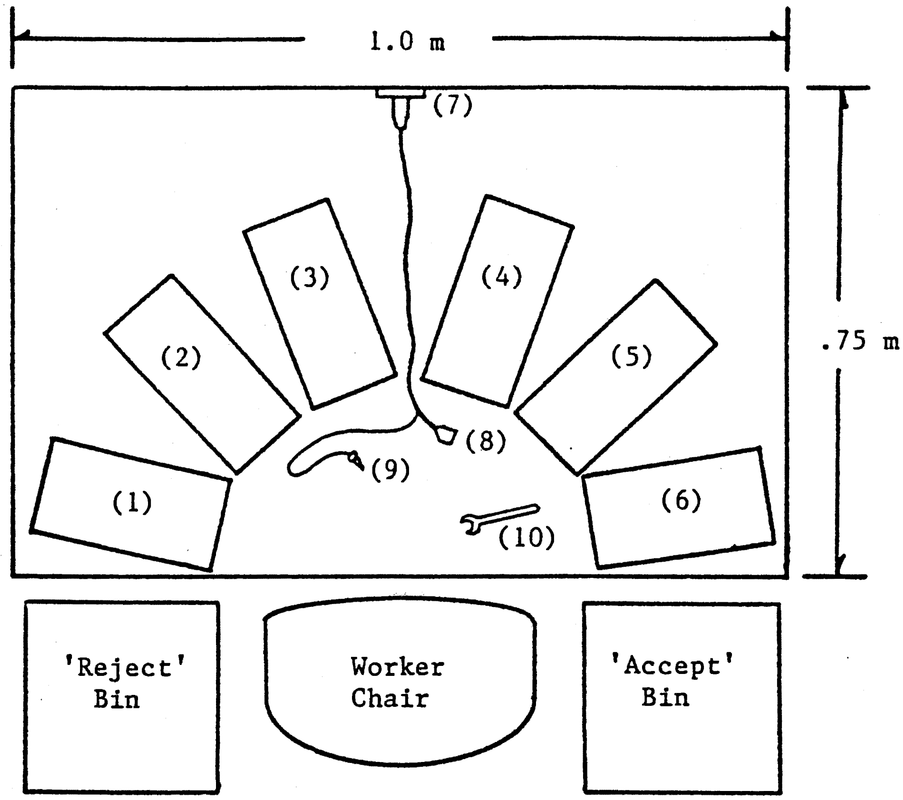
Figure 18. Operation Process Diagram for Manual Assembly

final assembly. The principles of workstation design were utilized in this respect by providing a worktable of adequate size and height, an adjustable chair for the worker, and individual motor parts located in separate bins providing easy identification of components. The part bins were arranged in a semi-circular fashion and using a sequential order matching the task order of the operations process diagram. This procedure provided a consistent motion which increased the efficiency of the operation. Figure 19 illustrates the layout of the workstation and provides a description of each item shown in the diagram.

After the manual assembly procedure had been defined and the layout of the workstation had been completed, the remaining step involved the actual timing of the assembly operation to obtain the component task times as well as the overall assembly time. A data collection form was prepared for the time study and the form was used to record the stop-watch measurements during the procedure. Spaces were also provided in the form for recording the average component times, average total assembly time, worker effort rating, and other pertinent information.

The assembly process began with the worker picking up the end bell in the left hand while the right hand picked up the stator assembly and placed the stator at a convenient location on the table. The time clock was started at the point of first hand motion. The first element ended when the stator power wires were threaded through the end bell grommet and the end bell was press fitted onto the stator.

The second element involved the left hand holding the end bell while the right hand grasped and placed the motor shaft into the appropriate bushing in the end bell. The third task element began with hand motion towards the bin which held the front bell components and ended



Description:

(1) - End Bell
(2) - Stator
(3) - Shaft
(4) - Front Bell
(5) - Bolts
(6) - Nuts
(7) - A/C Wall Outlet
(8) - A/C Test Lead
(9) - Ground Lead
(10) - Wrench

Figure 19. Layout of Manual Assembly Workstation

with the front bell placement over the shaft and press fitted on the stator. This step completed the major component assembly of the motor.

The fourth task element involved the insertion of the four 9.55 cm. long bolts through the end bell, stator, and front bell. To accomplish this task, the alignment of the major motor components (front/end bell and stator) was critical. At most, a cumulative tolerance of ±1 mm. was permitted for major components with regard to bolt-hole alignment.

Following bolt insertion, the fifth element involved threading a .313 cm. hexagonal nut onto each bolt and subsequently tightening each nut using a small open-end wrench. This action completed the assembly of the electric motor.

The final two task elements involved motor inspection and placement of the motor into the appropriate bin. Inspection of the motor for correct operation first involved connecting a ground lead to the motor frame, and then connecting the power leads of the motor to the A/C test lead. If the motor's shaft rotated counterclockwise when power was supplied, the motor was accepted and placed into the "accept" bin. If the motor did not perform in the described manner (i.e., did not rotate correctly, or rotate at all), the motor was placed into the "reject" bin for subsequent rework at a later period. After the worker had placed the completed motor into the appropriate bin, the entire manual assembly process was completed and assembly of a new motor was begun. In order to obtain a reasonable measure of both the element times and the total assembly time for the operation, a total of twenty observations were observed and recorded. The results of the manual assembly are discussed in the next chapter.

Robotic Assembly of the Motor

Assembly of the electric motor using single-arm and dual-arm robot configurations was approached in three main development phases. The first phase involved an analysis of the assembly sequence for both routines. The assembly sequence developed for robotic assembly of the motor could not utilize the identical sequence of operations developed for the manual assembly method due to the limitations imposed by the gripper/fixture interface in the assembly process. The limited dexterity of the robotic gripper arrangement introduced constraints upon the sequence of assembly.

For example, the decision to assemble the motor beginning with the front bell placed first followed by the stator, shaft, and end bell was constrained by the placement of the bolts into the completed main body of the motor. The bolts had to be inserted through the rear of the motor in order to exit out through the front bell, while at the same time the end bell, stator, and front bell positioning had to be correctly maintained in order to perform the operation. After careful consideration of alternative component sequencing to achieve correct bolt placement, the particular assembly sequence described above was selected for use in both the one-arm and dual-arm assembly procedures. The assembly procedures for both robot configurations are described in detail in the following two sections.

The second development phase involved the integration of the required hardware and fixtures into the assembly process. Based upon the assembly sequence and part geometrics, the necessary fixtures, hardware, and tooling were constructed and arranged in the work area. Since the hardware and tooling development has been previously discussed, the focus here shall be placed upon the integration of the hardware into the total assembly system.

Figure 20 illustrates a block diagram of the robotic assembly system. The diagram shows the physical relationship between each robot arm and its respective controller as well as the physical relationship between the robot controllers via the external signal communications lines. Control of the one-arm assembly routine was performed in the normal manner using one robot arm directed by its respective controller. Control of the two-arm assembly routine involved the exchange of signal communication between each robot controller to direct robot motion in a coordinated manner. The exchange of binary signals was directed by the VAL II robot control programs and the signals were transmitted through the external signal lines.

The arrangement of the assembly fixture was determined by consideration of the overlap between the work envelope of each robot. An overlap of approximately 60 cm. existed between the robot arms. The assembly fixture was located in the center of the overlap area, and was placed upon 20 cm. high concrete blocks to elevate the fixture to provide increased exposure in the work envelope. Maximum exposure in the combined work envelope was necessary so that all motor components could be reached and manipulated using straight-line motions by each robot arm.

The arrangement of other hardware was not as critical as the placement of the assembly fixture, however, the precaution was taken to place the additional hardware outside of the robot work envelope when possible. Figure 21 illustrates an overhead view of the robotic assembly area. The legend in the figure lists the hardware components used in

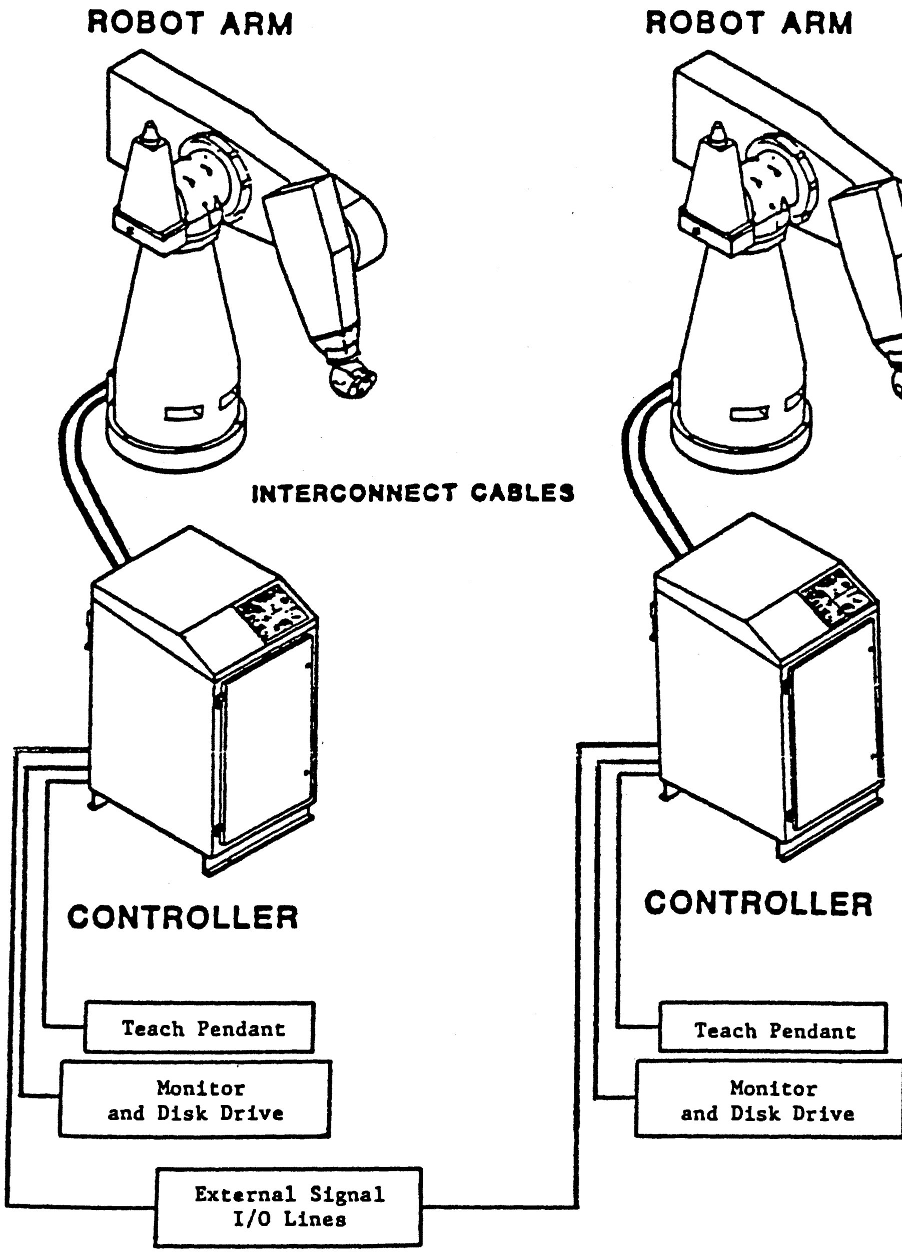
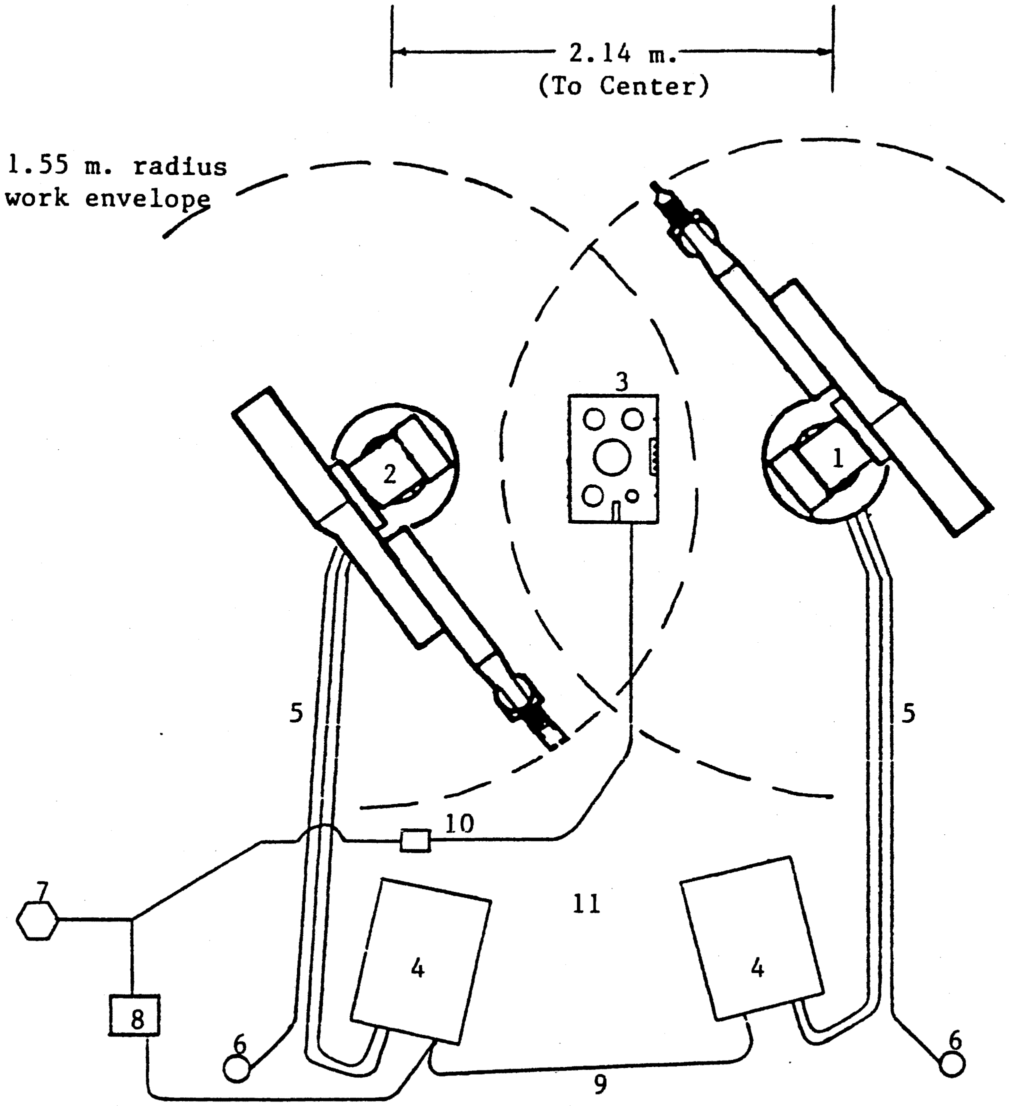


Figure 20. Block Diagram of Robotic Assembly System



Top View

Legend:

1. PUMA Robot #1 (PUMA.1)
2. PUMA Robot #2 (PUMA.2)
3. Assembly Fixture
4. Controller
5. Robot Power/Control Cables; Pneumatic Hose
6. Air Filter/Pressure Regulator/Lubricator
7. 110-Volt AC Outlet
8. 12-Volt DC Power Supply
9. External I/O Communications Cable
10. Toggle Switch for Control of Drill Motor in Fixture
11. Operator's Position

Figure 21. Physical Layout of Robotic Assembly Area

the experiment and the matching number on the diagram provides a reference to their location in the assembly cell. The Puma robot which handled the small motor components has been referred to as "PUMA.1", while the robot which handled the large motor components has been assigned the name "PUMA.2". This abbreviated form simplified the author's thought coordination in the creation of operations process diagrams, robot control programs, etc.

The third development phase involved the creation of both the single-arm and the dual-arm VAL II robot motion control programs to direct the assembly of the motor. Utilizing the information contained in the operations process diagram and with consideration of the assembly fixture dimensions, the robot control programs were produced. The location variables were subsequently taught utilizing the teach pendant, and the process of program debugging was accomplished.

The primary objective of the robot control programs was to provide the necessary instructions which would enable the robot(s) to accomplish successful assembly of the motor. A secondary objective was to obtain task element times and total assembly times which could then be compared to those task element times found in manual motor assembly. The task element times and total assembly times were obtained by the use of the "TIMER" function in the VAL II language. The use of this command enabled real-time motion data to be tracked throughout the program execution. The task data was summed at the end of each assembly cycle and provided a total arm movement time which was used in the analysis of the results. After the programs had been developed and were judged to be operating correctly, a "fine-tuning" process was initiated which attempted to decrease the total assembly time by increasing arm speeds, eliminating unnecessary arm movements, etc., until no further decrease could be achieved in the overall cycle time without sacrificing assembly quality. At this point, the assembly sequence was executed twenty times to obtain the same number of data sets achieved via manual assembly of the motor.

Assembly Procedure Utilizing One Robot Arm

The objective of the single-arm assembly routine was to complete final assembly of the motor using one robot arm in conjunction with a human worker to simulate a "production line" type of programmable assembly system. In other words, the product moves down an assembly "line" via conveyor, etc., and is assembled in sequential fashion by robots and human workers stationed along the line. In the actual experiment performed, the product remained stationery and was assembled with a single robot and a single human worker. This fact, however, did not detract from the usefulness of the data obtained from the one-arm assembly routine.

The assembly procedure developed for the single-arm routine is shown in the form of an operation process diagram in Figure 22. Prior to the start of the assembly process, several objectives were required: (a) all system components were switched on and judged to be functioning correctly, (b) all motor components were placed in their respective positions on the assembly fixture, (c) the VAL II operating system was placed in "RUN" mode to enable program execution, (d) the single-arm robot control program "ONEARM" was submitted for execution, and (e) all non-essential equipment and personnel were clear of the robot's work envelope. After these requirements were satisfied, the assembly se-

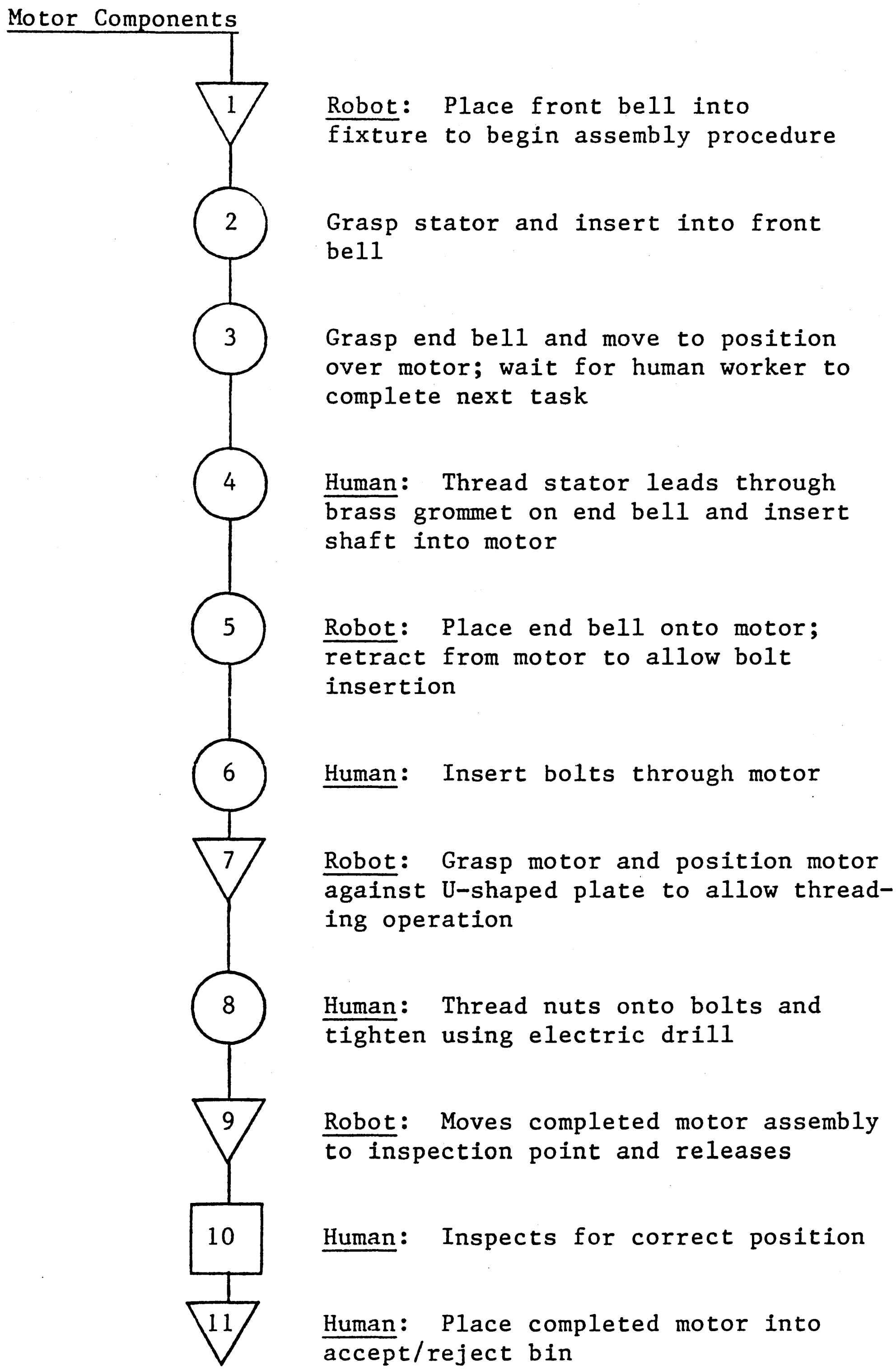


Figure 22. Operation Process Diagram (Single Robot Arm)

quence was initiated.

The assembly sequence began with the robot located at the sequence start point, designated "STRTP1" (refer to program listing, Appendix E). The robot arm is shown at this location in Figure 23.

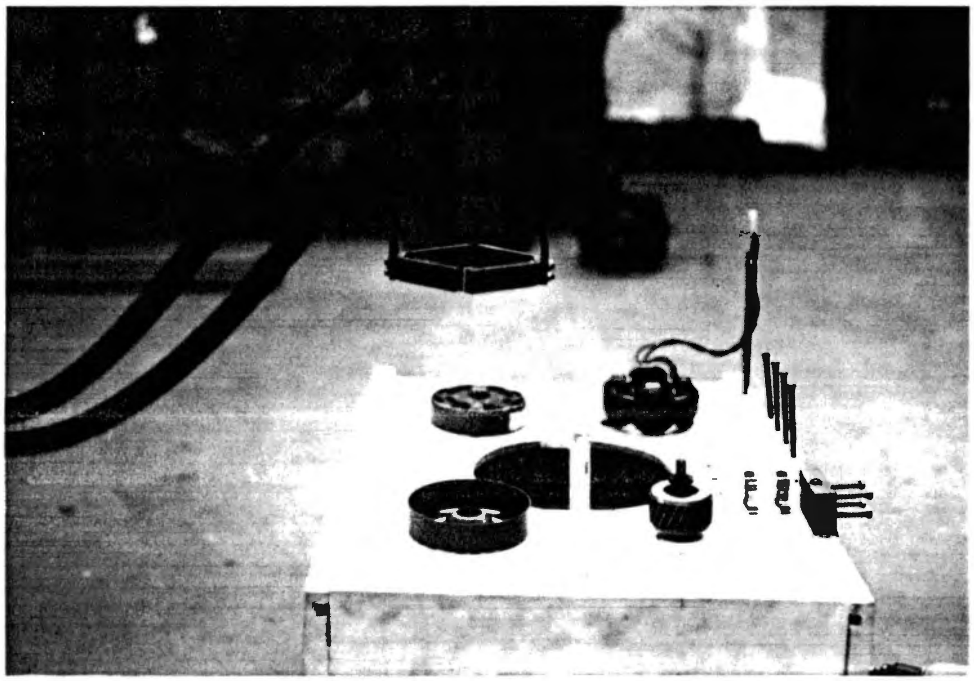


Figure 23. Robot at STRTP1 (Begin Sequence)

The first operation involved the placement of the front bell by the robot arm onto the assembly block set in the center of the assembly fixture (refer to Figure 22). This involved arm movement in a straight-line fashion from the start point down to the grasp point (P1), at which time the gripper closed around the front bell and the arm proceeded to place the front bell into position on the locator block at (P4). Figure 24 shows the placement of the front bell onto the block by the robot arm.

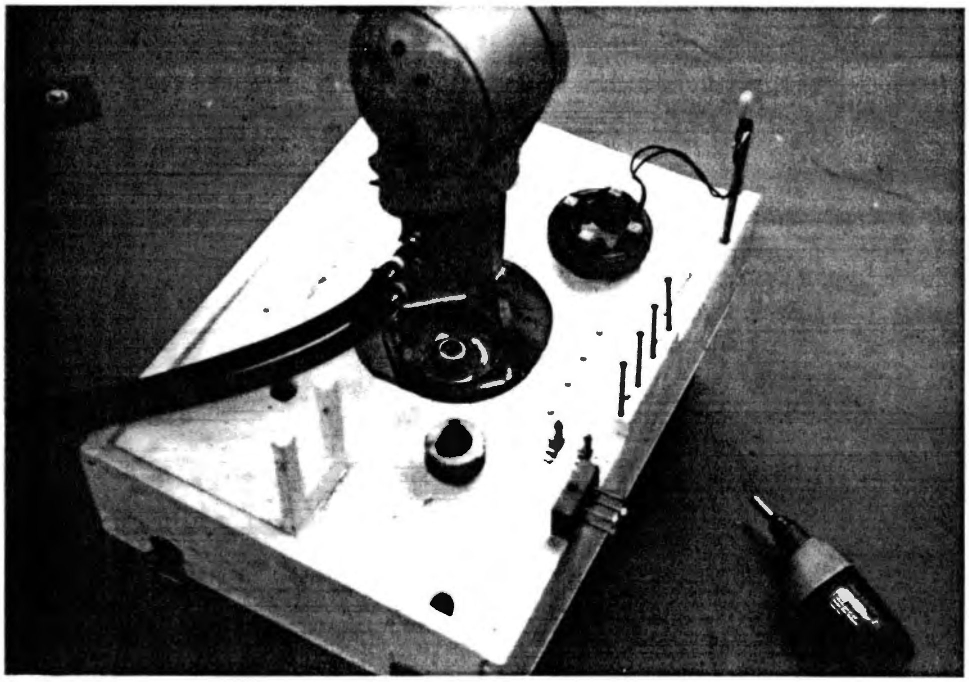


Figure 24. Robot Placing Front Bell onto Locator Block in Fixture

After placement of the front bell, the next objective for the robot arm was to place the stator assembly onto the front bell. The arm proceeded to move from the center locator block (P4) across the fixture to a point 3 cm. above the stator (P6) and then down to the grasp point (P7) where the end effector engaged the stator. Figure 25 illustrates the robot arm at the grasp point (P7) with the stator in the grasp of the end effector. The robot arm then proceeded to move into position above the end bell at which time the robot's speed was slowed down to allow for the delicate placement operation. Figure 26 shows the placement of the stator onto the front bell at the center locator block (P8).

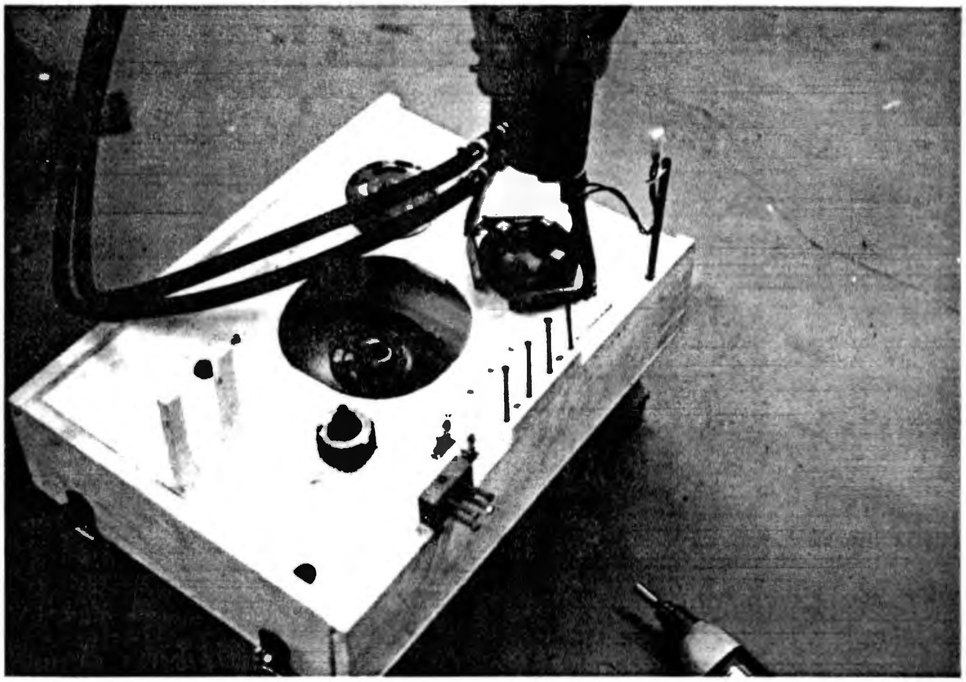


Figure 25. Robot Grasping Stator Assembly

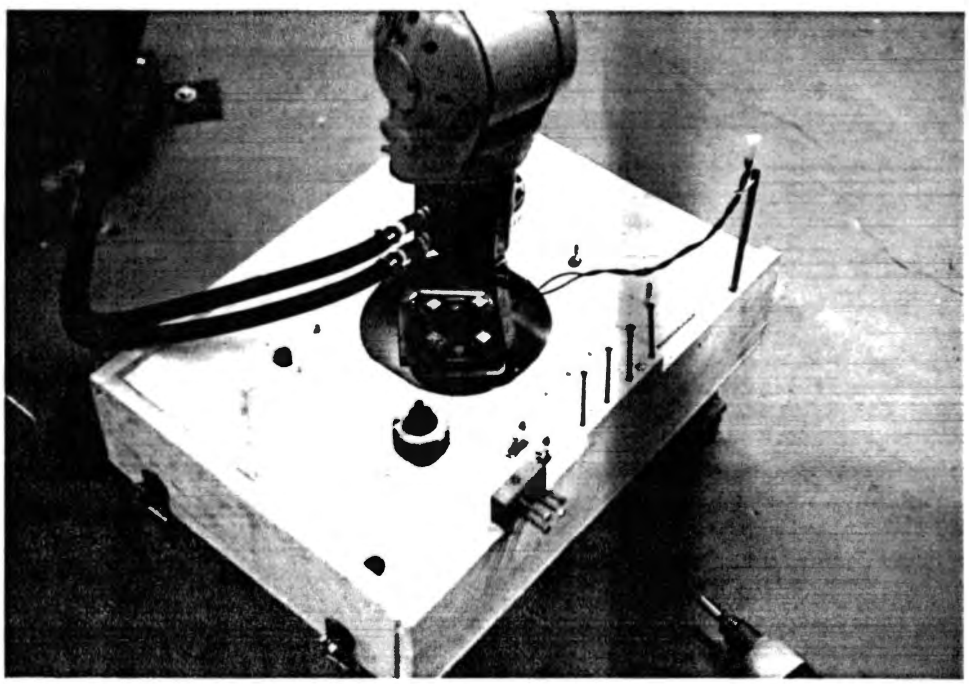


Figure 26. Robot Placing Stator onto Front Bell

After placement of the stator assembly, the speed of the arm was reset to the normal movement velocity and the arm proceeded to move on a course toward the end bell location (P9). From a point 3 cm. above the end bell (P9), a straight-line motion was executed to arrive at the grasp point (P10), where the end effector grasped the end bell and removed it to clear the locator pins. Figure 27 shows the end bell just as it was removed from the pins (P9) en route to a position over the stator assembly (P5).

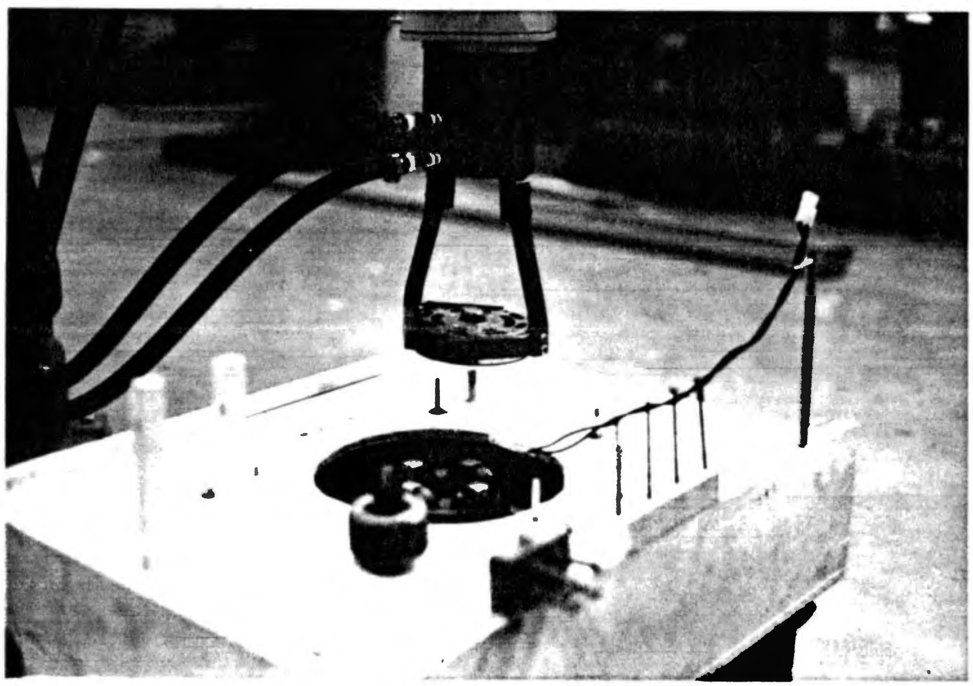


Figure 27. Robot Lifting End Bell from Fixture Locator Pins

Upon reaching the position above the stator (P5), program operation was suspended while task time statistics were gathered and at this time, the human operator was signalled via a message sent to the controller terminal to turn off arm power and perform the fourth assembly operation.

The operator proceeded to insert the motor shaft and thread the stator leads through the appropriate grommet in the end bell. Figure 28 illustrates the human worker performing this task. After completion, a task time was obtained, the operator returned to the controller, and program execution resumed.

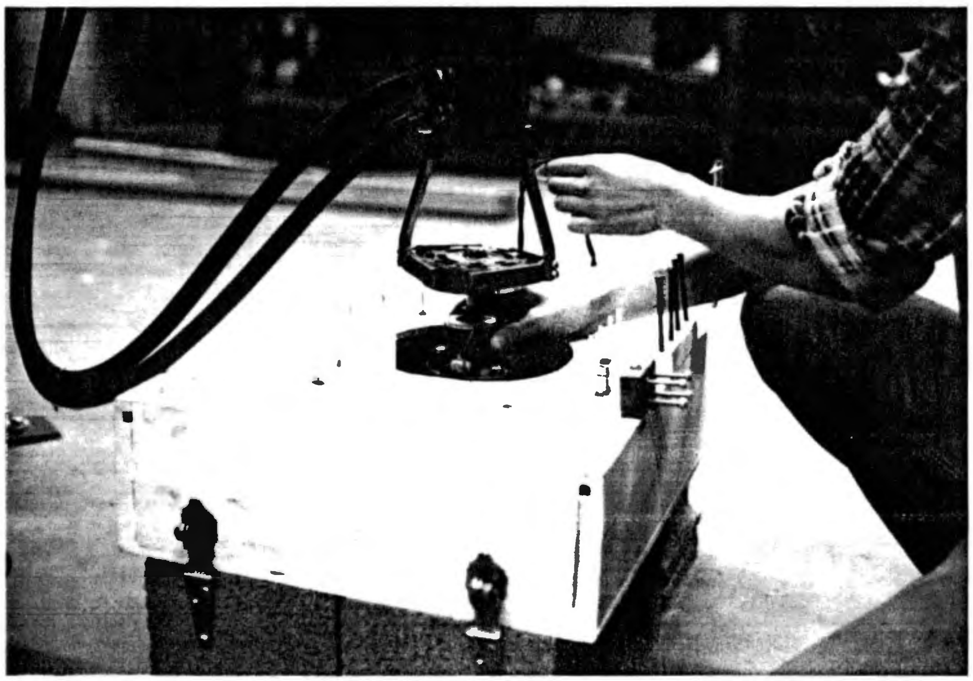


Figure 28. Human Worker Inserting Shaft Assembly and Threading Motor Leads through End Bell

The end bell was then moved into position by the robot arm, the arm speed reduced, and the end bell pressed onto the stator. This operation was the most difficult of any to accomplish due to the close part tolerances involved in both the shaft and end bell bushing. In addition to the close tolerances, another factor caused additional problems in the alignment process. It was discovered that upon application of force in particular areas of the end bell bushing, the bushing tended to misalign with respect to the motor shaft. This movement did not require a very large amount of force, and thus alignment problems occurred frequently until the assembly operator manually positioned the bushing to ensure that the bushing was correctly aligned prior to placement of the end bell by the robot arm.

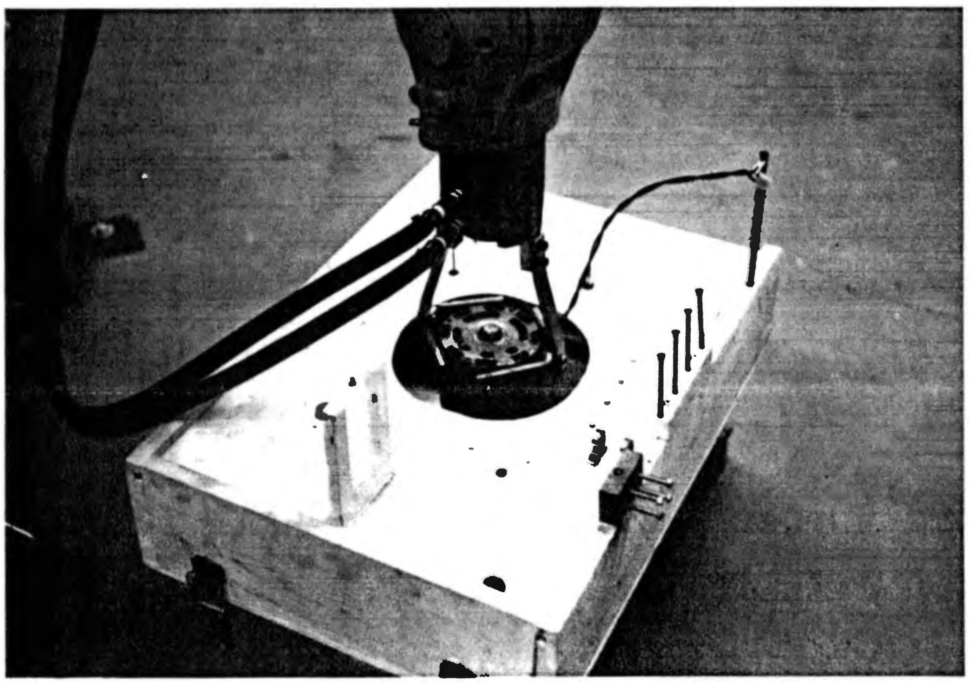


Figure 29. Robot Placing End Bell onto Shaft and Front Bell

Figure 29 shows the robot arm placing the end bell into position onto the shaft and stator. After completion of the task, the arm speed was reset to normal velocity and the wrist proceeded to rotate 45° to allow gripper clearance of the stator leads upon withdrawal from the location. The robot arm then retracted to a point 40 cm. above the motor (P13) and program execution halted to allow the human worker to perform the next task. At this point more time data was collected and the robot control program displayed instructions to the human worker on the system terminal.

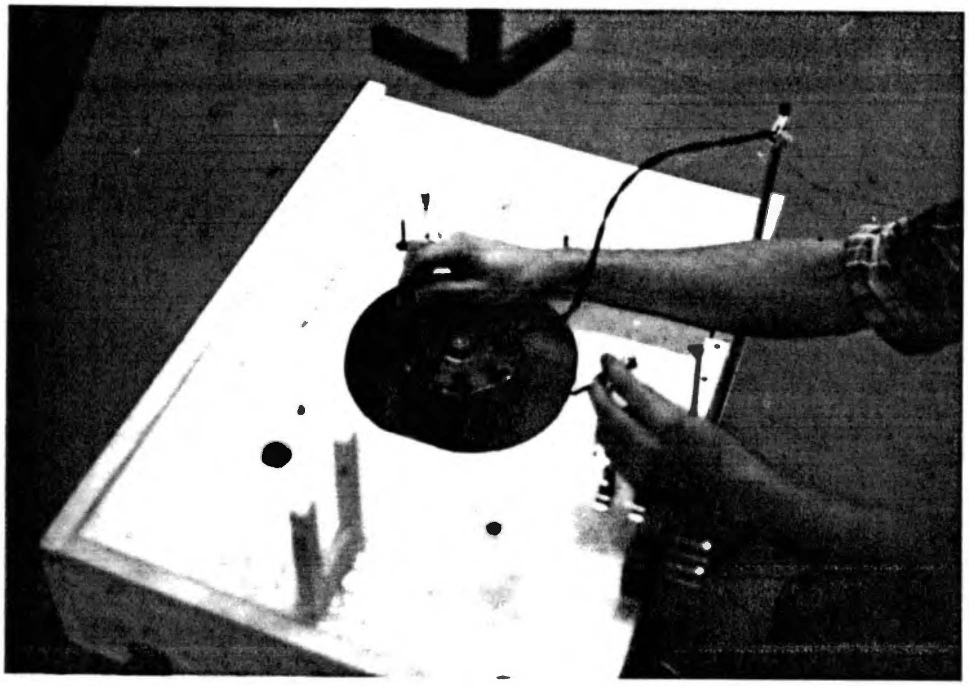


Figure 30. Human Worker Inserting Bolts into Motor

Figure 30 illustrates the human worker performing the task of bolt insertion while the robot arm remains idle a short distance from the motor. The human worker simply placed each bolt through the appropriate hole in the motor's end bell while at the same time checking to ensure correct alignment of the front bell, stator, and end bell. At the conclusion of the task, the operator returned to the system console, pressed the appropriate key on the keyboard and switched on robot arm power to resume program execution. A task element time was also obtained for the operation.

After insertion of the motor bolts, the next task for the robot was to grasp the motor (P14 and P15) and position it against the U-shaped bolt containment plate on the assembly fixture. The arm moved first through a series of straight-line motions (P14, P15, and P16), then through a joint-interpolated motion to arrive at a point just above the containment plate (P17), and finally through two more straight-line motions to arrive at the required position (P19).

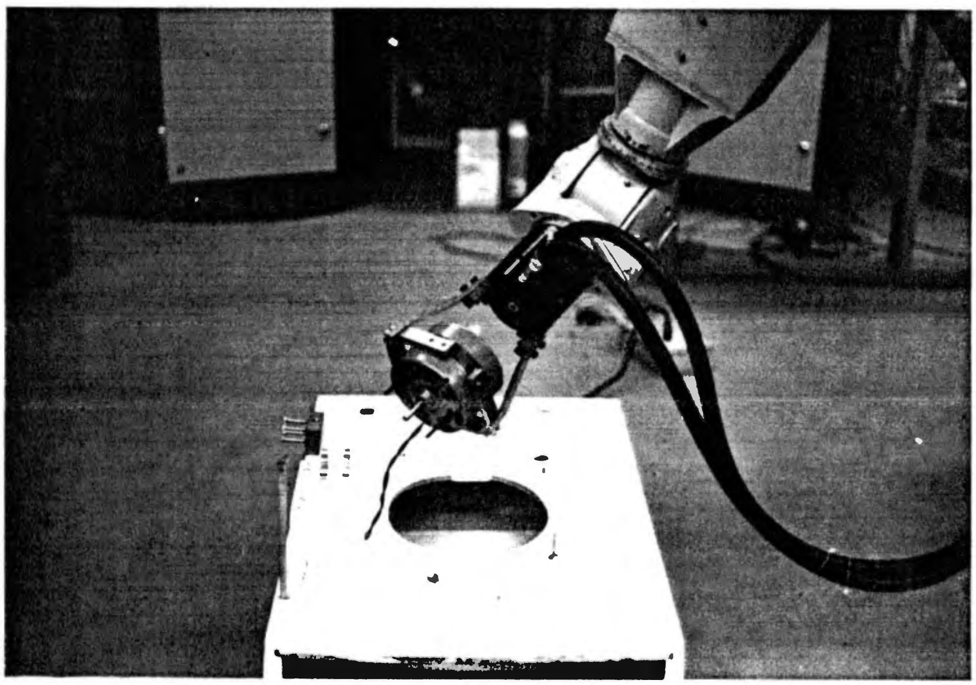


Figure 31. Robot Transporting Motor through Joint-Interpolated Movement

Figure 31 shows the robot arm during joint-interpolated movement before reaching its position above the containment plate (P17), while Figure 32 shows the final position obtained by the movement sequence (P19). After the final position was reached, a task element time was recorded and program execution was suspended while instructions were displayed to the operator via the system terminal.

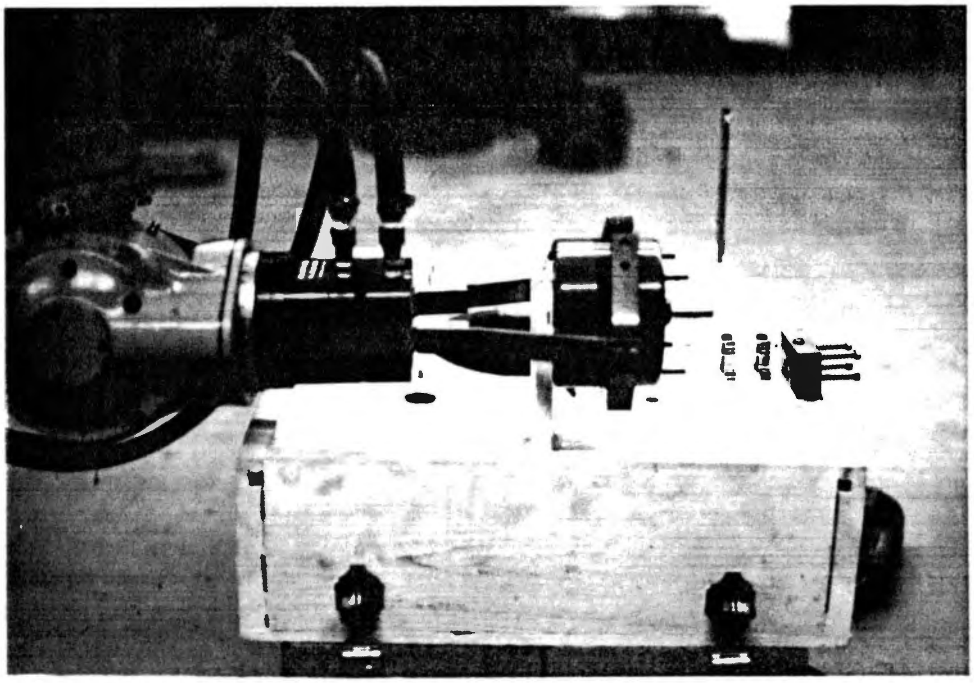


Figure 32. Robot Positioning Motor Against Bolt Containment Plate

The final task which remained in assembly of the motor was the nut threading and tightening operation performed by the human worker. Utilizing a small electric hand drill equipped with the magnetic extension socket, the worker pushed the socket over the nut which was positioned on the spring-loaded fixture block. The magnetic action of the socket held the nut in place while the worker initiated the threading procedure by placing the end of the socket against the end of an available bolt and started the drill motor. After the nut began to thread onto the bolt, the speed of the drill motor was increased to drive the nut the remaining distance on the bolt and provide sufficient torque to secure the nut onto the bolt. This procedure was repeated with the remaining nuts to complete the task. Figure 33 shows the human worker in the process of threading the nuts onto the bolts while the robot holds the motor in place against the bolt containment plate.

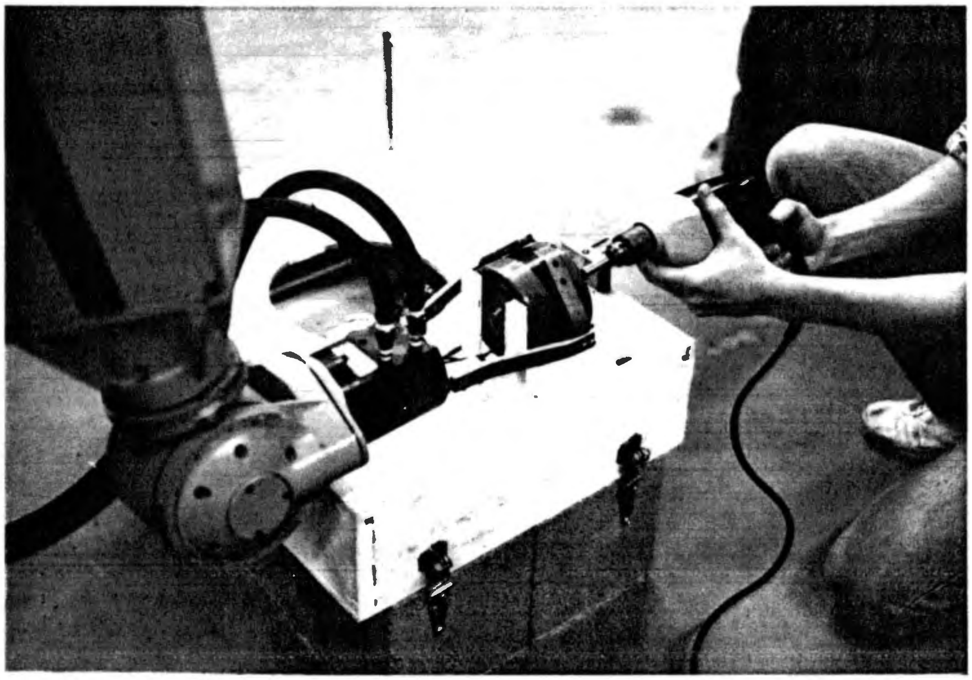


Figure 33. Human Worker Threading Nuts onto Motor Bolts

At the completion of the operation, the worker returned to the controller terminal, switched on arm power and pressed the "RETURN" key on the keyboard to resume program execution. A task element time was collected for the operation and assembly of the electric motor was completed.

The final task element which was recorded consisted of the elapsed arm movement time from the bolt containment plate (P19) to the final release point location in the inspection "bin" (P22). Figure 34 shows the completed motor in transit to the final release point (P20 and P21).

During the course of the move, the transition from normal straight-line movements to joint-interpolated movements was again required since the destination point could not be achieved with straight-line motion alone.

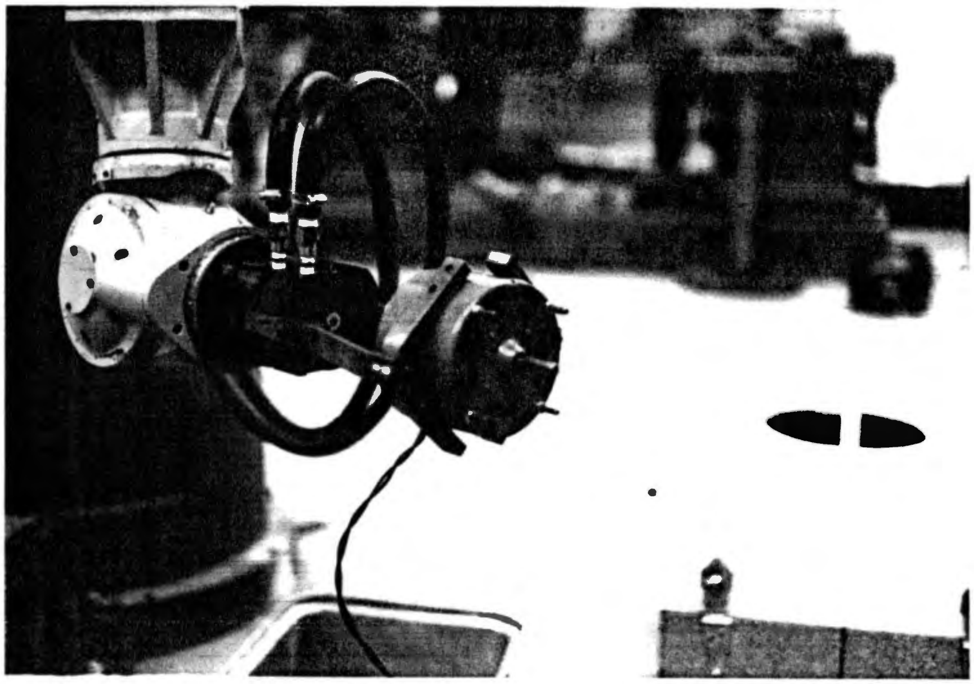


Figure 34. Robot Transferring Completed Motor to Release Point

Figure 35 shows the motor at the release point (P22). After the motor was deposited into the bin, the robot arm returned to the initial start point for the assembly sequence. At this point, the entire process was completed for the cycle, allowing all statistics to be collected and displayed on the terminal screen. In addition to the time statistics gathered during the cycle, the inspection for correct motor operation was performed by manual means. The motor was plugged into the appropriate power source and checked for correct operation, and the result recorded after each motor was assembled. Twenty single-arm assembly cycles were recorded to correspond with the cycles obtained during the manual assembly method. The results of the single-arm assembly procedure are discussed in Chapter IV.

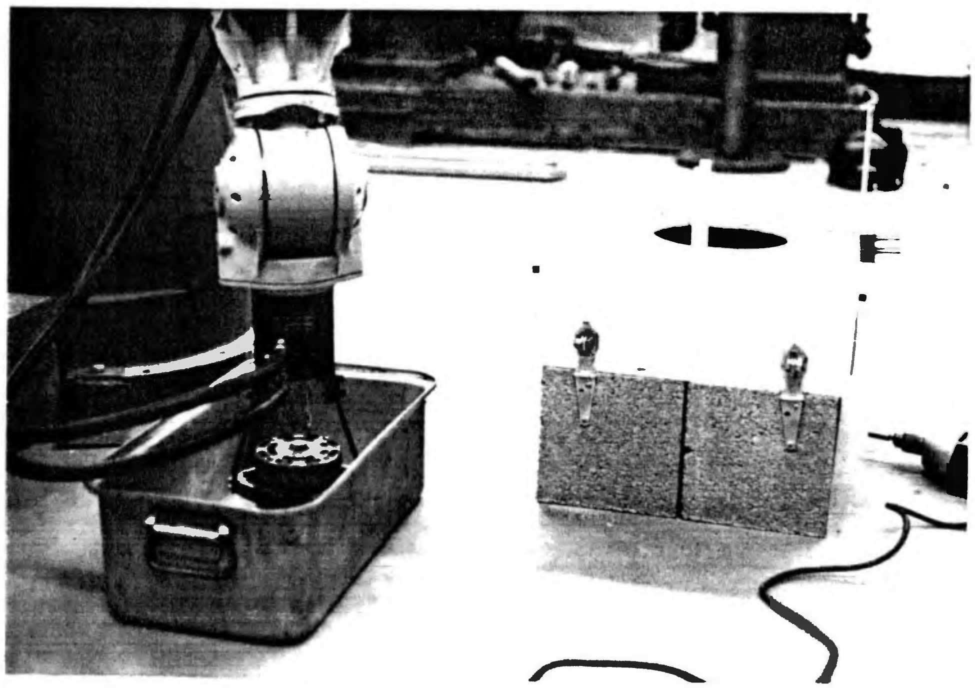


Figure 35. Robot Placing Completed Motor into Bin

Assembly Procedure Utilizing Two Robot Arms

The objective of the dual-arm assmbly procedure was to complete final assembly of the motor using coordinated motion between two robot arms to simulate a totally automated programmable assembly system. In an actual factory environment, the assembly system would be represented by a robotic "cell" which might receive the motor components from a conveyor belt arranged on a fixture "pallet" similar to the assembly fixture used in the study. Upon receipt of the pallet fixture, robots would proceed to assemble the motor and upon completion, the entire pallet fixture would leave the cell on the conveyor for further processing. In the actual experiment performed, however, the pallet fixture remained stationery and did not enter or exit the cell upon a conveyor system. In addition, the assembly procedure required a minor degree of human assistance (to throw a toggle switch on and off during the nut-threading operation). Thus, the assembly procedure was not fully automated; although it could have been completely automated if the proper input/output circuitry had been integrated with the robot controllers.

Prior to the start of the dual-arm assembly process, several objectives were required: (a) all system components were switched on and judged to be functioning correctly; (b) all motor components were placed in their respective positions on the assembly fixture; (c) the VAL II operating system was placed in "RUN" mode on both controllers to enable program execution; (d) the robot control "master" program entitled "MTR.PATH2" was submitted for execution on the controller directing the second robot arm (PUMA.2), while the robot control "slave" program entitled "MTR.PATH1" was submitted for execution on the controller directing the first robot arm (PUMA.1); and (e) all non-essential equipment and personnel were cleared of each robot's work envelope. After these requirements were met, the assembly sequence could be initiated.

Unfortunately, just prior to the dual-arm assembly procedure's initial trial run, an equipment malfunction caused damage to the first PUMA arm. The damage, although not major, was sufficient to cause a considerable delay in the repair effort and as a result, the dual-arm procedure could not be tested. However, an attempt has been made to describe the dual-arm assembly procedure as it would have been accomplished if the robot had not malfunctioned.

The assembly procedure developed for the dual-arm routine is illustrated in Figure 36 as an operation process diagram. The task elements closely matched those which composed the one-arm procedure, which would have served as a comparison of the task time results between the two procedures.

The assembly sequence would have begun with both robots positioned at their respective sequence starting points (refer to program listings in Appendix H). The motor components would have been positioned on the assembly fixture as shown in Figure 37. The flexible cable would have been attached to the drill motor located inside the assembly fixture for the dual-arm routine and can be seen in the figure.

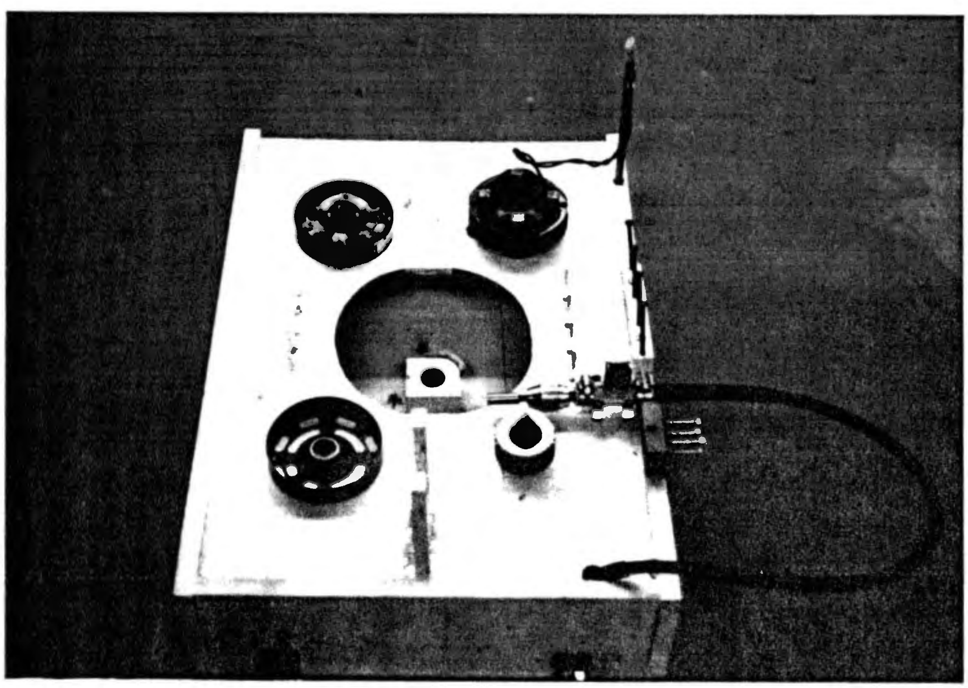
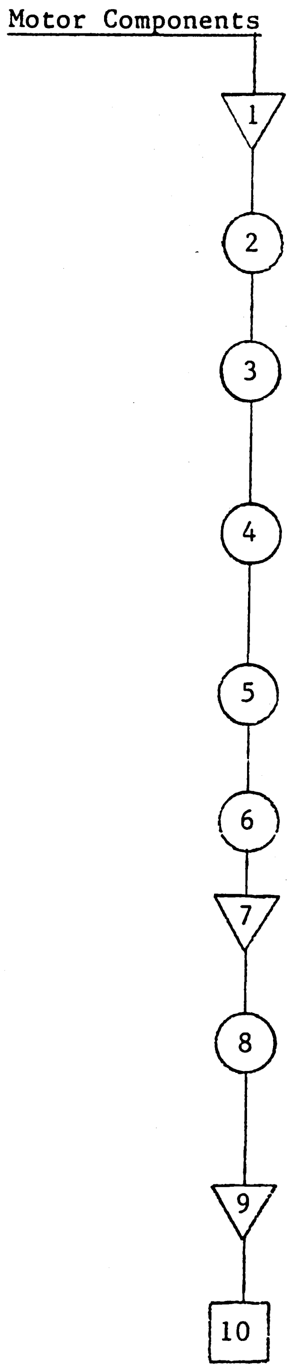


Figure 37. Assembly Fixture Prior to Start Dual-Arm Assembly Sequence



1. PUMA.2: Grasp front bell; place into assembly receptacle
2. PUMA.1: Grasp shaft; place into front bell
3. PUMA.2: Grasp stator; place onto front bell; grasp end bell and move to a point just above power plug
4. PUMA.1: Grasp power plug wire just below plug; feed through end bell grommet
5. PUMA.2: Place end bell onto front bell/stator
6. PUMA.1: Insert 4 bolts into motor
7. PUMA.2: Grasp motor and place against U-shaped bolt containment plate
8. PUMA.1: Grasp flexible drive cable and thread 4 nuts onto bolts to complete assembly
9. PUMA.2: Place completed motor into bin
10. HUMAN: Inspect motor for correct operation

Figure 36. Operation Process Diagram (Dual Robot Arm)

Figure 38 illustrates both robot arms at their respective sequence starting points (STRTP1). The starting point for PUMA.2 was identicial to the starting point in the single-arm routine. The starting point for PUMA.1 was a position approximately 20 cm. above the motor shaft and displaced midway between the shaft and the nut locator block (refer to Figure 38). The first operation would have involved placement of the front bell and then the stator by PUMA.2 onto the center locator block. The movement sequence of PUMA.2 was identical to that utilized in the "ONEARM" program (P1 through P8 in program "MTR.PATH2", Appendix H).

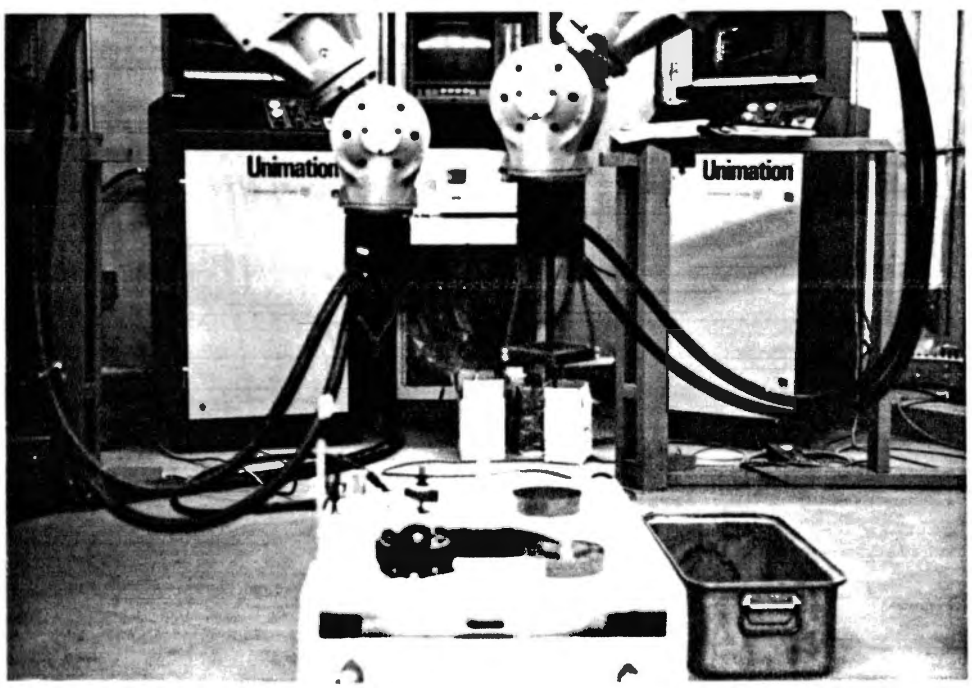


Figure 38. Robots at Sequence Starting Point

Figure 39 shows the dual-arm procedure during this stage of the assembly process. PUMA.2 is shown placing the front bell into the center locator block while PUMA.1 waits to grasp the motor shaft. After the stator had been placed, a signal would have been sent to PUMA.1 to begin movement towards the motor shaft (P1 in program "MTR.PATH1", Appendix H).

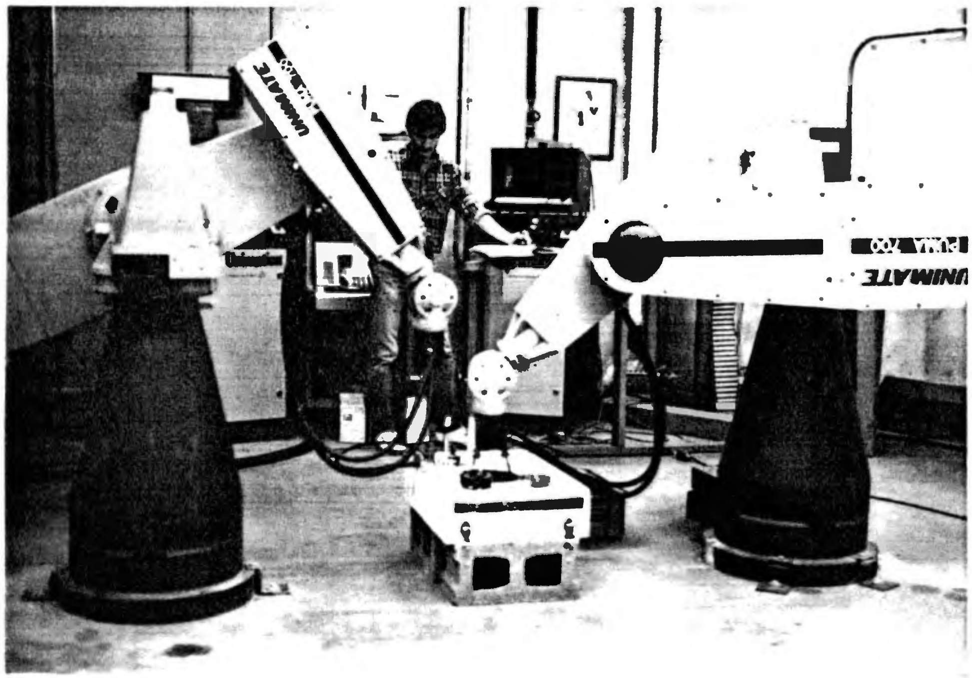


Figure 39. PUMA.2 Executing Placement of Front Bell onto Center Locator Block While PUMA.1 Waits to Grasp Motor Shaft

Upon receipt of the signal from PUMA.2, PUMA.1 would have grasped (P1) and then placed the motor shaft into the front bell/stator assembly (P4), while PUMA.2 grasped the end bell (P10) and lifted the bell off of the fixture locator pins (P9). The movement sequence of PUMA.2 was identical to that of the "ONEARM" program for these motions (refer to Appendix H). Instead of positioning the end bell over the motor, however, the PUMA.2 robot arm would have positioned the end bell approximately 2 cm. above the stator lead plug (P11) and then waited until PUMA.1 had moved to a nearby point (P5) before beginning a move which would have threaded the stator leads through the grommet in the end bell (P11). Just prior to the threading of the stator leads, a time for Task

1 would have been obtained from the elapsed times function.

The next task would have been the threading procedure for the stator leads combined with a coordinated move by both robots to position and place the end bell onto the motor. PUMA.1 would have held the stator leads at a secure position (P7) while PUMA.2 attached the end bell onto the stator (P12) to form the completed body of the electric motor. Task 2 time statistics would have been collected just prior to final placement of the end bell, while Task 3 time statistics would have been gathered just after placement of the end bell had been completed.

Task 4 involved the placement of the four bolts into the motor. After placement of the end bell, PUMA.2 would have withdrawn to a position approximately 50 cm. above the center locator block (P14). PUMA.1, upon receiving the "proceed" signal, would have begun to grasp and insert each of the four bolts into the motor. This movement essentially would have involved PUMA.1 approaching a bolt from a point 2 cm. above the bolt (P8), moving to grasp the bolt (P9), withdrawal (P8), moving to a point 2 cm. above the appropriate hole in the motor (P10), and a move to insert the bolt (P11). A withdrawal (P10) would have occurred, and the entire sequence repeated for the remaining bolts (P12-P15, P16-P19, and P20-P23, respectively). At the completion of the task, time statistics would have been collected and PUMA.1 would have withdrawn to a safe location to allow PUMA.2 to grasp the motor.

Task 5 would have begun with PUMA.2 grasping the motor (P16) and moving it through the joint-interpolated motion range (P17) into position against the U-shaped bolt containment plate (P20). Simultaneously, PUMA.1 would have grasped the flexible cable (P25) and moved to pick up the first nut on the nut locator block (P26). Upon the completion of these moves by each robot, time statistics would then be collected, and Task 6 would begin.

Task 6 would have involved the threading of each nut onto a corresponding motor bolt. Upon placing the motor against the bolt containment plate, PUMA.2 would have signaled PUMA.1 to begin the threading procedure. While PUMA.2 held the motor firmly against the containment plate, PUMA.1 would have begun the procedure by moving to pick up the first nut with the magnetic socket (P27), withdraw (P26), move to a position 1 cm. from the end of the appropriate bolt (P28), advance to thread the nut (P29) while power was applied to the drill motor, withdraw when threading completed (P28), and move to pick up the next nut in the series (P30-P33, P34-P37, and P38-P41, respectively). After the four nuts had been threaded onto the bolts, Task 6 would have been completed and time statistics collected.

The final assembly of the electric motor would then have been completed, and the final task element would have involved the deposit of the finished motor into the bin for subsequent inspection to ensure correct operation. PUMA.2 would have removed the motor from the bolt containment plate (P20), transferred the motor to the bin (P21, P22), and released it (P23). At the same time, PUMA.1 would have replaced the flexible drive cable into its position on the assembly fixture (P25) and moved to a sequence termination point (FINPT) to avoid a collision upon return of PUMA.2 to its sequence start point (STRTP1). After PUMA.2 had deposited the completed motor into the bin, the time statistics for Task 7 would then have been collected.

At this point, the entire assembly cycle could have been completed. All task times, robot arm times, and total assembly time would have been displayed on the system terminal following the same format as that of the "ONEARM" program. Had the dual-arm assembly procedure been achieved, the program would have been executed 20 times to achieve the same number of observations obtained in the manual and single-arm assembly methods.

CHAPTER IV

RESULTS

During the course of each assembly method, a time observation was collected for each task element involved in the method as well as a time observation of the total assembly time for the motor. These observations could then be treated in a statistical manner to provide a measure of the mean assembly time, the variance of the assembly time, and a mean and variance for each of the task elements. The various statistics for the methods could then be compared to each other and to their applications in actual manufacturing environments. This chapter presents a summary of the data obtained in the study. Further statistical analysis of the summarized data shall be discussed in the next chapter.

In addition to the presentation of the summary in a tabular format, this chapter also approaches results of the assembly methods from a "human factors" viewpoint. In essence, the impact of the human interaction in the assembly process is evaluated and the results are presented. The human interaction in each of the assembly methods certainly was a critical factor in the overall assembly time of the motor. In addition, human interation in the assembly methods ranged from 100% in the manual assembly method, to approximately 70% in the single-arm routine, to an estimate of less than 5% in the dual-arm procedure. These figures are based upon the percentage of time that the human performed assembly of the motor with respect to the total time in which the motor was assembled.

Manual Assembly

The results of the manual assembly method are presented in Table II. For each task element, a mean was calculated as well as a corresponding variance. In addition, the total assembly time mean and variance was computed.

TABLE II

RESULTS OF MANUAL MOTOR ASSEMBLY

Element Number	Mean	Variance	Std. Dev.
1	15.90	25.36	5.03
2	6.80	19.85	4.46
3	10.50	24.05	4.90
4	25.60	707.83	26.61
5	68.15	90.03	9.49
6	9.35	14.24	3.77
7	4.35	3.08	1.75
Total Assembly Time	140.65	1,201.33	34.66

NOTE: All times in seconds.

In each calculation of the mean, variance, and standard deviation, the number of observations remained fixed at 20. Two of the observations in the manual assembly sequence involved a high variance due to difficulty in motor alignment which was necessary for correct bolt insertion (refer to Table I, Appendix A). It was decided that these two abnormally long assembly times should be retained in the calculations, however, because the inclusion of these assembly times would serve to illustrate the difficulties with motor bolt hole alignment using human manipulation of these parts. In addition to the statistics listed in Table II, all 20 of the motors were inspected for correct operation after assembly. Eighteen motors operated correctly; two motors did not rotate properly and thus were placed in the "rework" bin for subsequent inspection and reconditioning.

From a "human factors" standpoint, the results of the manual assembly method suggest that human assembly of the motor results in worker fatigue beginning relatively early into the production cycle. From the time at which the third motor was completed, the subject who was performing the assembly began to complain about the weight of the motor becoming a burden on the assembly task. Although the data collected on the average assembly times does not indicate a significant increase in assembly time as the number of motors increases, if the observations were collected on the basis of an eight-hour work period, the results would most certainly indicate that productivity would decrease due to the handling of this weight for an extended period of time.

One Robot Arm

The results of the single-arm assembly method were collected by the variables assigned to the TIMER command as the robot control program was executed (refer to program "ONEARM", Appendix E). Since each task element time and final assembly time was obtained via the computer, the data collection process was simplified and reflected a greater accuracy in measurement than than obtained by manual methods using a stopwatch.

Table III represents the compilation of the data obtained during program execution (for a listing of the data, refer to Appendix I). The calculations listed in the table were made using 20 observations and the format closely parallels that of the manual assembly method, with the exception of the arm movement time included in the results.

TABLE III

RESULTS OF SINGLE-ARM ROBOT ASSEMBLY

Task Number	Mean	Variance	Std. Dev.
1	15.29	.35	.59
2	19.47	6.50	2.55
3	4.55	0	0
4	25.42	110.71	10.52
5	6.27	.008	.09
6	34.95	6.02	2.45
7	3.65	.008	.09
Arm Movement Time	32.29	.95	.97
Total Assembly Time	112.17	108.99	10.44

NOTE: All times in seconds.

With respect to the human element involved in the single-arm routine, it was obvious that a much larger amount of variance occurred within tasks which were performed by the human worker than those performed by the robot arm. Part of this variance was due to unequal or unsymmetric hand motions by the human when performing the task each time. There existed no specific assembly "pattern" to which the human worker conformed each time the task was performed. The other source of variance arose from the transit time involved when the human worker switched off the power supplied to the robot arm, walked over to the assembly fixture to perform the task, and subsequently walked back to the controller to switch on arm power. This action was repeated three times during the course of each assembly cycle and thus contributed significantly to an increase in both the total assembly time as well as the variance in assembly task element times.

Another result with regard to human factors found during the experiment was that there appeared to be no noticeable increase in worker fatigue during the course of the 20 assembly cycles. This was due to the decrease in the handling of heavy motor components by the worker and the infrequent handling of the drill motor. Thus, the worker was manipulating smaller, lighter parts which greatly contributed to the decrease in fatigue during motor assembly.

Two Robot Arms

The intent of this section was to provide a summary of results obtained from the dual-arm assembly of the motor. Unfortunately, these results could not be obtained due to the previously mentioned equipment malfunction. The hypothetical "results" could be discussed from the standpoint of an "educated guess" as to their probable outcome however, and from the standpoint of the human factors element as well.

During the course of the single-arm experiment, a general idea of the dual-arm assembly time could be envisioned from the correlation which existed between arm speed and accuracy in the placement of motor components. As arm speeds were increased faster than 800 millimeters per second, a corresponding decrease in placement accuracy occurred. Although the larger motor components could be adequately located with fast arm speeds, the smaller motor components such as the nuts and bolts would require much slower arm speeds during assembly. This would probably have resulted in a slightly slower total assembly time for the dual-arm routine when compared to the single-arm assembly of the motor. Dual-arm assembly times might, however, have been faster than the manual method of assembly, and would certainly have maintained a smaller degree of variance than that inherent to manual assembly.

In consideration of the human factors with regard to the dual-arm routine, two results were discovered. First, it was apparent that the human operator would not be involved in any physical manipulation of the motor components; thus, no fatigue would arise from the constant weight of the motor. Second, since the only human interaction during the dual-arm assembly cycle would have been to flip a toggle switch on or off to start and stop the drill motor for the nut threading operation, very little variance would be incorporated into the total assembly time of each cycle. Since only a small portion of the total assembly time would have been directly influenced by the human worker, a much more consistent assembly time would have resulted than those obtained from manual or single-arm assembly methods.

CHAPTER V

ANALYSIS OF RESULTS

The analysis of the results of the study is presented in two main areas: the first area being the evaluation of the summarized statistics presented in Chapter IV in order to more fully describe the characteristics of each assembly method, while the second area focuses upon the comparison between the assembly methods with regard to their use in an industrial environment.

After the raw data for the manual assembly method and the single-arm assembly method had been collected and summarized, further characteristics of each method were obtained by first calculating a confidence interval for the mean, then calculating a range for the production rate based upon the upper and lower confidence interval limits, and finally incorporating the effects of motors which required rework into the production rate to obtain a better estimate of true production output.

In order to obtain a confidence interval for the mean assembly time of the motor, an assumption was made that the data followed a normal distribution. Thus, the sample mean and sample standard deviation obtained from the data could be used to calculate a confidence interval about the true mean. A confidence interval of 90% was selected for use in the calculation. From equation (1) the two-sided confidence interval about the mean can be found (13):

$$\bar{X} - t_{\alpha/2,\ n-1}\left(\frac{s}{\sqrt{n}}\right) \qquad \bar{X} + t_{\alpha/2,\ n-1}\left(\frac{s}{\sqrt{n}}\right) \tag{1}$$

where

$\bar{X}$ = sample mean
S = sample variance
n = sample size
α = 1 - confidence coefficient (.90) = .10
t = percentage point of the t-distribution

The confidence interval about the mean for the manual assembly method is:

$$140.65 - 1.729\left(\frac{34.66}{4.47}\right) \leq \mu \leq 140.65 + 1.729\left(\frac{34.66}{4.47}\right)$$

or

$$127.25 \leq \mu \leq 154.05$$

Therefore, the "true mean" of the manual assembly time lies between 127.25 seconds and 154.05 seconds with a confidence of 90%.

The confidence interval about the mean for the single-arm assembly method is:

$$112.17 - 1.729\left(\frac{10.44}{4.47}\right) \leq \mu \leq 112.17 + 1.729\left(\frac{10.44}{4.47}\right)$$

or

$$108.13 \leq \mu \leq 116.20$$

Therefore, the "true mean" of the single-arm assembly time lies between 108.13 seconds and 116.20 seconds with a confidence of 90%.

From these calculations, a production range in motors per hour may be obtained from equation (2):

$$\frac{\text{motor}}{U_L \text{ or } L_L \text{ (sec.)}} \cdot \frac{60 \text{ sec.}}{\text{min.}} \cdot \frac{60 \text{ min.}}{\text{hour}} = \text{\# of motors/hr.} \tag{2}$$

For the manual assembly confidence limits, formula (2) yields a production range of 23.4 to 28.3 motors per hour. Using the values obtained for the single-arm confidence limits resulted in a production range of 31.0 to 33.3 motors per hour. Assuming a typical production run of 100 motors per "batch" (refer to Chapter I, page 10), the total time required to produce a single batch would range from 3.53 hours to 4.27 hours for manual assembly, and from 3.00 hours to 3.22 hours for single-arm assembly.

These production rate estimates do not consider the effects of defective motors however, and the inclusion of such possibilities must be done in order to obtain a more accurate description of the expected production rate. In Chapter IV of the study, the results of the post-assembly motor inspection were presented. In the case of the manual assembly experiment, two motors were found to be inoperable after assembly due to the worker applying excessive torque to the nuts, which in turn increased pressure upon the motor shaft from the front and end bell bushings. Thus, the shaft was not able to turn due to the increased friction applied by the bushings, and the motor was rejected at the inspection stage to be reworked. Based upon the sample data, the rejection rate for manual assembly is considered to be 2 out of every 20 motors produced. This translates into a motor rework rate of 10% for the assembly cycle. The revised production rate range would therefore need to be increased by the mean time necessary to complete two additional motors in order to compensate for the rework percentage.

The motor inspection results for the single-arm routine indicated no defective motors were found after assembly. Part of the difference between the results of the reject rates between the manual assembly and single-arm assembly can be traced to the problem of excessive torque applied to the nuts when threaded onto the motor bolts. Unlike manual threading of the nuts where a uniform torque cannot accurately be applied using a standard wrench, the single-arm assembly utilized a variable-speed drill motor to evenly apply torque to the nuts. This method greatly enhanced the correct uniform application of torque, which resulted in less bushing pressure upon the motor shaft and thus fewer motors failing to pass inspection.

Based upon the sample data for the single-arm assembly, the rejection rate was nonexistent, therefore, no adjustment in the overall production rate range was necessary. This is not to indicate that there would never exist any rework rate in an actual production cycle with single-arm assembly, but for comparative purposes in the study, it was accepted as negligible.

Table IV presents the compilation of the final results for both manual assembly and single-arm assembly methods. The table illustrates the differences in mean assembly times, standard deviation in assembly times, confidence intervals, production rates, and total batch production assembly times (adjusted to account for motor rework) between the two methods.

With respect to each method's performance in a manufacturing environment, the final assembly time and its associated variance are obviously two of the most important factors in selection of the most efficient manufacturing method. From the results summarized in Table IV, it is evident that single-arm robotic assembly combined with manual labor is faster and exhibits far less variance than that obtained using strictly manual assembly of the motor. The difference between the mean assembly time for the two methods (approximately 28 seconds), while not extraordinarily faster, is substantial when the total batch production

TABLE IV

SUMMARY OF RESULTS

Characteristics	Manual Assembly	Single-Arm Assembly
Mean Assembly Time	140.65 sec.	112.17 sec.
Standard Deviation	34.66 sec.	10.44 sec.
90% Confidence Interval on Mean Assembly Time	127.25 sec. to 154.05 sec.	108.13 sec. to 116.20 sec.
Production Rate Range Based upon Confidence Interval	23.4 motors/hr. to 28.3 motors/hr.	31.0 motors/hr. to 33.3 motors/hr.
Batch Production Time* (100 motors)	3.93 hrs. to 4.75 hrs.	3.00 hrs. to 3.22 hrs.

* Assuming a single assembly station.

times are considered. Time savings can be measured in hours when batch quantities of the motor are produced.

What is perhaps more important than the savings in assembly time, however, is the dramatic reduction in assembly time variance as well as task time variances of the single-arm method compared to the manual method. The benefits to manufacturing resulting from this reduction of assembly time variance are many. With smaller variance in the process, production becomes much more stable and predictable, assembly lead time is reduced, motor component delivery lead times are reduced, etc., all of which help to increase the productivity of the manufacturer.

There also exist other potential benefits of single-arm assembly over the manual assembly method. Although no attempt has been made to economically evaluate the performance of the two methods with respect to actual costs incurred, cost savings, etc., it is not difficult to project that a substantial cost savings would indeed occur in the long run if the semiautomated process was utilized rather than the manual assembly method. While the initial cost of implementing the manual assembly method would be lower due to the unsophisticated workstation, the savings resulting from faster, more efficient motor production using the single robot arm in conjunction with manual assembly would eventually pay back the high initial equipment cost and thereafter provide greater revenue earnings.

Not only would cost savings contribute to the advantage of the single-arm assembly over manual assembly, but also the savings with regard to worker fatigue would provide a distinct benefit to single-arm assembly. During the course of the manual assembly experiment, it was noted that the worker's arms began to tire after only three motors had been completed. During the course of a production run, it is highly probable that the worker's productivity would decrease as the number of motors assembled increased. This in turn would lower the production rate and increase batch production time. In the single-arm routine, however, only small motor components are handled, and the light weight of these parts do not contribute toward worker fatigue. The heaviest item which the worker must lift during the assembly process is the electric drill motor, but since the worker has both hands available for the task and the motor is only handled briefly during the cycle, the load does not accrue significant fatigue on the worker.

From the comparison of the benefits between purely manual assembly and single robot arm assembly of the motor, it is evident that assembly of the electric motor utilizing one robot arm in conjunction with human assistance holds many advantages over the assembly utilizing only manual means. Since a direct numerical comparison could not be made of the dual-arm assembly method with respect to the other methods, the author can only provide an educated guess concerning how the dual-arm assembly characteristics would have compared to the other methods.

Although a direct measurement of the mean assembly time was not obtained for the dual-arm method, it would not be unrealistic to place the range of the mean assembly time in between that of the single-arm assembly and that of the manual assembly. If improved component positioning accuracy were to be achieved in the assembly fixture through enhanced design and use of rigid material such as steel, mean assembly time might be reduced considerably, such that assembly times of less than 90 seconds might be possible. If a mean assembly time of less than 90 seconds were to be achieved, a much greater increase in productivity would be observed when compared with the other two methods.

A substantial benefit which would very likely have been exhibited by dual-arm assembly of the motor would be a very low assembly time variance. As stated previously, lower assembly time variances result in many benefits to the manufacturer. It is of great value to a manufacturer to be able to accurately predict when a product or a batch quantity is to be completed, and the utilization of the automated dual robotic arm assembly procedure certainly would have provided the lowest variance among the three methods.

Assuming that the dual-arm assembly procedure would be a completely automated and integrated manufacturing system or "cell" if the procedure were to be utilized by an actual manufacturer, two distinct advantages over the other assembly methods would be noted. Since the assembly procedure would be totally automated, no human interaction would be required in the assembly process, thus eliminating the position occupied by manual labor held for the process. This, in turn, would result in substantial cost savings which in many cases is enough to justify the cost of capital equipment purchased for the implementation of the project. Besides the cost savings incurred due to manual labor elimination, the other advantage would be the tremendous adaptability of the system to assemble a wide variety of motors.

The capability of the robotic system to be reprogrammed for each motor type produced by the manufacturer is a tremendous advantage. Once the assembly program is generated, it could then be loaded into the computer memory of the robotic controller. When the particular motor is to be assembled, the corresponding program can be executed immediately. Although manual assembly of the electric motors by a human worker exhibits the ultimate in adaptability, robotic assembly can be more than adequate for the task, especially if the variability of motor types which are assembled is not excessively large.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

The purpose of this study was to examine a product which would typically be produced in a batch quantity and perform a final assembly of the product using (a) human labor only, (b) human labor combined with the assistance of a single robot arm, and (c) coordination between two robot arms. These three methods of assembly could then be compared to determine various process characteristics which, in turn, could be compared to actual industry applications and conclusions drawn with regard to the use of each method in actual manufacture of the product.

The product selected for the assembly experiment was a single-phase alternating-current electric motor typically utilized for powering fans, air-conditioning compressor drives, and other home appliance applications. The motor was selected for its suitability with regard to robotic assembly by the UNIMATE Puma 762 series industrial robots available at the time of the study.

Although an equipment malfunction prevented completion of the dual-arm assembly procedure, relevant data was obtained from both manual assembly of the motor as well as single-arm assembly of the motor. The data obtained from the experiment was statistically analyzed and the results compared between these two assembly methods.

For the manual assembly method, a mean assembly time of 140.65 seconds was observed with a corresponding assembly standard deviation of 34.66 seconds. For single-arm assembly of the motor, a shorter mean assembly time of 112.17 seconds was observed with a small corresponding assembly standard deviation of 10.44 seconds. The analysis of the results indicates that single robot arm assembly of the electric motor combined with human labor is faster than assembly of the motor by human labor alone.

Assembly of the motor with the single-arm method also indicates less variability in assembly time when compared with assembly utilizing human labor alone. From this statistic, it can be concluded that single-arm assembly produces motors at a more consistent rate than that of the manual assembly method.

The results also indicated a higher fraction rejected rate of 10% in assembly of the motor by human labor alone, compared to a fraction rejected rate of 0% in assembly of the motor by the single robot arm method. In addition, worker fatigue was notably higher in assembly of the motor by human labor only as compared with very little fatigue in assembly using a single robot arm in conjunction with human labor.

Recommendations for Further Research

Although the experiment provided insight into the comparison between manual, semi-automated, and fully automated assembly of the product, much more research remains to be accomplished. Notably, the dual-arm assembly experiment should be completed to provide sample data for subsequent evaluation and comparison against the other assembly methods. Further investigation towards increased accuracy in motor component location using metal or molded fiber composite materials could be done. Machine vision techniques could be utilized to interface with the robot controller to enable the robot to locate and grasp motor parts which may be "randomly" located about the assembly fixture. These suggestions provide a starting point for further research into the robotic assembly project in hopes that the project shall be expanded upon in the future, and thus help to provide research discoveries which will advance automation technology to its full potential.

BIBLIOGRAPHY

(1) Groover, Mikell P. and Emory W. Zimmers, Jr. CAD/CAM: Computer-Aided Design and Manufacturing, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1984.

(2) Nitzan, David. "Assessment of Robotic Sensors." Proceedings of the First International Conference on Robot Vision and Sensory Controls. Bedford, England: IFS (publications) Ltd., (1981), pp.1-11.

(3) Industrial Robots: A Summary and Forecast. 2nd Ed. Naperville, Illinois: Tech Tran Corporation, 1983.

(4) "Multiple Arm Robot Control Systems." Robotics Age, October, 1985, p. 5.

(5) "Assembly Robots with Overlapping Work Envelopes." Robotics Age, October, 1985, p. 23.

(6) Movich, Richard C. "An Intelligent Robotic System for Aircraft Manufacturing." Proceedings of the Robots 8 Conference. Dearborn, Michigan: Robotics International of SME, 1984.

(7) Smith, Randall C. and David Nitzan. "A Modular Programmable Assembly Station." Conference Proceedings of the 13th International Symposium on Industrial Robots and Robots 7, Vol. 1. Dearborn, Michigan: Robotics International of SME, 1983.

(8) Robot Vision and Sensory Controls: Proceedings of the First International Conference. Bedford, England: IFS (publications) Ltd., (1981).

(9) Robillard, Mark J. Microprocessor Based Robotics. Indianapolis, Indiana: Howard W. Sams & Co., Inc., 1983.

(10) Safford, Edward L. Handbook of Advanced Robotics. Blue Ridge Summit, Pennsylvania: TAB Books, Inc., 1982.

(11) Unimation Industrial Robot User's Guide to VAL II. Danbury, CT: Unimation, Inc., 1985.

(12) Konz, Stephan. Work Design: Industrial Ergonomics, Second Ed. Columbus, Ohio: Grid Publishing, Inc., 1983.

(13) Hines, William W. and Douglas C. Montgomery. Probability and Statistics in Engineering and Management Science. 2nd Ed. New York, New York: John Wiley & Sons, 1980.

APPENDIXES

APPENDIX A

ELECTRIC MOTOR PART LIST AND DESCRIPTION

ELECTRIC MOTOR PART LIST

Motor Specifications

Manufacturer: Dayton Electric Manufacturing Company

Type: Model 3M569 1/15 H.P. Shaded Pole

Operation: 115V 60 Hz 2.3A 1550 RPM

Quantity	Description
1	End Bell
1	Front Bell
1	Stator Assembly
1	Shaft Assembly
4	10-32 NF x 9.55 cm bolts (3.75 in.)
4	.313 cm hexagonal nuts (5/16 in.)

APPENDIX B

MANUAL ASSEMBLY TIME STUDY DATA

TABLE I

TIME STUDY DATA FOR MANUAL ASSEMBLY

NO.	DATA SET 1 ELEMENTS	UPPER LINE: SUBTRACTED TIME 1	2	3	4	5	6	7	8	9	LOWER LINE: READING 10	MIN. TIME	AVG. TIME
	Thread wire leads into	20	32	22	16	12	13	17	11	16	17		
1	end bell & fit stator	:20	3:16	6:09	8:11	:20	:27	:31	:43	:33	:47	11	17.6
	Place shaft into end	10	7	6	3	6	5	10	6	23	5		
2	bell/stator	:30	:23	:15	:14	:26	:32	:41	:49	:56	:52	3	8.1
	Place front bell over	15	14	9	12	26	8	6	4	10	7		
3	shaft & into place	:45	:37	:24	:26	:52	:40	:47	:53	21:06	:59	4	11.5
	Insert 4 bolts through	17	31	19	15	102	27	20	16	15	11		
4	motor assembly	1:02	4:08	:43	:41	12:34	15:07	17:07	19:09	:21	23:10	11	27.3
	Thread nut onto each	74	79	57	71	81	56	75	56	60	74		
5	bolt & tighten w/wrench	2:16	5:27	7:40	9:52	13.55	16:03	18:22	20:05	22:21	24:24	56	68.3
	Clip plug/ground wire	20	16	10	11	15	7	7	8	6	7		
6	on motor & test	:36	:43	:50	10:03	14:10	:10	:29	:13	:27	:31	6	10.7
	Remove plug & ground	8	4	5	5	4	4	3	4	3	3		
7	wire; place in bin	:44	:47	:55	:08	:14	:14	:32	:17	:30	:34	3	4.3
	Sum of Elements	164	183	128	133	246	120	138	105	133	124	105	147.4

FOREIGN ELEMENTS:

TOOLS, JIGS, GAUGES, ETC: .794 cm. (5/16 in.) open-end wrench

EFFORT RATING	BEGIN	END	ELAPSED	UNITS FINISHED	TIME PER PIECE
100%	10:21:00	10:45:34	24:34	10	2:27

TABLE I (Continued)

NO.	DATA SET 2 ELEMENTS	UPPER LINE: SUBTRACTED TIME 1	2	3	4	5	6	7	8	9	LOWER LINE: READING 10	MIN. TIME	AVG. TIME
	Thread wire leads into	11	18	13	13	14	20	14	9	15	15		
1	end bell & fit stator	:11	:30	:27	:37	:26	12:04	:14	16:06	18:09	20:08	9	14.2
	Place shaft assembly in-	9	4	5	5	4	4	4	4	5	11		
2	to end bell/stator	:20	:34	:32	:42	:30	:08	:18	:10	:14	:19	4	5.5
	Place front bell over	12	6	8	7	10	8	13	7	16	12		
3	shaft & into place	:32	:40	:40	:49	:40	:16	:31	:17	:30	:31	6	9.9
	Insert 4 bolts through	13	15	12	20	102	13	14	22	13	15		
4	motor assembly	:45	:55	:52	7:09	10:22	:29	:45	:39	:43	:46	12	23.9
	Thread nut onto each	74	65	82	53	69	81	59	66	59	72		
5	bolt & tighten w/wrench	1:59	4:00	6:14	8:02	11:31	13:50	15:44	17:45	19:42	21:58	53	68.0
	Clip plug/ground wire	8	10	6	7	9	7	9	5	8	11		
6	on motor & test	2:07	:10	:20	:09	:40	:57	:53	:50	:50	22:09	5	8.0
	Remove plug/ground	5	4	4	3	4	3	4	4	3	10		
7	wire; place in bin	:12	:14	:24	:12	:44	14:00	:57	:54	:53	:19	3	4.4
	Sum of Elements	132	122	130	108	212	136	117	117	119	146	108	133.9

FOREIGN ELEMENTS:

TOOLS, JIGS, GAUGES, ETC: .794 cm. (5/16 in.) open-end wrench

EFFORT RATING	BEGIN	END	ELAPSED	UNITS FINISHED	TIME PER PIECE
100%	10:55:00	11:17:19	22:19	10	2:14

APPENDIX C

PROGRAM DESCRIPTION FOR SINGLE ARM ASSEMBLY

PROGRAM: ONEARM

Line Numbers	Description
1 - 35	Program Header section; includes program title, a brief description of purpose, the programmer's name and coding date, a listing of the variables used in the program and their corresponding function.
39 - 46	Initial parameters are defined in this section; robot configuration and operating parameters, such as arm speed, gripper delay, etc., are set. Variables used to store assembly time measurements are initialized.
51 - 56	Program prompts robot operator if fixture location/frame transformation are to be defined. If "yes" selected, program control continues with line 60; if "no" selected, program control branches to line 113.
60 - 78	Program directs robot arm to move to a orientation location at each end of the fixture where the operator is then prompted to position the appropriate end of the assembly fixture against the edge of the gripper to obtain an approximate alignment of the fixture.
79 - 104	Program prompts operator to position robot using the teach pendant to three different points which define an x - y coordinate system; a FRAME command is then executed to enable the robot to correctly reference all other points defined in the assembly routine.
105 - 109	Operator is prompted to either proceed with execution, or retry alignment procedure; if "retry" selected, program control returns to line 60; if "proceed" selected, program control advances to line 113.
113 - 118	Operator is prompted to either proceed with assembly sequence execution or abort the run; if "proceed" selected, program control continues at line 120; if "abort" selected, program control branches to line 255.

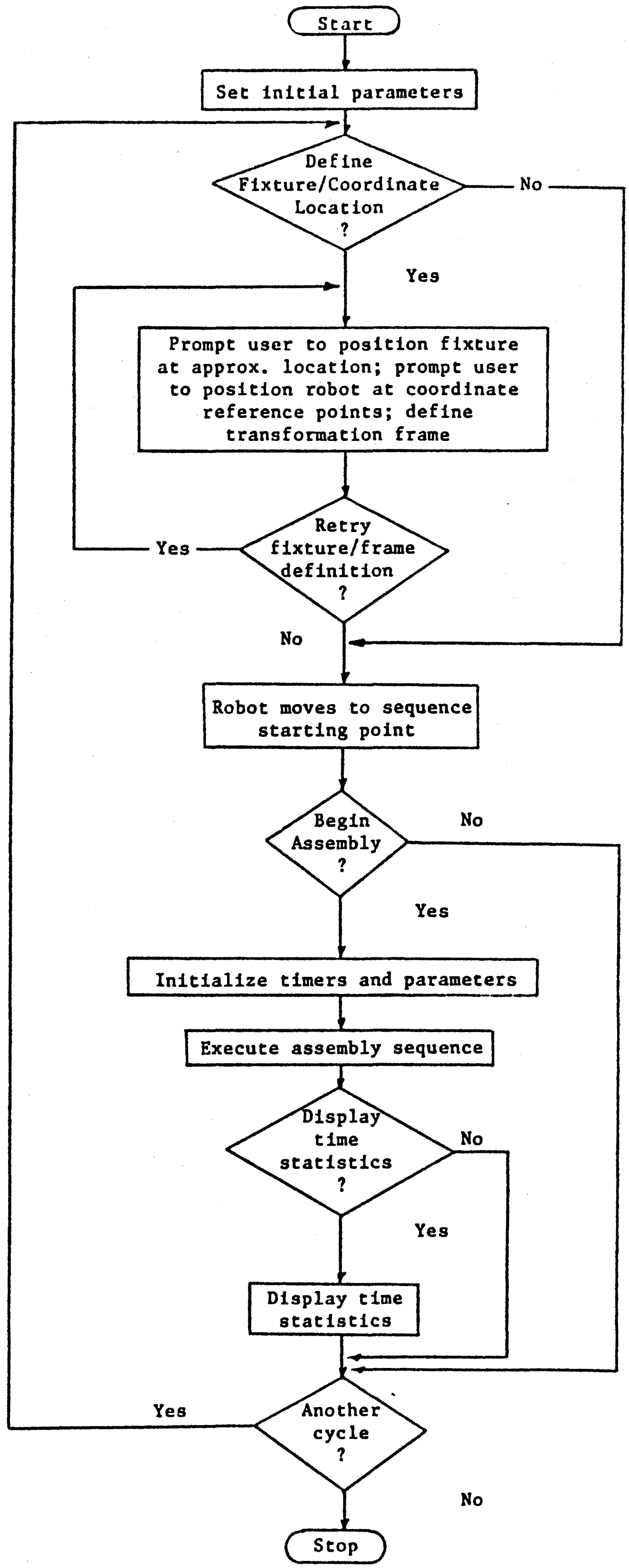
Line Numbers	Description
120 - 122	Program initializes timer function variables.
126 - 159	Assembly sequence initiated; program directs robot arm to grasp front bell, place it onto center locator block; then grasp stator and place onto front bell; then grasp end bell and position it above motor. Time measurements are then collected.
160 - 166	Program prompts human worker to turn off power and perform shaft insertion and stator lead positioning; program execution is suspended until task completed; time measurements collected.
167 - 181	Program directs robot arm to place end bell upon stator and withdraw to allow bolt placement; time measurements are collected.
182 - 187	Human worker prompted to turn off arm power and proceed with bolt insertion; program execution is suspended while task completed and time measurements obtained.
188 - 204	Program directs robot arm to grasp motor and move into position against bolt containment plate; time measurements recorded.
205 - 210	Human worker prompted to turn off arm power and proceed with bolt threading operation; program execution suspended until task completed and time measurements obtained.
211 - 229	Program directs robot arm to remove motor from bolt containment plate and place into finished assembly bin; time measurement recorded.
230 - 234	Assembly cycle completed; time measurements are summed for the assembly sequence; cycle count incremented.
236 - 240	Operator prompted if elasped time for assembly sequence is to be displayed; if "yes" selected, program control continues at line 241; if "no" selected, program control jumps to line 255.

Line Numbers	Description
241 - 254	Program displays time measurements obtained during the assembly cycle.
255 - 259	Operator prompted to repeat program execution; if "yes" selected, program control returns to line 53; if "no" selected, program execution ends at line 260.
260	Program termination.

APPENDIX D

PROGRAM FLOWCHART (SINGLE ARM ASSEMBLY)

PROGRAM: ONEARM



APPENDIX E

PROGRAM LISTING (SINGLE ARM ASSEMBLY)

```
.PROGRAM ONEARM
;◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
;
;       VAL II ROBOT CONTROL PROGRAM: "ONEARM"
;
;       DESCRIPTION: PROGRAM TO DIRECT PUMA 762 SERIES ROBOT IN
;                    CONJUNCTION WITH A HUMAN "WORKER" TO ASSEMBLE
;                    ELECTRIC MOTOR (DAYTON ELECTRIC MFG. STK#3M569)
;
;       PROGRAMMER:  J. REID            DATE: 11-4-86
;
;
;       VARIABLES LIST                  DESCRIPTION
;
;       ATIME (TIMER1)              VARIABLE USED TO TIME PUMA ARM MOTION
;       STIME (TIMER2)              VARIABLE USED TO TIME ASSY. CYCLE
;       HAND.TIME                   SETS DELAY TO ENABLE GRIPPER TIME
;                                   TO OPEN/CLOSE BETWEEN ARM MOVEMTS.
;       ANS, ANSR                   STORES USER RESPONSE TO PROMPTS
;       CYCLE                       COUNTER TO TRACK NUMBER OF
;                                   CONSECUTIVE ASSEMBLY CYCLES
;       TOTIME                      STORES TOTAL ASSY. TIME (ALL CYCLES)
;       ARMTIME                     STORES TOTAL ARM MOVEMENT TIME PER CYCLE
;       ARMTOT                      STORES TOTAL ARM MOVEMENT TIME (ALL CYCLES)
;       TASK1                       TIME TO PLACE FRONT BELL, STACK STATOR,
;                                     GRIP END BELL AND MOVE TO POINT 5
;       TASK2                       (HUMAN) TIME TO INSERT SHAFT INTO MOTOR,
;                                   AND THREAD POWER LEADS THROUGH END BELL
;       TASK3                       TIME TO PLACE END BELL ONTO MOTOR
;       TASK4                       (HUMAN) TIME TO PLACE 4 BOLTS IN MOTOR
;       TASK5                       TIME TO MOVE MOTOR INTO POSITION AGAINST
;                                   BOLT CONTAINMENT PLATE
;       TASK6                       (HUMAN) TIME TO THREAD NUTS ONTO BOLTS
;       TASK7                       TIME TO PLACE FINISHED MOTOR INTO BIN
;
;◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
;
;               ◆ ◆ ◆ SET INITIAL PARAMETERS ◆ ◆ ◆
;
        LEFTY;                                  SET CONFIGURATION
        READY;                                  MOVE TO READY POSITION
        HAND.TIME = 36;                         SET GRIPPER FOR 1-SEC DELAY
        RESET;                                  CLEAR ALL EXTERNAL SIGNALS
        SPEED 1000 MMPS ALWAYS;                 SET SPEED VALUE
        CYCLE = 0;                              SET CYCLE COUNT TO 0
        TOTIME = 0;                             SET TOT. ASSY. TIME = 0
        ARMTOT = 0;                             SET TOTAL ARM TIME = 0
;
;   ◆ ◆ ◆ DISPLAY INITIAL PROMPTS, ASK USER IF FIXTURE LOCATION IS
;         TO BE DEFINED ◆ ◆ ◆
;
        TYPE /B, /C10;                          SCROLL SCREEN
        TYPE /C8, "◆ ◆ ◆ VAL II ROBOT CONTROL PROGRAM: ONEARM ◆ ◆ ◆"
     5  PROMPT "FIXTURE ORIENTATION? (1=YES, 'RETURN'=NO)", ANS
        IF ANS == 0 GOTO 30
        IF ANS == 1 GOTO 10
        GOTO 5
;
```

```
;                ◆ ◆ ◆ FIXTURE  LOCATION ROUTINE ◆ ◆ ◆
;
10  MOVE INTP1;                      MOVE FIRST TO INTERMEDIATE PT.
    MOVEST FIXTR.P1, 0;              STOP AT APPROACH POINT
    SPEED 10 MMPS ALWAYS;            SLOW SPEED DOWN
    MOVEST FIXTR.P2, 0;              MOVE TO ORIENTATION POSITION
    BREAK;                           STOP CONTINUOUS PATH MOTION
    TYPE /C1, /B, "LOCATE FIXTURE AT FRONT POSITION, PRESS RETURN"
    PROMPT "", ANSR
    MOVEST FIXTR.P1, 0;              WITHDRAW SLIGHTLY
    BREAK
    SPEED 300 MMPS ALWAYS;           INCREASE SPEED TO NORMAL
    MOVEST FIXTR.P3, 0;              MOVE TO END OF FIXTURE
    BREAK
    SPEED 10 MMPS ALWAYS;            SLOW SPEED
    MOVEST FIXTR.P4, 0;              MOVE TO FINAL POSITION
    BREAK
    TYPE /C1, /B, "LOCATE FIXTURE AT REAR POSITION, PRESS RETURN"
    PROMPT "", ANSR
    MOVEST FIXTR.P3, 0;              ALIGNMENT COMPLETE, WITHDRAW
    BREAK
    SPEED 300 MMPS ALWAYS;           INCREASE SPEED TO NORMAL
    MOVEST INTP1, 0;                 GO BACK TO INTERMEDIATE POSITION
    SET Z = FRAME(P10, P1, P7, P10)
    DETACH
    TYPE "USE TEACH PENDANT TO MOVE ROBOT TO POSTITION AT POINT 10"
    TYPE "(GRIP END BELL) AND PRESS RETURN"
    PROMPT " "
    ATTACH
    HERE P10;                        DEFINE COORDINATE ORIGIN POINT
    DETACH
    TYPE "USE TEACH PENDANT TO MOVE ROBOT TO POSITION AT POINT 1"
    TYPE "(GRIP FRONT BELL) AND PRESS RETURN"
    PROMPT " "
    ATTACH
    HERE P1;                         DEFINE POINT ALONG X-AXIS
    DETACH
    TYPE "USE TEACH PENDANT TO POSTION AT POINT 7"
    TYPE "(GRIP STATOR) AND PRESS RETURN"
    PROMPT " "
    ATTACH
    HERE P7;                         DEFINE POINT ALONG Y-AXIS
    DETACH
    TYPE "USE TEACH PENDANT TO MOVE ROBOT ARM TO CLEAR POSITION"
    TYPE "ABOVE FIXTURE AND PRESS RETURN"
    PROMPT " "
    ATTACH
20  TYPE /B, "PROCEED OR RETRY ALIGNMENT? (1=RETRY, 'RETURN'=CONTINUE)"
    PROMPT "", ANSR
    IF ANSR == 1 GOTO 10
    IF ANSR == 0 GOTO 30
    GOTO 20
;
;   ◆ ◆ ◆ FIXTURE CORRECTLY ORIENTED, BEGIN ASSEMBLY SEQUENCE ◆ ◆ ◆
;
30  MOVE STRTP1;                     MOVE TO SEQUENCE STARTING POINT
35  TYPE /C2, /B, "SELECT: 1=ABORT RUN, 'RETURN'=EXECUTE SEQUENCE"
    PROMPT "", ANS
    IF ANS == 0 GOTO 40
    IF ANS == 1 GOTO 70
    GOTO 35
;
```

```
40  TIMER (2) = 0;                  INITIALIZE TIMERS
    TIMER (1) = 0
    ARMTIME = 0
;
;           * * * BEGIN ASSEMBLY SEQUENCE * * *
;
    MOVEST P1, 50.81;               MOVE TO GRASP FRONT BELL
    BREAK
    CLOSEI
    MOVEST P2, 0;                   LIFT FRONT BELL
    BREAK
    MOVEST P3, 0;                   MOVE OVER CENTER LOCATOR BLOCK
    MOVEST P4, 0;                   PLACE FRONT BELL ONTO BLOCK
    BREAK
    OPENI
    MOVEST P5, 50.81;               WITHDRAW
    MOVEST P6, 50.81;               MOVE TO PICK UP STATOR
    MOVEST P7, 50.81;
    BREAK
    CLOSEI
    MOVEST P6, 0;                   LIFT STATOR
    BREAK
    MOVEST P3, 0;                   MOVE BACK TO PLACE ONTO FRONT BELL
    BREAK
    SPEED 50 MMPS ALWAYS;           SLOW SPEED
    MOVEST P8, 0;                   PLACE STATOR ONTO FRONT BELL
    BREAK
    OPENI
    SPEED 1000 MMPS ALWAYS;         INCREASE TO NORMAL SPEED
    MOVEST P3, 50.81;               WITHDRAW
    MOVEST P9, 50.81;               MOVE TO PICK UP END BELL
    MOVEST P10, 50.81;
    BREAK
    CLOSEI
    MOVEST P9, 0;                   LIFT END BELL
    MOVEST P5, 0;                   MOVE TO PLACE ONTO STATOR
    BREAK
    ATIME = TIMER(1);               COLLECT TIME STATS.
    TASK1 = TIMER(1)
    ARMTIME = ARMTIME+ATIME
    TIMER (3) = 0
    DETACH
    TYPE /B, "SWITCH OFF ARM POWER. INSERT SHAFT INTO MOTOR ASSEMBLY"
    TYPE "AND INSERT WIRE LEADS THROUGH END BELL. AFTER COMPLETION,"
    TYPE "SWITCH ON ARM POWER AND PRESS RETURN ON KEYBOARD."
    PROMPT "", ANS
    TASK2 = TIMER(3);               COLLECT TIME STAT.
    WAIT STATE(3) == 5;             WAIT UNTIL COMP. MODE ACTIVATED
    ATTACH;
    TIMER (1) = 0;                  RESET ARM TIMER
    MOVEST INTP2, 0;                MOVE TO PT. JUST ABOVE STATOR
    SPEED 15 MMPS ALWAYS;           REDUCE SPEED
    MOVEST P11, 0;                  PLACE END BELL ONTO STATOR
    BREAK
    OPENI
    SPEED 1000 MMPS ALWAYS;         INCREASE SPEED TO NORMAL
    MOVEST P12, 50.81;              WITHDRAW 40 CM ABOVE MOTOR
    MOVEST P13, 50.81
    BREAK
```

```
        ATIME = TIMER(1)
        TASK3 = TIMER(1);               COLLECT TIME STATS.
        ARMTIME = ARMTIME+ATIME
        TIMER (3) = 0
        DETACH
        TYPE /B, "TURN OFF ARM POWER. INSERT BOLTS INTO MOTOR. TURN ARM"
        TYPE "POWER BACK ON AND PRESS RETURN WHEN TASK COMPLETED."
        PROMPT "", ANS
        TASK4 = TIMER(3);               COLLECT TIME STAT.
        WAIT STATE(3) == 5;             WAIT UNTIL COMP. MODE ACTIVATED
        ATTACH
        TIMER (1) = 0
        MOVEST P14, 50.81;              MOVE TO GRASP MOTOR
        MOVEST P15, 50.81
        BREAK
        CLOSEI
        MOVEST P16, 0;                  LIFT MOTOR
        MOVET P17, 0
        BREAK
        MOVEST P18, 0
        BREAK
        MOVEST P19, 0;                  POSITION MOTOR AGAINST BOLT
        BREAK;                          CONTAINMENT PLATE
        ATIME = TIMER(1)
        TASK5 = TIMER(1);               COLLECT TIME STATS.
        ARMTIME = ARMTIME+ATIME
        TIMER (3) = 0
        DETACH
        TYPE /B, "TURN OFF ARM POWER. THREAD NUTS ONTO BOLTS AND TIGHTEN."
        TYPE "WHEN COMPLETED, TURN ON ARM POWER AND PRESS RETURN."
        PROMPT "", ANS
        TASK6 = TIMER(3);               COLLECT TIME STAT.
        WAIT STATE(3) == 5;             WAIT UNTIL COMP. MODE ACTIVATED
        ATTACH
        TIMER (1) = 0
        MOVEST P18, 0;                  WITHDRAW FROM BOLT CONT. PLATE
        BREAK
        MOVEST P17, 0;                  MOVE FINISHED MOTOR TO BIN
        MOVET P20, 0
        MOVET P21, 0
        MOVEST P22, 0;                  AND DEPOSIT IN BIN
        BREAK
        OPENI
        TASK7 = TIMER(1)
        MOVEST P21, 50.81;              WITHDRAW FROM BIN
        MOVEST STRTP1, 0;               MOVE BACK TO SEQUENCE START POINT
        BREAK
;
;               * * * ASSEMBLY SEQUENCE COMPLETED * * *
;
        ATIME = TIMER(1);               OBTAIN ARM TIME
        STIME = TIMER(2);               OBTAIN CYCLE TIME
        ARMTIME = ARMTIME+ATIME;        GET TOTAL ARM TIME FOR CYCLE
        ARMTOT = ARMTOT+ARMTIME;        GET TOTAL ARM TIME (ALL CYCLES)
        TOTIME = TOTIME+STIME;          AND TOTAL ASSY. TIME (ALL CYCLES)
        CYCLE = CYCLE+1;                INCREMENT CYCLE COUNT
;
    50  TYPE /C1, /B, "DISPLAY ELAPSED TIME? (1=NO, 'RETURN'=YES)"
        PROMPT "", ANS
        IF ANS == 0 GOTO 60
        IF ANS == 1 GOTO 70
        GOTO 50
```

```
 60  TYPE /C3, "ELAPSED TIME FOR CYCLE # ", CYCLE
     TYPE /C5, " "
     TYPE "TASK1 TIME = ", TASK1
     TYPE "TASK2 TIME = ", TASK2
     TYPE "TASK3 TIME = ", TASK3
     TYPE "TASK4 TIME = ", TASK4
     TYPE "TASK5 TIME = ", TASK5
     TYPE "TASK6 TIME = ", TASK6
     TYPE "TASK7 TIME = ", TASK7
     TYPE /C3, " "
     TYPE "ASSEMBLY SEQUENCE ELAPSED TIME = ", STIME
     TYPE "ELAPSED ARM MOVEMENT TIME = ", ARMTIME
     TYPE "TOTAL ARM MOVEMENT TIME FOR ", CYCLE, " CYCLES = ", ARMTOT
     TYPE "TOTAL ASSEMBLY TIME FOR ", CYCLE, " CYCLES = ", TOTIME
 70  TYPE /C5, "SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN"
     PROMPT "", ANS
     IF ANS == 9 GOTO 100
     IF ANS == 0 GOTO 5
     GOTO 70
100  STOP
.END
.
```

APPENDIX F

PROGRAM DESCRIPTIONS FOR DUAL ARM ASSEMBLY

PROGRAM: MTR.PATH2

Line Numbers	Description
1 - 35	Program Header section; includes program title, a brief description of purpose, the programmer's name and coding date, a listing of the variables used in the program and their corresponding function.
39 - 47	Initial parameters are defined in this segment; robot configuration and operating parameters, such as arm speed, gripper delay, etc., are set. Variables used to store assembly time measurements are initialized.
52 - 57	Program prompts robot operator if fixture location/frame transformation are to be defined. If "yes" selected, program control continues at line 61; if "no" selected, program control branches to line 115.
61 - 79	Program directs robot arm to move to a fixture orientation location at each end of the fixture where the operator is then prompted to position the appropriate end of the assembly fixture against the edge of the gripper to obtain an approximate alignment of the fixture.
80 - 105	Program prompts operator to position robot arm with the teach pendant to three different points which define an x - y coordinate system; a FRAME command is then executed to enable the robot to correctly reference all other points defined in the assembly routine.
106 - 110	Operator is prompted either to proceed with execution, or retry alignment procedure; if "retry" selected, program control returns to line 61; if "proceed" selected, program control advances to line 115.
115 - 120	Program directs PUMA.2 to move to sequence starting point and sends external signal to PUMA.1 controller allowing PUMA.1 to move to its start point. Program execution halts until PUMA.1 has completed move. Communication channels are reset.

Line Numbers	Description
124 - 128	Operator is prompted to either proceed with assembly sequence execution or abort the run; if "proceed" selected, program control continues to line 131; if "abort" selected, program control branches to line 277.
131 - 134	Program initializes timer function variables.
135 - 176	Assembly sequence initiated; program directs PUMA.2 arm to grasp and place front bell onto center locator block, then grasp stator and place onto front bell, then move to grasp end bell. An external signal is sent to PUMA.1 controller to direct PUMA.1 to grasp and insert shaft into motor. Time measurements are collected upon task completion by PUMA.1 and PUMA.2.
177 - 192	Program directs PUMA.2 to move end bell down over stator lead plug, threading the stator leads through grommet in end bell. An external signal is then sent to PUMA.1 controller to direct coordinated PUMA.1 movement in the operation. Program execution is halted until a signal is received from PUMA.1 controller, then PUMA.2 arm is directed to move toward a position over center locator block; time measurements are obtained.
193 - 200	Program directs PUMA.2 to place end bell onto shaft and stator; signal sent to PUMA.1 to release stator leads; time measurements are obtained.
201 - 210	PUMA.2 directed to withdraw from motor; program execution suspended until PUMA.1 has inserted bolts into motor; time measurements collected.
211 - 231	PUMA.2 directed to grasp motor, place into location against bolt containment plate; signal sent to PUMA.1 and program execution halted until nut threading operation completed; time measurements collected.
232 - 244	PUMA.2 directed to remove motor from containment plate and place finished motor

Line Numbers	Description
232 - 244 (cont.)	into inspection bin; signal is sent to PUMA.1 to direct PUMA.1 towards sequence finish point location; time measurements recorded.
245 - 250	PUMA.2 directed to return to sequence start location; program execution suspended until PUMA.1 has completed move; time measurements are obtained.
254 - 256	Assembly cycle completed; time measurements are summed for the assembly sequence; cycle count incremented.
258 - 262	Operator prompted if elapsed time for assembly sequence is to be displayed; if "yes" selected, program control continues at line 263; if "no" selected, program control jumps to line 277.
263 - 276	Program displays time measurements obtained during the assembly cycle.
277 - 281	Operator prompted to repeat program execution; if "yes" selected, program control returns to line 54; if "no" selected, program execution ends at line 282.
282	Program termination.

PROGRAM: MTR.PATH1

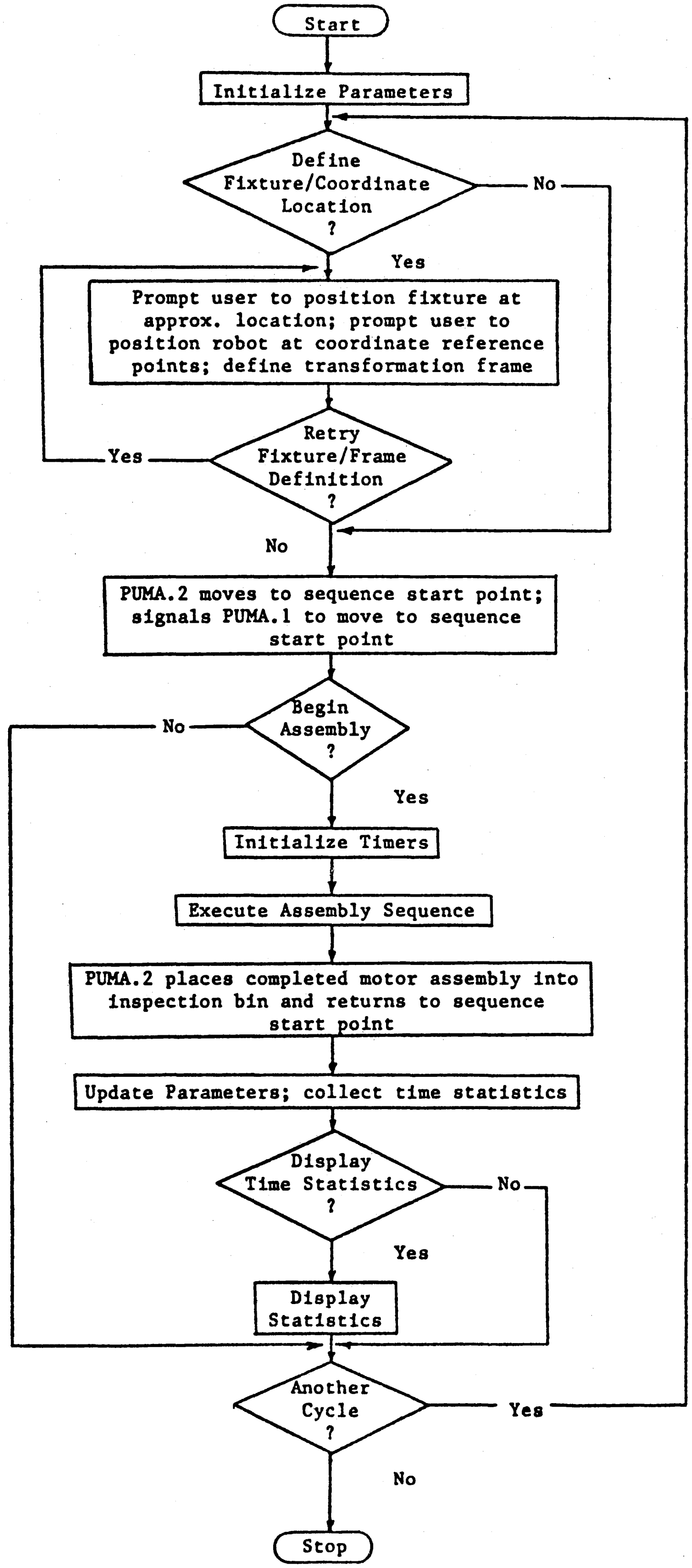
Line Numbers	Description
1 - 18	Program Header section; includes program title, a brief description of purpose, the programmer's name and coding date, a listing of the variables used in the program and their corresponding function.
22 - 26	Initial parameters are defined in this segment; robot configuration and operating parameters, such as arm speed, gripper delay, etc., are set.
32 - 36	Program prompts robot operator if coordinate reference frame is to be defined; if "yes" selected, program control continues at line 40; if "no" selected, program control branches to line 69.
40 - 60	Program prompts operator to position robot arm with the teach pendant to three different points which define an x - y coordinate system; a FRAME command is then executed to enable the robot to correctly reference all other points defined in the assembly routine.
61 - 65	Operator is prompted to either proceed with execution or retry coordinate definition procedure; if "retry" selected, program control returns to line 40; if "proceed" selected, program control advances to line 69.
69 - 73	Program directs PUMA.1 to move to sequence start point after receiving signal from PUMA.2 controller; after move is completed, a signal is returned to PUMA.2 so that assembly can proceed.
77 - 94	Program execution suspended until start signal received from PUMA.2; program then directs PUMA.1 to grasp and place shaft assembly into motor, then withdraw and position arm into location for the next task. After move completed, signal sent to PUMA.2 to begin task.
95 - 107	Program execution suspended while PUMA.2 manipulates end bell over stator leads.

Line Numbers	Description
95 - 107 (cont.)	Upon receipt of signal from PUMA.2, program directs PUMA.1 to grasp stator leads. A signal is then sent to PUMA.2 to initiate coordinated arm movement towards center locator block. Upon completion of the move, another signal is sent to initiate placement of the end bell by PUMA.2, while program execution is halted until operation completed; PUMA.1 is then directed to release stator leads.
108 - 163	PUMA.1 directed to repeat bolt insertion sequence; for each bolt inserted, the robot arm moves to grasp bolt, grasps bolt, withdraws, moves to insert bolt into motor, reduces speed, inserts bolt into motor and releases. Arm speed is then increased, and the insertion sequence is repeated for the remaining bolts.
164 - 235	PUMA.1 directed to grasp flexible cable and move to grasp first nut to be threaded; signal is sent to PUMA.2 to place motor against bolt containment plate. For each nut to be threaded, the robot arm moves to the nut pickup point, reduces speed, slips magnetic socket over nut, withdraws, increases arm speed, moves to thread nut onto appropriate bolt, decreases speed, threads nut onto bolt, increases speed, and withdraws. The threading sequence is then repeated for remaining nuts. After completion of the operation, PUMA.1 is directed to replace flexible cable into receptacle; signal sent to PUMA.2 that task completed.
237 - 240	Assembly cycle completed; PUMA.1 directed to move to sequence final point; signal sent to PUMA.2 that move completed.
244 - 248	Operator prompted to repeat program execution; if "yes" selected, program control returns to line 33; if "no" selected, program execution ends at line 249.
249	Program termination.

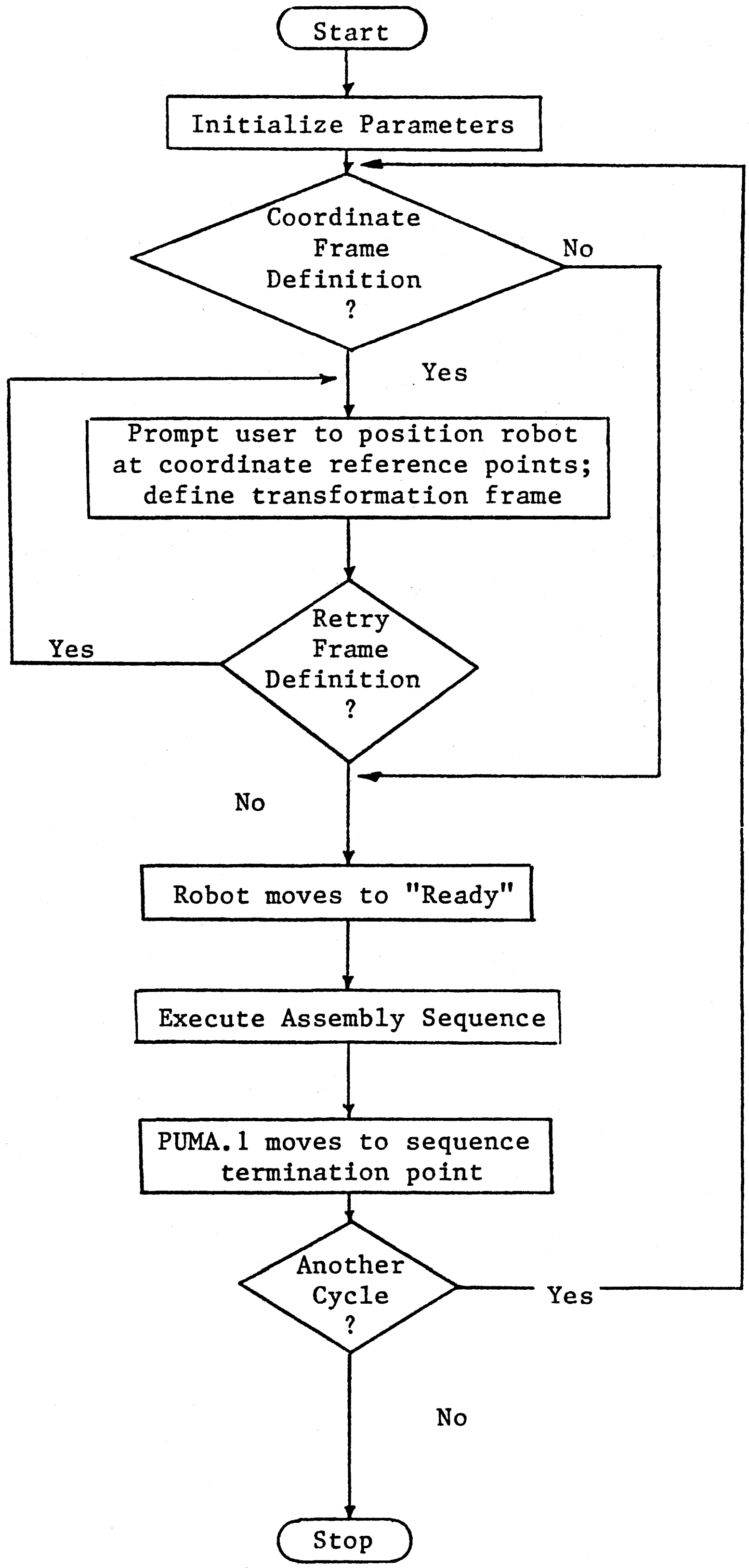
APPENDIX G

PROGRAM FLOWCHARTS (DUAL ARM ASSEMBLY)

PROGRAM: MTR.PATH.2



PROGRAM: MTR.PATH.1



APPENDIX H

PROGRAM LISTINGS (DUAL ARM ASSEMBLY)

```
.PROGRAM MTR.PATH2
;*****************************************************************
;
;       VAL II ROBOT CONTROL PROGRAM: "MTR.PATH2"
;
;       DESCRIPTION: PROGRAM TO DIRECT PUMA#2  ROBOT IN
;                    COORDINATION WITH PUMA#1 TO ASSEMBLE
;                    ELECTRIC MOTOR (DAYTON ELECTRIC MFG. STK#3M569)
;
;       PROGRAMMER:  J. REID              DATE: 11-4-86
;
;
;       VARIABLES LIST                      DESCRIPTION
;       R1TIME (TIMER1)          VARIABLE USED TO TIME PUMA.1 ARM TIME
;       R2TIME (TIMER2)           VARIABLE USED TO TIME PUMA.2 ARM TIME
;       STIME (TIMER3)           VAR. USED TO TIME ENTIRE ASSY. SEQUENCE
;       HAND.TIME                SETS DELAY TO ENABLE GRIPPER TIME
;                                TO OPEN/CLOSE BETWEEN ARM MOVEMTS.
;       ANS, ANSR                STORES USER RESPONSE TO PROMPTS
;       CYCLE                    COUNTER TO TRACK NUMBER OF
;                                CONSECUTIVE ASSEMBLY CYCLES
;       TOTIME                   STORES TOTAL ASSY. TIME (ALL CYCLES)
;       ARM1TIME                 STORES PUMA.1 ARM MOVEMENT TIME PER CYCLE
;       ARM2TIME                 STORES PUMA.2 ARM MOVEMENT TIME PER CYCLE
;       TASK1                    TIME TO PLACE FRONT BELL, STACK STATOR,
;                                  GRIP END BELL AND MOVE TO POINT 10
;       TASK2                    TIME TO THREAD STATOR PLUG INTO END BELL
;                                AND POSITION END BELL OVER STATOR
;       TASK3                    TIME TO PLACE END BELL ONTO MOTOR
;       TASK4                    TIME TO PLACE 4 BOLTS INTO MOTOR
;       TASK5                    TIME TO MOVE MOTOR INTO POSITION AGAINST
;                                BOLT CONTAINMENT PLATE
;       TASK6                    TIME TO THREAD NUTS ONTO BOLTS
;       TASK7                    TIME TO PLACE FINISHED MOTOR INTO BIN
;
;*****************************************************************
;
;               * * * SET INITIAL PARAMETERS * * *
;
        LEFTY;                          SET CONFIGURATION
        READY;                          MOVE TO READY POSITION
        HAND.TIME = 36;                 SET GRIPPER FOR 1-SEC DELAY
        RESET;                          RESET ALL EXTERNAL SIGNALS
        SPEED 1000 MMPS ALWAYS;         SET NORMAL SPEED VALUE
        CYCLE = 0;                      SET CYCLE COUNT TO 0
        TOTIME = 0;                     SET TOTAL ASSY. TIME = 0
        ARM1TIME = 0;                   SET ARM1 TOTAL TIME = 0
        ARM2TIME = 0;                   SET ARM2 TOTAL TIME = 0 0
;
;   * * * DISPLAY INITIAL PROMPTS, ASK USER IF FIXTURE LOCATION IS
;       TO BE DEFINED * * *
;
        TYPE /B, /C10;                  SCROLL SCREEN
        TYPE /C8, "* * * VAL II ROBOT CONTROL PROGRAM         * * *"
     5  PROMPT "FIXTURE ORIENTATION? (1=YES, 'RETURN'=NO)", ANS
        IF ANS == 0 GOTO 30
        IF ANS == 1 GOTO 10
        GOTO 5
;
```

```
;                  * * * FIXTURE  LOCATION ROUTINE * * *
;
    10  MOVE INTP1;                     MOVE FIRST TO INTERMEDIATE POINT
        MOVEST FIXTR.P1, 0;             STOP AT APPROACH POINT
        SPEED 10 MMPS ALWAYS;           SLOW SPEED DOWN
        MOVEST FIXTR.P2, 0;             MOVE TO ORIENTATION POSITION
        BREAK;                          STOP CONTINUOUS PATH MOTION
        TYPE /C1, /B, "LOCATE FIXTURE AT FRONT POSITION, PRESS RETURN"
        PROMPT "", ANSR
        MOVEST FIXTR.P1, 0;             WITHDRAW SLIGHTLY
        BREAK
        SPEED 300 MMPS ALWAYS;          INCREASE SPEED
        MOVEST FIXTR.P3, 0;             MOVE TO END OF FIXTURE
        BREAK
        SPEED 10 MMPS ALWAYS;           SLOW SPEED DOWN
        MOVEST FIXTR.P4, 0;             MOVE TO FINAL POSITION
        BREAK
        TYPE /C1, /B, "LOCATE FIXTURE AT REAR POSITION, PRESS RETURN"
        PROMPT "", ANSR
        MOVEST FIXTR.P3, 0;             ALIGNMENT COMPLETE, WITHDRAW
        BREAK
        SPEED 300 MMPS ALWAYS;          INCREASE SPEED
        MOVEST INTP1, 0;                MOVE BACK TO INTERMEDIATE POSITION
        SET Z = FRAME(P10, P1, P7, P10)
        DETACH
        TYPE "USE TEACH PENDANT TO MOVE ROBOT TO POSTITION AT POINT 10"
        TYPE "(GRIP END BELL) AND PRESS RETURN"
        PROMPT " "
        ATTACH
        HERE P10;                       DEFINE COORDINATE ORIGIN POINT
        DETACH
        TYPE "USE TEACH PENDANT TO MOVE ROBOT TO POSITION AT POINT 1"
        TYPE "(GRIP FRONT BELL) AND PRESS RETURN"
        PROMPT " "
        ATTACH
        HERE P1;                        DEFINE POINT ALONG X-AXIS
        DETACH
        TYPE "USE TEACH PENDANT TO POSTION AT POINT 7"
        TYPE "(GRIP STATOR) AND PRESS RETURN"
        PROMPT " "
        ATTACH
        HERE P7;                        DEFINE POINT ALONG Y-AXIS
        DETACH
        TYPE "USE TEACH PENDANT TO MOVE ROBOT ARM TO CLEAR POSITION"
        TYPE "ABOVE FIXTURE AND PRESS RETURN"
        PROMPT " "
        ATTACH
    20  TYPE /B, "PROCEED OR RETRY ALIGNMENT? (1=RETRY, 'RETURN'=CONTINUE)"
        PROMPT "", ANSR
        IF ANSR == 1 GOTO 10
        IF ANSR == 0 GOTO 30
        GOTO 20
;
;
;  * * * FIXTURE CORRECTLY ORIENTED, MOVE ROBOTS TO TASK READY POSTN * * *
;
    30  SPEED 1000 MMPS ALWAYS;         RESET SPEED TO NORMAL
        MOVE STRTP1;                    MOVE PUMA.2 TO SEQUENCE START POINT
        BREAK
        SIGNAL 1;                       SIGNAL PUMA.1 TO SEQUENCE START PT.
        WAIT SIG(1001);                 WAIT UNTIL PUMA.1 IN POSITION
        RESET;                          CLEAR ALL I/O SIGNAL CHANNELS
```

```
;
;    * * * ROBOTS AT THE READY, PROMPT FOR SEQUENCE EXECUTION * * *
;
     35  TYPE /C2, /B, "SELECT: 1=ABORT RUN OR 'RETURN'=EXECUTE SEQUENCE"
         PROMPT " ", ANS
         IF ANS == 0 GOTO 40
         IF ANS == 1 GOTO 70
         GOTO 35
;          * * *   BEGIN ASSEMBLY SEQUENCE * * *
;
     40  ARMTIME = 0;                     INITIALIZE TIMERS
         TIMER (2) = 0
         TIMER (3) = 0
         TIMER (4) = 0
         MOVEST P1, 50.81;                BEGIN ASSEMBLY, MOVE TO FRONT BELL
         BREAK
         CLOSEI
         MOVEST P2, 0;                    LIFT FRONT BELL
         BREAK
         MOVEST P3, 0;                    MOVE OVER CENTER LOCATOR BLOCK
         MOVEST P4, 0;                    PLACE FRONT BELL ONTO BLOCK
         BREAK
         OPENI
         MOVEST P5, 50.81;                WITHDRAW
         MOVEST P6, 50.81;                MOVE TO PICK UP STATOR
         MOVEST P7, 50.81
         BREAK
         CLOSEI
         MOVEST P6, 0;                    PICK UP STATOR
         BREAK
         MOVEST P3, 0;                    MOVE BACK TO PLACE ONTO FRONT BELL
         BREAK
         SPEED 60 MMPS ALWAYS;            SLOW SPEED
         MOVEST P8, 0;                    PLACE STATOR ONTO FRONT BELL
         BREAK
         OPENI
         SPEED 1000 MMPS ALWAYS;          INCREASE SPEED BACK TO NORMAL
         MOVEST P3, 50.81;                WITHDRAW
         MOVEST P9, 50.81;                MOVE TO PICK UP END BELL
         SIGNAL 1;                        SIGNAL PUMA.1 TO PICK UP SHAFT
         TIMER (1) = 0;                      SET PUMA.1 ARM TIMER
         MOVEST P10, 50.81
         BREAK
         CLOSEI
         MOVEST P9, 0;                    LIFT END BELL
         R2TIME = TIMER(2);                 GET ARM2 TIME
         ARM2TIME = ARM2TIME+R2TIME
         WAIT SIG(1001);                  WAIT UNTIL PUMA.1 IS READY
         R1TIME = TIMER(1);                 GET ARM1 TIME
         ARM1TIME = ARM1TIME+R1TIME
         TIMER (2) = 0;                      REINITIALIZE ARM2 TIMER
         RESET;                           CLEAR I/O CHANNEL
         MOVEST P10, 0;                   APPROACH STATOR LEAD PLUG FROM TOP
         BREAK
         TASK1 = TIMER(4);                  OBTAIN TASK1 TIME
         TIMER (4) = 0
         MOVEST P11, 0;                   MOVE END BELL DOWN TO THREAD PLUG
         BREAK;                           THROUGH GROMMET
         SIGNAL 1;                        SIGNAL PUMA.1 TO GRASP LEADS
         TIMER (1) = 0;                      REINITIALIZE ARM1 TIMER
         R2TIME = TIMER(2);                 GET ARM2 TIME
         ARM2TIME = ARM2TIME+R2TIME
```

```
WAIT SIG(1001);                 WAIT UNTIL PUMA.1 IS READY
TIMER (2) = 0
SIGNAL 2;                       SIGNAL PUMA.1 TO BEGIN A
MOVEST P5, 0;                   COORDINATED MOVEMENT TOWARDS
BREAK;                          CENTER LOCATOR BLOCK
MOVEST INTP2, 0
WAIT SIG(1002);                 STOP UNTIL PUMA.1 IS IN POSITION
R1TIME = TIMER(1);                GET ARM1 TIME
ARM1TIME = ARM1TIME+R1TIME
TASK2 = TIMER(4);               GET TASK2 TIME
TIMER (4) = 0
SPEED 20 MMPS ALWAYS;           REDUCE SPEED
RESET;                          CLEAR I/O CHANNELS
MOVEST P12, 0;                  PLACE END BELL ONTO STATOR
BREAK
SIGNAL 1;                       SIGNAL PUMA.1 TO RELEASE LEADS
TIMER (1) = 0;                     REINITIALIZE ARM1 TIMER
TASK3 = TIMER(4);                 GET TASK3 TIME
TIMER (4) = 0
OPENI
SPEED 1000 MMPS ALWAYS;         INCREASE SPEED BACK TO NORMAL
MOVEST P13, 50.81;
MOVEST P14, 50.81;              WITHDRAW FROM MOTOR
SIGNAL 2;                       SIGNAL PUMA.1 TO INSERT BOLTS
R2TIME = TIMER(2);                GET ARM2 TIME
ARM2TIME = ARM2TIME+R2TIME
WAIT SIG(1002);                 WAIT UNTIL PUMA.1 IS FINISHED
TASK4 = TIMER(4);                 GET TASK4 TIME
TIMER (4) = 0
RESET
TIMER (2) = 0;                     REINITIALIZE ARM2 TIMER
MOVEST P15, 50.81
MOVEST P16, 50.81;              GRASP MOTOR
BREAK
CLOSEI
MOVEST P17, 0;                  BEGIN MOVEMENT TOWARDS BOLT
MOVEST P18, 0;                  CONTAINMENT PLATE
BREAK
MOVEST P19, 0
BREAK
MOVEST P20, 0;                  PLACE MOTOR AGAINST BOLT
BREAK;                          CONTAINMENT PLATE
R2TIME = TIMER(2);                GET ARM2 TIME
ARM2TIME = ARM2TIME+R2TIME
TASK5 = TIMER(4);                 AND TASK5 TIME
TIMER (4) = 0
SIGNAL 1;                       SIGNAL PUMA.1 TO THREAD NUTS
WAIT SIG(1001);                 WAIT UNTIL TASK COMPLETE
TASK6 = TIMER(4);                 GET TASK6 TIME
TIMER (4) = 0
TIMER (2) = 0;                     REINITIALIZE ARM2 TIMER
MOVEST P19, 0;                  BEGIN MOVEMENT TOWARDS BIN
BREAK
MOVEST P18, 0
MOVEST P21, 0
MOVEST P22, 0
MOVEST P23, 0
BREAK;                          MOTOR ASSEMBLY COMPLETE, RELEASE
OPENI;                          MOTOR INTO BIN FOR SUBSEQUENT INSP.
```

```
        R2TIME = TIMER(2);                GET ARM2 TIME
        ARM2TIME = ARM2TIME+R2TIME
        TASK7 = TIMER(4);                 AND TASK7 TIME
        MOVEST P22, 50.81;              AND WITHDRAW
        MOVEST STRTP1, 0;               MOVE BACK TO SEQUENCE START POINT
        WAIT SIG(1001);                   WAIT UNTIL ARM1 HAS FINISHED MOVE
        R1TIME = TIMER(1);                GET ARM1 TIME
        ARM1TIME = ARM1TME+R1TIME
        BREAK
;
;                * * * ASSEMBLY SEQUENCE COMPLETED * * *
;
        STIME = TIMER(3);                 OBTAIN CYCLE TIME
        TOTIME = TOTIME+STIME;          AND TOTAL TIME (ALL CYCLES)
        CYCLE = CYCLE+1;                INCREMENT CYCLE COUNT
;
    50  TYPE /C1, /B, "DISPLAY ELAPSED TIME? (1=NO, 'RETURN'=YES)"
        PROMPT "", ANS
        IF ANS == 0 GOTO 60
        IF ANS == 1 GOTO 70
        GOTO 50
    60  TYPE /C3, "ELAPSED TIME FOR CYCLE # ", CYCLE
        TYPE /C5, " "
        TYPE "TASK1 TIME = ", TASK1
        TYPE "TASK2 TIME = ", TASK2
        TYPE "TASK3 TIME = ", TASK3
        TYPE "TASK4 TIME = ", TASK4
        TYPE "TASK5 TIME = ", TASK5
        TYPE "TASK6 TIME = ", TASK6
        TYPE "TASK7 TIME = ", TASK7
        TYPE /C3, " "
        TYPE "ASSEMBLY SEQUENCE ELAPSED TIME = ", STIME
        TYPE "ELAPSED ARM1 MOVEMENT TIME = ", ARM1TIME
        TYPE "ELAPSED ARM2 MOVEMENT TIME = ", ARM2TIME
        TYPE "TOTAL ASSEMBLY TIME FOR ", CYCLE, " CYCLES = ", TOTIME
    70  TYPE /C5, "SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN"
        PROMPT "", ANS
        IF ANS == 9 GOTO 100
        IF ANS == 0 GOTO 5
        GOTO 70
   100  STOP
.END
.
```

```
.PROGRAM MTR.PATH1
;*********************************************************************
;
;       VAL II ROBOT CONTROL PROGRAM: "MTR.PATH1"
;
;       DESCRIPTION: PROGRAM TO DIRECT PUMA#1  ROBOT IN
;                    COORDINATION WITH PUMA#2 TO ASSEMBLE
;                    ELECTRIC MOTOR (DAYTON ELECTRIC MFG. STK#3M569)
;
;       PROGRAMMER:  J. REID            DATE: 11-4-86
;
;
;       VARIABLES LIST                  DESCRIPTION
;
;       HAND.TIME               SETS DELAY TO ENABLE GRIPPER TIME
;                               TO OPEN/CLOSE BETWEEN ARM MOVEMTS.
;       ANS, ANSR               STORES USER RESPONSE TO PROMPTS
;
;*********************************************************************
;
;             * * * SET INITIAL PARAMETERS * * *
;
        LEFTY;                          SET CONFIGURATION
        READY;                          MOVE TO READY POSITION
        HAND.TIME = 36;                 SET GRIPPER FOR 1-SEC DELAY
        RESET;                          RESET ALL EXTERNAL SIGNALS
        SPEED 1000 MMPS ALWAYS;         SET NORMAL SPEED VALUE
;
;    * * * DISPLAY INITIAL PROMPTS, ASK USER IF COORD.  LOCATION IS
;         TO BE DEFINED * * *
;
        TYPE /B, /C10;                  SCROLL SCREEN
        TYPE /C8, "* * * VAL II ROBOT CONTROL PROGRAM: MTR.PATH1* *"
     5  PROMPT "COORD.  ORIENTATION? (1=YES, 'RETURN'=NO)", ANS
        IF ANS == 0 GOTO 30
        IF ANS == 1 GOTO 10
        GOTO 5
;
;             * * * COORDINATE LOCATION ROUTINE * * *
;
    10  SPEED 10 MMPS ALWAYS;           SLOW SPEED DOWN
        MOVE INTP1;                     MOVE FIRST TO INTERMEDIATE PT.
        DETACH;                         ALLOW TEACH PENDANT TO BE USED
        TYPE "USE TEACH PENDANT TO MOVE ROBOT TO POSITION ABOVE STATOR"
        TYPE "AND PRESS RETURN WHEN LOCATION ACHIEVED."
        PROMPT " "
        ATTACH;                         RETURN TO PROGRAM CONTROL
        HERE P50;                       DEFINE COORDINATE ORIGIN PT.
        DETACH
        TYPE "USE TEACH PENDANT TO MOVE ROBOT TO POSITION AT POINT 1"
        TYPE "(GRIP SHAFT) AND PRESS RETURN WHEN LOCATION ACHIEVED."
        PROMPT " "
        ATTACH
        HERE P1;                        DEFINE POINT ALONG X-AXIS
        DETACH
        TYPE "USE TEACH PENDANT TO MOVE ROBOT TO POSITION ABOVE END"
        TYPE "BELL AND PRESS RETURN WHEN LOCATION ACHIEVED."
        PROMPT " "
```

```
    ATTACH
    HERE P60;                         DEFINE POINT ALONG Y-AXIS
    SET Z = FRAME(P50, P1, P60, P50);  DEFINE TRANSFORMATION
20  TYPE /B, "PROCEED OR RETRY COORD. DEF? (1=RETRY,'RETURN'=PROCEED)"
    PROMPT " ", ANSR
    IF ANSR == 1 GOTO 10
    IF ANSR == 0 GOTO 30
    GOTO 20
;
;   * * * CORRECT COORDINATES ACHIEVED, MOVE TO TASK READY POSITION * * *
;
30  SPEED 1000 MMPS ALWAYS;          RESET SPEED TO NORMAL
    WAIT SIG(1001);                  WAIT UNTIL PUMA.2 AT READY POSN.
    MOVEST STRTP1, 0;                 MOVE TO SEQUENCE START POINT
    BREAK
    SIGNAL 1;                        SIGNAL PUMA.2 MOVE COMPLETED
;
;           * * * BEGIN ASSEMBLY SEQUENCE * * *
;
    WAIT SIG(1001);                  WAIT UNTIL PUMA.2 CLEAR OF AREA
    RESET;                           CLEAR I/O CHANNELS
    MOVEST P1, 50.81;                 MOVE TO PICK UP SHAFT
    BREAK
    CLOSEI;                          GRASP SHAFT
    MOVEST P2, 0;                     LIFT SHAFT
    BREAK
    MOVEST P3, 0;                     MOVE TO PLACE SHAFT INTO MOTOR
    BREAK
    SPEED 5 MMPS ALWAYS;             SLOW SPEED DOWN
    MOVEST P4, 0;                     INSERT SHAFT INTO STATOR/FRONT BELL

    BREAK
    OPENI
    SPEED 1000 MMPS ALWAYS;          INCREASE SPEED TO NORMAL
    MOVEST P3, 50.81;                 WITHDRAW
    MOVEST P5, 50.81;                 MOVE TO GRASP STATOR LEADS
    BREAK
    SIGNAL 1;                        SIG. PUMA.2 TO PROCEED W/TASK2
    WAIT SIG(1001);                  WAIT UNTIL LEADS CLEAR OF END BELL
    MOVEST P6, 50.81;
    RESET;                           CLEAR I/O
    BREAK
    CLOSEI;                          GRASP STATOR LEADS
    SIGNAL 1;                        SIGNAL PUMA.2 THAT LEADS SECURED
    WAIT SIG(1002);                  WAIT UNTIL PUMA.2 BEGINS MOVE
    MOVEST P7, 0;                     TOWARDS CENTER LOCATOR BLOCK
    BREAK
    SIGNAL 2;                        SIGNAL PUMA.2 MOVE COMPLETED
    WAIT SIG(1001);                  WAIT UNTIL END BELL PLACED
    RESET;                           CLEAR I/O
    OPENI;                           RELEASE STATOR LEADS
    MOVEST P8, 50.81;                 MOVE TO GRASP BOLT#1
    MOVEST P9, 50.81;
    BREAK
    CLOSEI;                          GRASP BOLT#1
    MOVEST P8, 0;                     WITHDRAW
    WAIT SIG(1002);                  WAIT UNTIL PUMA.2 CLEAR OF AREA
    MOVEST P10, 0;                    MOVE TO PLACE BOLT#1
    BREAK
    SPEED 5 MMPS ALWAYS;             SLOW SPEED
    MOVEST P11, 0;                    INSERT BOLT#1
    BREAK
```

```
OPENI
SPEED 1000 MMPS ALWAYS;          INCREASE SPEED
MOVEST P10, 50.81;                WITHDRAW
MOVEST P12, 50.81;                MOVE TO GRASP BOLT#2
BREAK
MOVEST P13, 50.81;
BREAK
CLOSEI;                          GRASP BOLT#2
MOVEST P12, 0;                    WITHDRAW
MOVEST P14, 0;                    MOVE TO PLACE BOLT INTO MOTOR
BREAK
SPEED 5 MMPS ALWAYS;             SLOW SPEED
MOVEST P15, 0;                    INSERT BOLT#2
BREAK
OPENI
SPEED 1000 MMPS ALWAYS;          INCR. SPEED
MOVEST P14, 50.81;                WITHDRAW
MOVEST P16, 50.81;                MOVE TO GRASP BOLT#3
BREAK
MOVEST P17, 50.81;
BREAK
CLOSEI;                          GRASP BOLT#3
MOVEST P16, 0;                    WITHDRAW
MOVEST P18, 0;                    MOVE TO PLACE BOLT INTO MOTOR
BREAK
SPEED 5 MMPS ALWAYS;             SLOW SPEED
MOVEST P19, 0;                    INSERT BOLT#3
BREAK
OPENI
SPEED 1000 MMPS ALWAYS;          INCR. SPEED
MOVEST P18, 50.81;                WITHDRAW
MOVEST P20, 50.81;                MOVE TO GRASP BOLT#4
BREAK
MOVEST P21, 50.81;
BREAK
CLOSEI;                          GRASP BOLT#4
MOVEST P20, 0;                    WITHDRAW
MOVEST P22, 0;                    MOVE TO PLACE INTO MOTOR
BREAK
SPEED 5 MMPS ALWAYS;             SLOW SPEED
MOVEST P23, 0;                    INSERT BOLT#4
BREAK
OPENI
SPEED 1000 MMPS ALWAYS;          INCR. SPEED
MOVEST P22, 50.81;                WITHDRAW
MOVEST P24, 50.81;                AND MOVE TO GRASP FLEX CABLE
BREAK
SIGNAL 2;                        SIGNAL PUMA.2 TO PICK UP MOTOR
MOVEST P25, 50.81;
BREAK
CLOSEI;                          GRASP FLEXIBLE CABLE
BREAK
MOVEST P24, 0;                    WITHDRAW WITH FLEX CABLE
MOVEST P26, 0;                    MOVE TO PICK UP NUT#1
BREAK
SPEED 5 MMPS ALWAYS;             SLOW SPEED
MOVEST P27, 0;                    INSERT NUT#1 INTO MAG. SOCKET
BREAK
MOVEST P26, 0;                    WITHDRAW
SPEED 1000 MMPS ALWAYS;          INCR. SPEED
MOVEST P28, 0;                    MOVE TO MOTOR BOLT#1
BREAK
SPEED 5 MMPS ALWAYS;             SLOW SPEED
MOVEST P29, 0;                    POWER APPLIED TO FLEX CABLE, NUT
BREAK;                            THREADED, DRILL MOTOR STOPPED
```

```
        SPEED 1000 MMPS ALWAYS;             INCR. SPEED
        MOVEST P28, 0;                       WITHDRAW
        MOVEST P30, 0;                       MOVE TO PICK UP NUT#2
        BREAK
        SPEED 5 MMPS ALWAYS;                SLOW SPEED
        MOVEST P31, 0;                       INSERT NUT#2 INTO MAG. SOCKET
        BREAK
        MOVEST P30, 0;                       WITHDRAW
        SPEED 1000 MMPS ALWAYS;             INCR. SPEED
        MOVEST P32, 0;                       MOVE TO MOTOR BOLT#2
        BREAK
        SPEED 5 MMPS ALWAYS;                SLOW SPEED
        MOVEST P33, 0;                       POWER APPLIED TO FLEX CABLE, NUT
        BREAK;                               THREADED. DRILL MOTOR STOPPED
        SPEED 1000 MMPS ALWAYS;             INCR. SPEED
        MOVEST P32, 0;                       WITHDRAW
        MOVEST P34, 0;                       MOVE TO PICK UP NUT#3
        BREAK
        SPEED 5 MMPS ALWAYS;                SLOW SPEED
        MOVEST P35, 0;                       INSERT NUT#3 INTO MAG. SOCKET
        BREAK
        MOVEST P34, 0;                       WITHDRAW
        SPEED 1000 MMPS ALWAYS;             INCR. SPEED
        MOVEST P36, 0;                       MOVE TO BOLT#3
        BREAK
        SPEED 5 MMPS ALWAYS;                SLOW SPEED
        MOVEST P37, 0;                       POWER APPLIED TO FLEX CABLE, NUT
        BREAK;                               THREADED. DRILL MOTOR STOPPED
        SPEED 1000 MMPS ALWAYS;             INCR. SPEED
        MOVEST P36, 0;                       WITHDRAW
        MOVEST P38, 0;                       MOVE TO PICK UP NUT#4
        BREAK
        SPEED 5 MMPS ALWAYS;                SLOW SPEED
        MOVEST P39, 0;                       INSERT NUT#4 INTO MAG. SOCKET
        BREAK
        MOVEST P38, 0;                       WITHDRAW
        SPEED 1000 MMPS ALWAYS;             INCR. SPEED
        MOVEST P40, 0;                       MOVE TO MOTOR BOLT#4
        BREAK
        SPEED 5 MMPS ALWAYS;                SLOW SPEED
        MOVEST P41, 0;                       POWER APPLIED TO FLEX CABLE, NUT
        BREAK;                               THREADED, DRILL MOTOR STOPPED
        SPEED 1000 MMPS ALWAYS;             INCR. SPEED
        MOVEST P40, 0;                       WITHDRAW
        MOVEST P24, 0;                       MOVE TO REPLACE FLEX CABLE
        BREAK;                               INTO RECEPTACLE
        MOVEST P25, 0;
        BREAK
        OPENI
        MOVEST P24, 50.81;                   AND WITHDRAW
        BREAK
        SIGNAL 1;                           SIGNAL PUMA.2 THREADING OPERATION
 ;                                           HAS BEEN COMPLETED
        MOVEST FINPT. 0;                     MOVE TO SEQUENCE TERMINATION POINT
        RESET;                              CLEAR I/O CHANNELS
        BREAK
        SIGNAL 1;                           SIGNAL PUMA.2 ASSY. CYCLE COMPLETED
 ;
 ;      * * * ALLOW OPERATOR TO RUN PROGRAM AGAIN IF DESIRED * * *
 ;
    40  TYPE "SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN PROG. AGAIN"
        PROMPT " ", ANS
        IF ANS == 9 GOTO 100
        IF ANS == 0 GOTO 5
        GOTO 40
   100  STOP
.END
```

APPENDIX I

PROGRAM OUTPUT FOR SINGLE ARM ASSEMBLY

```
ELAPSED TIME FOR CYCLE #  1.



TASK1 TIME =  12.7872
TASK2 TIME =  26.2656
TASK3 TIME =  4.550401
TASK4 TIME =  35.9136
TASK5 TIME =  6.3072
TASK6 TIME =  34.0992
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  125.9424
ELAPSED ARM MOVEMENT TIME =  29.664
TOTAL ARM MOVEMENT TIME FOR  1. CYCLES =  29.664
TOTAL ASSEMBLY TIME FOR  1. CYCLES =  125.9424


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN



ELAPSED TIME FOR CYCLE #  2.



TASK1 TIME =  15.4368
TASK2 TIME =  19.2096
TASK3 TIME =  4.550401
TASK4 TIME =  25.3152
TASK5 TIME =  6.3072
TASK6 TIME =  33.6672
TASK7 TIME =  3.2544


ASSEMBLY SEQUENCE ELAPSED TIME =  110.0736
ELAPSED ARM MOVEMENT TIME =  31.8816
TOTAL ARM MOVEMENT TIME FOR  2. CYCLES =  61.5456
TOTAL ASSEMBLY TIME FOR  2. CYCLES =  236.016


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN
```

```
ELAPSED TIME FOR CYCLE =  3.



TASK1 TIME =  15.4368
TASK2 TIME =  22.9824
TASK3 TIME =  4.550401
TASK4 TIME =  20.5056
TASK5 TIME =  6.3072
TASK6 TIME =  31.392
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  107.1936
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  3. CYCLES =  93.8592
TOTAL ASSEMBLY TIME FOR  3. CYCLES =  343.2096


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN


ELAPSED TIME FOR CYCLE =  4.



TASK1 TIME =  15.4368
TASK2 TIME =  16.0128
TASK3 TIME =  4.550401
TASK4 TIME =  28.512
TASK5 TIME =  6.3072
TASK6 TIME =  31.6224
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  108.4608
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  4. CYCLES =  126.1728
TOTAL ASSEMBLY TIME FOR  4. CYCLES =  451.6705


SELECT: 9=EXIT PROGRAM. 'RETURN'=RUN AGAIN
```

```
ELAPSED TIME FOR CYCLE #  5.


TASK1 TIME =  15.4368
TASK2 TIME =  21.0528
TASK3 TIME =  4.550401
TASK4 TIME =  20.304
TASK5 TIME =  6.3072
TASK6 TIME =  34.0992
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  107.7696
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  5. CYCLES =  158.4864
TOTAL ASSEMBLY TIME FOR  5. CYCLES =  559.4401


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN




ELAPSED TIME FOR CYCLE #  6.


TASK1 TIME =  15.4368
TASK2 TIME =  16.9056
TASK3 TIME =  4.550401
TASK4 TIME =  16.7616
TASK5 TIME =  6.019201
TASK6 TIME =  33.0048
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  98.6976
ELAPSED ARM MOVEMENT TIME =  32.0256
TOTAL ARM MOVEMENT TIME FOR  6. CYCLES =  190.512
TOTAL ASSEMBLY TIME FOR  6. CYCLES =  658.1377


SELECT: 9=EXIT PROGRAM,  RETURN'=RUN AGAIN
```

```
ELAPSED TIME FOR CYCLE =  7.



TASK1 TIME =  15.4368
TASK2 TIME =  21.4848
TASK3 TIME =  4.550401
TASK4 TIME =  64.2528
TASK5 TIME =  6.3072
TASK6 TIME =  30.2112
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  148.2624
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  7. CYCLES =  222.8256
TOTAL ASSEMBLY TIME FOR  7. CYCLES =  806.4001


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN


ELAPSED TIME FOR CYCLE =  8.



TASK1 TIME =  15.4368
TASK2 TIME =  21.6576
TASK3 TIME =  4.550401
TASK4 TIME =  18.7776
TASK5 TIME =  6.3072
TASK6 TIME =  35.9424
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  108.6912
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  8. CYCLES =  255.1392
TOTAL ASSEMBLY TIME FOR  8. CYCLES =  915.0913


SELECT: 9=EXIT PPOGPAM. 'RETURN'=PUN AGAIN
```

```
ELAPSED TIME FOR CYCLE #  9.



TASK1 TIME =  15.4368
TASK2 TIME =  18.9216
TASK3 TIME =  4.550401
TASK4 TIME =  20.6784
TASK5 TIME =  6.3072
TASK6 TIME =  34.9056
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  106.8192
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  9. CYCLES =  287.4528
TOTAL ASSEMBLY TIME FOR  9. CYCLES =  1021.911


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN


ELAPSED TIME FOR CYCLE #  10.



TASK1 TIME =  15.4368
TASK2 TIME =  20.3616
TASK3 TIME =  4.550401
TASK4 TIME =  21.3408
TASK5 TIME =  6.019201
TASK6 TIME =  35.3088
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  109.0368
ELAPSED ARM MOVEMENT TIME =  32.0256
TOTAL ARM MOVEMENT TIME FOR  10. CYCLES =  319.4784
TOTAL ASSEMBLY TIME FOR  10. CYCLES =  1130.947


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN
```

```
ELAPSED TIME FOR CYCLE =  11.



TASK1 TIME =  15.4368
TASK2 TIME =  17.4816
TASK3 TIME =  4.550401
TASK4 TIME =  18.6048
TASK5 TIME =  6.3072
TASK6 TIME =  36.576
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  107.3952
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  11. CYCLES =  351.792
TOTAL ASSEMBLY TIME FOR  11. CYCLES =  1238.343



SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN


ELAPSED TIME FOR CYCLE =  12.



TASK1 TIME =  15.4368
TASK2 TIME =  21.8592
TASK3 TIME =  4.550401
TASK4 TIME =  26.7264
TASK5 TIME =  6.3072
TASK6 TIME =  35.9136
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  116.8128
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  12. CYCLES =  384.1056
TOTAL ASSEMBLY TIME FOR  12. CYCLES =  1355.156



SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN
```

```
ELAPSED TIME FOR CYCLE =  13.



TASK1 TIME =  15.4368
TASK2 TIME =  17.712
TASK3 TIME =  4.550401
TASK4 TIME =  18.2016
TASK5 TIME =  6.3072
TASK6 TIME =  38.5056
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  106.7328
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  13. CYCLES =  416.4192
TOTAL ASSEMBLY TIME FOR  13. CYCLES =  1461.888


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN


ELAPSED TIME FOR CYCLE =  14.



TASK1 TIME =  15.4368
TASK2 TIME =  19.9872
TASK3 TIME =  4.550401
TASK4 TIME =  25.776
TASK5 TIME =  6.3072
TASK6 TIME =  38.5056
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  116.5824
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  14. CYCLES =  448.7328
TOTAL ASSEMBLY TIME FOR  14. CYCLES =  1578.471


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN
```

```
ELAPSED TIME FOR CYCLE #  15.



TASK1 TIME =  15.4368
TASK2 TIME =  19.296
TASK3 TIME =  4.550401
TASK4 TIME =  23.904
TASK5 TIME =  6.3072
TASK6 TIME =  36.0288
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  111.5424
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  15. CYCLES =  481.0464
TOTAL ASSEMBLY TIME FOR  15. CYCLES =  1690.013


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN


ELAPSED TIME FOR CYCLE #  16.



TASK1 TIME =  15.4368
TASK2 TIME =  16.0704
TASK3 TIME =  4.550401
TASK4 TIME =  24.0192
TASK5 TIME =  6.3072
TASK6 TIME =  35.3376
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  107.7408
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  16. CYCLES =  513.3601
TOTAL ASSEMBLY TIME FOR  16. CYCLES =  1797.754


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN
```

```
ELAPSED TIME FOR CYCLE #  17.



TASK1 TIME =  15.4368
TASK2 TIME =  18.0864
TASK3 TIME =  4.550401
TASK4 TIME =  20.0448
TASK5 TIME =  6.3072
TASK6 TIME =  40.2048
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  110.6496
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  17. CYCLES =  545.6737
TOTAL ASSEMBLY TIME FOR  17. CYCLES =  1908.404


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN


ELAPSED TIME FOR CYCLE #  18.



TASK1 TIME =  15.4368
TASK2 TIME =  18.5184
TASK3 TIME =  4.550401
TASK4 TIME =  32.976
TASK5 TIME =  6.3072
TASK6 TIME =  34.848
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  118.656
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  18. CYCLES =  577.9873
TOTAL ASSEMBLY TIME FOR  18. CYCLES =  2027.06


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN
```

```
ELAPSED TIME FOR CYCLE #  19.



TASK1 TIME =  15.4368
TASK2 TIME =  18.6336
TASK3 TIME =  4.550401
TASK4 TIME =  28.7136
TASK5 TIME =  6.3072
TASK6 TIME =  35.5968
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  115.2576
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  19. CYCLES =  610.3009
TOTAL ASSEMBLY TIME FOR  19. CYCLES =  2142.317


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN


ELAPSED TIME FOR CYCLE #  20.



TASK1 TIME =  15.4368
TASK2 TIME =  17.0208
TASK3 TIME =  4.550401
TASK4 TIME =  17.3088
TASK5 TIME =  6.3072
TASK6 TIME =  33.4944
TASK7 TIME =  3.6864


ASSEMBLY SEQUENCE ELAPSED TIME =  101.3184
ELAPSED ARM MOVEMENT TIME =  32.3136
TOTAL ARM MOVEMENT TIME FOR  20. CYCLES =  642.6145
TOTAL ASSEMBLY TIME FOR  20. CYCLES =  2243.636


SELECT: 9=EXIT PROGRAM, 'RETURN'=RUN AGAIN
```

VITA

Jeffrey Tate Reid

Candidate for the Degree of

Master of Science

Thesis: INVESTIGATION OF ROBOTIC ASSEMBLY OF AN ELECTRIC MOTOR UTILIZING SINGLE-ARM AND DUAL-ARM ROBOT CONFIGURATIONS

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Tulsa, Oklahoma, August 25, 1961, the son of James M. and Sally Reid.

Education: Graduated from East Central High School, Tulsa, Oklahoma, in May, 1979; attended Tulsa Junior College from August, 1979, to December, 1980, transferring to Oklahoma State University in January 1981; received Bachelor of Science degree at Oklahoma State University in December, 1984; completed requirements for the Master of Science degree at Oklahoma State University in May, 1987.

Professional Experience: Teaching Assistant, Department of Industrial Engineering and Management, Oklahoma State University, January, 1985, to June, 1985; Assistant Engineer, Moore Business Forms, Inc., June, 1985, to September, 1985; Research Assistant, Department of Industrial Engineering and Management, Oklahoma State University, September, 1985, to September, 1986; Teaching Assistant, Department of Industrial Engineering and Management, Oklahoma State University, September, 1986, to present.

Professional Organizations: Member of American Institute of Industrial Engineers, National Society of Professional Engineers, Oklahoma Society of Professional Engineers, Tau Beta Pi, Alpha Pi Mu, and Robotics International of the Society of Mechanical Engineers.