SPECIAL INDIAN EDITION

INDUSTRIAL ROBOTICS

Technology, Programming, and Applications

(SECOND EDITION)

About the Authors

MIKELL P. GROOVER received his B.A. in Applied Science, B.S. in Mechanical Engineering, and M.S. and Ph. D. in Industrial Engineering from Lehigh University. He is Professor of Industrial Engineering and Director of the Manufacturing Technology Laboratory at Lehigh. He is author and coauthor of two previous books on automation and CAD/CAM, respectively. His areas of specialization include manufacturing technology, automation and robotics.

MITCHELL WEISS received his B.S. in Mechanical Engineering from the Massachusetts Institute of Technology. He was employed as the applications engineer for the PUMA robot by Unimation. Inc., prior to cofounding United States Robots in 1980. He was involved in the design of robots for the company. He has started another company, ProgramMation, and is currently its President.

ROGER N. NAGEL received his B.S. from Stevens Institute of Technology and his M.S. and Ph. D. in Computer Science from the University of Maryland. His professional experience includes the National Bureau of Standards and International Harvester, Inc., as Corporate Director of Automation Technology. He is currently Director of the Institute for Robotics at Lehigh University, and Professor of Computer Science and Electrical Engineering.

NICHOLAS G. ODREY is currently the Director of the Robotics Laboratory within the Institute for Robotics at Lehigh University and is an Associate Professor of Industrial Engineering. His academic background includes a B.S. and M.S. in Aerospace Engineering. After considerable experience in the aerospace industry, he returned for his Ph. D. in Industrial Engineering with specialization in Manufacturing Systems at the Pennsylvania State University. Prior to joining Lehigh, he was associated with the University of Rhode Island, West Virginia University, the National Bureau of Standards, and was a faculty fellow with the U.S. Air Force ICAM program.

ASHISH DUTTA obtained his Ph.D. in Systems Engineering from Akita University, Japan. From 1994 to 2000, he was with Bhabha Atomic Research Center (Mumbai) where he worked on telemanipulator design and control for nuclear applications. He completed his B.Tech. in Mechanical Engineering form REC, Calicut in 1989, and M.E. in Production Engineering from Jadavpur University in 1994. During 2006 and 2007, he had also worked briefly with the Nagoya University, Japan.

Since 2002, Prof. Dutta is working as Associate Professor in the Department of Mechanical Engineering, IIT Kanpur. He won the "Japanese Ministry of Science and Technology Scholarship (MONBUSHO)" for research in Japan (1998–2002). He was listed in the Marquis "Who's Who in the World", 2009. He is also a member of several international professional bodies including IEEE and Japanese Ergonomics Society. He has published many articles in national and international journals. His research areas include humanoid robotics, grasping, micro sensors and actuators, intelligent control systems and rehabilitation engineering.

INDUSTRIAL ROBOTICS

Technology, Programming, and Applications

(Second Edition)

Mikell P. Groover
Professor of Industrial Engineering,
Lehigh University

Mitchel Weiss
ProgramMation Inc.
Cofounder of United States Robots, Inc.

Roger N. Nagel
Professor of Computer Science
and Electrical Engineering,
Lehigh University

Nicholas G. Odrey
Associate Professor of Industrial Engineering,
Lehigh University

Ashish Dutta
Associate Professor
Department of Mechanical Engineering
IIT Kanpur

Tata McGraw Hill Education Private Limited
NEW DELHI

McGraw-Hill Offices

New Delhi New York St Louis San Francisco Auckland Bogotá Caracas
Kuala Lumpur Lisbon London Madrid Mexico City Milan Montreal
San Juan Santiago Singapore Sydney Tokyo Toronto

Tata McGraw-Hill

Special Indian Edition 2012
Published by the Tata McGraw Hill Education Private Limited,
7 West Patel Nagar, New Delhi 110 008.

Industrial Robotics (Technology, Programming, and Applications), 2e (SIE)

ISBN (13): 978-1-25-900621-0
ISBN (10): 1-25-900621-2

Vice President and Managing Director—McGraw-Hill Education: *Ajay Shukla*

Head—Higher Education Publishing and Marketing: *Vibha Mahajan*
Publishing Manager—SEM & Tech Ed.: *Shalini Jha*
Sr Editorial Researcher: *Harsha Singh*
Copy Editor: *Preyoshi Kundu*
Sr Production Manager: *Satinder S Baveja*
Production Executive: *Anuj K. Shriwastava*
Sr Product Specialist —SEM & Tech Ed.: *Tina Jajoriya*
Marketing Manager—Higher Ed.: *Vijay Sarathi*
Graphic Designer (Cover): *Meenu Raghav*

General Manager—Production: *Rajender P Ghansela*
Production Manager: *Reji Kumar*

Typeset at Tej Composers, WZ-391, Madipur, New Delhi 110063, and printed at

Cover Printer:

The McGraw·Hill Companies

Contents

Foreword

In 1946 when I was a senior at Columbia University one course I would have revelled in is the one whose textbook might have been *Industrial Robotics: Technology, Programming, and Applications.* But, of course, in 1946 such a course would have been impossible. So much of the technology was not at hand and there was hardly any motivation, other than sheer fun, to build robots. Fun had already prompted the ingenious automatons that appear herein as background history.

Yet the beast was stirring. My friend, Isaac Asimov, was busy at Boston College

of that doomsday view owing to Capek.

Surely Asimov left a subliminal message with me as I put my Columbia training to industrial control problems. Servo theory was born in World War II, an esoteric discipline called boolean algebra became digital logic and the transistor was invented after I graduated.

The story of my fortuitous association with inventor Devol is recounted herein. Collectively if all welled up, as Victor Hugo would have had it, "an idea whose time had come."

all-in cost of an automotive worker was $3.50/hour. Ever since the cost of labor has increased to the current level of $21.00/hour in the automotive industry. Meanwhile, the cost of manufacturing a robot has tumbled from $60,000 in 1961 dollars to $25,000 in 1984 dollars. And a robot's capabilities today are so vastly superior to

Enter *Industrial Robotics: Technology, Programming, and Applications.* Here is the whole spectrum of the state of the art. My mind boggles to think how an engineering senior armed with this background would have been greeted by my

The authors bring it all together from design to use. And there is no shirking from

A would-be robot designer will learn the basic criteria. He or she will not have to

intelligently from the range of robots on the market.

and I sold Unimation, Inc., I have become a consultant. Will my clients still pay me dearly if they can read all about it in this exhaustive tutorial treatise?

Joseph F. Engelberger

Preface to the Special Indian Edition

About the Book

Robotics as a discipline is a mixture of computer science, mechanical, and electrical engineering. As there have been rapid advancements in these disciplines, robotics as a subject has also undergone many changes. The Special Indian Edition (SIE) is a takeoff from the original book by Groover, Weiss, Nagel, and Odrey published in 1986 and hence, most of the content has been revised and newer details added for the present edition. The book, *Industrial Robotics* is mainly focused on the industrial applications of robotics, but it also contains the basic theory required for programming and robot applications.

The Special Indian Edition caters to the requirements of undergraduate and Master's students in engineering disciplines of electrical, mechanical, computer science, mechatronics, etc. Engineers working in the industry can also use the book for refreshing their knowledge on the current robotics practices in industry. As the name indicates, the present edition is mainly focused on the industrial applications of robotics. The book focuses on industrial applications and hence, the book will be useful to students and practicing engineers.

New to this Edition

Chapters 1 and 10 have been updated in accordance to the universities' requirements. Topics like Circuit and Mechanical Model of DC Motors, Internal and External sensors, Joint Space Schemes Dynamics : Newton –Euler and Euler-Lagrangian Formulations, Trajectory via points, Jacobian and Singularity Functions, Force Sensors and their design, Human Centered robotics, and Rehabilitation Robots have been added in Chapters 3, 4, 5, 6, 19, and 20.

Salient Features of this Book

- Comprehensive analysis on fundamentals of robot technology
- New detailed discussion on robot dynamics
- Complete coverage of sensors and robot motions
- Emphasizes on robot programming and languages
- Detailed discussion on industrial and manufacturing applications of robots

Chapter Organization

The book contains 20 chapters and is organized into seven parts.

Part One is introductory. **Chapter 1** provides the motivation and rationale for learning about robotics. **Chapter 2** presents an overview of robot technology and

technical topics which must be placed into context relative to other topics. **Chapter 2**

Part Two examines the technical topics that relate to the robot and the peripheral hardware used with the robot. **Chapter 3** discusses the mechanical components of the robot and the control systems used to control the joints of the arm and wrist. **Chapter 4** presents some of the mathematical analysis that is used in robotics to study the motion of the manipulator. **Chapter 5** covers end effectors, the mechanical hands and other devices that are attached to the robot arm to perform useful work. **Chapters 6** and **7** are concerned with the sensors that are used in robotics, including

robotics work in the future.

Part Three deals with robot programming. **Chapter 8** is concerned with the fundamentals of how to program robots and what the requirements for robot programming are. **Chapter 9** and its appendixes cover some of the robot textual languages that are in common use. **Chapter 10**
intelligence and its relationship to robotics. We anticipate that robots of the future will possess far greater intelligence and reasoning power than current-day robots, and the

Part Four is concerned with applications engineering. What are the engineering and economic problems that must be addressed in installing robots, and what are some of the applications of robots today? **Chapter 11** describes work cell design and control-how to use the technology and programming of robotics in industrial applications. **Chapter 12** presents the methods that should be used to justify a robot investment. *Part Five* presents a survey of how robotics is used in industry. **Chapters 13** through **15** discuss the various types of robot applications in manufacturing today.

Closely related to applications engineering are the implementation issues associated with the introduction of robotics into the factory. *Part Six* surveys some of these issues. **Chapter 16** proposes a seven-step approach for the implementation of

of the robot work cell. **Chapter 17** discusses some of the additional problem areas that must be confronted during implementation. These areas include safety, training, maintenance, and quality control.

Part Seven deals with social issues and the future of robotics. **Chapter 18** explores the possible social impact of robotics, giving particular attention to the problems confronting labor. In **Chapters 19** and **20**, we speculate about the following questions. What will the technology of robotics be like in the future? And what kinds of applications will robots be performing in the future?

Acknowledgements

I would like to thank the following reviewers for taking out time in reviewing the script and providing valuable suggestions/opinions:

Arul Sanjivi
Department of Mechanical Engineering
Amrita School of Engineering, Tamil Nadu

P M Pathak
Department of Mechanical Engineering
IIT Roorkee, Uttaranchal

K Krishna Prasad
Department of Mechanical Engineering
NIT Warangal, Andhra Pradesh

R Sagar
Department of Mechanical Engineering
IIT Delhi, Delhi

K S Srinivasan
Department of Mechanical Engineering
Iswari Engineering College, Tamil Nadu

S S Roy
Department of Mechanical Engineering
NIT Durgapur, West Bengal

Manzoor Husain
Department of Mechanical Engineering
JNTU College of Engineering, Andhra Pradesh

Uday Shankar Dixit
Department of Mechanical Engineering
IIT Guwahati, Assam

Ashish Dutta

Preface

This book is intended to provide a comprehensive survey of the technical topics

the important automation areas for the 1980s and 1990's. Engineers, technicians, and managers must be educated and trained in order to realize the full potential of this technology. It is our hope that this book might help to satisfy the need for text materials to develop these technically educated people.

graduate engineering programs. It should be suitable for courses in several departments, including mechanical, industrial, manufacturing, and electrical engineering. The book includes mechanical joint-link analysis, control systems, sensors, machine vision, end effector design, and other topics of interest to these engineering disciplines. The text would also be appropriate for courses in computer science since a substantial portion of the book is devoted to robot programming. We have also designed the book for industrial training courses, and it contains much material that is relevant to those who must install robot systems. In short, it is a book on the technology, programming, and applications of industrial robots that should serve the student of robotics making the transition from the classroom and laboratory environment of academia into the applied and practical world of industry.

We began developing the outline for this text in 1981. The contract with McGraw-Hill to write the book was signed in summer 1982. A great deal has happened in the

corporation. We are beginning to see the fallout of the weaker companies in the industry. The technology has also developed dramatically during these several years.

more sophisticated yet easier to use technology.

Also during the last severer years we (the authors) have learned a great deal

developments.

Something else that has happened since 1982 is that Lehigh University has hired two new faculty members whose expertise includes robotics: Roger Nagel in Fall

1982 and Nick Odrey in Fall 1983. Accordingly, we have seen our way to invite them to add their expertise to this book Roger's education and interests are in computer science, and his professional experience includes research at the National Bureau of Standards in robot programming and machine vision. He was also Director of Automation as International Harvester where he managed projects in robotics before coming to Lehigh. Nick is an aerospace engineer turned industrial engineer. His background is heavily oriented toward the mathematical analysis of control systems and mechanical linkages (such as a robot's mechanical manipulator). Combining their knowledge with that of the two original coauthors we have a team whose expertise

include mechanical engineering, industrial engineering, computer science and electrical engineering. Their professional backgrounds include both academe and industry. We believe that the breadth and depth of knowledge and experience of this team has permitted us to provide a more complete and comprehensive coverage of industrial robotics than exists in any other available text on this subject.

The book contains 20 chapters, many of them technical with engineering problem sets at the end. Even the most ambitious and work-oriented instructor will have

Accordingly, what must be done is to cover the chapters that are most appropriate for the particular course being offered, and send the students on their way with the hope that they will read the other chapters if the need to do so subsequently arises in their work in robotics.

ACKNOWLEDGMENTS

There are many people and organizations to be acknowledged for their contributions and assistance in publishing this book. Our fear is that we may overlook some

overlooked, if there are any, we apologize in advance. For their technical input and/ or review of portions of the manuscript, we are indebted to the following: Robert

material for the book, was helpful during initial startup of our Robotics Laboratory at

David Hanan, MSE Don Hillman. Computer Science

Professor at L

of Grumman Aerospace; Ramadan Taher, IE graduate student; Gordon Vanderbrug of Automatix; West Vogel, a graduate student in IE who prompted our thinking on robot economic analysis.

In addition, we acknowledge with gratitude the reviews of our academic peers whose comments were helpful in shaping the final version of the text: Rashpal S. Ahluwalia (Ohio State University), Stephen J. Derby (Rensselaer Polytechnic Institute), Steven Dickerson (Georgia Institute of Technology), Lyman L. Francis (University of Missouri, Rolla), Herbert Freeman (Rutgers University), Ernest L. Hall (University of Cincinnati), R. T. Johnson (University of Missouri, Rolla), Donald J. McAleece and Edward E. Messal (Indiana University and Purdue University), Daniel Metz (University of Illinois). Wolfgang Sauer (University of Massachusetts), Holger J. Sommer (Pennsylvania State University), Allen Tucker (Colgate University), Richard A. Wysk (Pennsylvania State University).

We are also indebted to B. J. Clarke, Rodger Klas, and Anne Murphy of McGraw-Hill Book Co. for their wisdom and perception in selecting us to do the book, and their patience and tolerance with us during manuscript preparation.

For their help in preparing our solutions manual for the book we would like to thank Cemal Doyden, IE graduate student; Thomas Grycan, IE undergraduate; and W. Scott Sendel, IE undergraduate. We must also acknowledge Fern Sotzing, for her secretarial assistance and friendly disposition during manuscript preparation.

Finally, we would like to express our appreciation to our wives, respectively, Bonnie, Nancy, Arlene, and Sandy, for their understanding and encouragement during the many hours that their husbands spent on the book.

Mikell P. Groover
Mitchell Weiss

Fundamentals of Robotics

PART ONE

Introduction

Introduction

1.1 AUTOMATION AND ROBOTICS

Fig. l.1

1.2 ROBOTICS IN SCIENCE FICTION

1.

2.

3.

Fig. 1.2

1.3 A BRIEF HISTORY OF ROBOTICS

Table 1.1

Date	*Development*
mid-1700s	J. de Vaucanson built several human-sized mechanical dolls that played music.
1801	J. Jacquard invented the Jacquard loom, a programmable machine for weaving threads or yarn into cloth.
1805	H. Maillardet constructed a mechanical doll capable of drawing pictures.
1946	American inventor G. C. Devol developed a controller device that could record electrical signals magnetically and play them back to operate a mechanical machine U.S. patent issued in 1952.
1951	Development work on teleoperators (remote-control manipulators) for handling radioactive materials. Related U.S. patents issued to Goertz (1954) and Bergsland (1958),
1952	Prototype Numerical Control machine demonstrated at the Massachusetts Institute of Technology after several years of development. Pan programming language called APT (Automatically Programmed Tooling) subsequently developed and released in 1961,
1954	British inventor C. W. Kenward applied for patent tor robot design. British patent issued in 1957.
1954	G. C. Devol develops designs for “programmed article transfer.” U.S. patent issued tor design in 1961,
1959	First commercial robot introduced by Planet Corporation. It was controlled by limit switches and cams.

Contd.

1960	First "Unimate" robot introduced, based on Devol's "programmed ankle transfer." It used numerical control principles for manipulator control and was a hydraulic drive robot.
1961	Unimate robot installed at Ford Motor Company for tending a die casting machine.
1966	Trallfa, a Norwegian firm, built and installed a spray painting robot.
1968	A mobile robot named "Shakey" developed at SRI (Stanford Research Institute). It was equipped with a variety of sensors, including a vision camera and touch sensors, and can move on the floor.
1971	The "Stanford Arm," a small electrically powered robot arm, developed at Stanford University.
1973	First computer-type robot programming language developed at SRI for research called WAVE Followed by the language AL in 1974. The two languages were subsequently developed into the commercial VAL. language for Unimation by Victor Scheinman and Bruce Simano.
1974	ASEA introduced the all-electric drive IRb6 robot.
1974	Kawasaki, under Unimation license, installed arc-welding operation for motorcycle frames.
1974	Cincinnati Milacron introduced the T3 robot with computer control.
1975	Olivetti "Sigma" robot used in assembly operation–one of the very first assembly applications of robotics.
1976	Remote Center Compliance (RCC) device for part insertion in assembly developed at Charles Stark Draper Labs in United States.
1978	PUMA (Programmable Universal Machine for Assembly) robot introduced for assembly by Unimation, based on designs from a General Motors study.
1978	Cincinnati Milacron T3 robot adapted and programmed to perform drilling and routing operations on aircraft components, under Air Force ICAM (Integrated Computer-Aided Manufacturing) sponsorship.
1979	Development of SCARA type robot (Selective Compliance Arm for Robotic Assembly) at Yamanashi University in Japan for assembly. Several commercial SCARA robots introduced around 1981.
1980	Bin-picking robotic system demonstrated at University of Rhode Island. Using machine vision, the system was capable of picking parts in random orientations and positions out of a bin.
1981	A "direct-drive robot" developed at Carnegie-Mellon University. It used electric motors located at the manipulator joints without the usual mechanical transmission linkages used on most robots.

Contd.

1982	IBM introduces the RS-1 robot for assembly, based on several years of in-house development. It is a box-frame robot, using an arm consisting of three orthogonal slides. The robot language AML, developed by IBM, also introduced to program the RS-1.
1983	Report issued on research at Westinghouse Corp under National Science Foundation sponsorship on "adaptable-programmable assembly system" (APAS), a pilot project for a flexible automated assembly line using wools.
1984	Several off-line programming systems demonstrated at the Robots 8 show. Typical operation of these systems allowed the robot program 10 he developed using interactive graphics on a personal computer and then downloaded to the robot.
1990 s	Robot development diversified into walking robots at MIT, Honda, etc., rehabilitation robots for health care, as well as robots for defense and space applications.
2000 s	Micro and nano robots using smart materials, Unmanned Ariel vehicles and underwater robotics.

2000

Fig. 1.3

2,988,237
PROGRAMMED ARTICLE TRANSFER
George C. Devol, Jr., Brookside Drive, Greenwich, Conn.
Filed Dec. 10, 1954, Ser. No. 474,574
28 Claims. (Cl. 214—11)

Fig. 1.4

Fig. 1.5

Fig. 1.6

Fig. 1.7

Fig. 1.8

1.4 THE ROBOTICS MARKET AND THE FUTURE PROSPECTS

Fig. 1.9

Fig. 1.10

Fig. 1.11

Fig. 1.12

eview Questions

1.1

1.2

1.3
1.4
1.5

1.6

References

Fundamentals of Robot Technology, Programming, and Applications

Introduction

Robotics is an applied engineering science that has been referred to as a combination of machine tool technology and computer science. It includes such

theory. Robotics research and development are proceeding in all of these areas to

from simple arm mechanisms that can pick and place objects to intelligent

above are highly interdependent in the manner in which they are used in robotics.

of the way robots are applied in industry. In order to understand the use of

reader to relate the various topics in the chapters that follow.

features about the way the robot is constructed and the way it operates. Robots

programming of robots is accomplished in several ways.

following sections:

- Drive systems and sensors
- End effectors

Robot programming and communication with other systems

well beyond the basic introduction intended by this chapter. We will discuss these topics in greater depth in subsequent chapters of the book.

2.1 ROBOT ANATOMY

and wrist of the machine. Most robots used in plants today are mounted on a

consists of a number of components that allow it to be oriented in a variety of positions. Relative movements between the various components of
and wrist are provided by a series of joints. These joint movements usually involve

called the *manipulator*
robot's wrist is a hand or a tool called the "end effector". The end effector is not considered as part of the robot's anatomy. The arm and body joints of the manipulator
wrist joints of the manipulator are used to orient the end effector.

a of Fig. 2.1. It uses a telescoping

a rotating base.

Fig. 2.1 *The four basic robot anatomies: (a) Polar, (b) Cylindrical, (c) Cartesian, (d) Jointed-arm. (Reprinted from Reference [7])*

These various joints provide the robot with the capability to move its arm within

applied to this type.

.1(*b* uses a vertical column and a slide that can be moved up or down along the column. The robot arm is attached to the slide so that it can be moved radially with respect to the column. By rotating the

The cartesian coordinate robot illustrated in Fig. 2.1(*c*
slides to construct the *x* *y* *z*
xyz robot and rectilinear robot. By moving the three slides

envelope.

The jointed-arm robot as shown in Fig. 2.1(*d*

similar to the human-arm.

There are relative advantages and disadvantages to the four basic robot anatomies simply because of their geometries. In terms of repeatability of motion (the capability

robot probably possesses the advantage because of its inherently rigid structure. In

lift capacity of the robot is important in many applications. The cylindrical *xyz* robot can be designed for high rigidity and load-

into a small opening without interference with the sides of the opening is important.

advantage in terms of this capability.

The individual joint motions associated with the performance of a task are referred to by the term *degrees of freedom*

considered as a degree of freedom.

The robot's motions are accomplished by means of powered joints. Three joints

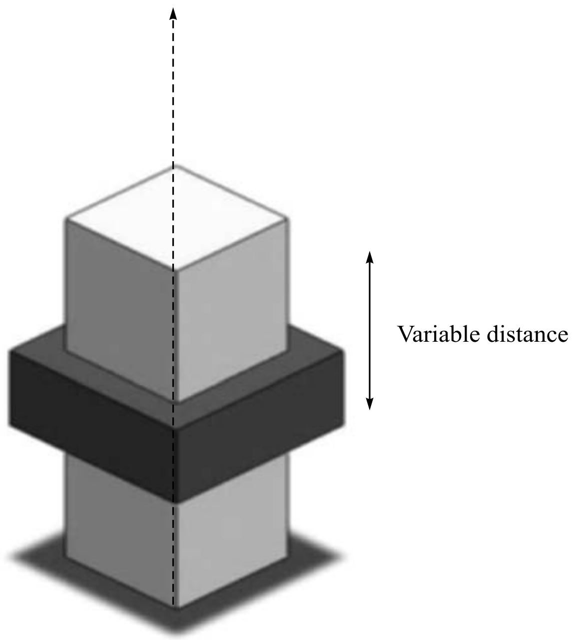
are generally used to actuate the wrist. Connecting the various manipulator joints together are rigid members that are called *links*. The links can be connected to form a serial chain or a parallel chain. Majority of industrial manipulators are serial chains.

The joints used in the design of industrial robots typically involve a relative motion of the adjoining links that is either linear or rotational. Linear joints involve a sliding or translational motion of the connecting links. This motion

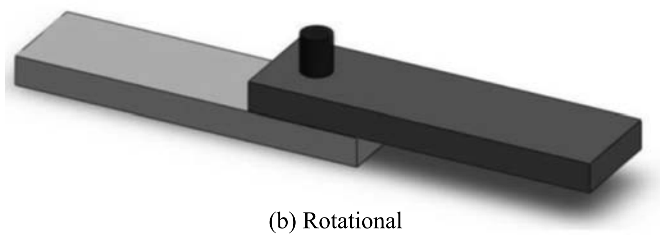
relative motion of the adjacent links. We shall refer to the linear joint or prismatic joint as a type *P* joint (*P*
Fig. 2.2(*a*

There are at least three types of rotating or revolute joints that can be distinguished in robot manipulators. The three types are illustrated in Fig. 2.2(*b c d*
-

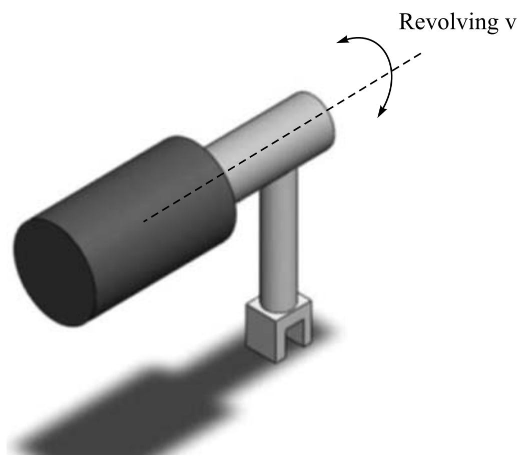
(*R stands for revolute* Both the prismatic joint and the revolute joint have one DOF each. Combining a revolute joint and a prismatic joint we can get a
e
type of joint that is not very common in robots but very common in biological systems is the ball and socket joint as shown in Fig. 2.2(*f*
largest range of motion and has three DOF.



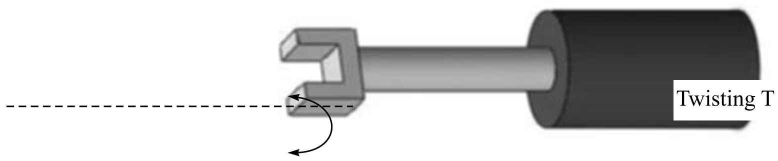
(a) Linear



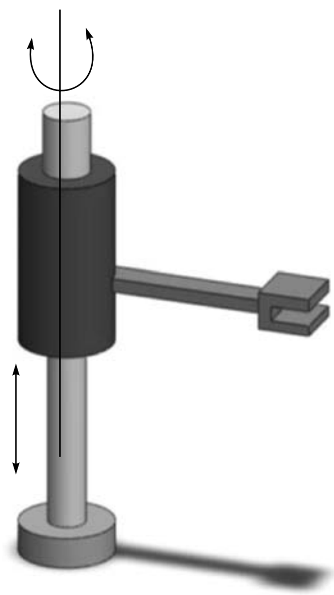
(b) Rotational



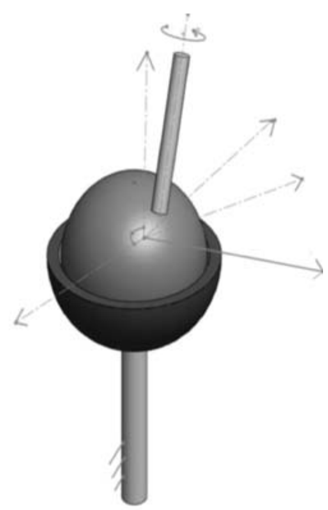
(c) Revolving



(d) Twisting



(e) Cylindrical (both linear and Revolute)



(f) Spherical

Fig. 2.2 *Different types of joints used in robots.*

The arm and body joints are designed to enable the robot to move its end

three degrees of freedom associated with the arm and body motions are:

1. Vertical traverse: This is the capability to move the wrist up or down to provide the desired vertical attitude.

2. Radial traverse:

3. Rotational traverse: .

The degrees of freedom associated with the arm and body of the robot are shown in

three degrees of freedom are vertical movement (z
(y x
corresponding movements of the three orthogonal slides of the robot arm.

Fig. 2.3 *Three degrees of freedom associated with arm and body of a polar coordinate robot.*

The wrist movement is designed to enable the robot to orient the end effector

To

1. Wrist roll: *wrist swivel* -

2. Wrist pitch:
involve the up or down rotation of the wrist. Wrist pitch is also sometimes called *wrist bend*.

3. Wrist yaw:

.

These degrees of freedom for the wrist are illustrated in Fig. 2.4. The reason

orientation of the pitch and yaw movements.

Fig. 2.4 *Three degrees of freedom associated with the robot wrist.*

PUMA (***P**rogrammable **U**niversal **M**achine for **A**ssembly, or **P**rogrammable **U**niversal **M**anipulation **A**rm*

that was developed in 1981 mainly for assembly operation jointly by Sankyo

one which uses rotary joints to access its work space and usually the joints are

Such robots are similar to the human hand and can be used to perform a variety

Fig. 2.5 *PUMA robot having six degrees of freedom.*

Fig. 2.6 *A SCARA Robot designed for assembly.*

Fig. 2.7 *Articulated robots used in machining, spray painting, welding etc.*

(L R T V .

The joint notation scheme permits the designation of more or less than the three

Table 2.1

Robot configuration (arm and body)	*Symbol*
Polar configuration	TRL
Cylindrical configuration	TLL,LTL,LVL
Cartesian coordinate robot	LLL
Jointed arm configuration	TRR, VVR
Robot configuration (wrist)	Symbol
Two-axis wrist (typical)	:RT
Three-axis wrist (typical)	:TRT

the joint closest to

are predominantly rotating joints of type *R* and *T*
with three rotational joints would be indicated by *TRR*

TRL: *TRT*.

The scheme can also provide for the possibility of robots that move on a track in *TRL*: *TRT* robot fastened to a platform on wheels that can be driven along a track between several machine tools would be designated by the following notation: *L*- *TRL*: *TRT*.

is linear.

2.2 WORK VOLUME

Work volume is the term that refers to the space within which the robot can

that might be attached to the robot's wrist. The end effector is an addition to the basic

effector attached to the wrist might not be capable of reaching certain points within the robot's normal work volume because of the particular combination of joint limits of the arm.

The work volume is determined by the following physical characteristics of the robot:

volume made up of two or more spheres on the inside and one sphere on the outside.

Fig. 2.8 *Work volumes for different types of robots: (a) Polar, (b) Cylindrical, (c) Cartesian. (Reprinted from Reference [7])*

determines the kinds of applications that the robot can accomplish. ln this and the

Commercially available industrial robots are powered by one of three types of drive systems. These three systems are:

1. Hydraulic drive
2. Electric drive
3. Pneumatic drive

Hydraulic drive and electric drive are the two main types of drives used on more

Hydraulic drive is generally associated with larger robots. The usual advantages of the hydraulic drive system are that it provides the robot with greater speed and strength. The disadvantages of the hydraulic drive system are that it typically adds to

oil which is a nuisance. Hydraulic drive systems can be designed to actuate either rotational joints or linear joints. Rotary vane actuators can be

accomplish linear motion.

Electric drive systems do not generally provide as much speed or power as

toward more precise work such as assembly.
Electric drive robots are actuated by dc stepping motors or dc servomotors. These motors are ideally suited to the actuation of rotational joints through appropriate drive train and gear systems. Electric motors can also be used to actuate linear joints
mechanisms.

The economics of the two types of drive systems are also a factor in the decision

the cost of a hydraulic drive system is somewhat less
be noted that there is a trend in the design of industrial robots toward all electric
above.

Pneumatic drive is generally reserved for smaller robots that possess fewer degrees of freedom (two- to
simple pick-and-place operations with fast cycles. These drives have the added advantage of having compliance or ability to absorb some shock during contact with the environment. Pneumatic power can be readily adapted to the actuation of piston devices to provide translational movement of sliding joints. It can also be used to operate rotary actuators for rotational joints.

than electric drive robots.

There is generally an inverse relationship between the accuracy and the speed of

reduce the location errors in its various joints to achieve the desired

must be operated more slowly to

robot manipulator is illustrated in Fig. 2.9. Because of acceleration and deceleration of short distances whose sum is equal to the long distance. The short distances may not permit the robot to ever reach the programmed operating speed.

Fig. 2.9

in the case of a it
than when the arms are held in close to the body.

The rated weight-carrying capacities of industrial robots ranges from less than a kilogram for some of the small robots up to several hundred kilograms for very large

weight-carrying capacity of the robot would be only 3 kg.

types of control systems and the associated performance characteristics which are

provided in Chs. 3 and 4.

-

cording to their control systems. The four categories are:

1. Limited-sequence robots
2. Playback robots with point-to-point control
3. Playback robots with continuous path control
4. Intelligent robots

control and intelligent robots are the most sophisticated.

Limited-sequence robots do not use servo-control to indicate relative positions of

stops to establish the endpoints of travel for each of their joints. Establishing the positions and sequence of these stops involves a mechanical set-up of the manipulator rather than robot programming in the usual sense of the term. With this method of

in a program for these robots. The sequence in which the motion cycle is played

to operate in the proper succession. There is generally no feedback associated with a

pneumatic drive seems to be the type most commonly

Playback robots use a more sophisticated control unit in which a series of positions

then repeated by the

robot under its own control. The term "playback" is descriptive of this general mode of operation. The procedure of teaching and recording into memory is referred to as programming the robot. Playback robots usually have some form of servo-control

by the robot

are the positions that have been taught.

performing motion cycles that consist of a series of desired point locations and

one point to another in the proper sequence. Point-to-point robots do not control

be done by programming a series of points along the desired path. Control of the

loading and unloading machines and spot welding.

Continuous-path robots are capable of performing motion cycles in which the path followed by the robot is controlled. This is usually accomplished by making the robot move through a series of closely spaced points which describe the

rather than the

programmer. Straight line motion is a common form of continuous-path control for

robot to follow a straight line trajectory. Some robots have the capability to follow a

arm through the desired motion cycle. To achieve continuous-path control to more

this usually involves the use of a digital computer (a microprocessor is typically used
robot controller. CP control is required for certain types of industrial applications such as spray coating and arc welding.

Intelligent robots constitute a growing class of industrial robot that possesses the capability not only to play back a programmed motion cycle but also to interact with

of a digital computer or similar device
robots can alter their programmed cycle in response to conditions that occur in the workplace. They can make logical decisions based on sensor data received from the operation. The robots in this class have the capacity to communicate during the work cycle with humans or computer-based systems. Intelligent robots are

and sophisticated activities that can be accomplished by these robots. Typical

smart materials-based actuators and sensors.

is ultimately measured by its ability to position and orient the end effector at the

three features:

1. Spatial resolution

3. Repeatability

to measure position and orientation in 3D space at all points the end effector moves through.

1. Spatial Resolution The spatial resolution of a robot is the smallest increment of movement into which the robot can divide its work volume. Spatial resolution depends on two factors: the system's control resolution and the robot's mechanical

degree of freedom.

The control resolution is determined by the robot's position control system and its feedback measurement system. It is the controller's ability to divide the total range of movement for the particular joint into individual increments that can be addressed in the controller. The increments are sometimes referred to as "addressable points." The ability to divide the joint range into increments depends on the bit storage capacity

$$2^n$$

where n

divided by the number of increments. We assume that the system designer will make all of the increments equal.

Example 2.1 we will assume it has one sliding joint with a full range of 1.0 m. The robot's control memory has a 12-bit storage capacity. The problem is to

The number of control increments can be determined as follows:

$$12$$

The total range of 1 m is divided into 4096 increments. Each position will be separated by

The control resolution is 0.244 mm.

would have a control resolution for each joint of motion. To obtain the control

summed vectorially. The total control resolution would depend on the wrist motions as well as the arm and body motions. Since some of the joints are likely to be rotary

to determine.

Mechanical inaccuracies in the robot's links and joint components and its feedback measurement system (if

factor that contributes to spatial resolution. Mechanical inaccuracies come from

These inaccuracies tend to be worse for larger robots simply because the errors are

The spatial resolution of the robot is the control resolution degraded by these mechanical inaccuracies. Spatial resolution can be improved by increasing the bit inaccuracies of the system become the dominant component in the spatial resolution.

2. Accuracy

of spatial resolution because the ability to achieve a given target point depends on

control increments. Ignoring for the moment the mechanical inaccuracies which

case assumption as one-half of the control resolution. This relationship is illustrated

spatial resolution as portrayed in Fig. 2.11.

affected by several factors.

Fig. 2.10 *Illustration of accuracy and control resolution when mechanical inaccuracies are assumed to be zero.*

Fig. 2.11 *Illustration of accuracy and spatial resolution in which mechanical inaccuracies are represented by a statistical distribution.*

is in the outer range of its work volume and better when the arm is closer to its base.

accuracy is improved if

range of motions. The robot's ability to reach a particular reference point within the limited work space is sometimes called its *local accuracy*. When the accuracy is

global accuracy

lower accuracy.

3. Repeatability Repeatability is concerned with the robot's ability to position its wrist or an end effector attached to its wrist at a point in space that had previously been taught to the robot. Repeatability and accuracy refer to two different aspects of

achieve a given target point. The actual programmed point will probably be different from the target point due to limitations of control resolution. Repeatability refers to the robot's ability to return to the programmed point when commanded to do so.

These concepts are illustrated in Fig. 2.12. The desired target point is denoted by the letter T

point T

becomes point P. The distance between points T and P is a manifestation of the

programmed point P;

it returns to position R. The difference between P and R is a result of limitations on the robot's repeatability. The robot will not always return to the same position R on subsequent repetitions of

both sides of the position P as shown in Fig. 2.12.

Fig. 2.12 *Illustration of repeatability and accuracy.*

Repeatability errors form a random variable and constitute a statistical

principally responsible for repeatability errors do not form the nice symmetric bell-

the central limit theorem in probability. This theorem states that the sums of random if

point P

work volume that are further away from the center of the robot. It is likely that the

due to compliance of the robot arm.

4. Compliance

compliance. The compliance of the robot manipulator refers to the displacement of

means that the wrist is displaced a large amount by a relatively small force. The

compliance means that the manipulator is relatively stiff and is not displaced by a

the robot arm will be greater in certain directions than in other directions because of the mechanical construction of the arm.

Compliance is important because it reduces the robot's precision of movement

because of compliance when it operates under loaded conditions.

by means of additional devices. We might refer to these devices as the robot's peripherals. They include the tooling which attaches to the robot's wrist and the sensor systems which allow the robot to interact with its environment. We provide a

attached to the wrist. The end effector represents the special tooling that permits the general-purpose robot to perform a particular application. This special tooling must

End effectors can be divided into two categories: grippers and tools. Grippers

work cycle. There are a variety of holding methods that can be used in addition to the

tool would be used as an end effector in applications where the robot is required to to the robot's wrist to accomplish the application. With the recent need for holding

Sensors are not only required for working of the robot and interacting with

last decade with the drop of prices of vision systems and with the availability of

can be used for several applications. Vision systems can be used to locate objects

depends on several sensor information. Simple sensors are used to detect the

2.8 ROBOT PROGRAMMING AND WORK CELL CONTROL

which the manipulator is directed to move. This path also includes other actions such as controlling the end effector and receiving signals from sensors. The purpose of robot programming is to teach these actions to the robot.

There are various methods used for programming robots. The two basic categories

language programming.

Leadthrough programming consists of forcing the robot arm to move through the required motion sequence and recording the motions into the controller memory. Leadthrough methods are used to program playback robots. In the case of point-

and record the points into memory for subsequent playback. The teach pendant is equipped with a series of switches and dials to control the robot's movements during the teach procedure. Owing to its ease and convenience and the wide range of

method for playback-type robots.

pendant can be employed to teach the locations of the two points; and the robot controller then computes the trajectory to be followed in order to

spray-

it is usually more convenient for the programmer to physically move the robot arm and end effector through the desired motion path and record the positions at closely spaced sampling intervals. Certain parameters of the motion

would be controlled independently when the job is

these aspects of the program. The programmer's principal concern is to make sure that the motion sequence is correct.

input the program instructions into the controller but a teach pendant is also used to

programming language names the points as symbols in the program and these

more

and greater

corresponds largely to the so-called intelligent robots.

SPEED 35 IPS
MOVE P1
CLOSE 40 MM
WAIT 1 SEC
DEPART 60 MM

The series of commands tells the robot that its velocity at the wrist should

the robot is to move its gripper to point P1 and close to an opening of 40 mm. It is directed to wait 1.0 sec. before departing from P1 by a distance of 60 mm above the point.

the

point locations in the program. The potential advantage of this method is that the programming can be accomplished without taking the robot out of

of the current methods of programming require the participation of the robot in

program can be entered into a computer for later downloading to the robot. Off-line programming would hasten the changeover from one robot work cycle to a new certain technical problems associated with off-line programming. These problems

in
robot languages.

is another form of programming for the low-technology-limited sequence robots.

similar means to establish the endpoints of travel for each of the joints. This is sometimes called *mechanical programming;* it really involves more of a manual set-up procedure rather than a programming method. The work cycles for these kinds of

Work cell control deals with the problem of coordinating the robot to operate with

control is accomplished either by the robot controller or a separate small computer

to the equipment in the cell and receives signals from the equipment. These signals are sometimes called *interlocks*. By communicating back and forth with the different

proper sequence.

perform simple reprogrammable functions are already commonplace.

following categories:

1. Material-handling and machine-loading and -unloading applications In

in the work cell to some other location.

2. Processing applications

a tool to accomplish some manufacturing process in the work cell. Spot welding represents a particularly important application in the processing category.

3. Assembly and inspection These are two separate operations which we

is showing great interest because of its economic potential.

4. Advanced applications
etc.

2.1 The notation scheme described in Sec. 2.1 provides a shorthand method of

the robot.

(a) *LLR*

(b) RLR

(c) *LRR*

(d) *LVR*

(e) *LL- TRL*

2.2

2.3

(a) :*TR*

(b) :*RT*

(c) :*TRT*

position and orient an end effector.

2.4

all of the robots in the lab in terms of their respective anatomy notations. That

2.5 arm

’s control memory has an 8-bit storage

2.6

in its control memory:

(a)

(b)

2.7

of bits of storage capacity which the robot’s control memory must possess to provide this level of precision.

2.8 *RRL* robot is a Sliding mechanism with a total of 0.70 m

it has been observed that the mechanical inaccuracies associated with moving the arm to any given programmed point form a normally distributed random variable with the mean at the taught point and the standard deviation equal to

(a)
(b)
(c)
(d) The repeatability of the robot.

2.9 The telescoping arm of a certain industrial robot obtains its vertical motion by rotating (type *R*

When fully retracted the arm measures 30 in. from the pivot point.

(a)
rotation.
(b) Determine the robot's control resolution on a linear scale in both the

(c) Sketch the side view of the robot's work volume as determined by this

2.10 The mechanism connecting the wrist assembly is a type *T* (twisting joint

to the other. It is desired to have a control resolution of plus or minus 0.2

order to achieve this resolution?

2.11 You are required to design a robot that can pick up an object of arbitrary shape

(a) What is the minimum number of degrees of freedom it should have?
(b)
(c) Draw their corresponding work volumes.

2.12
drive systems.

2.13

in assembly?

2.14

Report on Robot Joint Notation Scheme

2. *Robotics* in *Practice* *chaps*. 1 and 2.
3. M. P. on *Industrial Engineering* *CAD/CAM*: *Computer-Aided Design and Manufacturing* *IEEE Spectrum* *lndustrial Engineering* *Technical Report AFWAL-TR-80-4042*

Robot Technology: The Robot and Its Peripherals

PART TWO

Part Two of the book is focused on the mechanical and electronic technology of industrial robots. This includes the technology of both the manipulator itself and the peripheral devices and systems that work with the robot.

The design of the robot manipulator represents a significant challenge to mechanical and electrical engineers. As suggested by our discussion in Chapter 2, the problem is to configure a physical system that is capable of positioning its end effector to within several thousandths of an inch of a desired target location, is relatively lightweight, possesses high-lift capacity, and moves at high speeds between positions in the workspace. Chapters 3 and 4 consider this design problem. Chapter 3 examines the various components of the individual joints that make up the manipulator. Chapter 4 treats the mathematical analysis of the arm position and motion. These mathematical methods are utilized to aid engineers in the design of robots, and by the robot control computer to calculate motion trajectories and frame transformations during the work cycle.

Chapter 5 considers one of the important peripheral devices used with robots in industrial applications—their end effectors. The different types of end effector are discussed, along with the engineering considerations for their design. We will concentrate in this chapter on the grippers used to handle workparts. Chapters 6 and 7 examine another type of peripheral used in industrial robotics—sensors. Chapter 6 discusses the variety of sensors commonly found in robot applications, with emphasis on touch and force sensors. Chapter 7 surveys the technology of machine vision. Sensors, combined with a sufficiently powerful computer brain to use them, provide robots with significant capabilities to perform useful work.

3 Control Systems and Components

Introduction

A robot manipulator consists of several mechanical links connected in series or parallel by joints. Actuators at each joint supply the necessary energy to move the links, while sensors measure the position of each link. Software controls the position and orientation of the end effector by regulating each joint motion, to move the robot to a desired goal point. In this chapter, we will study the modeling of a robot manipulator using a simple spring mass damper system for each joint. First, we will review some of the basic concepts of mathematical modeling techniques to analyze control systems. Then we will show how to develop control methods to move a joint to a desired location or to follow a desired trajectory. Finally, modeling and control of a single joint manipulator consisting of a joint, motor with gear and a link will be studied.

3.1 BASIC CONTROL SYSTEMS CONCEPTS AND MODELS

This and the following two sections will review some of the basic concepts and mathematical modeling techniques used to analyze control systems. The emphasis will be on mechanical systems since a robot manipulator falls within this class. Readers already familiar with linear feedback systems may want to skim quickly over these sections and go to Sec. 3.4.

When studying a mechanical system, we are concerned with the response of the system to certain inputs. These inputs include commands to 'drive the system and disturbances from the environment. We can divide a system into five major components:

1. The input (or inputs) to the system
2. The controller and actuating devices
3. The plant (the mechanism or process being controlled)
4. The output (the controlled variable)
5. Feedback elements (sensors)

By analyzing the effects that the controller and the plant have on the inputs, we can predict what the outputs will be under certain conditions. In order to do this analysis, we must be able to model the system mathematically.

3.1.1 Mathematical Models

Mathematical models are simply mathematical representations of real world systems. They are developed by applying the known rules of behavior for the elements in a system. Hook's law for the operation of a spring is an example:

$$F = K_s x \tag{3.1}$$

where F is the force applied to the spring, x is the displacement due to the application of the force, and K_s is the 'spring constant.' Using physical relationships of this sort we can develop models of more complex systems than just a spring. As an example, let us formulate the model for a familiar mechanical system: the spring–mass–damper system.

The system is illustrated in Fig. 3.1 and consists of a block with a certain mass suspended from a fixed wall by a dashpot. The mass is connected to a spring, the other end of which is given some displacement from its equilibrium position. We will use the following symbols to represent the various parameters of system behavior:

y = the displacement of the mass
M = the mass of the block
K_s = the spring constant
K_d = the damping coefficient of the dashpot
x = the displacement of the end of the spring

The operation of the system can be described as a sum of the forces on the mass. The force due to acceleration of the mass is

$$M \frac{d^2y}{dt^2}$$

The force due to the dashpot is

$$K_s \frac{dy}{dt}$$

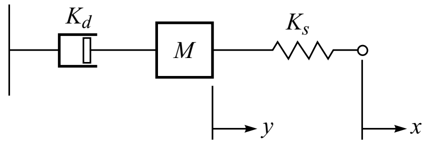


Fig. 3.1 *Schematic diagram of a spring-mass damper system.*

The force due to the spring is

$$K_s y - K_s x$$

Summing all of the forces, we get

$$M\frac{d^2y}{dt^2} + K_d\frac{dy}{dt} + K_s y = K_s x \tag{3.2}$$

In this system, the input is x, representing the displacement of the end of the spring, and the system output is y, representing the displacement of the block. The system has been described by a second-order linear differential equation which relates the input and the output. This mathematical description of the system allows us to analyze its behavior. Before beginning the analysis, however, we will develop other useful tools for building the models.

3.1.2 Transfer Functions

Linear differential equations can be rewritten using the differential operator, s. The variable, s, is used to represent the mathematical operation of taking the derivative of a time-dependent variable with respect to time. Thus, functions which are variables of time [e.g., $x(t)$ and $y(t)$] become functions of the variable s [e.g., $X(s)$ and $Y(s)$]' By using s with Laplace transforms, linear differential equations can be converted to equivalent expressions which are functions of s. (It is assumed that the reader is familiar with the Laplace transform and its s operator for linear systems analysis.) Using s, Eq. (3.2) can be written as

$$Ms^2Y(s) + K_d sY(s) + K_s Y(s) = K_s X(s) \tag{3.3}$$

The transfer function relates the output of the system to an input. The spring–mass–damper transfer function can be derived by rewriting Eq. (3.3) as

$$\frac{Y(s)}{X(s)} = \frac{K_s}{Ms^2 + K_d s + K_s} \tag{3.4}$$

3.1.3 Block Diagrams

It is often useful to provide a schematic representation of the system in addition to the mathematical model. A common means of graphically representing the relationships among the components of the system is the block diagram. Block diagrams are constructed from four basic elements:

1. Function blocks
2. Signal arrows
3. Summing junctions
4. Takeoff points

The four components are illustrated in Fig. 3.2.

A function block, shown in Fig. 3.2(a), represents one of the components of the system and contains the transfer function for the component. The signal arrows, illustrated in Fig. 3.2(*b*), indicate the direction of the signals and variables in the diagram and are used to connect function blocks and other system components. Summing junctions (also called summing points) permit two or more signals to be added (algebraically) as shown in Fig. 3.2(*c*). Takeoff points are pictured in part (*d*) of the figure and permit signals and variables to be shared among than a single component.

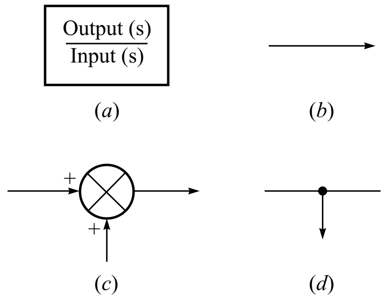


Fig. 3.2 *Block diagram elements: (a) Function block, (b) Signal arrow, (c) Summing junction, (d) Takeoff point.*

By assembling these components, it is possible to describe any linear system in the form of a block diagram. By convention, block diagrams are usually read from left to right, with inputs coming in from the left and outputs going to the right. Summing junctions may have any number of arrows entering, but only one leaving. Feedback loops generally run from right (the output side) to left (the input side).

Example 3.1 Draw the block diagram that corresponds to the spring–mass–damper system represented by Eqs. (3.2) and (3.3) in the text.

The resulting block diagram is displayed in Fig. 3.3.

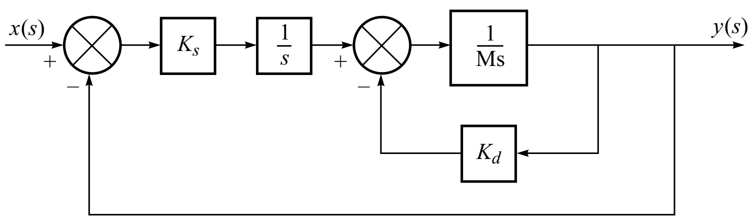


Fig. 3.3 *Block diagram for Example 3.1.*

Example 3.2 For the set of equations below, develop the block diagram which uses $X(s)$ as the system input and $Y(s)$ as the system output:

$$
\begin{aligned}
W(s) &= X(s) - Y(s) \\
V(s) &= W(s) - Z(s) \\
Z(s)(s+5) &= V(s)(s+2) \\
Y(s)(s^2+5s+6) &= Z(s)
\end{aligned}
$$

The resulting block diagram is shown in Fig. 3.4.

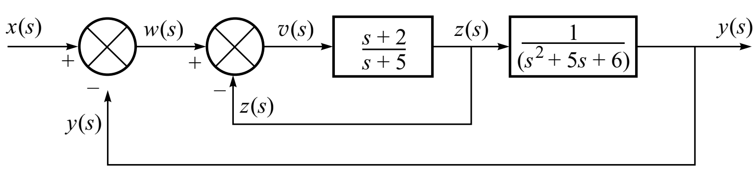


Fig. 3.4 *Block diagram for Example 3.2.*

Complicated block diagrams may be simplified by using reduction techniques called block diagram algebra. For example, two function blocks in series may be combined into a single block. Figure 3.5 illustrates these reduction techniques. The usual procedure is to first combine all series blocks in the diagram into the corresponding single block; then any parallel blocks are combined; next the basic feedback loops are reduced to equivalent single blocks and finally, the summing points are shifted to the left and the takeoff points to the right. For very complex block diagrams, the above procedures may have to be repeated in order to simplify the diagram to a single block representing the system transfer function.

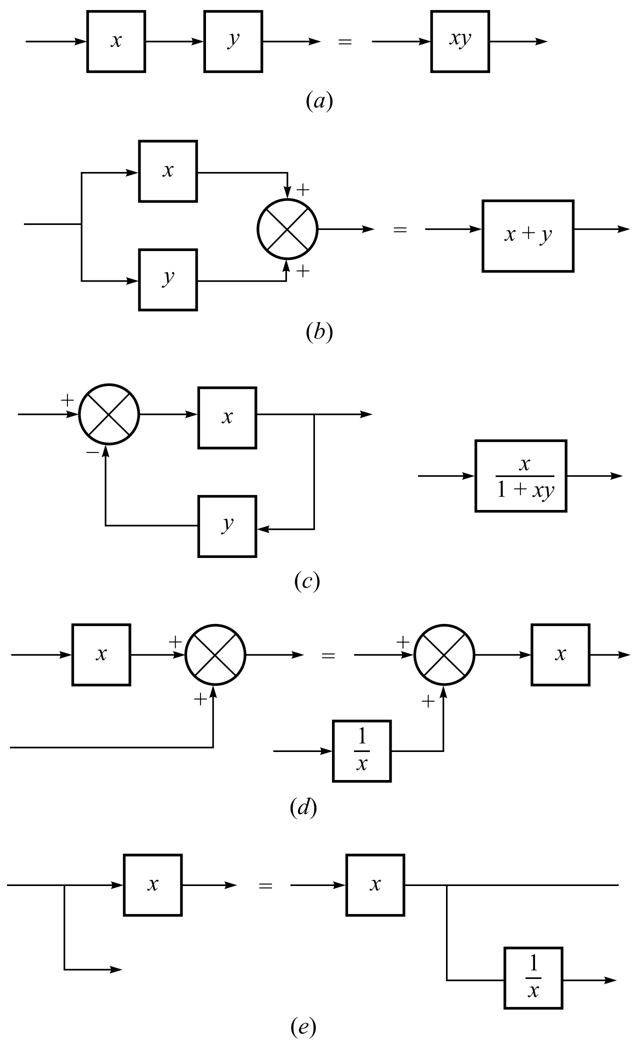


Fig. 3.5 *Block diagram algebra: (a) Blocks in series, (b) Blocks in parallel, (c) Elimination of a feedback loop, (d) Shifting a summing junction, (e) Shifting a takeoff point.*

Example 3.3 Using block diagram algebra, reduce the block diagram of Fig. 3.3 (Example 3.1) to a single block.

Starting with Fig. 3.3, we reduce the diagram in two steps as shown in Fig. 3.6. The system transfer function is in the resulting block.

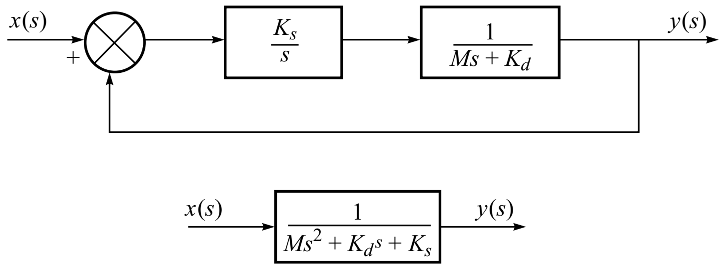


Fig. 3.6 *Block diagram reduction for Example 3.3.*

3.1.4 The Characteristic Equation

The characteristic equation for the spring–mass–damper system can be written as

$$Ms^2 + K_d s + K_s = 0 \tag{3.5}$$

and the roots of the characterstic equation, Eq. (3.5), are given by

$$s_{1,2} = -\frac{K_d}{2M} \quad \frac{\sqrt{K_d^2 - 4MK_s}}{2M} \tag{3.6}$$

The performance of the system is dependent on the values of M, K_d, and K_s. One aspect of the system performance that can he determined by analyzing the roots of the characteristic equation is the 'damping' of the system. Depending on the values of the parameters in the characteristic equation, the system may respond in one of four ways. The four responses classify the system into one of the following types:

1. Undamped system
2. Underdamped system
3. Critically damped system
4. Overdamped system

We will now briefly describe these four types of system response.

1. Undamped system In order for the system to be undamped, the damping coefficient, K_d, must be equal to zero. In this case the roots of the characteristic equation are given by

$$s_{1,2} = \pm\sqrt{\frac{K_s}{M}} \tag{3.7}$$

These are imaginary roots. Assuming a step input X to the system, the response can be decribed as

$$y = C_1 \sin(\omega_n t) + C_2 \cos(\omega_n t) + X \tag{3.8}$$

where $\omega_n = \sqrt{K_s/M}$ and is called the *natural frequency of the system*. The response represented by Eq. 3.8 is shown is Fig. 3.7(a) where it can be seen that the undamped response is oscillatory.

2. Underdamped system When there is a small amount of damping in the system, that is, where

$$K_d^2 < 4\ MK_s$$

the roots may be rewritten as

$$s_{1,2} = -\frac{K_d}{2M} \pm j\frac{\sqrt{4MK_s - K_d^2}}{2M} \tag{3.9}$$

Substituting $a = kd/2M$ and $\omega_d = \sqrt{(4MK_s - K_d^2)}\ \pm$ /2M. Equation (3.9) may be writen as

$$a_{1,2} = -\,a \pm j\omega_d \tag{3.10}$$

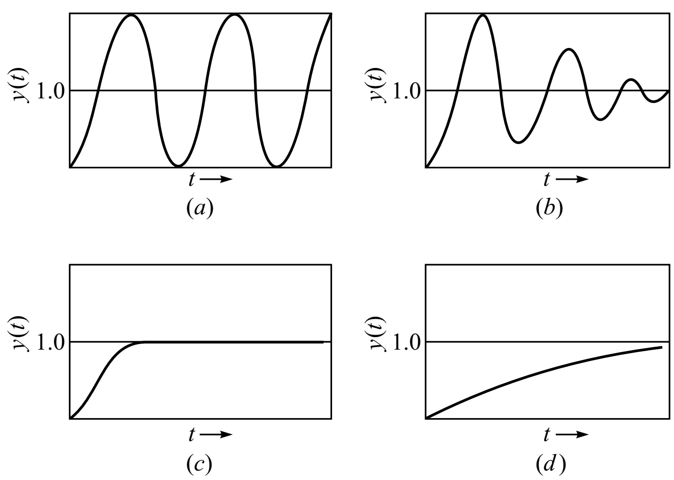


Fig. 3.7 *Response curves for the four possible cases in a second order linear system: (a) Undamped, (b) Underdarnped, (c) Critically damped, (d) Overdamped.*

The response of the system is described by

$$y = e^{-at}\,[\,C_1 \sin(\omega_d t) + C_2 \cos(\omega_d t)] + X \tag{3.11}$$

which is similar to the previous response except that ω_d represents the damped natural frequency and the term e^{-at} results in a decaying amplitude envelope for the oscillations as represented in Fig. 3.7(b).

3. Critically damped system This situation occurs in the special case where

$$K_d^2 = 4\ MK_s$$

which results in the roots of the characteristic equation being

$$-\frac{K_d}{2M}$$

In this case, the response of the system is given by

$$y = C_1 e^{-at} + C_2 t\, e^{-at} + X \tag{3.12}$$

and is represented in Fig. 3.7(*c*). The critically damped response provides the fastest response without overshoot to the input of the four types.

4. Overdamped system In the case of an overdamped system

$$K_d^2 > 4MKs$$

and the roots are

$$s_{1,2} = -a \pm b \tag{3.13}$$

where $b = \sqrt{K_d^2 - 4MK_s}/2M$. The response of the system is

$$y = C_1 e^{(-a+b)t} + C_2 e^{(-a-b)t} + X \tag{3.14}$$

This is illustrated in Fig. 3.7(*d*). As with the critically damped system this response does not oscillate, but the time to reach the desired steady-state response is longer.

Although a robot manipulator is mechanically far more complex than the spring–mass–damper system we have analyzed here, it exhibits many of the same operating features. It has mass, it has stiffness that can be likened to the spring constant, and the joints possess damping. The resulting motions of the robot arm behave in a manner similar to the performance of a system described by a second-order linear differential equation with constant coefficients. The complications of the robot manipulator are: (i) it represents a higher-order equation than second order; (ii) its degrees of freedom are greater than the single degree of freedom of the spring–mass–damper example, and these degrees of freedom interact to some extent; and (iii) its behavior includes features that are non-linear.

Notwithstanding the complications listed above, the design of a robot must address the same kinds of performance characteristics discussed in our example. For instance, the robot stability problem discussed in Chapter 2 (Secs. 2–4) takes on additional meaning in the light of the discussion in this section. There are trade-offs that must be made between the response of the system and the stability of the system. In some cases, it would be undesirable for the system to overshoot the target point and, therefore, it might be necessary to overdamp with the result that the speed of response would be slower.

Figure 3.8 illustrates a general block diagram of the control system components for one joint of a robot manipulator. The input command is the defined position (and possibly speed) to which the joint is directed to move. The output variable is the actual position (and speed) of the joint. In nearly all robots today, most of the computational functions of the joint controller are carried out by a microprocessor as portrayed in Fig. 3.8.

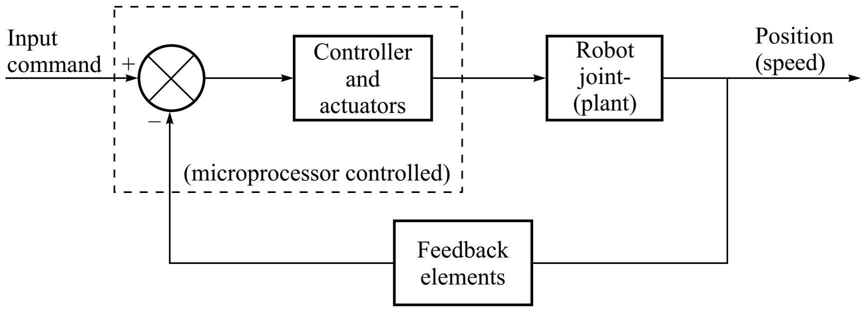


Fig. 3.8 *Typical block diagram configuration of a control system for a robot joint.*

Up to this point, we have focused on the modeling of physical systems and the ways in which a system may respond. Let us turn our attention to the methods of controlling the response of a system.

3.2 CONTROLLERS

As indicated in Fig. 3.8, the components of a control system include the controller and actuator. The purpose of the controller is to compare the actual output of the plant with the input command and to provide a control signal which will reduce the error to zero or as close to zero as possible. In this section, we will present the theoretical operation of these controllers and in the next section, we will discuss the physical devices which are used to implement this control in a robot joint.

The controller generally consists of a summing junction where the input and output signals are compared, a control device which determines the control action, and the necessary power amplifiers and associated hardware devices to accomplish the control action in the plant. The actuator is used in robotics to convert the control action into physical movement of the manipulator. The controller and actuator may be operated by pneumatic, hydraulic, mechanical, or electronic means, or combinations of these.

There are four basic control actions which are used singly or in combination to provide six common types of controller: on-off control, proportional control, derivative control and integral control. The six controller types are:

1. On-off
2. Proportional
3. Integral
4. Proportional-plus-integral (P-I)
5. Proportional-plus-derivative (P-D)
6. Proportional-plus-integral-plus-derivative (P-I-D)

Each one of these controller types is best suited to certain applications. The following subsections describe the operation of each type of controller.

1. On-Off Control In the on–off controller, the control element provides only two levels of control, full-on or full-off. An example of a common implementation of this type of controller is the household thermostat. If the error which is present at the controller is $e(t)$ and the control signal which is produced by the controller is $m(t)$, then the on–off controller is represented by

$$\begin{aligned} m(t) &= M_1 \quad \text{for } e(t) > 0 \\ &= M_2 \quad \text{for } e(t) < 0 \end{aligned} \tag{3.15}$$

In most on–off controllers either M_1 or M_2 is zero. The practical use of an on–off controller usually requires that the error must move through some range before switching actually takes place. This prevents the controller from oscillating at too high a frequency. This range is referred to as the differential gap.

2. Proportional Control In cases where a smoother control action is required a proportional controller may be used. Proportional control provides a control signal that is proportional to the error. Essentially, it acts as an amplifier with a gain K_p. Its action is represented by

$$m(t) = K_p e(t) \tag{3.16}$$

Using the differential operator notation introduced earlier the transfer function would be

$$\frac{M(s)}{E(s)} = K_p \tag{3.17}$$

3. Integral Control In a controller employing an integral control action the control signal is changed at a rate proportional to the error signal. That is, if the error signal is large, the control signal increases rapidly, if it is small, the control signal increases slowly. This may be represented by

$$m(t) = K_i \int e(t)\, dt \tag{3.18}$$

where K_i is the integrator gain. The corresponding transfer function is

$$\frac{M(s)}{E(s)} = K_i/s \tag{3.19}$$

using $1/s$ as the operator for integration. If the error were to go to zero, the output of the controller would remain constant. This feature allows integral controllers to be used when there is some type of constant load on the system. Even if there is no error the controller would still maintain an output signal to counteract the load.

4. Proportional-plus-Integral Control Sometimes it is necessary to combine control actions. A proportional controller is incapable of counteracting a load on the system without an error. An integral controller can provide zero error but usually provides slow response.

One way to overcome this is with the P-I controller. This is represented by

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int e(t) dt \tag{3.20}$$

where T_i adjusts the integrator gain and K_p adjusts both the integrator and the proportional gain. The transfer function is

$$\frac{M(s)}{E(s)} = K_p\left(1+\frac{1}{T_i s}\right) \tag{3.21}$$

5. Proportional-plus-Derivative Control Derivative control action provides a control signal proportional to the rate of change of the error signal. Since this would generate no output unless the error is changing, it is rarely used alone. The P-D controller is represented by

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \tag{3.22}$$

and the transfer function is

$$\frac{M(s)}{E(s)} = K_p(1 + T_d s) \tag{3.23}$$

The effect of derivative control action is to anticipate changes in the error and provide a faster response to changes.

6. Proportional-plus-Integral-plus-Derivative Control Three of the control actions can be combined to form the P-I-D controller. The P-I-D controller can be represented by

$$m(t) = K_p e(t) + \frac{K_p}{T_i}\int e(t)dt + K_p T_d \frac{de(t)}{dt} \tag{3.24}$$

and the transfer function is

$$\frac{M(s)}{E(s)} = K_p\left(1+\frac{1}{T_i s}+T_d s\right) \tag{3.25}$$

P-I-D control is the most general control type and probably the most commonly used type of controller. It provides quick response, good control of system stability and low steady-state error. As indicated previously, the computations associated with any of the above controllers are typically performed by microcomputers in a modern robot controller.

3.3 CONTROL SYSTEM ANALYSIS

Analysis of a control system may be divided into two parts: transient response and steady-state response. The transient response of a system is the behavior of the system during the transition from some initial state to the final state. The steady-state response is the behavior of the system as time approaches infinity.

3.3.1 Transient Response of Second-Order Systems

Second-order linear systems are frequently used in control systems analysis, even when it is known that the particular system of interest may be of higher order. Second-order systems can often approximate complex physical systems with reasonable fidelity. Let us return to our transfer function for the secondorder system derived in Sec. 3.1.

$$\frac{Y(s)}{X(s)} = \frac{K_s}{Ms^2 + K_d s + K_s} \tag{3.26}$$

The natural frequency of the system was represented by

$$\omega_n = \sqrt{\frac{K_s}{M}}$$

The damping ratio of a second-order system can he defined as

$$z = \frac{Kd/2M}{\omega_n}$$

If the damping ratio is equal to zero then the system will oscillate continuously, if $z < 1$ but greater than zero then the system is underdamped, If $z = 1$ then the system is critically damped and if $z > 1$ the system is overdamped. Fig. 3.9 illustrates the transient response of a second-order system with different damping ratios to a unit step input. There are other parameters of interest in the transient response of a system. These are:

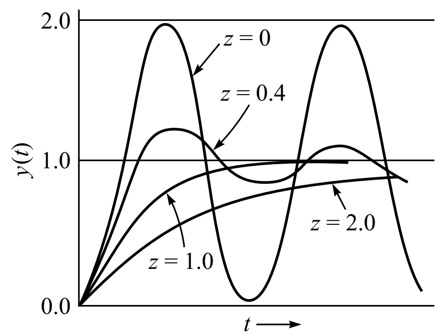


Fig. 3.9 *Transient responses for different damping ratios.*

1. Delay time, t_d Delay time is the time that it takes the system to reach one-half of the final value for the first time.

2. Rise time, t_r Rise time is the time that it takes the system to go from 10 to 90 per cent, 5 to 95 per cent, or 0 to 100 per cent of the final value.

3. Peak time, t_p Peak time is the time that it takes the system to reach the maximum overshoot for the first time.

4. Maximum overshoot, M_p Maximum overshoot is the maximum peak value measured from the steady state value in Fig. 3.10.

5. Settling time, t_s Settling time is the time required for the system to stay within a range about the final value. This is usually within 2 and 5 per cent.

Figure 3.10 illustrates these system parameters. In some cases, certain of the parameters are not relevant. In the case of a critically damped system there is no overshoot and hence M_p and t_p do not apply. In robotics, it is sometimes critical that the system not be allowed to overshoot, while in other applications it may be necessary for the sake of speed, to allow overshoot. The balancing of these parameters when designing the system is the responsibility of the controls engineer.

Included within the scope of the transient response is the question of whether the system will be stable for all inputs. System stability is interpreted to mean that the

input. Stability is assured if the transients gradually go toward zero as time increases. System instability occurs when the transient response increases with time.

The stability of any linear system can be determined if the characteristic equation of the system is known and if it is possible to factor the equation. Referring back to our example of a second-order linear system, let us relate our discussion of stability to the four system responses described in Section 3.1. In the case of the underdamped, critically damped, and overdamped systems, the transient responses gradually decrease with time as the output assumes some steady-state value. These systems are all stable. The common feature which makes each of these systems stable is the fact that their characteristic equations have roots that are negative real numbers or complex numbers with negative real parts. This is the requirement for stability. If the roots are negative real numbers or complex numbers with negative real parts, the transient response will always approach zero with time.

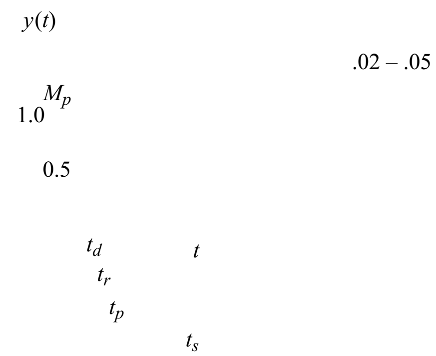


Fig. 3.10 *Transient response parameters.*

3.3.2 Steady-State Response

The steady-state analysis of a control system is concerned with determining the response of the system after the transient response has disappeared. It is assumed that the system of interest is one which is stable. In steady-state analysis, the system

time of operation increases. One approach to the problem would be to solve the differential equation of the system as it is subjected to some appropriate input.

theorem from control theory which uses the Laplace transform of the system output.

$$\lim_{t} f(t) = \lim_{s \to 0} sF(s) \tag{3.27}$$

where $F(s)$ is the Laplace transform of the function $f(t)$. Implicit in the above statement of the final value theorem is that the limit of $f(t)$ exists as time approaches infinity.

Example 3.4 Suppose that in Eq. (3.2) the values of the constants are $M = 2$, $K_d = 6$, and $K_s = 5$. Further suppose that the input, x, to the system is a unit step function. Determine the steady-state response of the system according to the final value theorem.

The unit step response has a Laplace transform = l/s. The transfer function for the system represented by Eq. (3.2) for the values given is

$$\frac{Y(s)}{X(s)} = \frac{5}{2s^2 + 6s + 5}$$

Substituting the unit step input, $1/s$, for $X(s)$ gives us the Laplace transform of the system response, $y(t)$. Expressed in the s domain, we have

$$Y(s) = \frac{1}{s}\frac{5}{(2s^2 + 6s + 5)}$$

Substituting $Y(s)$ into Eq. (3.27), we get

$$\lim_{t\to\infty} = \lim_{s\to 0}\frac{5s}{s(2s^2 + 6s + 5)} = 1$$

3.4 ROBOT SENSORS AND ACTUATORS

A robot moves in space to perform tasks and hence it needs actuators to move the links and sensors to know where each joint is. Sensors inform the controller by how much each joint has moved and thus enables the controller to enforce a particular velocity or position during motion. As we have seen in the previous section each joint has an independent control structure in which there is an actuator, reduction gear mechanism and a sensor at each joint.

Sensors can be divided into two parts:

1. Internal sensors These are responsible for the internal working of the robot and are mainly used for closing the loop in feedback control e.g., position sensors. A robot cannot function properly without these if it is using a closed loop feedback control system. The main internal sensors are position and velocity sensors.

2. External sensors These are responsible for interaction with the environment. A robot can use external sensors like touch sensor for interaction with the environment. In case any of these sensors fail the robot can still function but its ability to interact with the external world is reduced. External sensors are of many different types depending on the kind of interaction with the environment. The main external sensors are force/torque sensors, vision, touch, pressure sensors, etc.

We will discuss about internal sensors like potentiometers, resolvers and encoders in this chapter. External sensors will be discussed in later chapters.

3.4.1 Position and velocity sensors

Potentiometers Potentiometers are analog devices whose output voltage is proportional to the position of a wiper. Fig. 3.11 illustrates a typical pot. A voltage is applied across the resistive element. The voltage between the wiper and ground is proportional to the ratio of the resistance on one side of the wiper to the total resistance of the resistive element. Essentially the pot acts as a voltage divider network. That is, the voltage across the resistive element is divided into two parts by a wiper. Measuring this voltage gives the position of the wiper. The function of the potentiometer can be represented by the following function:

$$V_o(t) = K_p\theta(t) \tag{3.28}$$

where $V_o(t)$ is the output voltage, K_p is the voltage constant of the pot in volts per radian (or volts per inch in the case of a linear pot) and $\theta(t)$ is the position of the pot in radians (or inches). Since a pot requires an excitation voltage, in order to calculate V_o, we can use

$$V_o = V_{ex}\frac{\theta_{act}}{\theta_{tot}} \tag{3.29}$$

where V_{ex} is the excitation voltage, θ_{tot} is the total travel available of the wiper, and θ_{act} is the actual position of the wiper.

Potentiometers can be single turn in which the rotating wiper can move only by 360° or they can also be multi turn in which the rotating wiper can move by several 360° turns. Potentiometers suffer from disadvantages like non-linearity and low life due to the continuous friction between the wiper and the variations in the resistive element. In addition, the variation in wiper contact between the coil and the wiper can lead to noise in position measurement.

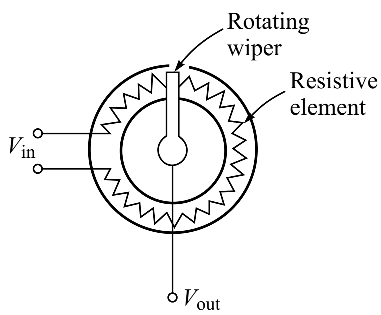


Fig. 3.11 *Potentiometer.*

Example 3.5 Find the output voltage of a potentiometer with the following characteristics. Also determine the K_p. The excitation voltage = 12 V; total wiper travel = 320°; wiper position = 64°.

Solution The $K_p = V_{ex}/\theta_{tot}$ which is 12 V/320° = 0.0375 V/deg. The output voltage is

$$(64°)(0.0375 \text{ V/deg}) = 2.4 \text{ V}.$$

Resolvers A resolver is another type of analog device whose output is proportional to the angle of a rotating element with respect to a fixed element. In its simplest form, a resolver has a single winding on its rotor and a pair of windings on its stator. The stator windings are 90° apart as shown in Fig. 3.12. If the rotor is excited with a signal of the type $A \sin(\omega t)$ the voltage across the two pairs of stator terminals will be

$$V_{s1}(t) = A \sin(\omega t) \sin \theta \tag{3.30}$$

and

$$V_{s2}(t) = A \sin(\omega t) \sin \theta \tag{3.31}$$

where θ is the angle of the rotor with respect to the stator. This signal may be used directly, or it may be converted into a digital representation using a device known as a 'resolver-to-digital' converter. Since a resolver is essentially a rotating transformer, it is important to remember that an ac signal must be used for excitation. If a dc signal were used there would be no output signal

Example 3.6 At time t the excitation voltage to a resolver is 24 V. The shaft angle is 90°. What is the output signal from the resolver?

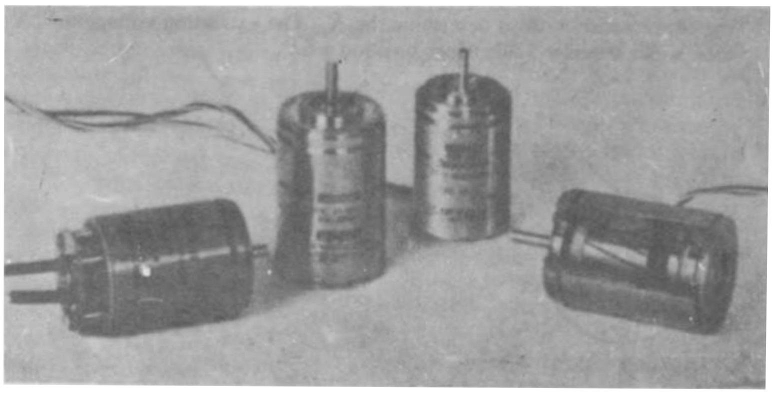


Fig. 3.12 *Resolver. (Courtesy: Litton Systems, Incorporated, Clifton Precision Division)*

Solution

$$V_{s1} = (24 \text{ V}) (\sin 90°) = 24 \text{ V}$$
$$V_{s2} = (24 \text{ V}) (\cos 90°) = 0 \text{ V}.$$

Example 3.7 At time t the excitation voltage to a resolver is 24 V and V_{s1} = 17 V and V_{s2} = – 17 V. What is the angle?

Solution

$$\arcsin\left(\frac{17}{24}\right) = 45° \text{ or } 135°$$

$$\arccos\left(-\frac{17}{24}\right) = 135° \text{ or } 225°$$

The shaft angle must be 135°.

Encoders As microprocessors have become cheaper and with a move towards digital electronics, the encoder is virtually used everywhere for position measurement. Almost all industrial robots, NC machines, etc., use encoders to measure the position and velocity of motion. Encoders are available as two basic types: incremental and absolute. There are various categories of encoding devices, but we will limit our discussion to those that are most commonly used in robots, i.e., optical encoders. A simple incremental encoder is illustrated in Fig. 3.13.

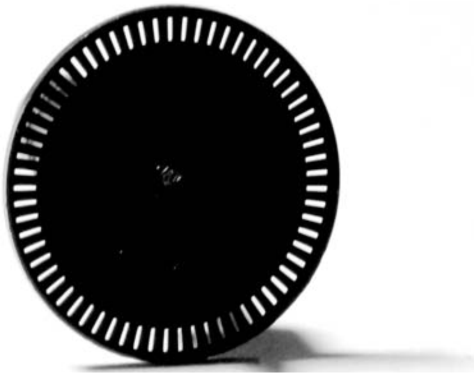


Fig. 3.13 *Incremental encoder.*

An incremental encoder consists of a disk marked with alternating transparent and opaque stripes aligned radially. A phototransmitter (a light source) is located on one side of the disk and a photo receiver is on the other Fig. 3.14. As the disk rotates, the light beam is alternately completed and broken. The output from the photoreceiver is a pulse train whose frequency is proportional to the speed of rotation of the disk. In a typical encoder, there are two sets of phototransmitters and receivers aligned 90° out of phase. This phasing provides direction information, that is, if signal A leads to signal B by 90° the encoder disk is rotating in one direction, if B leads A then it is going in the other direction. By counting the pulses and by adding or subtracting based on the sign, it is possible to use the encoder to provide position information with respect to a known starting location. Normally, two incremental encoders are used in parallel so that the resolution of measurement is increased. These two signals are passed through an XOR gate. It can be seen that the resolution of the resulting

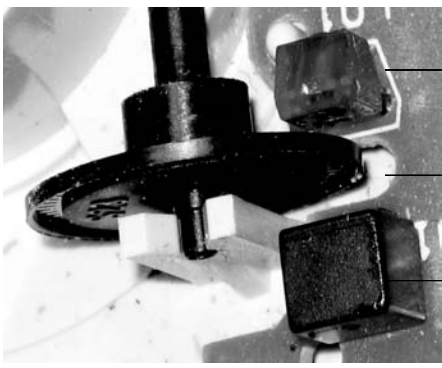


Fig. 3.14 *Photo transmitter and receiver place on a incremental encoder.*

signal is now increased two times, as we now have two pulses in place of only one pulse from each encoder. Most modern position control systems have two or more encoders in parallel to increase the resolution of the systems. The rate at which the pulses are generated by the encoder can also be counted to get an estimate of the velocity of the rotating shaft. Hence, an encoder can also be used as a velocity sensor. In some cases, it is desirable to know the, position of an object in absolute terms, that is, not with respect to a starting position. For this an absolute encoder could be used. Absolute encoders employ the same basic construction as incremental encoders except that there are more tracks of stripes and a corresponding number of receivers and transmitters. Usually the stripes are arranged to provide a binary number proportional to the shaft angle. The first track might have two stripes, the second four, the third eight and so on. In this way the angle can be read directly from the encoder without any counting being necessary. Figure 3.15 illustrates an absolute encoder. The resolution of an absolute encoder is dependent on the number of tracks and is given by

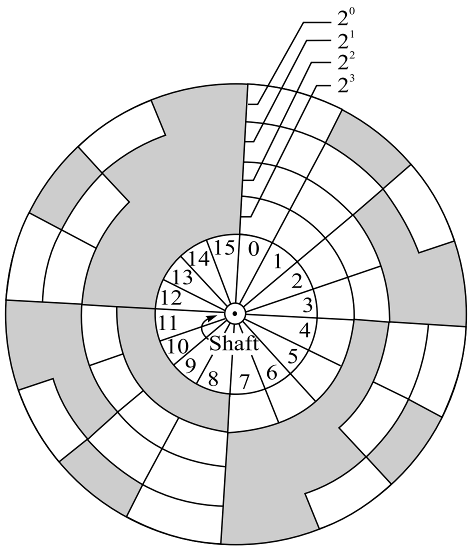


Fig. 3.15 *Absolute optical encoder.*

$$\text{resolution} = 2^n \tag{3.32}$$

where n is the number of tracks on the disk.

Example 3.8 What is the resolution, in degrees, of an encoder with 10 tracks?

The number of increments per revolution is 2^{10}

$$= 1024 \text{ increments/rev}$$

The angular width of each control increment is therefore

$$\frac{360°}{2^{10}} = \frac{360°}{1024} = 0.3515°$$

The output of an absolute encoder or of an incremental encoder and counter combination is represented by

$$\text{out}(t) = K_e\theta(t) \tag{3.33}$$

where out (t) is a number, K_e is the number of pulses per radian and θ is the shaft angle, expressed in radians.

Example 3.9 What is the output value of an absolute encoder if the shaft angle is 1 rad. and the encoder has 8 tracks?

The resolution is 256 parts/rev. There are 27π rad/rev. Therefore, the output is

$$\frac{256}{2\pi} = 41$$

3.6 VELOCITY SENSORS

Velocity information is required for closed loop feedback control using a PD or PID controller. One of the most commonly used devices for the feedback of velocity information is the dc tachometer. A tachometer is essentially a dc generator providing an output voltage proportionat to the angular velocity of the armature.Velocity information can also be obtained from an incremental encoder in which the rate at which the dark and transparent slots cross the emitter receiver pair indicate the velocity of the rotating shaft. Figure 3.16 illustrates a typical dc tachometer. A tachometer can be described by the relation,

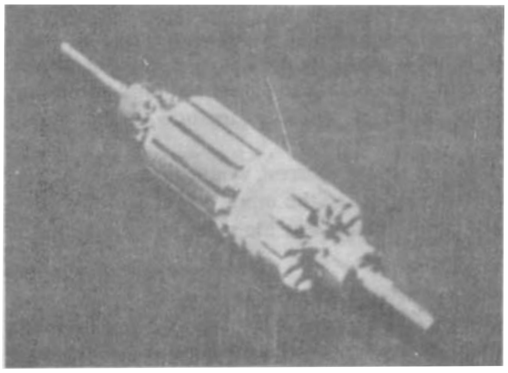


Fig. 3.16 *Tachometer mounted on a motor armature. (Courtesy: Litton Systems, Incorporated, Clifton Precision Division)*

$$V_0(t) = K_t(t)\omega \tag{3.34}$$

where $V_o(t)$ is the output voltage of the tachometer in volts, $K_t(t)$ is the tachometer constant, usually in V/rad/s and ω is the angular velocity in radians per second.

Tachometers are generally used to provide velocity information to the controller. This can be used for performing velocity control of a device or, in many cases, to increase the value of K_d in a system, thereby improving the stability of the system and its response to disturbances.

Direct current tachometers provide a voltage output proportional to the armature rotational velocity, hence they are analog devices. There is a digital equivalent of the dc tachometer which provides a pulse train output of a frequency proportional to the angular velocity. They are in effect, encoders which were described in the previous section.

3.7 ACTUATORS

Actuators are the devices which provide the actual motive force for the robot joints. They commonly get their power from one of three sources: compressed air,

actuators respectively. We will discuss all three types in this section.

3.7.1 Pneumatic and Hydraulic Actuators

pressurized oil. The operation of these actuators is generally similar except in their

about 100 lb/in.^{2} and hydraulic systems at 1000 to 3000 lb/in.^{2} We discussed the relative advantages and disadvantages of these types of drive systems in Chap. Two.

could be used to actuate a linear joint by means of a moving piston. This example is called a *single-ended cylinder* as the piston rod only comes out of the cylinder at one end. Other types of cylinders include double-ended cylinders and rodless cylinders.
There are two relationships of particular interest when discussing actuators: the velocity of the actuator with respect to the input power and the force of the actuator with respect to the input power. For the cylinder type actuator these relationships are given by

$$V(t) = \frac{f(t)}{A} \tag{3.35}$$

$$F(t) = P(t)A \tag{3.36}$$

Cylinder

Fluid port

Piston

Piston rod

Fig. 3.17 *Cylinder and piston.*

where $V(t)$ is the velocity of the piston, $f(t)$
$F(t)$ is the force, $P(t$ A is the area of the piston. Since the requirements of a robot are to carry a payload at a given speed, we can use the relations described for choosing the appropriate actuator.

3.7.2 Electric Motors

As their capabilities improve, electric motors are becoming more and more the actuator of choice in the design of robots. They provide excellent controllability with a minimum of maintenance required. There are a variety of types of motors in use in robots; the most common are dc servomotors, stepper motors, and ac servomotors.

DC motors that are used in closed loop position control are called servo motors. As servo motors are required to have very large acceleration or deceleration in order to have faster response, the armatures of these motors are specially designed to have low inertia. DC servo motors can be either brushed motors or brushless motors. The basic construction difference between a brushed and brushless motor is that in a brushless motor, the armature winding are in the stator and the permanent magnets are in the rotor. While in a brushed motor, the armature windings are in the rotor and the permanent magnets are in the stator.

Figure 3.18 shows a typical dc servomotors with a permanent magnet stator and winding on rotor. The main components of the dc servomotor are the rotor and the stator. Usually, the rotor includes the armature and the commutator assembly and

set up by the magnets. This produces a torque on the rotor. As the rotor rotates, the

remains opposed to the one set up by the magnets. In this way, the torque produced by the rotor is a function of the current through it, it can be shown that for a dc servomotor

$$T_m(t) = K_m I_a(t) \tag{3.37}$$

Fig. 3.18 *Brushed DC motor showing the armature (rotor) and permanent magnet (stator).*

where T_m is the torque of the motor, I_a is the current flowing through the armature, and K_m is the motor's torque constant.

Another effect associated with a dc motor is the back-emf. A dc motor is similar to a dc generator or tachometer. Spinning the armature in the presence of a magnetic field produces a voltage across the armature terminals. This voltage is proportional to the angular velocity of the rotor:

$$e_b(t) = K_b \omega(t) \tag{3.38}$$

where e_b is the back-emf (voltage), K_b is called the voltage constant of the motor, and ω is the angular velocity. The effect of the back-emf is to act as viscous damping for the motor: as the velocity increases the damping increases proportionately. If we were to supply a voltage across the motor terminals of V_{in}, and the resistance of the armature were R_a, then the current through the armature would be V_{in}/R_a. This current produces a torque on the rotor and causes the motor to spin. As the armature spins it generates a back-emf equal to $K_b\omega(t)$, or $e_b(t)$. This voltage must be subtracted from V_{in} in order to calculate the armature current. The actual armature current is therefore

$$I_a(t) = \frac{V_{\text{in}}(t) - e_b(t)}{R_a} \tag{3.39}$$

As the motor velocity increases, and the back-emf voltage increases accordingly, the current available to the armature decreases. The decreasing current reduces the torque generated by the rotor. As the torque decreases the acceleration of the rotor decreases as well. At the point at which $e_b = V_{\text{in}}$ the motor maintains a steady-state velocity (assuming there are no external disturbances on the motor). The block diagram in Fig. 3.19 illustrates the effects of the torque constant and back-emf on the model of a motor. Note that this simplified model discounts such effects as friction or inductance of the armature windings.

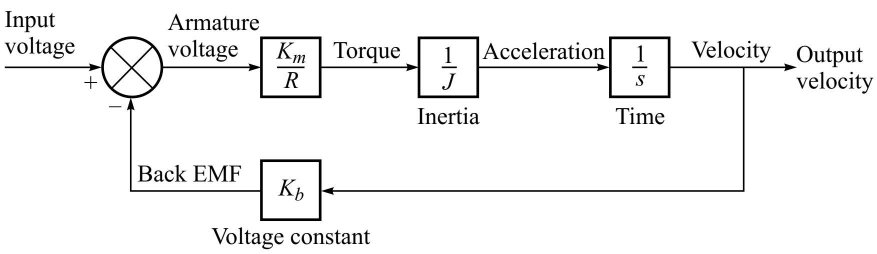


Fig. 3.19 *DC motor block diagram.*

Example 3.10 A motor has a torque constant, K_m = 10 oz-in./A and a voltage constant of 12 V/Kr/min (1 Kr/min = 1000 r/min). The armature resistance is 2 Ω. If 24 V were applied to the terminals what would be: (*a*) the torque at stall (0 r/min), (*b*) the speed at 0 load (torque = 0), and (*c*) the torque at 100 r/min? Plot the results on a speed versus torque graph.

Solutions

(*a*) At 0 r/min the value of e_b = 0. Therefore, the armature current is

$$= 24 \text{ V}/2\ \Omega = 12 \text{ A}$$

and the torque is

$$= (12 \text{ A})\ (10 \text{ oz-in./A}) = 120 \text{ oz-in.}$$

(*b*) At no load the output voltage is equal to the input voltage so that

$$24\text{ V} = (12\text{ V/Kr/min})\, w(t)$$
$$w(t) = 2\text{ Kr/min} = 2000\text{ r/min}$$

(*c*) At 1000 r/min the output voltage is 12 V. Therefore the current through the armature is

$$\frac{24\text{ V} - 12\text{ V}}{2\ \Omega} = 6\text{ A}$$

and the torque is

We can see that the relationship between speed and torque in Fig. 3.20 is a straight line. The feature is one of the desirable feature of dc servomotors.

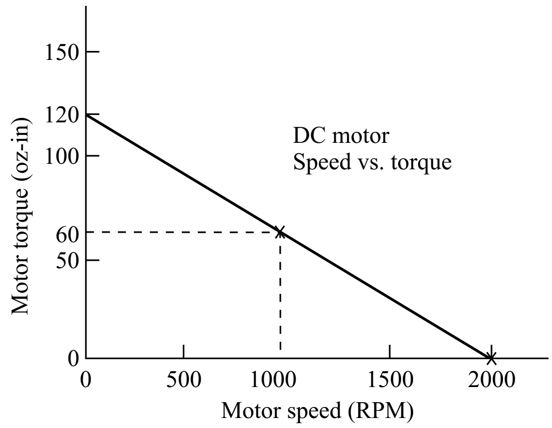


Fig. 3.20 *Plot for Example 3.11.*

3.7.3 Stepper Motors

Stepper motors (also called stepping motors) are a unique type of actuator and have been used mostly in computer peripherals. A stepper motor provides output in the form of discrete angular motion increments. It is actuated by a series of discrete electrical pulses. For every electrical impulse there is a single-step rotation of the motor shaft. In robotics, stepper motors are used for relatively light duty applications. Also, stepper motors are typically used in open-loop systems rather than the closed-loop systems on which we have been concentrating in this chapter.

Figure 3.21 provides a schematic representation of one type of stepper motor. The stator is made up of four electromagnetic poles and the rotor is a two-pole permanent magnet. If the electromagnetic stator poles are activated in such a way that pole 3 is N (magnetic North) and pole I is S then the rotor is aligned as illustrated. If the stator is excited so that pole 4 is N and pole 2 is S, the rotor makes a 90° turn in the clockwise direction. By rapidly switching the current to the stator electronically, it is possible to make the motion of the rotor appear continuous.

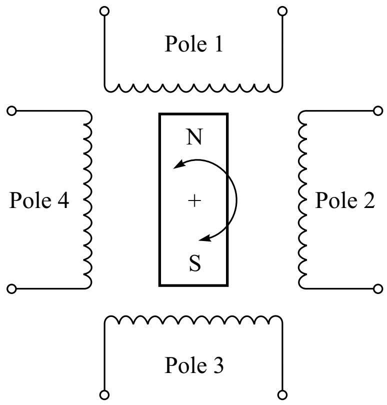


Fig. 3.21 *Stepper motor schematic.*

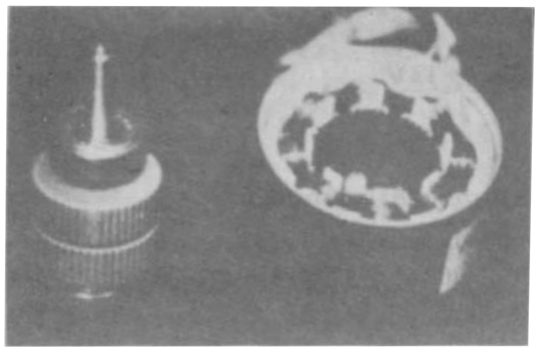


Fig. 3.22 *Toothed stepper motor. (Courtesy: Litton System, Incorporated, Clifton Precision Division)*

The resolution (number of steps per revolution) of a stepper is determined by the number of poles in the stator and rotor. Figure 3.22 show a commercially available stepper motor which has a notched stator and rotor. This effectively increses the number of poles and hence the resolution of the device. The relation between a stepper motor's resolution and its step angle is given by

$$n = A/360° \tag{3.42}$$

where n is the resolution and A is the step angle.

Unlike the dc servomotor, the relation between a stepper motor's speed and torque is not necessarily a straight line. Because of the discrete nature of the stepper motor's construction the torque is also a function of the angle between the stator and rotor poles. The torque is greatest when the poles are aligned. This maximum torque is known as the *holding torque of the motor*. It is possible to increase the resolution of a stepper by using a technique known as *half-stepping* or *microstepping*. By applying current to more than one set of field windings, it is possible to make the rotor seek out an 'average' position. Of course, when using this technique the holding torque is reduced.

The control of a stepper motor is dependent on the ability of the switching electronics to switch the windings at precisely the right moment. If the windings are switched too quickly, for example, it is possible that the motor will not be able to 'keep up' with the command signals and will perform erratically, in some cases oscillating. With some steppers, the speed-torque relation degrades badly at certain frequencies of operation, and operation of the motors at these frequencies must be avoided.

3.7.4 AC Servomotors and Other Types

There are numerous other aspects of electric motors which may be investigated. Recent advances in control electronics are producing ac servomotors. These motors have the advantages of being cheaper to manufacture than dc motors; they have no brushes, and they possess a high power output. With proper electronics package, however, their performance can be made to look very much like the performance of a dc motor.

Another type of electric motor is the brushless dc motor. It is constructed like an 'inside-out' dc motor. It has a permanent magnet rotor and an electromagnetic stator. Instead of using brushes, however, commutation is performed electronically using an encoder to inform the electronics of the relative positions of the stator and rotor. Also available are linear electric motors. Their construction is similar to a dc servomotor that has been cut open and flattened out.

In almost all cases of electric motors the limiting factor on power output is heat dissipation. Some of the current used in the motor must be dissipated as heat. Two ways to increase the performance of a motor is to remove heat more quickly or to reduce the current requirements. The latter may be done by increasing the magnetic flux of the permanent magnets. Recent advances in magnetic materials are allowing for performance improvements of almost 10 times with the same power requirements.

3.8 POWER TRANSMISSIONS SYSTEMS

In many cases, it is not possible to find an actuator with the exact speed-force or speed-torque characteristics to perform the desired tasks. In other cases it is necessary to locate the actuator away from the intended joint of the manipulator. For these reasons it becomes necessary to use some type of power transmission. Power transmissions perform two functions: transmit power at a distance and act as a power transformer. There are a number of ways to perform mechanical power transmission. These include belts and pulleys, chains and sprockets, gears, transmission shafts, and screws. In this section, we will discuss a number of power transmission devices that are used for industrial robots.

3.8.1 Gears

The use of gears for power transmission in robots is very common. Gears are used to transmit rotary motion from one shaft to another. This transfer may be between parallel shafts, intersecting shafts, or skewed shafts. The simplest types of gears are for transmission between parallel shafts and are known as *spur gears*.

Figure 3.23 illustrates a simple two-gear spur gear train. The driving gear, in this case the smaller one, is known as the *pinion* and the other gear is the *driven gear*. For example, if the pinion is one-fourth the size of the gear, for every revolution made by the pinion the driven gear turns only one-fourth of a revolution. This gear train is referred to as a *speed reducer*. The torque applied by the pinion however is multiplied by four times at the gear shaft. Since the speed is quartered and the torque is quadrupled the power out of the gear train equals the power into it.

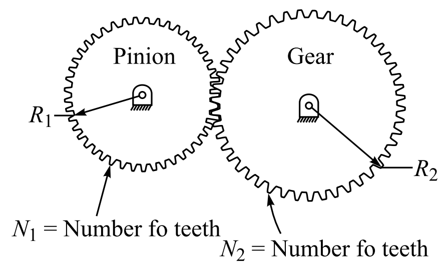


Fig. 3.23 *Spur gear train.*

The number of teeth in a gear is proportional to its diameter. If we let the number of teeth in a pinion be N_1 and the teeth in the gear equal N_2 then the gear ratio is given by

$$n = \frac{N_1}{N_2} \tag{3.43}$$

and the speed of the output with respect to the input is

$$\omega_o = n\omega_{\text{in}} \tag{3.44}$$

where ω_o is the output speed and ω_{in} is the input speed. The output torque is

$$T_o = \frac{T_{\text{in}}}{n}$$

3.8.2 Power Screws

In robotics and many other applications, power screws are often used to convert rotary motion to linear motion. The parameter of a screw which is analogous to a gear ratio is the screw pitch, p, often called the *lead*. The pitch defines the distance that the screw travels in a single rotation. The conversion for the screw's angular rotation to linear motion is given by

$$v(t) = p\omega(t) \tag{3.45}$$

where $v(t)$ is the linear velocity in inches per minute, $\omega(t)$ is the angular velocity in rotations per minute, and p is the screw pitch expressed in inches per rotation. In most cases, the screw is rotating and a nut is moving along the length of the screw. The conversion from torque, T, applied to the screw to force, F, on the nut is obtained by the following Eq. (3.46):

$$F = \frac{2T}{d_m}\frac{\pi d_m - \mu p \sec\beta}{p + \mu\pi d_m \sec\beta} \tag{3.46}$$

where μ

β

d_m = the mean diameter of the screw

angle, β. For square threads, the value of β is 0, and the secant terms = 1.0 in Eq. (3.46).

Example 3.11 A power screw mechanism is used to actuate a linear (type *L*) joint in a new robot design. Determine the maximum force that can be transmitted to the nut moving along the power screw if the torque available to turn the screw is 2.0 in.-lb. The screw has square threads (β = 0) whose pitch is

threads is 0.25.

Solution

$$F = \frac{2(2.0)}{0.5}\,\frac{\pi(0.5) - 0.25(0.1)}{0.1 + 0.25\pi(0.5)} = 25.1 \text{ lb}$$

Because of the relatively high friction in a typical screw thread, ball bearing screws are often used to actuate the linear joints of a robot manipulator. In a ball bearing screw, the nut rides on ball bearings as the screw rotates, rather than directly on the

from screw torque *T* to force *F* resulting at the nut is given by

$$F = \frac{2\pi TE}{P} \tag{3.47}$$

where *E*

Example 3.12 Let us compare the force resulting from a ball bearing screw with the force in the conventional screw mechanism of Example 3.12. The same values apply: Torque = 2.0 in-lb, pitch = 0.1 in., and we

Solution

$$F = \frac{2\pi(2.0)(0.9)}{0.1} = 113.1 \text{ lb}$$

because of the lower friction.

3.8.3 Other Transmission Systems

Other power transmission devices include pulley systems, chain drives, and harmonic drives. Pulley systems are usually used to transmit power from actuators located in

from synthetic materials such as nylon. The rotational joints may be connected to a pulley which is driven by a rope attached to a rotary actuator (e.g., electric motor). Similarly, ropes may be used to activate linear joints. In either application, the rope

the application then this may result in stretching or even failure. If the rope stretches this results in degrading the accuracy of the robot. To maintain desired performance ropes must be maintained according to the manutacturer's instructions.

Chain drives operate with a constant ratio. Due to the positive interaction between the chain and sprockets there is no slipping. The pitch of a chain is the distance between adjacent roller centers. The driving sprocket and the driven sprocket each have a number of teeth designed to match the size and pitch of the chain. The transmission of rotational speed and power between the sprockets follows relationships similar to those developed for gears which we discussed earlier. Lubrication is an important factor in chain drive maintenance. A properly lubricated chain can last 100 times longer than an identical, improperly lubricated chain.

Harmonic drives are proprietary products of USM, Inc. They can be used as speed reducers or increasers. The input and output shafts lie along the same axis, so that a harmonic drive could be mounted to the face of a motor with the output shaft coming out the same end. Harmonic drives can provide any reduction ratio from

100 : 1. Harmonic

drives require little maintenance and can operate with no noticeable wear over their

3.8.4 Some General Comments

With any power transmission one is liable to introduce two unwanted effects to the performance of the control system. These are compliance and backlash. We used the

of the transmission. For example, in a gear train, this might be due to the bending

The compliance of a robot is largely a result of the transmission system compliance.

Backlash represents the hysteresis in the transmission. In gears, this is normally due to spaces between the gear teeth mesh. The result of backlash is that the output of the system does not correspond directly to the input of the system. This clearly

moved to a desired location with a precision of only a few thousandths of an inch.

links, and other imperfections inherent in the mechanical system are the principal contributors to the mechanical inaccuracies that degrade a robot's ability to position its end effector.

3.9 MODELING AND CONTROL OF A SINGLE JOINT ROBOT

Most industrial manipulators are independent joint controlled and each joint has an

model of actuation of a rotary joint of a manipulator. A DC motor connected to a gear box is assumed to actuate the joint. A DC motor consists of a stator that does not rotate and a rotor that provided the rotation motion. The physical principle that causes a motor to rotate is the force experienced by a charge moving in a coil through is those of the electrons moving through the coil. The torque produced in a motor is proportional to the current passing through the coils. The relation may be written as

$$\tau_m = k_m i \tag{3.48}$$

where τ_m is the torque developed by the motor, k_m is the motor torque constant and i is the current. A moving conductor in a magnetic field develops a voltage across it and this is called the back emf. The back emf is proportional to the motor rotational velocity and the relation between the three are given by

$$v = k_b \dot{\theta} \tag{3.49}$$

where 'v' is the voltage generated due to back emf, 'K_b' is the back emf constant and $\dot{\theta}$ is the rotational velocity of the motor. The electrical circuit of the motor armature can be represented by a voltage source V, inductance l, resistance of the winding as r and back emf as v. The arrangement of these elements are as shown in Fig. 3.24

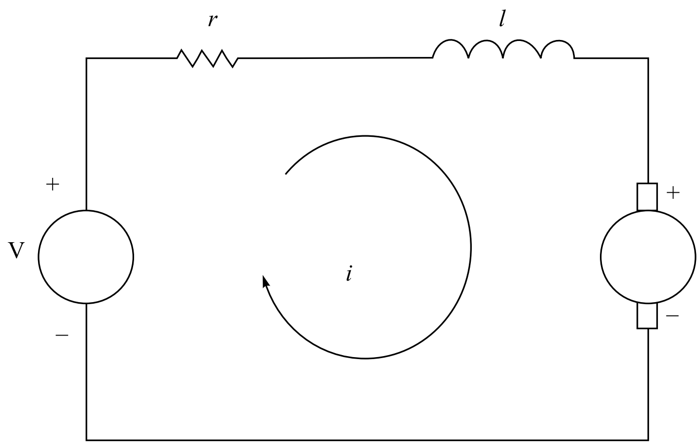


Fig. 3.24 *Circuit of a DC motor armature.*

The circuit is described by the first order equation given by

$$l\dot{i} + ri = V - k_b\dot{\theta}$$

This equation shows that one way of controlling the torque developed by a motor is to control the current by changing the voltage applied. Hence, generally, drive circuits measure the current flowing through the armature and continuously adjust the voltage source V so that the current passing through the armature is a constant. After determining the electrical model of the armature we will now develop the mechanical model of torque required by a motor to drive a load. We can then equate the electrical model with the mechanical model to study the control of the complete system.

Figure 3.25 shows the mechanical model of a DC motor connected to a link through a gear reduction. The torque generated by the motor was earlier shown to be proportional to the current in the armature coil. A gear reduction is used to increase the torque and reduce the speed at the output link. Hence, the output torque at the load side and link speed are given by

$$\tau_l = \eta \tau_m \tag{3.51}$$

$$\dot{\theta}_l = (1/\eta)\,\dot{\theta}_m \tag{3.52}$$

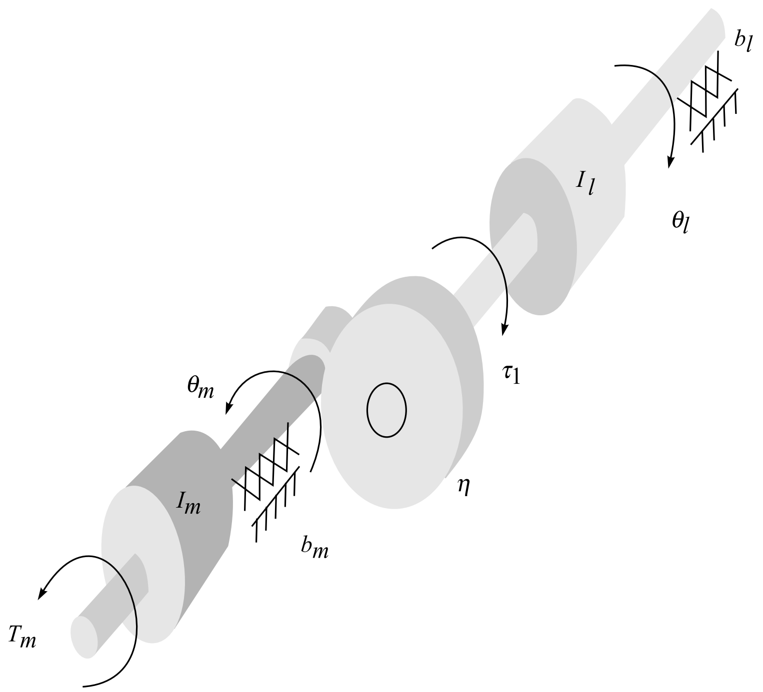


Fig. 3.25 *Mechanical model of a DC motor driving a robot link via. a gear reduction.*

where θ_l, θ_m are the rotational angles at the motor and load sides, respectively. Further writing a torque balance for the system in terms of the torque at the motor end, we have:

$$\tau_m = I_m\ddot{\theta}_m + B_m\dot{\theta}_m + (1/\eta)(I_l\,\ddot{\theta}_l + B_1\dot{\theta}_l) \tag{3.53}$$

where I_m and I_l are the inertias at the motor and load (link), while B_m and B_l are the viscous damping coefficients at the motor and load ends. Using relation (3.53), we can write

$$\tau_m = (I_m + I_l/\,\eta^2)\,\ddot{\theta}_m + (B_m + B_l/\eta^2)\,\dot{\theta}_m \tag{3.54}$$

This equation clearly shows that the load inertia and viscous damping coefficient reflected to the motor side is reduced by η^2. In terms of the load variable we can write this as

$$\tau_l = (I_l + I_m\eta^2)\,\ddot{\theta}_l + (B_l + B_m\eta^2)\,\dot{\theta}_l \tag{3.55}$$

The term $(I_l + I_m\eta^2)$ is also known as the effective inertia at the output side. In a highly geared motor, the inertial of the motor rotor can contribute significantly to the effective inertia, as seen from the equation above. This allows us to make the assumption that the effective inertia is a constant. This means that the variation of the load inertia due to change in configuration or due to change in grasping load forms a small fraction as compared to the variation in the case of direct drive robots for which the gear reduction is 1.

Example 3.13 A single link robot has link parameter: $I_m = 0.02$ and $I_l = 1.5$ and gear ratio $\eta = 100$. Where I_m and I_l are the motor and link inertia in kg.m^2 respectively. Find the effective inertia's at the output and input side.

Solution

The effective inertia at the output side

$$= (I_l + I_m\,\eta^2) = (1.5 + 0.02\,(100)^2)$$
$$= 201.5 \text{ kg.m}^2$$

The effective inertia at the input side

$$= (I_l/\eta^2 + I_m) = \left(\frac{1.5}{100^2} + 0.02\right)$$
$$= 0.02 \text{ kg.m}^2$$

Example 3.14 A single link robot arm has link parameters of Moment of Inertia 0.01 and moment of intertia 2.0 kg.m^2. The robot arm lifts a weight and the link inertia changes to 5.50 kg.m^2. Find the percentage change in effective inertia at the output side due to the object lifted by the robot. (gear reduction $\eta = 100$)

Solution

Effective inertia of robot at output side without weight

$$= (I_l + I_m\,\eta^2)$$
$$= (2.0 + 0.01(100)^2) = 102 \text{ kg.m}^2$$

Effective inertia of robot after lifting the weight

$$= (I_l + I_m\,\eta^2) = (5.50 + 0.01(100)^2)$$
$$= 105.5 \text{ kg.m}^2$$

The difference in the two inertias

$$= (105.5 - 102.0) = 3.5 \text{ kg.m}^2$$

Hence change in effective inertia as percentage

$$= \frac{3.5}{102} \times 100 = 3.43\%$$

This example illustrates that even when a robot lifts a very heavy object, its effective inertia changes by a very small amount. This is one of the reason why robot controllers function properly even when there is change in effective inertia due to grasping and releasing of an object by the gripper.

Problems

3.1 Using block diagram reduction techniques, simplify the block diagram of Fig. 3.19 in order to determine the transfer function for the dc motor.

3.2 A certain rotational joint design (including feedback controller) to be used on a new robot model has been studied to determine its response characteristics. It is known that the joint behaves very much like a second-order system. In one part of the study, measurements were taken on the position of the joint output link in response to a defined position input command. The table below presents the response data.

Table P3.2 *Joint response data.*

Time, ms	*Input position command*	*Output position*
0	45°	0°
50	45°	24.9°
100	45°	39.2°
150	45°	45.4°
200	45°	46.9°
250	45°	46.6°
300	45°	45.9°
400	45°	45.1°
500	45°	44.9°
1000	45°	45.0°

(***a***) Plot the output data on a piece of graph paper.

(***b***) Make your best estimate as to which of the following types of system the response data come from: (1) undamped. (2) underdamped, (3) critically damped, or (4) overcemped.

(***c***) What additional data would be needed to determine the second order differential equation of the form of Eq. (3-2)?

3.3 A mechanical joint design for a certain robot manipulator has the following difterential equation which describes the position of the output link as a function of time:

$$\frac{3.26d^2y}{dt^2} + \frac{17.5dy}{dt} + 44.2y = X$$

where X equals the forcing function and y represents the position response of the joint.

(***a***) Write the characteristic equation for the differential equation above.

(***b***) Determine the roots of the characteristic equation.

(***c***) Based on the roots of the characteristic equation, will the response be (1) undamped, (2) underdamped, (3) critically damped, or (4) cverdamped?

3.4 Write the transfer function of the differential equation from Prob. 3.3 above.

3.5 For the following set of equations, rewrite each equation using the *s*-operator notation. Then, construct the block diagram that relates the equations, using *x* as the input and *y* as the output.

$$\frac{dz}{dt} + 3.2z = w$$

$$\frac{dy}{dt} + 5.0y = 2.6z$$

$$w = x - 1.5y$$

3.6 Using the block diagram reduction techniques of Fig. 3.5 in the text, reduce the block diagram developed in Prob. 3.5 to a single block thus yielding the transfer function for the system.

3.7 For the differential equation of Prob. 3.3, calculate the natural frequency and the damping ratio of the system.

3.8 For a step input X = 5.0, solve the differential equation of Prob. 3.3. Plot your solution on a piece of graph paper, and determine the following transient

(***a***) Delay time.
(***b***)

(***c***) Peak time.
(***d***) Maximum overshoot.
(***e***) Settling time,

3.9 Using the response data tor the rotational joint design of Prob. 3.2, determine

text of Sec. 3.3.
(***a***) Delay time.
(***b***)

(***c***) Peak time.
(***d***) Maximum overshoot.
(***e***) Settling time.

3.10

joint design of Prob. 3.3, determine the steady-state response of the system to a step input of X = 5.0. (*Hint*: Recall that a step input of value X = 5.0 would have a Laplace transform = 5/*s*.)

3.11 A certain potentiometer is to be used as the feedback device to indicate position of the output link of a rotational robot joint. The excitation voltage of the potentiometer equals 5.0 V, and the total wiper travel of the potentiometer is 300°. the wiper arm is directly connected to the rotational joint so that a given rotation of the joint corresponds to an equal rotation of the wiper arm.

(***a***) Determine the voltage constant of the potentiometer, K_p.
(***b***) The robot joint is actuated to a certain angle, causing the wiper position to be 38°. Determine the resulting output voltage of the potentiometer.

(*c*) In another actuation of the joint, the resulting output voltage of the potentiometer is 3.75 V. Determine the corresponding angular position of the wiper and the output link.

3.12 A resolver is used to indicate angular position of a rotational wrist joint. The excitation voltage to the resolver is 24 V. The resolver is connected to the wrist joint so that a given rotation of the output link corresponds to an equal rotation of the resolver. At a certain moment in time, the movement of the wrist joint results in voltages across the two pairs of stator terminals to be V_{s1} = 10.0 V and V_{s2} = –21.82 V. Determine the angle of the rotational joint.

3.13 What is the resolution of an absolute optical encoder that has six tracks? Nine tracks? Twelve tracks?

3.14 For an absolute optical encoder with 10 tracks, determine the value of K_e, the encoder constant. If the shaft angle of the encoder were 0.73 rad, determine its output value.

3.15 A dc tachometer is to be used as the velocity feedback device on a certain twisting joint. The joint actuator is capable of driving the joint at a maximum velocity of 0.75 rad/s, and the tachometer constant is 8.0 V/rad/s. What is the maximum output voltage that can be generated by the device, if the tachometer is geared with the joint so that it rotates with twice the angular velocity of the joint? If the joint rotates at a speed of 25°/s, determine the output voltage of the dc tachometer.

3.16 A hydraulic single-ended piston cylinder is to be used to actuate the linear arm joint for a polar configuration robot. The size of the cylinder is 10.0 in.^{2} on the forward stroke (piston extension), and 9.0 in.^{2} on the reverse stroke (piston retraction). The hydraulic power source can generate up to 1000 lb/in.^{2} of pressure for delivery to the cylinder at a rate of 100 in.^{3}/min.

(***a***) Determine the force that can be applied by the piston on the forward stroke and the reverse stroke.

(***b***) Determine the maximum velocity at which the piston can operate in the forward and reverse strokes.

3.17 A hydraulic rotary vane actuator is to be used for a twist joint with the same hydraulic power source used for Prob. 3.16. The outer and inner radii (R and r) of the vane are 2.5 in. and 0.75 in., respectively. The thickness of each vane (h) is 0.20 in. Determine the angular velocity and the torque that can be generated by the actuator.

3.18 A dc servomotor is used to actuate a robot joint. It has a torque constant of 10 in.-lb/A, and a voltage constant of 12 V/Kr/min (1 Kr/min = l000 r/min), The armature resistance = 2.5 Ω. At a particular moment during the robot cycle, the joint is not moving and a voltage of 25 V is applied to the motor.

(***a***) Determine the torque of the motor immediately after the voltage is applied.

(***b***) As the motor accelerates, the effect of the beck-emt is to reduce the torque. Determine the back-emf and the corresponding torque of the motor at 250 and 500 r/min.

3.19 A certain dc servomotor used to actuate a robot joint has a torque constant of 25 in.-lb/A, and a voltage constant of 15 V/Kr/min. The armature resistance =

3.0 Ω. At a particular moment during the robot cycle, the joint is not moving and a voltage of 30 V is applied to the motor.

(***a***) Determine the torque of the motor immediately after the voltage is applied.

(***b***) Determine the back-emf and the corresponding torque of the motor at 500 and l000 r/min.

(***c***) If there were no resisting torques and no inductance of the armature windings operating to reduce the speed of the motor, determine the maximum theoretical speed of the motor when the input voltage is 30 V.

(***d***) If the resisting torques due to friction and the payload being carried by the robot total 72 in.-lb, determine the maximum theoretical speed of the motor when the input voltage is 30 V. Assume no eftect of inductance from the armature windings.

3.20 A stepping motor is to be used to actuate one joint of a robot arm in a light duty pick-end-place application. The step angle of the motor is 10°. For each pulse received from the pulse train source, and motor rotates through a distance of one step angle.

(***a***) What is the resolution of the stepping motor?

(***b***) Relate this value to the definitions of control resolution, spatial resolution, and accuracy, as these terms were defined in Chap. 2.

3.21 For the stepping motor described in Prob. 3.20, a pulse train is to be generated by the robot controller.

(***a***) How many pulses are required to rotate the motor through a total of three complete revolutions?

(***b***) If it is desired to rotate the motor at a speed of 25 r/min, what pulse rate must be generated by the robot controller?

3.22 A power screw mechanism is used to convert rotational motion into linear motion for a robot joint The screw has 12 threads/in. and the thread angle is 10°. The diameter of the screw is 0.375 in., and the coefficient of friction between threads on the screw and the moving nut is 0.30. If the torque applied to the screw is 10 in.-lb. determine the force that will be transmitted to the nut moving along the screw.

3.23 Solve Problem 3.22 except that a ball bearing screw will be used instead of a conventional screw thread. The applied torque is 10 in-lb, there are 12 threads/in., and the efficiency factor is assumed 10 be 90%.

3.24 A stepping motor is to be used to drive each of the three linear axes of a cartesian coordinate robot. The motor output shaft will be connected to a screw thread with a screw pitch of 0.125 in. It is desired that the control resolution of each of the axes be 0.025 in.

(***a***) To achieve this control resolution, how many step angles are required on the stepping motor?

(***b***) What is be corresponding step angle?

(***c***) Determine the pulse rate that will be required to drive a given joint at a velocity of 3.0 in./sec.

3.25 List the different internal and external sensors that can be used in designing a robot arm?

3.26 A single link robot is driven by a DC motor via a gear reducer. The robot parameters are given as $I_m = 0.01$, $I_l = 1.0$, and $\eta = 80$.

(***a***) Find the effective inertias at the output and input sides.

(***b***) If the robot now picks up a mass and its link inertia increases to 6.50, find the percentage variation in the effective inertias at the output side.

References

1. G. S. Boyes, *Synchro and Resolver Conversion*, Memory Devices Ltd., Surrey, United Kingdom, 1980.
2. Electro-Craft Corp., *DC Motors, Speed Controls, Servo Systems*, Hopkins, MN, 1975.
3. M. P. Groover, *Automation, Production Systems and Computer-Aided Manufacturing*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
4. E. Kafrissen and M. Stephans, *Industrial Robots and Robotics*, Reston, Reston, VA, 1984.
5. K. Ogata, *Modern Control Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1970
6. J. E. Shigley, *Mechanical Engineering Design*, 3rd ed., McGraw-Hill, New York, 1977.
7. Stock Drive Products, *Design and Application of Small Standardized Components*, New Hyde Park, New York, 1983.

Robot Motion Analysis and Control

Introduction

required to perform welding. It is also necessary to control the path which the

the mathematical techniques used to analyze manipulator positions and motions.

4.1 INTRODUCTION TO MANIPULATOR KINEMATICS

In order to develop a scheme for controlling the motion of a manipulator it is necessary to develop techniques for representing the position of the arm at points

RR notation and the manipulator *LL* rotation.

Fig. 4.1 *Two different two-jointed manipulators: (a) two-rotational joints (RR), (b) two-linear joints (LL).*

L_n to indicate the length of the link in some of our

a_n to denote the length of a manipulator link.

4.1.1 Position Representation

RR *LL*

RR

θ and θ

P_j θ θ

Fig. 4.2 *A two-dimensional 2-degree of freedom manipulator (type RR).*

$P_w \quad x \;\; y$

$P_w \quad x \;\; y \;\; z$

inverse kinematics.

4.1.2 Forward Transformation of a 2-Degree of Freedom Arm

$$\mathbf{r} = [L \;\; \cos\theta \;\; L \;\; \sin\theta$$

$$\mathbf{r} = [L \qquad \theta + \theta \quad L \qquad \theta + \theta$$

x and y of the end of the

P_w

$$x = L \;\; \cos\theta \; + L \qquad \theta + \theta$$

$$y = L \;\; \sin\theta \; + L \qquad \theta + \theta$$

4.1.3 Reverse Transformation of the 2-Degree of Freedom Arm

$x \;\; y$

θ

θ is positive as shown in

$$A + B \qquad A \cos B - \sin A \sin R$$

$$A + B \qquad A \cos B + \sin B \cos A$$

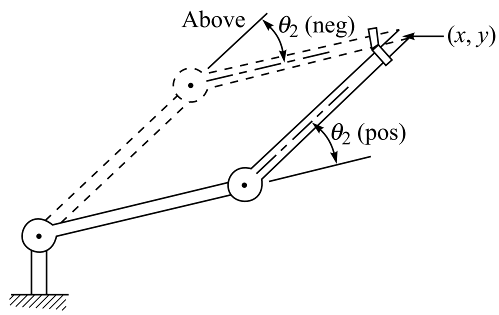


Fig. 4.3 *The arm at point P (x, y), indicating two possible configurations to achieve the position.*

we can rewrite Eqs. (4.3) and (4.4) as

$$x = L_1 \cos \theta_1 + L_2 \cos \theta_1 \cos \theta_2 - L_2 \sin \theta_1 \sin \theta_2$$

$$y = L_1 \sin \theta_1 + L_2 \sin \theta_1 \cos \theta_2 + L_2 \cos \theta_1 \sin \theta_2$$

Squaring both sides and adding the two equations yields

$$\cos \theta_2 = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2} \tag{4.5}$$

Defining α and β as in Fig. 4.4 we get

$$\tan \alpha = \frac{L_2 \sin \theta_2}{L_2 \cos \theta_2 + L_1} \tag{4.6}$$

$$\tan \beta = \frac{y}{x}$$

Using the trigonometric identity

$$\tan (A - B) = \frac{\tan A - \tan B}{1 + \tan A \tan B}$$

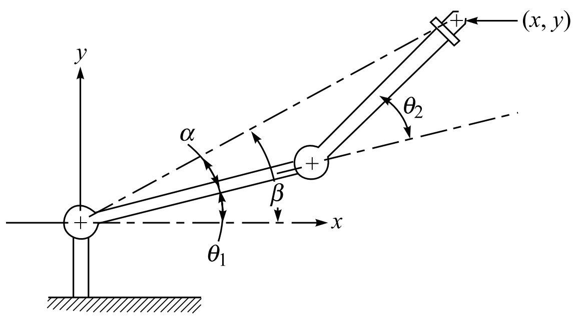


Fig. 4.4 *Solving for the joint angles.*

we get

$$\tan \theta = \frac{[y(L_1 + L_2 \cos \theta_2) - xL_2 \sin \theta_2]}{[x(L_1 + L_2 \cos \theta_2) + yL_2 \sin \theta_2]}$$

Knowing the link lengths L and L

x y

4.1.4 Adding Orientation: A 3-Degree of Freedom Arm in Two Dimensions

RR:*R*

$$\left.\begin{aligned} x &= L \cos\theta + L \quad \theta + \theta \quad L \quad \theta + \theta + \theta \\ y &= L \sin\theta + L \quad \theta + \theta \quad L \quad \theta + \theta + \theta \\ \psi & \quad \theta + \theta + \theta \end{aligned}\right\}$$

x y ψ ψ is the

θ θ θ

$$x = x - L \cos \psi$$

$$y = y - L \sin \psi$$

θ and

Fig. 4.5 *The two-dimensional 3-degree of freedom manipulator with orientation (type RR:R).*

θ

4.1.5 A 4-Degree of Freedom Manipulator in Three Dimensions

T

z R z

a type R

TRL: R manipulator.

θ

ϕ

L L

$x - y$ ψ

P

$$x = \cos\theta \; L \cos\phi + L \quad \cos\psi$$

$$y = \sin\theta \; L \cos\phi + L \quad \cos\psi$$

$$z = L \quad + L \sin\phi + L \quad \sin\psi$$

P x y z ψ

Fig. 4.6 *A three-dimensional 4 degree-of-freedom manipulator (type TRL:R).*

x y z

$$x = x - r\cos\theta \; L \quad \cos\psi$$

$$y \quad = y - \sin\theta \; L \quad \cos\psi$$

$$z \quad = z - L \quad \sin\psi$$

L ϕ θ

$$L = [x \quad + y \qquad z \; - L \quad]$$

$$\sin \phi = \frac{z_4 - L_1}{L}$$

$$\cos \theta = \frac{y}{L}$$

at a solution.

4.2 HOMOGENEOUS TRANSFORMATIONS AND ROBOT KINEMATICS

$$\mathbf{v} = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

where $a = x/w$ $b = y/w$ $c = z/w$ w

$\mathbf{v}$ $\mathbf{i}$ $\mathbf{j}$ $\mathbf{k}$.

$$\begin{bmatrix} 25 \\ 10 \\ 20 \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} 50 \\ 20 \\ 40 \\ 2 \end{bmatrix} \text{ or } \begin{bmatrix} 12.5 \\ 5.0 \\ 10.0 \\ 0.5 \end{bmatrix}$$

w

to notice that homogenous transformation is not unique as the coordinates of a point

$\times$ **H** **v** is

transformed into the vector **u**

$$\mathbf{u} = \mathbf{Hv}$$

a

in the *x* *b* in the *y* *c* in the *z*

$$\mathbf{H} = \mathbf{Trans}\ a\ b\ c \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example 4.1 **v** **i** **j** **k** *x* *y*

z

$$\mathbf{H} = \mathbf{Trans}\ a\ b\ c \begin{bmatrix} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Hv} = \begin{bmatrix} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 25 \\ 10 \\ 20 \\ 1 \end{bmatrix} = \begin{bmatrix} 33 \\ 15 \\ 20 \\ 1 \end{bmatrix}$$

θ

x

$$\mathbf{Rot}\ x\ \theta \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

y

$$\mathbf{Rot}\ y\ \theta \quad \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

z

$$\mathbf{Rot}\ z\ \ \theta \quad \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

K

where **K** $x\ y \quad z$

$$\mathbf{Rot}\ \mathbf{K}\ \ \theta$$

Example 4.2 Rotate the vector **v** **i** **j** **k**
x

$$\mathbf{H} = \mathbf{Rot}\ x \qquad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90 & -\sin 90 & 0 \\ 0 & \sin 90 & \cos 90 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Hv} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 8 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ -8 \\ 3 \\ 1 \end{bmatrix}$$

important to note that performing two or more transformations in a row will only yield the same result if the transformations are carried out in the same sequence. In

AB does not equal **BA**

$$\mathbf{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T **T**

$$\mathbf{T} \quad = \begin{bmatrix} n_x & n_y & n_z & -p \cdot n \\ o_x & o_y & o_z & -p \cdot o \\ a_x & a_y & a_z & -p \cdot a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where **p** · **n** **p** · **o** and **p** · **a** represent the dot products of the column vectors **n** **o** **a** and **p** **p** · **n** is the scalar $P_x n_x + p_y n_y + p_z n_z$ apply to **p** · **o** and **p** · **a** transformation **T** itself.

Example 4.3 **v** *i* *j* *k* *z* *x*

$$\mathbf{H} = R\ z \quad \begin{bmatrix} \cos 60 & -\sin 60 & 0 & 0 \\ \sin 60 & \cos 60 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H} = R\ x \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 30 & -\sin 30 & 0 \\ 0 & \sin 30 & \cos 30 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}\ \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & -0.8 & 0 \\ 0 & 0.8 & 0.5 & 0 \\ 0 & & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & -0.8 & 0 & 0 \\ 0.8 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 0.5 & -0.8 & 0 & 0 \\ 0.4 & 0.25 & -0.8 & 0 \\ 0.64 & 0.4 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 7 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.1 \\ -3.9 \\ 6.22 \end{bmatrix}$$

Example 4.4 **v** *i* *j* *k* *z*

x

$$\mathbf{H} = R\ z \quad \begin{bmatrix} \cos 60^\circ & -\sin 60 & 0 & 0 \\ \sin 60 & \cos 60 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_2 = R(x, 30°) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 30 & -\sin 30 & 0 \\ 0 & \sin 30 & \cos 30 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_1\mathbf{H}_2 = \begin{bmatrix} 0.5 & -0.8 & 0 & 0 \\ 0.8 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & -0.8 & 0 \\ 0 & 0.8 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & -0.4 & +0.6 & 0 \\ 0.8 & 0.25 & 0 & 0 \\ 0 & 0.8 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 0.5 & -0.4 & 0.6 & 0 \\ 0.8 & 0.25 & 0 & 0 \\ 0 & 0.8 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 7 \\ 1 \end{bmatrix} = \begin{bmatrix} 4.9 \\ 2.9 \\ 5.1 \end{bmatrix}$$

Example 4.5 A vector $\mathbf{v} = 2i + 5j + 3k$ is rotated by 60° about the z-axes and translated by 3, 4 and 5 units in the x, y and z directions respectively. Find the vector with reference to the reference frame.

The homogeneous transformation matrix is given by

$$\mathbf{H} = \begin{bmatrix} \cos 60 & -\sin 60 & 0 & 3 \\ \sin 60 & \cos 60 & 0 & 4 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The vector is given as

$$= \begin{bmatrix} 0.5 & -0.8 & 0 & 3 \\ 0.8 & 0.5 & 0 & 4 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 8.1 \\ 8.0 \\ -1 \end{bmatrix}$$

4.2.1 Kinematic Equations Using Homogeneous Transformations

The transformation **T** in the previous subsection is of the form

$$\mathbf{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We can consider **T**

p

n **o** **a**

a

o

n **o** and **a**

n **o** **a** vectors constitute the x y z

Fig. 4.7 ***o***, ***a***, ***n***, *and* ***p*** *for a robot manipulator.*

the product of n n

a_n

a_n is the

angle t_n d_n a_n and a_{n-1} of the two

θ_n

a_n and a_n

n n

the four parameters a_n t_n θ_n and d_n.

origin of the coordinate frame for link n z_n

n a_n n n

using:

θ_n z_n

d_n along z_n

x_n a_n

x_n t_n

develop the transformation which relates the coordinate frame of link n

Fig. 4.8

notation are: l. Axis z_{n-1} $_n$, z_n for $_{n+1}$, and so forth. 2. Axis x_{n-1} is selected to be an extension for the common perpendicular line of length a_{n-1} $_{n-2}$ and z_{n-1}. 3. The axis y_{n-1}

4. Axis x_n is an extension of the common perpendicular line of length a_n.

with link n **A**

position and orientation of the coordinate frame of the second link with respect to the

$$\mathbf{A}_n = \mathbf{Rot}\ z\ \ \theta\ \ t\ \mathbf{Trans}\ \ \ d\ \ \mathbf{Trans}\ a\ \ \ \mathbf{Rot}\ x\ \ t$$

A

A

$$\mathbf{A}\ = \mathbf{Rot}\ z\ \ \theta\ \ \mathbf{Trans}\ \ \ d\ \ \mathbf{Trans}\ a\ \ \ \mathbf{Rot}\ x\ \ t$$

$$\mathbf{A}n = \begin{bmatrix} \cos\theta & -\sin\theta\cos t & \sin\theta\sin t & a\cos\theta \\ \sin\theta & \cos\theta\cos t & -\cos\theta\sin t & a\sin\theta \\ 0 & \sin t & \cos t & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

a

$\mathbf{T}_n$ where

$$\mathbf{T}_n = \mathbf{A}\ \mathbf{A}\ \ldots \mathbf{A}_n$$

In order to calculate the **T** n **A**

a θ d t

A matrices are

$$\mathbf{A} = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 4.9 *A three-dimensional 3-degree of freedom manipulator (type TRL).*

Table 4.1 *Link parameters*

Link	*variable*	*t*	*a*	*d*
1	θ_n	—90°	0	0
2	θ_n	90	0	0
3	θ_n	0	0	d_3

$$\mathbf{A} = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & 0 \\ \sin\theta_2 & 0 & -\cos\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

n with respect to any other

preceding link *m*.

$$^{m}\mathbf{T}_n = \mathbf{A}_m \quad \dots \mathbf{A}_n \quad \mathbf{A}_n$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T} = \mathbf{A} \quad \mathbf{T}$$

we get

$$\mathbf{T} = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & d_3\sin\theta_2 \\ \sin\theta_2 & 0 & -\cos\theta_2 & -d_3\cos\theta_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{T} = \begin{bmatrix} \cos\theta_1\cos\theta_2 & -\sin\theta_1 & \cos\theta_1\sin\theta_2 & d_3\cos\theta_1\sin\theta_2 \\ \sin\theta_1\cos\theta_2 & \cos\theta_1 & \sin\theta_1\sin\theta_2 & d_3\sin\theta_1\sin\theta_2 \\ -\sin\theta_2 & 0 & \cos\theta_2 & d_3\cos\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

C

A $\mathbf{G}_n$

$$\mathbf{T} = \mathbf{CAG}_n$$

Fig. 4.10 *A simple robot workcell.*

4.2.2 Solving the Kinematic Equations

introduce an analysis technique known as the transform graph.

0
T and the other is along the path **A A A** . We always travel along the

T to get to the end of the arm.

$$\mathbf{A} \quad \mathbf{T} = \mathbf{T}$$

Fig. 4.11

or

$$\mathbf{A} \quad \mathbf{A} \quad \mathbf{T} = \mathbf{T}$$

and so forth.

$$\mathbf{A} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} q1 & q2 & q3 & q4 \\ -n_z & -o_z & -a_z & -p_z \\ q5 & q6 & q7 & q8 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & d_3\sin\theta_2 \\ \sin\theta_2 & 0 & -\cos\theta_2 & -d_3\cos\theta_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$q \quad n_x \cos\theta + n_y \sin\theta$$
$$q \quad o_x \cos\theta + o_y \sin\theta$$
$$q \quad a_x \cos\theta + a_y \sin\theta$$
$$q \quad p_x \cos\theta + p_y \sin\theta$$
$$q \quad n_x \sin\theta + n_y \cos\theta$$
$$q \quad o_x \sin\theta + o_y \cos\theta$$
$$q \quad a_x \sin\theta + a_y \cos\theta$$
$$q \quad p_x \sin\theta + p_y \cos\theta$$

we can use this to solve for θ θ d

$$q \quad o_x \cos\theta + o_y \sin\theta$$

so that

$$\tan\theta = -\frac{o_x}{o_y}$$

we can also see that

$$\sin\theta = -n_z$$

and

$$-\cos\theta = -a_z$$

so that

$$\tan\theta = -\frac{n_z}{a_z}$$

$$d = \frac{p_z}{\cos\theta_2}$$

4.2.3 A Discussion on Orientation

o a n

z y x

constitutes a coordinate reference frame.

a o n

using the following equations:

$$\text{Roll} = \arctan \frac{n_y}{n_x}$$

Fig. 4.12 *Roll, pitch, and yaw for a manipulator wrist mechanism (Refer to Fig. 2.11 in Chap. 2).*

$$\text{Pitch} = \arctan \frac{-n_z}{n_x \cos(\text{roll}) + n_y \sin(\text{roll})}$$

$$\text{Yaw} = \arctan \frac{a_x \sin(\text{roll}) - a_y \cos(\text{roll})}{o_y \cos(\text{roll}) - o_x \sin(\text{roll})}$$

4.3 MANIPULATOR PATH CONTROL

to another in the workspace.

4.3.1 Motion Types

straight line motion.

to travel from point *A* to point *B*

4.3.2 Joint Space Schemes

avoidance etc.

θ_i θ_f

functions.

Fig. 4.13

θ t a $+ a$ $t + a$ t $+ a$ t

$\dot{\theta}$ t_f a a t a t

$\ddot{\theta}$ t a a t

t_i t_f.

$\dot{\theta}$

$\dot{\theta}$ t_f

$$\theta_i = a$$
$$\theta_f = a \;+ a\; t_f + a\; t_f \;+ a\; t_f.$$
$$a$$
$$a \quad a\; t_f \quad a\; t_f$$

$$a \;= \theta_i$$
$$a$$
$$a \;= \frac{3}{t_f^2}\;\theta_f - \theta_i$$
$$\theta \;= -\frac{2}{t_f^3}\;\theta_f - \theta_i$$

Example 4.6 θ_f θ_i

θ_f

Example 4.7 is stationary at θ θ

Initial angle θ_i
θ_f
t

$$\theta\; t \quad a \;+ a\; t + a\; t \;+ a\; t$$

$$a$$
$$a$$
$$a \;= \frac{3}{t^2}\;\theta_f - \theta_i \quad \frac{3}{25}$$
$$a \;= -\frac{2}{t^3}\;\theta_f - \theta_i \quad \frac{2}{125}$$

$$\theta\; t \quad t \quad t$$

LLL

motions are unnatural and the controller must compute the sequence of incremental

A to point *B*

straight line path are computed relative to the arm velocity.

programming.

4.4 ROBOT DYNAMICS

4.4.1 Static Analysis

F at

we get

$$\mathbf{F} \quad - \mathbf{F}$$

and

$$\mathbf{F} \quad - \mathbf{F}$$

$$\mathbf{F} \quad = \mathbf{F} \quad = \mathbf{F}$$

Fig. 4.14 *Two-link arm forces and torques.*

r as

$$\mathbf{T} \;\; = \mathbf{T} \;\; + \mathbf{r} \;\; \times \mathbf{F}$$

and

$$\mathbf{T} \;\; = \mathbf{r} \;\; \times \mathbf{F}$$

$\mathbf{T} \quad \mathbf{r} \; + \mathbf{r} \quad \times \mathbf{F}$

If $\mathbf{F} \quad \mathbf{F}_x \; \mathbf{F}_y$

$\mathbf{T} \;\; = [L \;\; \cos\theta \; + L \qquad \theta \; + \theta \qquad \sin\theta \; + L \qquad \theta \; + \theta \quad \times \; F_x \; F_y$

$a \;\; b \; \times \; c \;\; d \qquad ad - bc$ we get

$\mathbf{T} \;\; = [L \;\; \cos\theta \; + L \qquad \theta \; + \theta \quad \times F_x - [L \;\; \sin\theta \; + L \qquad \theta \qquad \theta \quad \times F_y$

and

$\mathbf{T} \;\; = L \qquad \theta \qquad \theta \quad \times F_x - L \qquad \theta \; + \theta \quad \times F_y$

$\mathbf{F} \quad F_x \; F_y$

torques we must solve the equations for F_x and F_y

F_y using the value in

$F_x \;\; T \quad T \;\; \theta \qquad \theta$

using trigonometric identities as outlined earlier we get

$$F_x = \frac{T_1 L_2 \cos(\theta_1 + \theta_2) - T_2(L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2))}{L_1 L_2 \sin\theta_2}$$

F_y is determined as

$$F_y = \frac{T_1 L_2 \sin(\theta_1 + \theta_2) - T_2(L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2))}{L_1 L_2 \sin\theta_2}$$

of forces applied at the wrist or tool are useful for controlling the arm when it must

4.4.2 Compensating for Gravity

$\mathbf{F}_g$ and $\mathbf{F}_g$ acting at the

m and m

to gravity only we get

Fig. 4.15

$$\mathbf{F} = m\ \mathbf{g}$$

$$\mathbf{F} = \mathbf{F} + m\ \mathbf{g}$$

$\boldsymbol{r}$ and the forces give us the

$$\mathbf{T}_g = -m\ r \times \mathbf{g}$$

$$= \frac{g[m_2 L_2 \cos(\theta_1 + \theta_2)]}{2}$$

and $$\mathbf{T}_g = g\left[\left(\frac{m_1}{2} + m_2\right) L_1 \cos\theta_1 + \frac{m_2 L_2 \cos(\theta_1 + \theta_2)}{2}\right]$$

4.4.3 Robot Arm Dynamics

Fig. 4.16 *1* *1·*

Fig. 4.17 *Dynamic forces and torques for a TRR robot.*

Dynamics

acting at the centre of each link is calculated as:

$$F_i = m \dot{\ }_i$$
$$N_i = I\dot{w}_i + w_i \times Iw_i$$

where F_i is the force and N_i
i $_i$ and w_i

Z τ

$$\tau_i = N_i Z_i$$

In the Lagrangian formulation we derive the equation of motion using a scalar

$$L = KE - PE$$

where KE and PE
of a link is given as

$$KE_i = \frac{1}{2} m_i \ {}_i + \frac{1}{2} w_i^T I w_i$$

$$PE = -mg\,P_i$$

g
P_i is the position of the centre of the link.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau$$

τ
equation of motion takes the form.

$$D\ \theta\ \ddot{\theta} + H\ \theta\ \dot{\theta} \quad \theta \quad F = Z.$$

where $D\ \theta\ \ddot{\theta}$ = inertia forces

$H\ \theta\ \dot{\theta}$

$C\ \theta$

F

manipulator.

4.5 CONFIGURATION OF A ROBOT CONTROLLER

Fig. 4.18 *General robot controller element.*

operator input or program memory. Either an operator inputs commands to the commands are downloaded to the system from program memory under control

Microprocessors are typically utilized in several of the components of a modern

roblems

4.1

L

L θ θ

x and y P_w.

4.2 It is desired to determine the values to which the angles θ and θ

L L

point P_w x

and y

determine the angles. θ and θ

4.3

4.4 It is desired to determine the values to which the angles θ and θ

L L

point P_w x

and y

determine the angles θ and θ

4.5

4.6

the manipulator had the following settings:

Length of link L

L

L

θ

Elevation angle ϕ

Pitch angle ψ

Determine the coordinates of the resulting point P

4.7 x y z

4.8

x y z

y

P

L θ and ϕ.

4.9

x y z

ψ

P

L θ and ϕ.

4.10 Write a computer program that will calculate and print out the x y z

of the following parameters: the length of link L

link L L θ ϕ

ψ.

4.11 Write a computer program that will calculate and print out the values of the manipulator parameters L θ ϕ

x y z ψ

L L

4.12

L

calculation routine that will determine whether the input values of x y

z constitute a point that is outside the work volume of the manipulator. It the

appropriate error message.

4.13 i j k perform the following operations:

(a) X

(b) Y

(c) Z

(d) Y

(e) X Y.

(f) Y X.

4.14 i j k.

4.15

(a)

$$\begin{bmatrix} 0 & 10 & 5 & 3 \\ 1 & 5 & 3 & 2 \\ 0 & 1 & 5 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 5 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4.16

4.17 e f

4.18

4.19

Fig. P4.19

Table P4.19

Link	*Variable*	*t*	*a*	*d*
1	θ_1	— 90°	0	0
2	θ_2	0	0	d_2
3	θ_3	0	a_3	d_3
4	θ_4	— 90°	a_4	0
5	θ_5	— 90°	0	0
6	θ_6	90°	0	0

4.20

4.21

(a) **T** *(b)* **T** *(c)* **T**

(d) **T** *(e)* **T** *(f)* **T**

4.22 **O** **A** **B** **C**
O
A **B**
and **C** with respect to **O**.

Fig. P4.22

4.23 x y z
x y
x y z

x y z direction from the tip of the end effector.

(a) Make a sketch of the workcell.

(b) Identify all transforms numerically.

(c)

effector.

4.24

z

y

in the x direction.

(a)

$\times$

(b) What are the new coordinates of the vertices of the prism after the move?

(c) W

Fig. P4.24

4.25 *VVR*

(*a*)

(*b*) Determine the time required to move the arm to the desired position and

4.26

4.27 *LLL*

x y z *x y z*

(*a*)

(*b*) Determine the time required to move the arm to the new position and the

4.28

4.29

4.30

that are connected at the via point with continuous acceleration at the via

References

Robot Motion: Planning and Control

Conference Papers

Conference
Papers

Robotics for Engineers

Conference Proceedings

Conference Papers

Conference
Proceedings

Robot Manipulators: Mathematics, Programming, and Control

Industrial Robots: Computer Interfacing and Control

Robot end Effectors

Introduction

An end effector is a device that attaches to the wrist of the robot arm and enables

effector is part of that special-purpose tooling for a robot. Usually, end effectors must be custom engineered for the particular task which is to be performed. This can be accomplished either by designing and fabricating the device from scratch, or by purchasing a commercially available device and adapting it to the application. The company installing the robot can either do the engineering work

robot manufacturers have special engineering groups whose function is to design end effectors and to provide consultation services to their customers. Also, there

engineering work to install robot systems. Their services would typically include end effector design.

5.1 TYPES OF END EFFECTORS

There are a wide assortment of end effectors required to perform the variety of different work functions. The various types can be divided into two major categories:

1. Grippers
2. Tools

Grippers are end effectors used to grasp and hold objects. The objects are generally workparts that are to be moved by the robot. These part-handling applications include machine loading and unloading, picking parts from a conveyor, and arranging parts onto a pallet. In addition to workparts, other objects handled by robot grippers include cartons, bottles, raw materials, and tools. We tend to think of grippers as mechanical grasping devices, but there are alternative ways of holding objects involving the use of magnets, suction cups, or other means. In this chapter we will divide grippers according to whether they are mechanical grasping devices or some other physical

use a means of retention other than mechanical.

by the fact that only one grasping device is mounted on the robot's wrist. A double gripper has two gripping devices attached to the wrist and is used to handle two separate objects. The two gripping devices can be actuated independently. The double gripper is especially useful in machine loading and unloading applications. To illustrate, suppose that a particular job calls for a raw workpart to be loaded from a

up the raw part. This would consume valuable time in the production cycle because the machine would have to remain open during these handling motions. With a double gripper, the robot can pick the part from the incoming conveyor with one

available grasping device, and insert the raw part into the machine with the other grasping device. The amount of time that the machine is open is minimized.

The term multiple gripper is applied in the case where two or more grasping mechanisms are fastened to the wrist. Double grippers are a subset of multiple grippers. The occasions when more than two grippers would be required are somewhat rare. There is also a cost and reliability penalty which accompanies an increasing number of gripper devices on one robot arm.

Another way of classifying grippers depends on whether the part is grasped on its

is attached to the robot's wrist. One of the most common applications of industrial robots is spot welding, in which the welding electrodes constitute the end effector

effectors include spray painting and arc welding. Section 5.4 discusses the various end effectors in this category.

It was mentioned above that grippers are sometimes used to hold tools rather than workparts, The reason for using a gripper instead of attaching the tool directly to the robot's wrist is typically because the job requires several tools to be manipulated by

debarring operation in which several different sizes and geometries of deburring tool must be held in order to reach all surfaces of the workpart. The gripper serves as a quick change device to provide the capability for a rapid changeover from one tool

and other forms of holding devices to position the workpart or tooling during the work cycle.

5.2 MECHANICAL GRIPPERS

are either attached to the mechanism or are an integral part of the mechanism. If the

for use with the same gripper mechanism can be designed to accommodate different

The function of the gripper mechanism is to translate some form of power input

from the robot and can be pneumatic, electric, mechanical, or hydraulic. We will discuss the alternatives in Sec. 5.5. The mechanism must be able to open and close the

Fig. 5.l

There are two ways of constraining the

the motion of the part. This is usually accomplished by designing the contacting

shape of the part geometry. This method of

Fig. 5.2

The second way of holding the part is by

retain the part against gravity, acceleration, and any other force that might arise scratching or other damage.

The friction method of holding the part results in a less complicated and therefore

variety of workparts. However, there is a problem with the friction method that is

a), the part might slip out of the gripper. To resist this slippage,

b),

covers the simpler case in which weight alone is the force tending to cause the part to slip out of the gripper.

$$\mu n F_g =$$

where μ

surface

n

F_g = gripper force

= weight of the part or object being gripped

Fig. 5.3

This equation would apply when the force of gravity is directed parallel to the

Engelberger [3] suggests that in a high-speed handling operation the acceleration

He reduces the problem to the use of a g

follows:

$$\mu n F_g =$$

where g = the g factor. The g factor is supposed to take account of the combined effect of gravity and acceleration. If the acceleration force is applied in the same direction as the gravity force, then the g value = 3.0. If the acceleration is applied in the opposite direction, then the g × the weight of the part due to acceleration minus 1 × the weight of the part due to gravity). If the acceleration is applied in a horizontal direction, then use g
illustrate the use of the equations.

Example 5.1 Suppose a stiff cardboard carton weighing 10 lb is held in a

0.25. The orientation of the carton is such that the weight of the carton is directed
g factor of 3.0
should be applied to calculate the required gripper force. Determine the required
F_g.

$$F_g$$

$$F_g = \frac{30}{0.5} = 60 \text{ lb}$$

carton surface. There is an assumption implicit in the preceding calculations that should be acknowledged. In particular, it is assumed that the robot grasps the carton at its center of mass, so that there are no moments that would tend to rotate the carton in the gripper.

× 60 =
90 lb as the required gripper force. This safety factor would help to compensate for the potential problem of the carton being grasped at a position other than its center of mass.

It is possible to determine the actual value of the g factor without resorting to the rough rule of thumb suggested above. However, data must be available concerning the accelerations and decelerations of the part and the direction of the acceleration

Example 5.2

acceleration of 40 ft/sec/sec in a vertical direction when it is lifted by the gripper

direction as the weight, and that this direction is parallel to the contacting surfaces
g factor for this
situation.

The value of g would be 1.0 + the ratio of the actual acceleration divided by gravity acceleration of 32.2 ft/sec/sec. This ratio is 40/32.2 = 1.24. The g value would be 1.0 + 1.24 = 2.24.

There are various ways of classifying mechanical grippers and their actuating

1. Pivoting movement
2. Linear or translational movement

and close. The motion is usually accomplished by some kind of linkage mechanism.

orientation to each other during actuation.

Fig. 5.4

Fig. 5.5

Mechanical grippers can also be classed according to the type of kinematic device

types:

1. Linkage actuation
2. Gear-and-rack actuation
3. Cam actuation
4. Screw actuation

6. Miscellaneous

The linkage category covers a wide range of design possibilities to actuate the

design of the linkage determines how the input force F_a to the gripper is converted into the gripping force F_g

and how quickly the gripper will actuate.

Fig. 5.6

-

to a piston or some other mechanism that would provide a linear motion. Movement of the rack would drive two partial pinion gears, and these

The cam actuated gripper includes a variety of possible designs, one of which is shown in

Fig. 5.7

cam in one direction would force the gripper to open, while movement of the cam in the opposite direction would cause the spring to force the gripper to close. The advantage of this arrangement is that the spring action would accommodate different

Fig. 5.8

-

screw is turned by a motor, usually accompanied by a speed reduction mechanism. When the screw is rotated in one direction, this causes a threaded block to be translated in one direction. When the screw is rotated in the opposite direction, the threaded block moves in the opposite direction. The threaded block is, in turn,

the corresponding opening and closing action.

Fig. 5.9

designed to open and close a mechanical gripper. Because of the nature of these mechanisms, some form of tension device must be used to oppose the motion of

in one direction to open the gripper, and the tension device would take up the slack in the rope and close the gripper when the pulley system operates in the opposite direction.

The miscellaneous category is included in our list to allow for gripper-actuating

As indicated previously, the purpose of the gripper mechanism is to convert input power into the required motion and force to grasp and hold an object. Let us illustrate the analysis that might be used to determine the magnitude of the required input power in order to obtain a given gripping force. We will assume that a friction-type grasping action is being used to hold the part, and we will therefore use the

other books such as Beer and Johnson [1] and Shigley and Mitchell [6].

Example 5.3 Suppose the gripper is a simple pivot-type device used for holding the

5.10. The gripper force, calculated in

gripper is to be actuated by a piston device to apply an actuating force F_a. The corresponding lever arms for the two forces are shown in the diagram of the

Fig. 5.10

The analysis would require that the moments about the pivot arms be summed and made equal to zero.

$$F_g L_g - F_a L_a = 0$$

$$F_a$$

$$F_a = \frac{720}{3} = 240 \text{ lb}$$

The piston device would have to provide an actuating force of 240 lb to close the gripper with a force against the carton of 60 lb.

Example 5.4 dimensions of a gripper used to handle a workpart for a machining operation. Suppose it has been determined that the gripper force is to be 25 lb. What is required is to compute the actuating force to deliver this force of 25 lb?

Fig. 5.11

$$F \qquad\qquad F$$
$$96.6 = F \qquad\qquad F$$
$$F$$

Fig. 5.12

$$F_a = 2 \times \quad \times$$
$$F_a = 105.2 \text{ lb}$$

Some power input mechanism would be required to deliver this actuating force of 105.2 lb to the gripper.

5.3 OTHER TYPES OF GRIPPERS

In addition to mechanical grippers there are a variety of other devices that can be designed to lift and hold objects. Included among these other types of grippers are the following:

1. Vacuum cups
2. Magnetic grippers

3. Adhesive grippers
4. Hooks, scoops, and other miscellaneous devices

Vacuum cups, also called suction cups, can be used as gripper devices for handling certain types 01 objects. The usual requirements on the objects to be handled are that

Fig. 5.13

The suction cups used in this type of robot gripper are typically made of elastic

handled is composed of a soft material. In this case, the suction cup would be made

round. Some means of removing the air between the cup and the part surface to create the vacuum is required. The vacuum pump and the venturi are two common devices used for this purpose. The vacuum pump is a piston-operated or vane-driven device powered by an electric motor. It is capable of creating a relatively high vacuum.

of 'shop air pressure.' Its initial cost is less than that of a vacuum pump and it is relatively reliable because of its simplicity. However, the overall reliability of the vacuum system is dependent on the source of air pressure.

Fig. 5.14

The lift capacity of the suction cup depends on the effective area of the cup and the negative air pressure between the cup and the object. The relationship can be summarized in the following equation:

$$F = PA$$

where F = the force or lift capacity, lb.

P = the negative pressure, lb/in^2.

A ^{2}.

undeformed area determined by the diameter of the suction cup, The squashing action of the cup as it presses against the object would tend to make the effective area slightly larger than the undeformed area. On the other hand, if the center portion of the cup makes contact against the object during deformation, this would reduce the effective area over which the vacuum is applied. These two conditions tend to cancel each other out. The negative air pressure is the pressure differential between

the operation of the vacuum cup as a robotic gripper device:

Example 5.5

stainless steel, Each piece of steel is $\frac{1}{4}$ in, thick and measures 2.0 by 3.0 ft. The gripper will utilize two suction cups separated by about 1.5 ft for stability. Each suction cup is round and has a diameter of 5.0 in. Two cups are considered a requirement to overcome the problem that the plates may be off center with respect to the gripper. Because of variations in the positioning of the end effector or in the positions of the steel plates before pick up, the suction cups will not always operate on the center of mass of the plates. Consequently, static moments and inertia will result which must be considered in the design of the end effector. We are attempting to compensate for these moments by providing two pressure points on the part, separated by a substantial distance.

^{2}) required to lift the stainless steel plates is to be determined. A safety factor of 1.6 is to be used to allow for acceleration of the plate and for possible contact of the suction cup against the plate which would reduce the effective area of the cup.

We must begin by calculating the weight of the stainless steel plate. Stainless steel ^{3}. The weight of the plate would therefore be

$$\times \frac{1}{4} \times 24 \times$$

This would be equal to the force F which must be applied by the two suction cups, g factor used before). The area of each suction cup would be

$$A = 3.142 \left(\frac{5}{2}\right)^2 = 19.63 \text{ in.}^2$$

The area of the two cups would be $2 \times 19.63 = 39.26$ in.^{2}
negative pressure required to lift the weight can be determined by dividing the weight by the combined area of the two suction cups.

$$P = \frac{w}{A} \qquad {}^2 = 1.54 \text{ lb/in.}^2$$

Applying the safety factor of 1.6, we have

$$P = 1.6 \times 1.54 \text{ lb/in.}^2 = 2.461\text{b/in.}^2 \text{ negative pressure}$$

Some of the features and advantages that characterize the operation of suction cup grippers used in robotics applications are:

2. Applies a uniform pressure distribution on the surface of the part.
   elatively light-weight gripper.
3. Applicable to a variety of different materials.

Magnetic grippers can be a very feasible means of handling ferrous materials.

Other steels, however, including certain types of stainless steel, would be suitable candidates for this means of handling, especially when the materials are handled in sheet or plate form.

In general, magnetic grippers offer the following advantages in robotic-handling applications:

1. Pick up times are very fast.
2. Variations in part size can be tolerated. The gripper does not have to be designed for one particular workpart.

   vacuum grippers).
4. They require only one surface for gripping.

Disadvantages with magnetic grippers include the residual magnetism remaining in the workpiece which may cause a problem in subsequent handling, and the possible side slippage and other errors which limit the precision of this means of handling. Another potential disadvantage of a magnetic gripper is the problem of picking up only one sheet from a stack. The magnetic attraction tends to penetrate beyond the top sheet in the stack, resulting in the possibility that more than a single sheet will be

grippers can be designed to limit the effective penetration to the desired depth, which would correspond to the thickness of the top sheet. Second, the stacking device used to hold the sheets can be designed to separate the sheets for pick up by the robot. One

to induce a charge, in the ferrous sheets in the stack. Each sheet toward the top of the stack is given a magnetic charge, causing them to possess the same polarity and repel each other. The sheet most affected is the one at the top of the stack. It tends to rise above the remainder of the stack, thus facilitating pick up by the robot gripper.

Magnetic grippers can be divided into two categories, those using electromagnets, and those using permanent magnets. Electromagnetic grippers are easier to control, but require a source of dc power and an appropriate controller unit. As with any other robotic-gripping device, the pan must be released at the end of the handling cycle. This is easier to accomplish with an electromagnet than with a permanent magnet. When the part is to be released, the controller unit reverses the polarity at a reduced power level before switching off the electromagnet. This procedure acts to cancel the residual magnetism in the workpiece and ensures a positive release of the part.

source to operate the magnet. However, there is a loss of control that accompanies

the handling cycle, some means of separating the part from the magnet must be provided. The device which accomplishes this is called a stripper or stripping device. Its function is to mechanically detach the part from the magnet. One possible stripper

Fig. 5.15

Permanent magnets are often considered for handling tasks in hazardous environments

operate the magnet reduces the danger of sparks which might cause ignition in such an environment.

Gripper designs in which an adhesive substance performs the grasping action can be used to handle fabrics and other lightweight materials, The requirements on the items to be handled are that they must be gripped on one side only and that other forms of grasping such as a vacuum or magnet are not appropriate. One of the potential limitations of an adhesive gripper is that the adhesive substance loses its tackiness on repeated usage. Consequently, its reliability as a gripping device is diminished with each successive operation cycle. To overcome this limitation, the adhesive material is loaded in the form of a continuous ribbon into a feeding mechanism that is attached to the robot wrist. The feeding mechanism operates in a manner similar to a typewriter ribbon mechanism.

A variety of other devices can be used to grip parts or materials in robotics applications. Hooks can be used as end effectors to handle containers of parts and to load and unload parts hanging from overhead conveyors. Obviously, the items to be handled by a hook must have some sort of handle to enable the hook to hold it.

Scoops and ladles can be used to handle certain materials in liquid or powder form. Chemicals in liquid or powder form, food materials, granular substances, and

this method of holding. One of its limitations is that the amount of material being

cycle is also a problem.

rubber or other elastic material which makes it appropriate for gripping fragile objects. The gripper applies a uniform grasping pressure against the surface of the object rather

Fig. 5.16

universal gripper capable of grasping and handling a variety of objects with differing geometries. If such a universal device could be developed and marketed at a relatively

for each new robot application, Most of the gripper models under consideration are patterned after the human hand which turns out to possess considerable versatility.

directions in end effector design.

Fig. 5.17

5.4 TOOLS AS END EFFECTORS

In many applications, the robot is required to manipulate a tool rather than a workpart. In a limited number of these applications, the end effector is a gripper that is designed to grasp and handle the tool. The reason for using a gripper in these applications is that there may be more than one tool to be used by the robot in the work cycle. The use of a gripper

In most of the robot applications in which a tool is manipulated, the tool is attached

of tools used as end effectors in robot applications include:

1. Spot-welding tools
2. Arc-welding torch
3. Spray-painting nozzle

5. Liquid cement applicators for assembly
6. Heating torches

robot must coordinate the actuation of the spot-welding operation as part of its work cycle. This is controlled much in the same manner as the opening and closing of a mechanical gripper. We will discuss the interface between the robot and its end effector in the following section. Design and application considerations of most of the robot tools listed above will be considered in Chaps. 14 and 15 of the book.

5.5 THE ROBOT/END EFFECTOR INTERFACE

An important aspect of the end effector applications engineering involves the interfacing of the end effector with the robot. This interface must accomplish at least some of the following functions:

1. Physical support of the end effector during the work cycle must be provided. Power to actuate the end effector must be supplied through the interface. Control signals to actuate the end effector must be provided. This is often accomplished by controlling the actuating power.

the robot controller.

In addition, certain other general-design objectives should be met. These include high reliability of the interface, protection against the environment, and overload

The physical support of the end effector is achieved by the mechanical connection between the end effector and the robot wrist. This mechanical connection often consists of a faceplate at the end of the wrist to which the end effector is bolted. In other cases, a more complicated wrist socket is used. Ideally, there should be three characteristics taken into consideration in the design of the mechanical connection strength, compliance, and overload protection. The strength of the mechanical connection refers to its ability to withstand the forces associated with the operation of the end effector. These forces include the weight of the end effector, the weight of the objects being held by the end effector if it is a gripper, acceleration and deceleration

support the end effector against these various forces.

The second consideration in the design of the mechanical connection is compliance. Compliance refers to the wrist socket's ability to yield elastically when subjected to a force. In effect, it is the opposite of rigidity. In some applications, it is desirable to design the mechanical interface so that it will yield during the work

operations require the insertion of an object into a hole where there is very little clearance between the hole and the object to be inserted. If an attempt is made to insert the object off center, it is likely that the object will bind against the sides of the hole. Human assembly workers can make adjustments in the position of the object as designed to provide high lateral compliance for centering the object relative to the hole in response to sideways forces encountered during insertion. We will discuss the

The third factor which must be considered relative to the mechanical interface between the robot wrist and the end effector is overload protection. An overload results when

die, or a tool getting caught in a moving conveyor. Whatever the cause, the consequences involve possible damage to the end effector or maybe even the robot itself. Overload protection is intended to eliminate or reduce this potential damage. The protection can be provided either by means of a breakaway feature in the wrist socket or by using sensors to indicate that an unusual event has occurred so as to somehow take preventive action to reduce further overloading of the end effector.

A breakaway feature is a mechanical device that will either break or yield when subjected to a high force. Such a device is generally designed to accomplish its

the robot. The disadvantage of a device that breaks is that it must be replaced and this generally involves downtime and the attention of a human operator. Some

used to hold structural components in place during normal operation. When abnormal conditions are encountered, these mechanisms snap out of position to release the structural components. Although more complicated than shear pins and other similar devices that fail, their advantage is that they can be reused and in some cases reset by the robot without human assistance.

Sensors are sometimes used either as an alternative to a breakaway device or

signal the robot controller that an unusual event is occurring in the operation of the end effector and that some sort of evasive action should be taken to avoid or reduce damage. Of course, the kinds of unusual events must be anticipated in advance so that the robot controller can be programmed to respond in the appropriate

the end effector becomes caught in a part that is fastened to the conveyor, the most appropriate response might be simply to stop the conveyor and call for help. In other cases, the robot might be programmed to perform motions that would remove the end effector from the cause of the unusual force loading.

End effectors require power to operate. They also require control signals to regulate their operation. The principal methods of transmitting power and control signals to the end effector are:

1. Pneumatic
2. Electric

3. Hydraulic
4. Mechanical

The method of providing the power to the end effector must be compatible

pneumatically operated gripper if the robot has incorporated into its arm design the facility to transmit air pressure to the end effector. The control signals to regulate the end effector are often provided simply by controlling the transmission of the actuating power. The operation of a pneumatic gripper is generally accomplished in this manner. Air pressure is supplied to either open the gripper or to dose it. In some

the gripper might possess a range of open/close positions and there is the need to

from sensors in the end effector are required to operate the device. These feedback signals might indicate how much force is being applied to the object held in the gripper, or they might show whether an arc-welding operation was following the

and signal transmission to the end effector.

Pneumatic power using shop air pressure is one of the most common methods of operating mechanical grippers. Actuation of the gripper is controlled by regulating the incoming air pressure. A piston device is typically used to actuate the gripper. Two air lines feed into opposite ends of the piston, one to open the gripper and the other to close it. This arrangement can be accomplished with a single shop air line by providing a pneumatic valve to switch the air pressure from one line to the other. A

into the opposite end of the chamber, the piston ram is retracted. The force supplied

of the piston diameter. Because of the diameter of the piston ram, the force supplied

piston forces can be calculated as follows:

$$F \quad = P_a \frac{D_p^2}{4}$$

$$F_{\text{retract}} = \frac{P_a}{4} (D_P^2 - D_r^2)$$

Fig. 5.18

where F

F_{retract} = the piston force on the retraction stroke, lb

D = the piston diameter, in.

D_r = the ram diameter, in.

P_a = air pressure, lb/in.^{2}

We will illustrate the kind of engineering analysis required to design a pneumatic

Example 5.6

actuating force of 240 lb is required to provide the desired gripper force. We are now concerned with the problem of designing a piston which can supply the acuating force of 240 lb. It is logical to orient the piston device in the gripper so that this force

^{2} Our problem, therefore, is to determine the required diameter of the

can solve for this diameter.

$$2\ D_p^2$$

$$4$$

$$D^2 = \frac{4 \times 240}{75}$$

$$D = 2.02 \text{ in.}$$

Another use of pneumatic power in end effector design is for vacuum cup grippers. When a venturi device is used to provide the vacuum, the device can be actuated by shop air pressure. Otherwise, some means of developing and controlling the vacuum must be provided to operate the suction cup gripping device.

A second method of power transmission to the end effector is electrical. Pneumatic actuation of the gripper is generally limited to two positions, open and closed.

control over the actuation of the gripper and of the holding force applied. Instead of merely two positions, the gripper can be controlled to any number of partially closed positions. This feature allows the gripper to be used to handle a variety of objects of different sizes, a likely requirement in assembly operations. By incorporating

Other uses of electric power for end effectors include electromagnet grippers, spot-welding and arc-welding tools, and powered spindle tools used as robot end effectors.

Hydraulic and mechanical power transmission are less common means of actuating the end effector in current practice. Hydraulic actuation of the gripper has the potential to provide very high holding forces, but its disadvantage is the risk of oil leaks. Mechanical power transmission would involve an arrangement in which use of pulleys. The possible advantage of this arrangement is a reduction of the weight and mass at the robot's wrist.

5.6 CONSIDERATIONS IN GRIPPER SELECTION AND DESIGN

Most of this chapter has been concerned with grippers rather than tools as end effectors. As indicated in Sec. 5.4, tools are used for spot welding, arc welding,

tooling used with these operations when we discuss the corresponding applications in Chaps. 14 and 15. In this section, let us summarize our discussion of grippers by enumerating some of the considerations in their selection and design.

Certainly one of the considerations deals with determining the grasping

be considered in assessing gripping requirements. The following list is based on Engelberger's discussion of these factors:

2. The size variation of the part must be accounted for, and how this might

   problem in placing a rough casting or forging into a chuck for machining operations.
3. The gripper design must accommodate the change in size that occurs between

   and forging operations.
4. Consideration must be given to the potential problem of scratching and distorting the part during gripping, if the part is fragile or has delicate surfaces.
5. If there is a choice between two different dimensions on a part, the larger dimension should be selected for grasping. Holding the part by its larger surface will provide better control and stability of the part in positioning.

   This provides better part control and physical stability. Use of replaceable

   models.

A related issue is the problem of determining the magnitude of the grasping force that can be applied to the object by the gripper. The important factors that determine the required grasping force are:

The weight of the object. Consideration of whether the part can be grasped consistently about its center of mass. If not, an analysis of the possible moments from off-center grasping should be considered.

The speed and acceleration with which the robot arm moves (acceleration and deceleration forces), and the orientational relationship between the direction of movement and the position of the fingers on the object (whether the movement is parallel or perpendicular to the finger surface contacting the part).

Whether physical constriction or friction is used to hold the part.
Coefficient of friction between the object and the gripper fingers.

We have discussed the methods for dealing with these factors and analyzing the gripping forces in Secs. 5.2 and 5.3. Table 5.1 provides a checklist of the many different issues and factors that must be considered in the selection and design of robot gripper.

Table 5.1 *Checklist of factors in the selection and design of grippers.*

Factor	*Consideration*
Part to be handled	Weight and size
	Shape
	Changes in shape during processing
	Tolerances on the part size
	Surface condition, protection of delicate surfaces
Actuation method	Mechanical grasping
	Vacuum cup
	Magnet
	Other methods (adhesives, scoops, etc.)
Power and signa transmission	Pneumatic
	Electrical
	Hydraulic
	Mechanical
Gripper force	Weight of the object
(mechanical gripper)	Method of holding (physical constriction or friction)
	Coefficent of firction between fingers and object
	Speed and acceleration during motion cycle
Positioning problems	Length of finger's
	Inherent accuracy and repeatability of robot
	Tolerances on the part size
Service conditions	Number of actuations during lifetime of gripper
	Replaceabilty of wear components (fingers)
	Maintenance and serviceability

Contd.

Operating environment	Heat and temperature
	Humidity, moisture, dirt, chemicals
Temperature protection	Heat shields
	Long fingers
	Forced cooling (compressed air, water cooling, etc.)
	Use of heat-resistant materials
Fabrication materials	Strength, rigidity, durability
	Fatigue strength
	Cost and ease of fabrication
	Friction properties for finger surfaces
	Compatibility with operating environment
Other considerations	Use of interchangeable fingers
	Design standards
	Mounting connections and interfacing with robot
	Risk of product design changes and their effect on the gripper design
	Lead time for design and fabrication
	Spare parts. maintenance, and service
	Tryout of the gripper in production

5.1

surface is estimated to be 0.3. The orientation of the gripper will be such that the weight of the part will be applied in a direction parallel to the contacting *g* factor to be used in force calculations should be 3.0. Compute the required gripper force for the

5.2

5.3 *g* factor of 3.0, the following information is given to make the force computation. The robot motion cycle has been analyzed and it has been determined that the largest acceleration

of 1.5, compute the required gripper force.

5.4 A part weighing 15 lb is to be grasped by a mechanical gripper using friction

Why? Compute the required gripper force assuming that a *g* factor of 2.0 is applicable.

5.5 -
mine the required actuating force if the gripper force is to be 25 lb.

Fig. P5.5

5.6 -
culate the required actuating force if the gripper force is to be 20 lb.

Fig. P5.6

5.7

and the diameter of the suction cup is 6.0 in. Determine the negative pressure

^{2}) to lift each plate.

Use a safety factor of 1.5 in your calculations.

5.8 A vacuum pump to be used in a robot vacuum gripper application is capable of drawing a negative pressure of 4.0 lb/in.^{2} compared to atmospheric pressure. The gripper is to be used for lifting stainless steel plates, each plate having dimensions of 15 by 35 in. and weighing 52 lb. Determine the diameter of the suctions cups to be used for the robot gripper if it has been decided that two suction cups will be used for the gripper for greater stability. A factor of safety of 1.5 should be used in the design computations.

5.9

stroke. The inside diameter of the piston is 2.0 in. and the ram diameter is

force? Use a safety factor of 1.3 in your computations.

5.10

the anticipated g factor is 2.0. The gripper is to be operated by a piston whose

Fig. P5.10

(*a*) Determine the required gripping force to retain the part.

(*b*) Determine the actuation force that must be applied to achieve this gripping force for this mechanical design.

(*c*) Determine the air pressure needed to operate the piston so as to apply the required actuation force.

(*d*) If a safety factor of 1.5 were to be used for this design, at what point in the computations would it be appropriate to apply it?

, 3rd ed.,

4. M. P. Groover and E. W. Zimmers, Jr., :

,

6. J. E. Shigley and L. D. Mitchell, , McGraw-Hill.

Sensors in Robotics

Introduction

In Chap. 3, various sensor devices used as components of the robot control system

encoders, and velocity sensors such as tachometers. Sensors can also be used as peripheral devices for the robot, just as end effectors discussed in Chap. 5 are used as peripheral devices. Most industrial applications require that sensors

In this chapter, the various types of sensors that are used as peripheral

a

in the chapter describes the four major categories of uses of sensor systems in robotics.

6.1 TRANSDUCERS AND SENSORS

A transducer is a device that converts one type of physical variable (e.g., force,

conversion is to electrical voltage, and the reason for making the conversion is that the converted signal is more convenient to use and evaluate using a digital computer. A sensor is a transducer that is used to make a measurement of a physical variable of interest. Some of the common sensors and transducers include strain gauges

Any sensor or transducer requires calibration in order to be useful as a measuring

variable and the converted output signal is established.

based on their functions.

1. Analog transducers
2. Digital transducers

Analog transducers provide a continuous analog signal such as electrical voltage or current. This signal can then be interpreted as the value of the physical variable that is being measured. Digital transducers produce a digital output signal, either in the form of a set of parallel status bits or a series of pulses that can be counted. In either form, the digital signal represents the value of the measured variable. Digital

he read as separate measuring instruments. In addition, they offer the advantage in

digital computer than analog-based sensors.

In order to be useful as measuring devices, in robotics and in other applications, sensors must possess certain features. Some of the desirable engineering features of

these desirable features, and a compromise must be made among them to select the best sensor for a given application.

Table 6.1 *Desirable features of sensors.*

1. *Accuracy:* The accuracy of the measurement should be as high as possible. Accuracy is interpreted to mean that the true value of the variable can be sensed with no systematic positive or negative errors in the measurement. Over many measurements of the variable, the average error between the actual value and the sensed value will tend to be zero.
2. *Precision:* The precision of the measurement should be as high as possible. Precision means that there is little or no random variability in the measured variable. The dispersion in the values of a series of measurements will be minimized.
3. *Operating range:* The sensor should possess a wide operating range and should be accurate and precise over the entire range.
4. *Speed of response:* The transducer should be capable of responding to changes in the sensed variable in minimum time. Ideally, the response would be instantaneous.
5. *Calibration:* The sensor should be easy to calibrate. The time and trouble required to accomplish the calibration procedure should be minimum. Further, the sensor should not require frequent recatibration. The term 'drift' is commonly applied to denote the gradual loss in accuracy of the sensor with time and use, and which would necessitate recalibration.
6. *Reliability:* The sensor should possess a high reliability. It should not be subject to frequent failures during operation.
7. *Cost and ease of operation:* The cost to purchase, install, and operate the sensor should be as low as possible. Further, the ideal circumstance would be that the installation and operation of the device would not require a specially trained, highly skilled operator.

6.2 SENSORS IN ROBOTICS

1. Tactile sensors
2. Proximity and range sensors
3. Miscellaneous sensors and sensor-based systems
4. Machine vision systems

We will discuss the first three of these categories in the sections that follow. The fourth category, machine vision, is examined in the next chapter. Table 6.2 provides a listing of some of the common sensors that are applicable in robotic workcells.

Table 6.2 *Sensor devices used in robot workcells.*

Ammeter—(miscellaneous) Electrical meter used to measure electrical current.
Eddy current detectors—(proximity sensor) Device that emits an alternating magnetic field at the tip of a probe, which induces eddy currents in any conductive object in the range of the device. Can be used to indicate presence or absence of a conductive object.
Electrical contact switch—(touch sensor) Device in which an electrical potential is established between two objects, and when the potential becomes zero, this indicates contact between the two objects. Not a commercial device. Can be used to indicate presence or absence of a conductive object.
Infrared sensor—(proximity sensor) Transducer which measures temperatures by the infrared light emitted from the surface of an object. Can be used to indicate presence or absence of a hot object.
Limit switch—(touch sensor) Electrical on–off switch actuated by depressing a mechanical lever or button on the device. Can be used to measure presence or absence of an object.
Linear variable differential transformer—(miscellaneous) Electromechanical transducer used to measure linear or angular displacement.
Microswitch—(touch sensor) Small electrical limit switch (see, limit switch). Can be used to indicate presence or absence of an object.
Ohmmeter—(miscellaneous) Meter used to measure electrical resistance.
Optical pyrometer—(proximity sensor, miscellaneous) Device used to measure high temperatures by sensing the brightness of an object's surface. Can be used to indicate presence or absence of a hot object.
Photometric sensors—(proximity sensor, miscellaneous) Various transducers used to sense light. Category includes photocells, photoelectric transducers, phototubes, photodiodes. phototransistors, and photoconductors. Can be used to indicate presence or absence of an object.
Piezoelectric accelerometer—(miscellaneous) Sensor used to indicate or measure vibration.
Potentiometer—(miscellaneous) Electrical meter used to measure voltage.
Pressure transducers—(miscellaneous) Various transducers used to indicate air pressure and other fluid pressures.

Contd.

Radiation pyromete—(proximity sensor, miscellaneous) Device used to measure high temperatures by sensing the thermal radiation emitting from the surface of an object. Can be used to indicate presence or absence of a hot object.

Strain gage—(force sensor) Common transducer used to measure force, torque, pressure, and other related variables. Can be used to indicate force applied to grasp an object.

Thermistor–(miscellaneous) Device based on electrical resistance used to measure temperatures.

Thermocouple–(miscellaneous) Commonly used device used to measure temperatures. Based on the physical principle that a junction of two dissimilar metals will emit an emf which can be related to temperature.

Vacuum switches—(proximity sensor, miscellaneous) Device used to indicate negative air pressures. Can be used with a vacuum gripper to indicate presence or absence of an object.

Vision sensors—(vision system) Advanced sensor system used in conjunction with pattern recognition and other techniques to view and interpret events occurring in the robot workplace.

Voice sensors—(voice and speed recognition) Advanced sensor system used to communicate commands or information orally to the robot.

6.3 TACTILE SENSORS

6.3.1 Touch Sensors

simpler devices are frequently used in the design of interlock systems in robotics.

coordinate measuring machine, the inspection system normally considered for such

6.3.2 Force Sensors

The capacity to measure forces permits the robot to perform a number of tasks. These include the capability to grasp parts of different sizes in material handling, machine

have become cross-threaded or if the parts are jammed.

of the joint motors. Finally, a third technique is to form an array of force-sensing elements so that the shape and other information about the contact surface can be

Force-sensing wrist

information about the three components of force (F_x, F_y, and F_z

(M_x, M_y, and M_z

to a moment applied to the bracket due to forces and moments on the tool.

Fig. 6.1

computation can be carried out by the robot controller (if it has the required

Based on these calculations, the robot controller can obtain the required information

could be programmed to accomplish these kinds of applications. The procedure

2. Calculate the force offsets required. The force offset in each direction is determined by subtracting the desired force from the measured force.

combined effects of the various joints and links of the robot.

4. Then the robot must provide the torques calculated in step 3 so that the desired forces are applied in each direction.

problems that may be encountered. The end-of-the-arm is often in a relatively hostile environment. This means that the device must

of the robot arm. At the same time the device must be sensitive enough to detect small forces. This design problem is usually solved by using overtravel limits. An

Fig. 6.2

considerable computation time. Also, for an arm traveling at moderate-to-high

limited by the dynamic performance of the arm. The momentum of the arm makes it in the force sensing process itself. Force sensors are made using strain gages that

x

z

Fig. 6.3

Fig. 6.4 *Internal structure of sensor.*

Fig. 6.5.

Fig. 6.5

$$\begin{pmatrix} F_x \\ F_y \\ F_2 \\ M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \end{pmatrix}$$

$$F = CW$$

F

‘C W’
is the strain gage readings corresponding to the forces in each beam. As can be
C’ is non-square and hence a unique

$w_1 \ldots w_8$, for a given force and moment input. Hence, force sensors need to be accurately calibrated before

Joint sensing

torques required to accelerate the links of the arm and to overcome the friction and transmission losses of the joints. In fact, if the joint friction is relatively high (and it

accompany these losses are thereby reduced.

Tactile array sensors A tactile array sensor is a special type of force sensor

number of characteristics about the impression contacting the array sensor surface. Among these characteristics are 1. the presence of an object, 2. the object's contact area, shape, location, and orientation, 3. the pressure and pressure distribution, and

The device is typically composed of an array of conductive elastomer pads. As each pad is squeezed its electrical resistance changes in response to the

By measuring the resistance of each pad, information about the shape of the object against the array of sensing elements can be determined. The operation of a tactile $\times$ pictured in Fig. 6.6. In the background is the CRT monitor display of the tactile impression made by the object placed on the surface of the sensor device, As the number of pads in the array is increased the resolution of the displayed information improves.

Fig. 6.6

Example 6.1 A possible use of a tactile array sensor is to measure the force and moments being applied to an object by the robot. A typical application of this capability is the case of a robot inserting a peg into

of forces on the tactile array sensor surface.

number of interconnections if the number of sensors in the array is very large.

on the surface. The principal application foreseen for this product is for location and

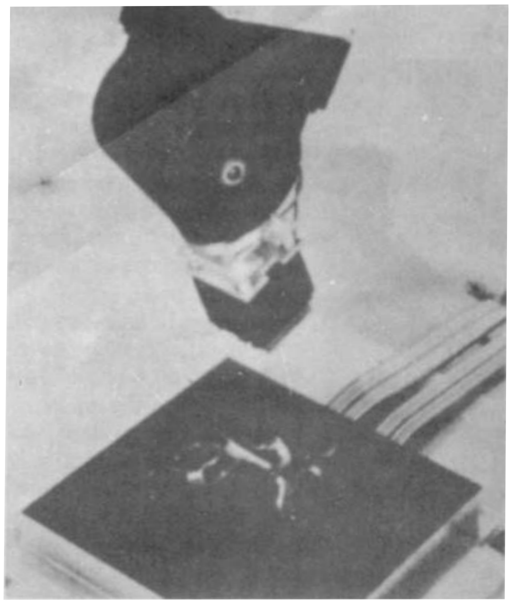


Fig. 6.7 *Tactile array sensor mounted on a flat work surface. (Photo courtesy: Lord Corporation)*

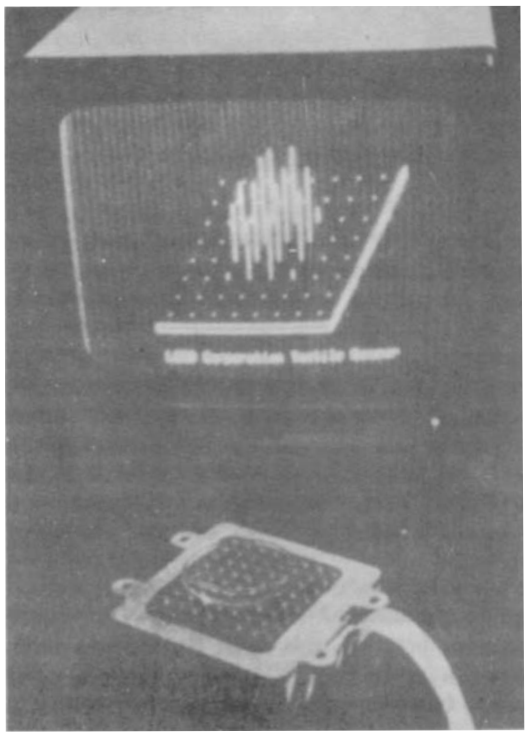


Fig. 6.8 *Tactile array sensor using 8 × 8 sensor array with display of tactile impression. (Photo courtesy: Lord Corporation)*

Fig. 6.9

Research into potential materials for tactile sensors has lead to the development

chemicals. Another interesting feature of this material is that it is pyroelectric,

possibility for designing a tactile sensor that provides simultaneous force and temperature sensing.

6.4 PROXIMITY AND RANGE SENSORS

in relation to the robot.

that is commercially available. The active infrared sensor can be used to indicate not

are often utilized in security systems to detect the presence of bodies giving off heat

areas in building interiors.

surface of the object, the location of the object can be determined from the position

Fig. 6.10

x $y\tan(A$

x = the distance of the object from the sensor

y

the linear array. This distance corresponds to the number of elements contained

A

the object must be parallel to the sensing array.

chamber is used to sense the change in the sound pattern. This kind of device can also be used as a range sensor.

the region so long as the object is made of a conductive material. These eddy currents

presence of the object.

being detected or it can be part of the sensor device. In either case, the device can be designated so that the presence of the object in the region of the sensor completes the

is required for its operation.

6.5 MISCELLANEOUS SENSORS AND SENSOR BASED SYSTEMS

The miscellaneous category covers the remaining types of sensors and transducers

sensors used for these variables are listed in Table 6.2.

An area of robotics research that might be included in this chapter is voice sensing or voice programming. Voice-programming systems can be used in robotics for oral communication of instructions to the robot. Voice sensing relies on the techniques of

6.6 USES OF SENSORS IN ROBOTICS

The major uses of sensors in industrial robotics and other automated manufactur-

1. Safety monitoring

4. Determining positions and related information about objects in the robot cell

-

in Chap. 17.

The second major use of sensor technology in robotics is to implement interlocks

The third category is quality control. Sensors can be used to determine a variety of part quality characteristics. Traditionally, quality control has been performed using manual inspection techniques on a statistical sampling basis. The use of sensors

is that the sensor system can only inspect for a limited range of part characteristics

of robotics. The reason for including this category in our discussion of robotic sensors is that robots are, in fact, often used to implement

applications in Chap. 15.

The fourth major use of sensors in robotics is to determine the positions and

the robot. Feedback information is required to improve the accuracy of the robot's positioning.

that station. Simple optical sensors are typically used to indicate the presence or be required to pick up parts moving along a conveyor in random orientation and

systems represent an important category of sensor system that might be employed to determine such characteristics as part location and orientation. We consider machine

In some applications, the accuracy requirements in the application are more stringent than the inherent accuracy and repeatability of the robot. Certain assembly

possible solution that might be used is a

All four categories of sensor applications (safety monitoring, interlocks, inspection,

function. That control system, in turn, is a component of a larger control system

roblems

6.1

system use one or more sensors to detect the presence of humans in the cell.

(a)

(b) From the list of sensors in Table 6.2, select several alternative sensors that

the safety monitoring system. Use sketches if necessary to illustrate the

(c) Select the best sensor alternative, and justify your selection.

6.2

6.3

for determining the presence or absence of a non-metallic part in a

=

6.4

in a robot cell. Assume that the robot could be used to implement the sensor system, or that the sensor operation could be independent of the robot. Use sketches as necessary to illustrate your proposal.

6.5

bins depending on their size. Select a sensor system from Table 6.2 that

candidates and compare their relative attributes for the application.

6.6

6.7 A linear array of light sensors is to be used to determine the distance x in

A representing the

the object reaches a position y

x.

References

1. J. F. Engelberger, *Robotics in Practice*

3. E. Kafrissen and M. Stephans, *Industrial Robots and Robotics*, Reston, Reston,

, Marketing/Technical Brochure, Cary,

‘ ’ *Robotics Today*, June

Machine Vision

Introduction

an important sensor technology with potential applications in many industrial operations. Many of the current applications of machine vision are in inspection; however, it is anticipated that vision technology will play an increasingly

manufacturing world. The systems are used to perform tasks which include

improved

Com

stages of development. Advances in vision technology and related disciplines

tions

navigation, cartography, and medical

image analysis.

7.1 INTRODUCTION TO MACHINE VISION

Machine vision is concerned with the sensing of vision data and its interpretation a

a digital computer, and hardware and software necessary to interface them. This interface hardware and software is often referred to as a preprocessor. The operation of the vision system consists of three functions:

2. Image processing and analysis
3. Application

Fig. 7.1.

a camera focused on the scene of interest. Special lighting techniques are frequently

is called a *frame of vision data,*
called a *frame grabber*

picture
elements, or *pixels*

the scene which reduces that portion to a single value. The value is measure of the

Sec. 7.2.

processing and analysis functions for data reduction and interpretation of the image.

will

the data representation of the image. This data reduction can change the representation

Fig. 7.1 *Functions of a machine vision system.*

previously computed values stored in the computer. These descriptors include shape and analysis will discussed in Sec. 7.3.

To accomplish image processing and analysis, the vision system frequently must

computer models. The information gathered during training consists of features such

determine if a match has occurred, Sec. 7.4 will discuss training of a vision system.

The third function of a machine vision system is the applications function.

require two cameras in order to achieve a stereoscopic view of the scene, while other

range of which is called a gray scale.

7.2 THE SENSING AND DIGITIZING FUNCTION IN MACHINE VISION

of machine vision.

Image sensing requires some type of image formation device such as a camera

the image of the scene with the vision camera. The image consists of relative light intensities corresponding to the various portions of the scene. These light intensities

form.

the vision controller.

7.2.1 Imaging Devices

Figure 7.2 illustrates the vidicon camera. In the operation of this system, the lens forms an image on the glass faceplate of the camera. The faceplate has an inner

The second layer is, a thin photosensitive material deposited over the conducting

electrical resistance in response to increasing illumination. A charge is created in each small area upon illumination. An electrical charge pattern is thus generated corresponding to the image formed on the faceplate. The charge accumulated for an

Fig. 7.2

(Reprinted with permission of McGraw-Hill, Inc. [10])

current at the video signal electrode. The magnitude of the signal is proportional to the light intensity and the amount of time with which an area is scanned. The current is then directed through a load resistor which develops a signal voltage which is

of the impinging light is considered. In the United States, the entire faceplate is

considered. The output of the camera is a continuous voltage signal for each line

positive control electrodes in isolated wells due to voltages applied to the central

illustrated in Figs. 7.3(*a* and (*b* .

Fig. 7.3 *Basic principle of charge-coupled device: (a) Accumulation of an electron charge in a pixel element: (b) Movement of accumulated charge through the silicon by changing the voltages on the electrodes A, B, and C. (Reprinted with permission of McGraw-Hill, Inc. [10])*

Figure 7.4 indicates one type of CCD imager. Charges are accumulated for the

cycle $\frac{1}{60}$th of a second.

Fig. 7.4 *One type of charge-coupled device imager. Register A accumulates the pixel charges produced by photoconductivity generated by the light image. The B register stores the lines of pixel charges and transfers each line in turn into register C. Register C reads out the charges laterally as shown*

7.2.2 Lighting Techniques

An essential ingredient in the application of machine vision is proper

lighting makes the

the following categories:

1. **Diffuse surface** *devices*

2. **Condenser projectors**
   source into a condensing light source. This is useful in imaging optics.
3. **Flood or spot projectors** Flood lights and spot lights are used to illuminate surface areas.
4. **Collimators**

5. **Imagers**

is to direct the path of light from the lighting device to the camera so as to display the

Table 7.1 *Illumination techniques.*

Technique	*Function/use*
A. Front light source	
1. Front illumination	Area flooded such that surface is defining feature of image.
2. Specular illumination (dark field)	Used for surface defect recognition (background dark).
3. Specular illumination (light field)	Used for surface defect recognition; camera in-line with reflected rays (background light).
4. Front imager	Structured light application; imaged light superimposed on object surface–light beam displaced as function of thickness.
B. Back light source	
1. Rear illumination (lighted field)	Uses surface diffusor to silhouette features; used in parts inspection and basic measurements.
2. Rear illumination (condenser)	Produces high-contrast images; useful for high magnification application.
3. Rear illumination (collimator)	Produces parallel light ray source such that the features of object do not lie in same plane.
4. Rear offset illumination	Useful to produce feature highlights when feature is in transparent medium.
C. Other miscellaneous devices	
1. Beam splitter	Transmits light along same optical axis as sensor; advantage is that it can illuminate difficult-to-view objects.
2. Split mirror	Similar to beam splitter but more efficient with lower intensity requirements.
3. Non-selective redirectors	Light source is redirected to provide proper illumination
4. Retroreflector	A device that redirects incident rays back to sensor; incident angel capable of being varied; provides high contrast for object between source and reflector.
5. Double density	A technique used to increase illumination intensity at sensor; used with transparent media and retroreflector.

Front lighting simply means that the light

back lighting, the light source

7.2.3 Analog-to-Digital Signal Conversion

conversion process involves taking an analog input voltage signal and producing an

Sampling

twice the highest [14]

frequency in the video signal if

Fig. 7.5 *Sampling and digitizing an analog waveform: (a) Analog waveform indicating sampling interval, t, and sampled voltage points, (b) Digital approximation to analog signal.*

Example 7.1 signal is generated for each line of the 512 lines comprising the

×

10^{-9}

milliseconds $\left(\frac{1}{30}\text{ s}\right)$ to scan the entire faceplate consisting of 512 lines, determine the sampling rate and

100×10^{-9} s

$= 0.1 \times 10^{-6}$

The scanning rate for the 512 lines in the faceplate is $\frac{1}{30}$ s = 33.33×10^{-3} s

Accordingly, the scanning rate for each line is = (33.33×10^{-3}

$= 65.1 \times 10^{-6}$

$$= \frac{65.1 \times 10^{-6}\text{ s/line}}{0.1 \times 10^{-6}\text{ s/pixel}}$$

in home television. For a 512 ×

sampling

one of the limiting factors in the development of machine vision. Systems with a

× ×

impose lower computational requirements: however, the image resolution of these systems is much lower than for systems possessing a greater density. For a

vertical resolution.

Quantization

$= 2^n$

where n

=

2^4

Encoding

code. This process, termed encoding, involves representing an

various amplitude levels is a function of the spacing of

Full-scale range

2^n

$= \pm \frac{1}{2}$

Example 7.2 A continuous

error.

converter resolution = $\frac{1}{256}$ = 0.0039 or 0.39 per cent. For the 5 V range,

$$\text{pacing} = \frac{(5\text{ V})}{(2^8)}\ 0.0195\text{ V}$$

$$\pm \frac{1}{2} \qquad 0.00975\text{ V}$$

Example 7.3

levels as fellows:

Voltage range, V	*Binary number*	*Gray scale*
0–0.0195	0000 0000	0 (black)
0.0195–0.0390	0000 0001	1 (dark gray)
0.0390–0.0585	0000 0010	2
4.9610–4.9805	1111 1110	254 (light gray)
4.9805–5.0	1111 1111	255 (white)

If we ± 5 V range, we could

^{7} =

7.2.4 Image Storage

conversion, the image is stored in computer memory, typically called

Ideally, one would want to acquire a single frame of data in real time. The frame

picture and acquire it in = $\frac{1}{30}$ s

uffer is adequate since the average camera system cannot

^{6} = 64 gray

7.3 IMAGE PROCESSING AND ANALYSIS

and stored in a computer. For use of the stored image in industrial applications, the

represent various gray ×

$\frac{1}{30}$ s

short period of time and has led to various techniques to reduce the magnitude of the

1. Image data reduction
2. Segmentation

7.3.1 Image Data Reduction

step in the data analysis, the following two schemes have found common usage for data reduction:

1. Digital conversion
2. Windowing

the large volume of data in image processing.

= 256 gray

Example 7.4

level values required if converter is used to indicate various shades of

(a) For gray scale imaging with 2 = 256 levels of gray

× ×

(b)

× ×

Windowing involves using only a portion of the total image stored in the frame and analysis. This portion is called the *window*. For

of the total scene.

7.3.2 Segmentation

Segmentation is a general term which applies to various methods of data reduction. In

There are many ways to segment an image.

Three important techniques that we will discuss are:

1. Thresholding

3. Edge detection

In its simplest form, *thresholding*

accomplished

shows a regular image with each

are trying to differentiate

To improve the

a high contrast.

Fig. 7.6 *Obtaining a binary image by thresholding: (a) Image of object with all gray-levels present, (b) Histogram of image, (c) Binary image of object after thresholding. (Photos courtesy: Robotics Laboratory, Lehigh University)*

employed.

Thresholding is the most widely used technique for segmentation in industrial vision applications. The reasons are that it is fast and easily implemented and that the

a region.

Region growing is a
grouped in regions called *grid elements*

other regions means of an analysis of the difference in their average properties and spatial connectiveness. For instance, consider an image as depicted in Fig. 7.7(*a*

Fig. 7.7(*b*

simple procedure did not identify the hole in the key of Fig. 7.7(*a*
resolved
with which the original image is represented.

Fig. 7.7 *Image segmentation: (a) Image pattern with grid, (b) Segmented image after runs test.*

provide an adequate partition of an image into a set of meaningful regions. Such

images could have the following procedure:

a region. In the simplest

if

consider only edge detection or simple thresholding. This is due to the fact that light implementation is simpler.

Fig. 7.8 *Edge following procedure to detect the edge of a binary image.*

7.3.3 Feature Extraction

means of features that uniquely

these features is that the features should not depend on position or orientation. The

Table 7.2

Gray level (maximum, average, or minimum)
Area
Perimeter length
Diameter
Minimum enclosing rectangle
Center of gravity—For all pixels (n) in a region where each pixel is specified by (x, y) coordinates, the x and y coordinates of the center of gravity are defined as

$$C.G._x = \frac{1}{n}\sum_x x$$

$$C.G._y = \frac{1}{n}\sum_y y$$

Eccentricity: A measure of 'elongation' Several measures exist of which the simplest is

$$\text{Eccentricity} = \frac{\text{Maximum chord length } A}{\text{Maximum chord length } B}$$

where maximum chord length B is chosen perpendicular to A.

Aspect ratio—The length-to-width ratio of a boundary rectangle which encloses the object. One objective is to find the rectangle which gives the minimum aspect ratio.

Thinness—This is a measure of how thin an object is. Two definitions are in use

(a) $$\text{Thinness} = \frac{(\text{Perimeter})^2}{\text{Area}}$$

This is also referred to as compactness.

Diameter (b) $$\text{Thinness} = \frac{\text{Diameter}}{\text{Area}}$$

The diameter of an object, regardless of its shape, is the maximum distance obtainable for two points on the boundary of an object.

Contd.

Holes—Number of holes in the object.

Moments—Given a region, R, and coordinates of the points (x, y) in or on the boundary of the region, the pqth order moment of the image of the region is given as

$$M_{pq} = \sum_{x,\, y} x^p y^q$$

Example 7.5 Consider the schematic of the image in Fig. 7.9. Determine the area, the minimum aspect ratio, the diameter, the centroid, and the thinness measures of the image.

calculation, the origin is translated to O' with x', y'
determined from the moment, $M_{o'o'}$ as

$$M_{o'o'} = \sum_{x',\, y'} x'y'$$

$$\text{Minimum aspect ratio} = \frac{\text{Length}}{\text{Width}} = \frac{9}{4}$$

$n =$

$$\text{C.G.}_{x'} = \frac{1}{n}\sum_{x'} x'$$

$$\text{C.G.}_{y'} = \frac{1}{n}\sum_{y'} y'$$

$$\text{C.G.}_{x'} = \frac{1}{24}\left[4\left(\frac{1}{2}\right) + 4\left(\frac{3}{2}\right) + 4\left(\frac{5}{2}\right) + 2\left(\frac{7}{2}\right) + \ldots + 2\left(\frac{17}{2}\right)\right]$$

$$= \frac{1}{24}(90) = \frac{15}{4} \text{ units}$$

$$\text{C.G.}_{y'} = \frac{1}{24}\left[3\left(\frac{1}{2}\right) + 9\left(\frac{3}{2}\right) + 9\left(\frac{5}{2}\right) + 3\left(\frac{7}{2}\right)\right]$$

$$= \frac{96}{48} = 2 \text{ units}$$

Fig. 7.9 *Schematic of pixel pattern for Example 7.5.*

$$\text{Compactness} = \frac{(\text{Perimeter})^2}{\text{Area}} = \frac{26^2}{24}$$

$$\text{Thinness} = \frac{\text{Diameter}}{\text{Area}} = \frac{9}{24} = \frac{3}{8}$$

7.3.4 Object Recognition

2. Structural techniques

during the training procedure in which the vision system is programmed for known

are compared to the corresponding stored values. These values constitute the stored template. When a match is found, allowing for certain statistical variations in the rectangle. This kind of technique, known *as* syntactic pattern recognition, is the most

Accordingly, it is often more appropriate to search for simpler regions or edges

7.4 TRAINING THE VISION SYSTEM

The purpose of vision system training is to program the vision system with known

Vision system manufacturers have developed application software for each

7.5 ROBOTIC APPLICATIONS

Many of the current applications of machine vision are inspection tasks that do not

device that is communicating with the vision system.

vision applications in an industrial setting are:

.

that the recognition process is facilitated if

to control the appearance.

1. Inspection

3. Visual servoing and navigation

holes and other features in a part. When these kinds of inspection operations are performed manually, there is a tendency for human error. Also, the time required in

using 100 percent inspection, and usually in much less time. In Chap. 15, we will

machine vision.

The second category, is concerned with applications in which the

In the third application category, *visual servoing and navigational control,* the devices

servoing is where the machine vision system is used to control the

of this application include part positioning, retrieving parts moving along a

deal of intelligence is required in the controller to use the data for navigation and collision avoidance. This and the visual servoing tasks remain important

The *bin-picking*

the container, and then it must direct the end effector to a position to permit grasping

the target and its surroundings are far from ideal for part recognition.

distortion to determine the path and other parameters required for a successful

arc welding in Chap. 14.

Fig. 7.10 *i-bot robot-vision system. (Photo courtesy: Object Recognition Systems, Inc.)*

In

Machine vision, coupled with the force and torque sensors discussed in Chap.

application in Chap. 15.

roblems

7.1 Consider a vision system which provides one frame of 256 lines every $\frac{1}{2}$ s. The system is a raster scan system. Assume that the time for the electro

line. Determine the sampling rate for the system if it

7.2

7.3

7.4 ×

60	59		57	59	45	25	15
55	60	59	61	55	40	12	11
59		60	60	11	12	10	10
54	55		25	10	11	11	
59	60	20	15	11	10	55	59
60	59	15	12	15	10	60	60
60	15	10	11	10	12	60	59
55	14	9		11		62	60
10	11	15	11	12	59	61	60
9	10	11	12	60	57	59	55

Fig. P7.4

7.5

technique.

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	0	0	0	1	1	1	1	0
0	1	1	1	1	0	0	0	1	1	1	0	0
0	1	1	1	1	1	1	1	1	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	0
	1	0	0	1	1	1	1	0	0	1	1	0
0	1	0	0	1	1	1	1	1	1	1	1	0
0	1	0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	1	1	0

Fig. P7.5

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.6

a and

b

elements and compare the results.

7.7

a *b*

aspect ratio, (*c* (*d* the centroid of the image. Choose the

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. P7.7

References

, Chapman and Hall, New York,

,

, U.S. Department of Commerce,

Computers in Mechanical Engineering,

Tutorial on Robotics, C. S.

CAD CAM: Computer-Aided Design and Manufacturing,

,
pp.

Computers in Mechanical Engineering,
Computers in Mechanical Engineering, pp. 59–69.
McGraw-Hill Encyclopedia of Electronics and Computers,

Robotics,
pp. 3–22.
Fundamentals in Computer
,
Robotics Today, June
pp. 53–57.

14. W. E. Snyder, *Industrial Robots: Computer Interfacing and Control,*

Robotics Today,
pp. 63–67.

Robot Programming and Languages

PART THREE

Robot programming is concerned with teaching the robot its work cycle. A large portion of the program involves the motion path that the robot must execute in moving parts or tools from one location in the work space to another. These movements are often taught by showing the robot the motion and recording it into the robot's memory. However, there are other portions of the program that do not involve any movement of the arm. These, other parts of the program include interpreting sensor data, actuating the end effector, sending signals to other pieces of equipment in the cell, receiving data from other devices, and making computations and decisions about the work cycle. Many of these other activities are best taught by programming the robot using a computer-like language.

Chapters 8 and 9 consider the two fundamental methods for programming today's industrial robots. Chapter 8 details the 'teach-by-showing' methods of programming. Chapter 9 presents what we consider to be a comprehensive discussion of how robots are programmed with a computer-like robot language. There are several appendixes to Chap. 9, which present summaries of some of the commercially available robot languages.

Advanced technology robots of the future with versatile end effectors and sophisticated sensors, will be capable of responding to very high-level commands-higher, more general commands than we have in roday's commercially available languages. The robots will have to interpret these high-level commands and act upon them. To do this, robots of the future must possess more intelligence than today's machines. In Chap. 10, the field of artificial intelligence to see what promise this technology holds for robotics will be discussed.

8 Robot Programming

Introduction

In Chap. 2, we defined a robot program to be a path in space. It is really more than that. A robot today can do much more than merely move its arm through a series of points in space. Current technology robots can accept input from sensors and other devices. They can send signals to pieces of equipment operating with them in the cell. They can make decisions. They can communicate with computers to receive instructions and to report production data and problems. All of these capabilities require programming.

8.1 METHODS OF ROBOT PROGRAMMING

Robot programming is accomplished in several ways. Consistent with current industrial practice we divide the programming methods into two basic types:

1. Leadthrough methods
2. Textual robot languages

The leadthrough methods require the programmer to move the manipulator through the desired motion path and that the path be committed to memory by the robot controller. The leadthrough methods are sometimes referred to as 'teach-by-showing' methods. Chronologically, the leadthrough methods represent the first real robot programming methods used in industry. They had their beginnings in the early 1960s when robots were first being used for industrial applications.

Robot programming with textual languages is accomplished somewhat like computer programming. The programmer types in the program on a CRT (cathode ray tube) monitor using a high-level English-like language. The procedure is usually augmented by using leadthrough techniques to teach the robot the locations of points in the workspace. The textual languages started to be developed in the 1970s, with the first commercial language appearing around 1979.

In addition to the leadthrough and textual language programming, another method of programming is used for simple, low-technology robots. We referred to these types of machines in Chap. 2 as limited sequence robots which are controlled by means of mechanical stops and limit switches to define the end points of their joint motions. The setting of these stops and switches might be called a programming method. We prefer to think of this kind of programming as a manual set-up procedure.

In this chapter, the leadthrough methods will be discussed along with the basic features and capabilities of these programming methods. What functions must a typical robot be able to do, and how is it taught to do these functions using leadthrough programming? In the following chapter, the textual programming languages and their capabilities will be examined.

8.2 LEADTHROUGH PROGRAMMING METHODS

In leadthrough programming, the robot is moved through the desired motion path in order to record the path into the controller memory. There are two ways of accomplishing leadthrough programming:

1. Powered leadthrough
2. Manual leadthrough

The powered leadthrough method makes use of a teach pendant to control the various joint motors, and to power drive the robot arm and wrist through a series of points in space. Each point is recorded into memory for subsequent play back during the work cycle. The teach pendant is usually a small handheld control box with combinations of toggle switches, dials, and buttons to regulate the robot's physical movements and programming capabilities. Among the various robot programming methods, the powered leadthrough method is probably the most common today. It is largely limited to point-to-point motions rather than continuous movement because of the difficulty in using the teach pendant to regulate complex geometric motions in space. A large number of industrial robot applications consist of point-to-point movements of the manipulator. These include part transfer tasks, machine loading and unloading, and spot welding.

The manual leadthrough method (also sometimes called the 'walkthrough' method) is more readily used for continuous-path programming where the motion cycle involves smooth complex curvilinear movements of the robot arm. The most common example of this kind of robot application is spray painting, in which the robot's wrist, with the spray painting gun attached as the end effector, must execute a smooth, regular motion pattern in order to apply the paint evenly over the entire surface to be coated. Continuous arc welding is another example in which continuous-path programming is required and this is sometimes accomplished with the manual leadthrough method.

In the manual leadthrough method, the programmer physically grasps the robot arm (and end effector) and manually moves it through the desired motion cycle. If the robot is large and awkward to physically move, a special programming apparatus is often substituted for the actual robot. This apparatus has basically the same geometry as the robot, but it is easier to manipulate during programming. A teach button is often located near the wrist of the robot (or the special programming apparatus) which is depressed during those movements of the manipulator that will become part of the programmed cycle. This allows the programmer the ability to make extraneous

is divided into hundreds or even thousands of individual closely spaced points along the path and these points are recorded into the controller memory.

The control systems for both leadthrough procedures operate in either of two modes: teach mode or run mode. The teach mode is used to program the robot and the run mode is used to execute the program.

The two leadthrough methods are relatively simple procedures that have been developed and enhanced over the last 20 years to teach robots to perform simple, repetitive operations in factory environments. The skill requirements of the programmers are relatively modest and these procedures can be readily applied in the plant.

8.3 A ROBOT PROGRAM AS A PATH IN SPACE

This and the following sections of this chapter will examine the programming issues involved in the use of the leadthrough methods, with emphasis on the powered

sequence of positions through which the robot will move its wrist. In most applications, an end effector is attached to the wrist, and the program can be considered to be the path in space through which the end effector is to be moved by the robot.

of the path in space in effect requires that the robot move its axes through various positions in order to follow that path. For a robot with six axes, each point in the path consists of six coordinate values. Each coordinate value corresponds to the position

effector and the wrist determines its orientation. If we think of a point in space in the robot program as a position and orientation of the end effector, there is usually

to reach that point.

each point in the path.

Fig. 8.1

discussing here. For the sake of simplicity, let us assume that we are programming a point-to-point Cartesian robot with only two axes, and only two addressable points for each axis. An addressable point is one of the available points (as determined

be commanded to go to that point. Figure 8.2 shows the four possible points in the robot's rectangular workspace. A program for this robot to start in the lower left-hand corner and traverse the perimeter of the rectangle could be written as follows:

Fig. 8.2

Example 8.1

Step	Move	Comments
1	1,1	Move to lower left corner
2	2,1	Move to lower right corner
3	2,2	Move to upper right corner
4	1,2	Move to upper left corner
5	1,1	Move back to start position

The point designations correspond to the x, y coordinate positions in the cartesian

Using the same robot, let us consider its behavior when performing the following program:

Example 8.2

Step	Move	Comments
1	1,1	Move to lower left corner
2	2,1	Move to lower right corner
3	1,2	Move to upper left corner
4	1,1	Move back to start position

corner (2,2) has not been listed. Before explaining the implications of this missing point, let us recall that in Example 8.1, the move from one point to the next required

moved. The question that arises is what path will the robot follow in getting from the

same time, and the robot will therefore trace a path along the diagonal line between the two points. The other possibility is that the robot will move only one axis at a time and trace out a path along the border of the rectangle, either through point 2,2 or through point 1,1.

The question of which path the robot will take between two programmed points is not a trivial one. It is important for the programmer to know the answer in order to plan out the motion path correctly. Unfortunately, there is no general rule that all robots follow. Limited-sequence non-servo robots, which are programmed using manual setup procedures rather than leadthrough methods, can usually move both

(as described in Chap. 4), which is along the diagonal in our illustration. Other

Usually, these robots that move one axis at a time do so by moving the lower

However, there are no industry standards on this issue, and the programmer must make this kind of determination either from the user's manual or by experimentation with the actual robot. Servocontrolled robots, which are programmed by leadthrough and textual language methods, tend to actuate all axes simultaneously. Hence, with servocontrol, the robot would likely move approximately along the diagonal path between points 2,1 and 1,2. The differences between the paths for Example 8.2 are illustrated in Fig. 8.3.

As illustrated by the preceding discussion of Example 8.2, it is possible for the programmer to make certain types of robots pass through points without actually including the points in the program. The key phrase is 'pass through.' These are not addressable points in the program and the robot will not actually stop at them in the sense of an addressable point.

Fig. 8.3

programmer during the teach mode to actuate the robot arm and wrist. We list the following three methods:

1. Joint movements
2. *x-y-z* coordinate motions (also called world coordinates)
3. Tool coordinate motions

usually by means of a teach pendant. The teach pendant has a set of toggle switches

the end effector has been positioned to the desired point. This method of teaching

way of programming the robot.

To overcome this disadvantage, many robots can be controlled during the teach mode to move in *x-y-z* coordinate motions. This method, called the world coordinate

coordinate system with origin at some location in the body of the robot. In the case

the robot into the Cartesian coordinate system. These conversions are carried out in such a way that the programmer does not have to be concerned with the substantial computations that are being performed by the controller. To the programmer, the wrist (or end effector) is being moved in motions that are parallel to the *x*, y, and *z*

almost always rotational, and while programming is being done in the *x-y-z* system to

in a constant orientation. The *x-y-z*

Fig. 8.4

robot. This is a Cartesian coordinate system in which the origin is located at some point on the wrist and the *xy* plane is oriented parallel to the faceplate of the wrist. Accordingly, the *z* axis is perpendicular to the faceplate and pointing in the same direction as a tool or other end effector attached to the faceplate. Hence, this method of moving the robot could be used to provide a driving motion of the tool. Again, a

Figure 8.5 shows the tool coordinate system.

Fig. 8.5

The preceding examples and discussion are intended to argue that there are some

2. To avoid obstacles

is programmed to pick up a part at a given location or to perform a spot-welding

This category also includes safe positions that are required in the work cycle. For

which the robot would start the work cycle.

workcell. Machines, conveyors, and other pieces of equipment in the work volume

the collisions can be prevented.

Most robots allow for their motion speed to be regulated during the program execution. A dial or group of dials on the teach pendant are used to set the speed for different portions of the program. It is considered good practice to operate the robot at a relatively slow speed when the end effector is operating close to obstacles in the workcell, and at higher speeds when moving over large distances where there are no obstacles. This gives rise to the notion of 'freeways' within the cell. These are possible pathways in the robot cell which are free of obstructions and therefore permit operation at the higher velocities.

The speed is not typically given as a linear velocity at the tip of the end effector for robots programmed by leadthrough methods. There are several reasons for this. First, the robot's linear speed at the end effector depends on how many axes are moving at one time and which axes they are. Second, the speed of the robot depends

robot will be much greater with its arm fully extended than with the arm in the fully retracted position. Finally, the speed of the robot will be affected by the load it is carrying due to the force of acceleration and deceleration. All of these reasons lead to considerable computational complexities when the control computer is programmed to determine wrist end velocity.

languages so that the wrist or even the end effector velocity can be programmed in more conventional units (e.g., millimeters per second or inches per second). This capability is not available with all computer-controlled robots because of the reasons mentioned above. However, we will assume that it is available for our purposes in Chap. 9.

8.4 MOTION INTERPOLATION

Suppose we were programming a two-axis servocontrolled cartesian robot with eight addressable points for each axis. Accordingly, there would be a total of 64 addressable points that we can use in any program that might be written. The work volume is illustrated in Fig. 8.6. Assuming the axis sizes to be the same as our previous limited sequence robot, a program for the robot to perform the same work cycle as Example 8.1 would be as follows:

Fig. 8.6 ×

Example 8.3

Step	Move
1	1,1
2	8,1
3	8,8
4	1,8
5	1,1

If we were to remove step 3 in this program (similar to Example 8.2), our servocontrolled robot would execute step 4 by tracing a path along the diagonal line from point 8,1 to point 1,8. This process is referred to as interpolation. This internal algorithm followed by the robot controller to get between the two points is somewhat more complicated than it appears from our simple illustration. Also, as indicated in

to get from one point to another. Before discussing these differences, let us describe the most basic interpolation process, called .

In

that requires the longest time. This determines the time it will take to complete the

the movement required for the other axes, the controller subdivides the move into

Consider, for example, the move from point 1,1 to point 7,4 in the grid of Fig. 8.6.

a set of intermediate addressable points along the path between 1,1 and 7,4 which would be followed by the robot. The following program illustrates the process:

Example 8.4

Step	Move	Comments
1	1,1	
2	2,2	Internally generated interpolation point
3	3,2	Internally generated interpolation point
4	4,3	Internally generated interpolation point
5	5,3	Internally generated interpolation point
6	6,4	Internally generated interpolation point
7	7,4	

one axis. Also, for each move requiring actuation of both axes, the two axes start and stop together. This kind of actuation causes the robot to take a path as illustrated in Fig. 8.7. The controller does the equivalent of constructing a hypothetically perfect

points as close to that line as possible. The resulting path is not a straight line, but is rather an approximation. The controller approximates the perfect path as best it can within the limitations imposed by the control resolution of the robot (the available addressable points in the work volume). In our case, with only 64 addressable points in the grid, the approximation is very rough. With a much larger number of addressable points and a denser grid, the approximation would be better.

Fig. 8.7

The reader might have noticed that the interpolation procedure used above created a straight line approximation. This is usually referred to as straight line interpolation,

and straight line interpolation are the same. For other robots with a combination of

the preceding discussion. In , the robot controller computes the straight line path between two points and develops the sequence of addressable points along the path for the robot to pass through. As indicated the procedure is identical to the example given in Example 8.4.

Consider a robot that has one rotational axis (axis 1) and one linear axis (axis 2), where each axis has eight addressable points. This creates a total of 64 addressable points which form the grid shown in Fig. 8.8. The grid is polar rather than rectilinear. During an interpolation procedure, this has the effect of creating moves of different lengths (from the viewpoint of euclidean geometry). For example, compare the move from 1,1 to 3,2 with the move from 1,7 to 3,8. The addressability of a robot with rotational axes is not uniform in euclidean space. Moves that are made close to the

schemes used by the controller. Although the descriptions given above still apply for

robot will be affected by the change in anatomy. The incremental moves executed by the robot consist of combinations of rotational moves (along axis 1) and linear moves (along axis 2). We leave the visualization of these effects to the reader as exercises

Fig. 8.8

On many robots, the programmer can specify which type of interpolation scheme to use. The possibilities include:

1. Joint interpolation
2. Straight line interpolation
3. Circular interpolation
4. Irregular smooth motions (manual leadthrough programming)

interpolated path between two points unless the programmer specifies straight line (or some other type of) interpolation.

Circular interpolation requires the programmer to define a circle in the robot's workspace. This is most conveniently done by specifying three points that lie along the circle. The controller then constructs an approximation of the circle by selecting a series of addressable points that lie closest to the defined circle. The movements that are made by the robot actually consist of short-straight-line segments. Circular interpolation therefore produces a linear approximation of the circle. If the gridwork of addressable points is dense enough, the linear approximation looks very much like a real circle. Circular interpolation is more readily programmed using a textual programming language than with leadthrough techniques.

In manual leadthrough programming, when the programmer moves the manipulator wrist to teach spray painting or arc welding, the movements typically consist of combinations of smooth motion segments. These segments are sometimes approximately straight, sometimes curved (but not necessarily circular), and sometime back-and-forth motions. We are referring to these movements as *irregular smooth motions*, and an interpolation process is involved in order to achieve them. To approximate the irregular smooth pattern being taught by the programmer, the motion path is divided into a sequence of closely spaced points that are recorded into the controller memory. These positions constitute the nearest addressable points to the path followed during programming. The interpolated path may consist of thousands of individual points that the robot must play back during subsequent program execution.

8.5 WAIT, SIGNAL, AND DELAY COMMANDS

Robots usually work with something in their work space. In the simplest case, it may be a part that the robot will pick up, move, and drop off during execution of its work cycle. In more complex cases, the robot will work with other pieces of equipment in the workcell, and the activities of the various equipment must be coordinated. This situation introduces the subject of workcell control, which we shall discuss in more detail in Chap. 11. For the moment, let us introduce the kinds of basic programming commands that must be employed in workcell control.

Nearly all industrial robots can be instructed to send signals or wait for signals during execution of the program. These signals are sometimes called *interlocks*, and their various applications in workcell control will be discussed in Chap. 11. The most common form of interlock signal is to actuate the robot's end effector. In the case of a gripper, the signal is to open or close the gripper. Signals of this type are usually binary; that is, the signal is on-off or high-level-low-level. Binary signals are not readily capable of including any complex information such as force sensor measurements. The binary signals used for the robot gripper are typically implemented by using one or more dedicated lines. Air pressure is commonly used to actuate the gripper. A binary valve to actuate the gripper is controlled by means of two interlock signals, one to open the gripper and the other to close it. In some cases, feedback signals can be used to verify that the actuation of the gripper had occurred, and interlocks could be designed to provide this feedback data.

In addition to control of the gripper, robots are typically coordinated with other devices in the cell also. For example, let us consider a robot whose task is to unload a press. It is important to inhibit the robot from having its gripper enter the press before the press is open, and even more obvious, it is important that the robot remove its hand from the press before the press closes.

To accomplish this coordination, we introduce two commands that can be used during the program. The first command is

SIGNAL M

which instructs the robot controller to output a signal through line M (where M is one of several output lines available to the controller). The second command is

WAIT N

which indicates that the robot should wait at its current location until it receives a signal on line N (where N is one of several input lines available to the robot controller).

Let us suppose that the two-axis robot of Fig. 8.2 is to be used to perform the unloading of the press in our example. The layout of the workcell is illustrated in Fig. 8.9, which is similar to Fig. 8.6. The platten of the press (where the parts are to be picked up) is located at 8,8. The robot must drop the parts in a tote pan located at 1,8. One of the columns of the press is in the way of an easy straight line move from 8,8 to 1,8. Therefore, the robot must move its arm around the near side of the column in order to avoid colliding with it. This is accomplished by making use of points 8,1 and 1,1. Point 8,1 will be our position to wait for the press to open before entering the press to remove the part, and the robot will be started from point 1,1, a point in space known to be safe in the application. We will use controller ports 1 to 10 as output (SIGNAL) lines and ports 11 through 20 as input (WAIT) lines. Specifically, output line 4 will be used to actuate (SIGNAL) the press, and output lines 5 and 6 will be used to close and open the gripper, respectively. Input line 11 will be used to receive the signal from the press indicating that it has opened (WAIT). The following is our program to accomplish the press unloading task (the sequence begins with the gripper in the open position).

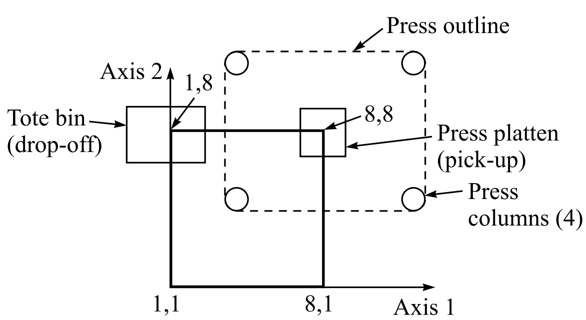


Fig. 8.9 *Robot work space for press unloading operation of Example 8.5.*

Example 8.5

Step	Move or signal	Comments
0	1,1	Start at home position
1	8,1	Move to wait position
2	WAIT 11	Wait for press to open
3	8,8	Move to pickup point
4	SIGNAL 5	Signal gripper to close
5	8,1	Move to safe position
6	SIGNAL 4	Signal press to actuate
7	1,1	Move around press column
8	1,8	Move to tote pan
9	SIGNAL 6	Signal gripper to open
10	1,1	Move to safe position

Each step in the program is executed in sequence, which means that the SIGNAL and WAIT commands are not executed until the robot has moved to the point indicated in the previous step.

The operation of the gripper was assumed to take place instantaneously so that its actuation would be completed before the next step in the program was started. Some grippers use a feedback loop to ensure that the actuation has occurred before the program is permitted to execute the next step. A WAIT instruction can be programmed to accomplish this feedback. One of the exercises at the end of the chapter deals with this problem.

An alternative way to address this problem is to cause the robot to delay before proceeding to the next step. In this case, the robot would be programmed to wait for a specified amount of time to ensure that the operation had taken place. The form of the command for this second case has a length of time as its argument rather than an input line. The command

DELAY X SEC

indicates that the robot should wait X seconds before proceeding to the next step in the program. Below, we show a modified version of Example 8.5, using time as the means for assuring that the gripper is either opened or closed.

Example 8.6

Step	Move or signal	Comments
0	1,1	Start at home position
1	8,1	Move to wait position
2	WAIT 11	Wait for press to open
3	8,8	Move to pickup point
4	SIGNAL 5	Signal gripper to close

Contd.

5	DELAY 1 SEC	Wait for gripper to close
6	8,1	Move to safe position
7	SIGNAL 4	Signal press that hand is clear
8	1,1	Move around press column
9	1,8	Move to tote pan
10	SIGNAL 6	Signal hand to open
11	DELAY 1 SEC	Wait for gripper to open
12	1,1	Move to home position

The reader is cautioned that our programs above are written to look like computer programs. This is for convenience in our explanation of the programming principles. The actual teaching of the moves and signals is accomplished by leading the arm through the motion path and entering the non-motion instructions at the control panel or with the teach pendant. In the majority of industrial applications today, robots are programmed using one of the lead through methods. Only with the textual language programming do the programs read like computer program listings.

8.6 BRANCHING

Most controllers for industrial robots provide a method of dividing a program into one or more branches. Branching allows the robot program to be subdivided into convenient segments that can be executed during the program. A branch can be thought of as a subroutine that is called one or more times during the program. The subroutine can be executed either by branching to it at a particular place in the program or by testing an input signal line to branch to it. The amount of decision logic that can be incorporated into the program varies widely with controllers. However, most controllers allow for a branch to be identified or assigned one of a pre-established group of names. They permit the use of an incoming signal to invoke a branch. Most controllers allow the user to specify whether the signal should interrupt the program branch currently being executed, or wait until the current branch completes. The interrupt capability is typically used for error branches. An error branch is invoked when an incoming signal indicates that some abnormal event (e.g., an unsafe operating condition) has occurred. Depending on the event and the design of the error branch, the robot will either take some corrective action or simply terminate the robot motion and signal for human assistance.

A frequent use of the branch capability is when the robot has been programmed to perform more than one task. In this case, separate branches are used for each individual task. Some means must be devised for indicating which branch of the program must be executed and when it must be executed. A common way of accomplishing this is to make use of external signal which are activated by sensors or other interlocks. The device recognizes which task must be performed, and provides the appropriate signal to call that branch. This method is frequently used on spray painting robots which have been programmed to paint a limited variety of parts moving past the workstation of a conveyor. Photoelectric cells are frequently employed to identify the part to be sprayed by distinguishing between the geometric features (e.g., size, shape, the presence of holes, etc.) of the different parts. The photoelectric cells are used to generate the signal to the robot to call the spray painting subroutine corresponding to the particular part.

Given the concept of a branch or subroutine that might be repeated in a program, an additional concept is readily introduced. Robot programs have thus far been

executes the program or the branch of the program, each point is visited at exactly the same location every time. The new concept involves the use of a relocatable branch.

A relocatable branch allows the programmer to specify a branch involving a set

point for the branch. This would permit the same motion subroutine to be performed at various locations in the workspace of the robot. Many industrial robots have the capacity to accept relocatable branches as part of a program. The programmer

from robot to robot), and the controller records relative or incremental motion points rather than absolute points.

Let us illustrate these branching concepts by developing two versions of a robot program to perform a palletizing operation. A pallet is a common type of container used in a factory to hold and move parts. Suppose that the operation required the robot to pick up parts from an input chute, and place them on a pallet with 24 positions as depicted in Fig. 8.10. When a start signal is given, the robot must begin picking up parts and loading them into the pallet, continuing until all 24 positions on the pallet

wait for the start signal to begin the next cycle.

must go to during the execution of the palletizing program. We will use names for convenience to identify these points, with the understanding that the names must be

Point name	Explanation
SAFE	Safe location to start and stop
PICKUP	Location of part pickup at end of chute
INTER	Intermediate point above chute to pass through
LOC 1	
LOC 2	Location of second pallet position
LOC 24	Location of 24th pallet position
ABOVE 1	
ABOVE 24	Location above 24th pallet position

Fig. 8.10

In creating robot programs for pelletizing operations of this type, the robot is programmed to approach a given part from a direction chosen to avoid interference with other parts. Accordingly, the various pallet locations would be approached from points at some distance directly above each location. The 24 points named ABOVE 1 through ABOVE 24 have been designated for this purpose. Similarly, the point

The speed at which the program is executed should be varied during the program. When the gripper is approaching a pick up or drop off point, the speed setting should be at a relatively slow value. When the robot moves larger distances between the chute and the pallet, higher speeds would be programmed.

As our example programs in this chapter have become more complicated,

as variable names, and the use of commands such as WAIT and SIGNAL, we have seen the need to change the format of our program listing. Once again we are faced with the need to expand the way we express the program. In the programs below, we introduce the MOVE command followed by the name of the point to which the command refers. The instruction

MOVE SAFE

means that the robot should move its end effector to the point in space called SAFE.

instruction in the program.

The program is initiated upon receipt of a start signal on input line 11. The robot is interlocked so that it must await a signal on input line 12 indicating that a part has been delivered by the chute and is ready for pickup. To operate the gripper, our program will use output line 5 to open the gripper, and output line 6 to close the gripper. We assume that the gripper is fast acting, and that no feedback signals or waiting time are needed (as suggested by Examples 8.5 and 8.6) to ensure that the operation of the gripper has taken place. Finally, output line 7 will be used to indicate that the pallet is full.

We begin our program development by presenting a robot program in which no use is made of the branching capability. This will serve as a starting point to show

The reader will note that our program contains a total of 243 commands and that portions of the program are very repetitious. Obviously, the repetition could be reduced by means of branches (subroutines) that can be written and called during the program. Let us consider the task of fetching and placing the parts that is repeated 24 times in Example 8.7. The task can be divided into two subtasks, one to fetch the part from the chute position, and the second to place the part in the correct pallet position. This division can be seen if we examine a portion of the program corresponding to the actions involved in fetching and placing any given part (we will use the second part to illustrate).

Example 8.7

Step	Command	Comments
1	MOVE SAFE	Move to the starting safe position
2	WAIT 11	Wait for start signal on line 11
(The following portion of the program directs the robot to pick up first part.)		
3	MOVE INTER	Go to the intermediate point above chute
4	WAIT 12	Wait for next part from chute
5	SIGNAL 5	Open gripper
6	MOVE PICK UP	Move gripper to pick-up part
7	SIGNAL 6	Close gripper
8	MOVE INTER	Depart to intermediate point above chute
9	MOVE ABOVE l	Move to point above first pallet location
10	MOVE LOC 1	Position part in first pallet location
11	SIGNAL 5	Open gripper
12	MOVE ABOVE 1	Depart slowly from pic-kup point
(The next portion of the program directs the robot to pick up second part.)		
13	MOVE INTER	Go to the intermediate point above chute
14	WAIT 12	Wait for next part from chute
15	SIGNAL 5	Open gripper
16	MOVE PICK UP	Move gripper to pick-up part
17	SIGNAL 6	Close gripper
18	MOVE INTER	Depart to intermediate point above chute
19	MOVE ABOVE 2	Move to point above second pallet location
20	MOVE LOC 2	Position part in second pallet location
21	SIGNAL 5	Open gripper
22	MOVE ABOVE 2	Depart slowly from pick-up point
(The preceding portions of the program are repeated for the next 21 part 5.)		
⋮		
(The next portion of the program directs the robot to pick up 24th part.)		
232	MOVE INTER	Go to the intermediate point above chute
233	WAIT 12	Wait for next part from chute
234	SIGNAL 5	Open gripper
235	MOVE PICK UP	Move gripper to pick-up part
236	SIGNAL 6	Close gripper
237	MOVE INTER	Depart to intermediate point above chute
238	MOVE ABOVE 2	Move to point above second pallet location
239	MOVE lOC 2	Position part in second pallet location
240	SIGNAL 5	Open gripper
241	MOVE ABOVE 2	Depart slowly from pick-up point
(The pallet is now full.)		
242	MOVE INTER	Go to the intermediate safe position
243	SIGNAL 7	Signal that pallet is full

This portion of the program is repeated below, indicating where the subdivision occurs. (The following is the 'fetch' subtask.)

13	MOVE INTER	Go to the intermediate point above chute
14	WAIT 12	Wait for next part from chute
15	SIGNAL 5	Open gripper
16	MOVE PICK UP	Move gripper to pick-up part
17	SIGNAL 6	Close gripper
18	MOVE INTER	Depart to intermediate point above chute

(The following is the 'place' subtask.)

19	MOVE ABOVE 2	Move to point above second pallet location
20	MOVE LOC 2	Position part in second pallet location
21	SIGNAL 5	Open gripper
22	MOVE ABOVE 2	Depart slowly from pick-up point

Notice that the coding for the 'fetch' portion of the task is identical regardless of which part number is being picked up. We will identify this portion of the program as a branch and we will name it 'FETCH.' It can be expressed as follows:

BRANCH FETCH	Indicates the following is branch FETCH
MOVE INTER	Go to the intermediate point above chute
WAIT 12	Wait for next part from chute
SIGNAL 5	Open gripper
MOVE PICK UP	Move gripper to pick-up part
SIGNAL 6	Close gripper
MOVE INTER	Go to intermediate point above chute
END BRANCH	This is the end of the branch

The second subtask to place the part in a numbered pallet position is nearly the same for the 24 repetitions except for the fact that the location changes on each replication. It would therefore constitute a relocatable branch. We will name it 'PLACE' and to use it for each part we must somehow move the end effector to the point above the pallet location corresponding to that part. One way to do this is to use incremental positioning in which the robot is directed to move a certain distance rather than to a particular point. The PLACE subroutine might be expressed as follows for this method:

BRANCH PLACE	Indicates the following is branch PLACE
MOVE Z(–50)	Position part in pallet
SIGNAL 5	Open gripper to release part
MOVE Z(+50)	Depart from pick-up point
END BRANCH	This is the end of the branch

The MOVE commands indicate that the robot should move its end effector in the z-axis direction by a distance of 50 mm. MOVE Z(–50) directs the robot to move down 50 mm and the command MOVE Z(+50) directs it to move in a positive z direction by 50 mm. In this case the robot would first have to be moved to the point above the required pallet location. The sequence for a particular part placement (we will use part number 2 for illustration) would be as follows:

MOVE ABOVE 2	Move to point above second pallet location
PLACE	Calls branch PLACE for execution

In the incremental positioning approach, it is very important that each ABOVE point (ABOVE 1 through ABOVE 24) be located very precisely relative to the respective pallet position. Otherwise, the PLACE branch will position the parts inaccurately on the pallet.

below.

Example 8.8

Step	Command	Comments
1	BRANCH FETCH	Indicates the following is branch FETCH
2	MOVE INTER	Go to the intermediate point above chute
3	WAIT 12	Wait for next part from chute
4	SIGNAL 5	Open gripper
5	MOVE PICK UP	Move gripper to pick-up part
6	SIGNAL 6	Close gripper
7	MOVE INTER	Depart to intermediate point above chute
8	END BRANCH	This is the end of the branch
9	BRANCH PLACE	Indicates the following is branch PLACE
10	MOVE Z(–50)	Position part in pallet
11	SIGNAL 5	Open gripper to release part
12	MOVE Z(+50)	Depart from pick up point
13	END BRANCH	This is the end of the branch
14	MOVE SAFE	Move robot to the starting safe position
15	WAIT 11	Wait for start signal on line 11
19	FETCH	Fetch second part
20	MOVE ABOVE 2	Move to second position
21	PLACE	Place second part
85	FETCH	Fetch 24th part
86	MOVE ABOVE 24	Move to 24th position
87	PLACE	Place 24th part
88	MOVE INTER	Go to intermediate safe position
89	SIGNAL 7	Signal that pallet is full

been reduced from 243 to 89 for the palletizing operation described above. This is a substantial reduction in the programming effort.

There are other ways in which the use of branches can make programming of robots easier. A good example is in the editing of a robot program. The problem of editing an existing program can be facilitated when the program is divided into branches, even if the branches are used only once during the work cycle. Instead of editing the entire program when an error is discovered, the individual branch can be edited. For instance, when a robot is programmed using the manual leadthrough method, the editing of the program is never easy and sometimes impossible. This is due to the fact that these programs are recorded into the robot control memory second. Accordingly, to correct an individual point or series of points would require the user to know which of thousands of points must be changed. Rather than make the editorial corrections in the control memory, it is common practice to divide such a program into logical sections (branches). When a program is found to be in error, the particular branch containing the error is deleted from the program and retaught using the manual leadthrough method.

8.7 CAPABILITIES AND LIMITATIONS OF LEADTHROUGH METHODS

Some of the teach pendants used for commercially available robots possess a wide

done using the toggle switches and dials of a simple teach pendant. For WAIT, SIGNAL, and DELAY commands, special buttons must be added to the pendant

programming of branches can also be accomplished with a teach pendant in various ways. For example, a branch can be executed by using a toggle switch mounted either on the teach pendant or on the controller itself. When this method is used, there are several toggle switches corresponding to a predetermined set of branches. For

regular program at the desired times by manipulating the toggle switches to indicate when each of the branches are to be executed. Some of the teach pendants are quite sophisticated in terms of the kinds of instructions that can be programmed into the controller. In Appendix A9 after Chap. 9, we will describe the operation of the teach pendant supplied by United States Robots for the MAKER 110 model. This is an example of a programming device with features that are somewhat beyond those

additional programming terms in Chap. 9 usually associated with robot programming languages before discussing the MAKER 110 system.

Although the leadthrough programming controls offer the above capabilities, there are certain inherent limitations with the leadthrough methods. These limitations can be summarized as follows:

1. The robot cannot be used in production while it is being programmed.

   accomplish leadthrough programming using the currently available methods.
3. Leadthrough programming is not readily compatible with modern computer-based technologies such as CAD/CAM, data communications networking, and integrated manufacturing information systems.

The fact that leadthrough programming requires the presence of the robot precludes the use of the robot in production. This has important economic implications. Since programming takes time away from production, it means that the batch size of parts

of the programming cost. If the lot size is too small, it might take longer to prepare the program than to run it.

The second limitation deals with the fact that robots are being employed in production applications of increasing complexity, and being called on to perform

to program these kinds of functions. As we have progressed through this chapter in our discussion of the various robot programming functions, it has become more

programming language.

The third limitation is concerned with the problem of interfacing robots to other computer-based systems in the corporation. One of the important goals in manufacturing is to establish computer-integrated manufacturing (CIM) systems in which the data and information necessary to make a product is captured originally on a CAD/CAM data base, and downloaded through the various manufacturing planning

The various components of the CIM system need to be able to communicate with each other and with the central plant computer. The use of leadthrough programming procedures does not lend itself to the communications and data base requirements of this kind of computer-integrated factory. Textual robot languages are more suited to these needs. We continue our study of robot programming in the next chapter by examining the opportunities and capabilities offered by these languages.

8.1 Using the 8 × 8 square grid illustrated in Fig. 8.6, show the path taken by a cartesian coordinate robot if it is directed to move between the following sets of points in the grid using linear interpolation:

- ***(a)*** Point (l,l) and point (6,6).
- ***(b)*** Point (2,1) and point (8,2).
- ***(c)*** Point (2,2) and point (7,5).

8.2 Using the gridwork illustrated in Fig. 8.8 for a robot with one rotational axis and one linear axis, show the path taken by the robot if it is directed to move

- ***(a)*** Point (l,l) and point (6,6).
- ***(b)*** Point (2,1) and point (8,2).
- ***(c)*** Point (2,2) and point (7,5).

8.3 Using the gridwork illustrated in Fig. 8.8 for a robot with one rotational axis and one linear axis, show the path taken by the robot if it is directed to move between the following sets of points in the grid using linear interpolation:

- ***(a)*** Point (1,1) and point (6,6).
- ***(b)*** Point (2,1) and point (8,2).
- ***(c)*** Point (2,2) and point (7,5).

8.4 Rewrite the program in Example 8.5 in the text so that it includes the use of WAIT instructions to make sure that the gripper has opened and closed properly before the next step in the program has been executed. Use the coding format we have adopted in this chapter to write the program.

The following programming exercises are intended to illustrate the programming features of a robot that is programmed using the powered leadthrough method. This type of robot includes many of the small teaching robots, such as the Microbot TeachMover and the Rhino robot.

8.5 Using a pen mounted in the robot's end effector, program the robot to write

sheet of paper attached to the surface of the work table.

8.6 The robot is to be programmed to pick up a part from point A and move it to point B, followed by a move to a neutral position. Points A and B are to be

should pick up the part at point B and move it back to point A, followed by a move to the previous neutral position. The robot can be operated continually in the 'run' mode to repeat the motion pattern over and over.

8.7 As an enhancement to Problem 8.6, if the capability exists on the robot used (e.g., Microbot Teachmover), program the robot to check each time it attempts to pick up the part to determine whether or not it has closed on the part. If the part is in the gripper, then continue the program. If the part is not in the gripper, the robot should move to the neutral position, provide a signal of some kind (e.g., light or buzzer), and wait 5s. It should then attempt the pick up again.

8.8 Program the robot to pick up two blocks (the blocks are of different sizes)

in the center position. The larger block will always be on one side of the center and the smaller block will always be on the other side of the center position. The smaller block is to be placed on top of the larger block.

8.9 This exercise is similar to the Prob. 8.8 except that the positions of the two blocks can be exchanged at random. It is not known whether the larger block is on one side of the center or the other. The robot must be programmed to

pick up the smaller block and place it on top of the larger block.

8.10

on a conveyor and to place it at an upstream location on the conveyor so

position is established by means of a mechanical stop along the conveyor so that the part is always in the same orientation and location for the robot.

1. M. P. Groover and E. W. Zimmers, Jr., *CAD/CAM*: , Prentice-Hall, Englewood Cliffs, NJ, 1984, Chap. 10.
2. E. Kafrissen and M. Stephens, , Reston, Reston, VA, 1984, Chap. 9.

3. R. P. Paul, *Robot Manipulators*: *Mathematics*, *Programming*, *and Control*, The MIT Press, Cambridge, MA, 1981, Chap. 10.
4. R. R. Schreiber, ‘How to Teach a Robot,’ *Robotics Today*, June 1984, pp. 51–56.
5. R. Thomas, ‘Programming Expands Limits of Robot Controllers,’ *Industrial Enginnering*, April, 1983, pp. 34–40.
6. L. L. Toepperwein, M. T. Blackman, et al., ‘ICAM Robotics Application Guide,’ *Technical Report AFWAL-TR-80-4042*, Vol. II, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Ohio, April 1980.

Robot Languages

Introduction

9.1 THE TEXTUAL ROBOT LANGUAGES

9.2 GENERATIONS OF ROBOT PROGRAMMING LANGUAGES

9.2.1 First Generation Languages

9.2.2 Second Generation Languages

:

1. Motion control

2. Advanced sensor capabilities

3. Limited intelligence

4. Communications and data processing

9.2.3 Future Generation Languages

9.3 ROBOT LANGUAGE STRUCTURE

Fig. 9.1 *Diagram of robot system showing various components of the system that must be coordinated by means of the language.*

9.3.1 Operating Systems

9.3.2 Robot Language Elements and Functions

9.4 CONSTANTS, VARIABLES, AND OTHER DATA OBJECTS

9.4.1 Constants and Variables

z

9.4.2 Aggregates and Location Variables

$\langle \quad \rangle$

$\langle \quad \rangle$
$\langle \quad \rangle$

$\langle \quad \rangle$

$x\ y\ z$

9.5 MOTION COMMANDS

9.5.1 Move and Related Statements

z

z

〈 〉〈 〉

9.5.2 Speed Control

$\langle \qquad \rangle$

$x\ y\ z$

$\langle \qquad \rangle$

x xy

z

xy

9.6 END EFFECTOR AND SENSOR COMMANDS

9.6.2 Sensor Operation

9.7 COMPUTATIONS AND OPERATIONS

9.8 PROGRAM CONTROL AND SUBROUTINES

9.8.1 Program Sequence Control

Example 9.1

xy

x y

x

y

Fig. 9.2

y

x

$\langle \quad \rangle$

9.8.2 Subroutines

Example 9.2

y
x

$\langle \quad \rangle$

Example 9.3

9.9 COMMUNICATIONS AND DATA PROCESSING

9.10 MONITOR MODE COMMANDS

roblems

9.1

9.2

Fig. P9.2(a)

Fig. P9.2(b)

9.3

x *y*

x *y*

9.4

9.5

x *y*

x *y*

9.6

x *y*

9.7
9.8

9.9

Fig. P9.9(a)

Fig. P9.9(b)

9.10

9.11

9.12

9.13

9.14

xy *z*

xy

xy

eview Questions

9.1

9.2

9.3

9.4

(a)

(b)

Proceedings

RAIL Software Reference Manual

Conference Proceedings

IBM 7565 Manufacturing System, A Manufacturing Language Concepts. and User's Guide,
A Manufacturing Language Reference

Final Report
Robotics Today
Industrial Robots: Computer Interfacing and Control
Conference Proceedings

The International Journal of Robotics Research
Industrial

Proceedings

Conference Papers

Programming Manual *User's Guide to VAL II*

Conference Proceedings

Appendix

9A

Programming the Maker Robot

The MAKER robot system is produced by United States Robots, Incorporated, a subsidiary of Square D Company. The appendix is based on the *MAKER Robot System Operation Manual*, copyright 1984 by United States Robots. Portions of the manual are reprinted with permission of United States Robots, Inc.

The MAKER robot system is somewhat different to program than the other systems presented in this chapter. The MAKER is designed to be operated with a minimum of training of the operator. To accomplish this the MAKER is programmed using only the teach pendant. The teach pendant displays menus which allow the operator to select the necessary program commands. By responding to the menu options available, the operator is able to define points or to develop programs.

9A.1 THE TEACH PENDANT

The teach pendant is shown in Fig. 9A.1. Table 9A.l describes the function of each of the major elements of the teach pendant.

Table 9A.1 *Teach pendant functions.*

Item	Function
Display	Error messages, status, operator prompts
Function 1–5	Buttons with associated displays for menu choices
Numeric	To enter numeric data
Prev Lev	To move to the previous menu level
Status	Displays status of overrides
Halt	Stops arm motion and puts the system in ‘halt’ mode
Vert	Align Aligns the tool Z axis with the world Z axis
JT W/T	Selects joint or world/tool motions in manual mode
Relax	Relaxes the selected joint
Speed	Used to set monitor speed of arm
Grip 1,2	Used to operate the gripper controls
Motion switches	Used to move the joints or the arm in world or tool mode depending on JT W/T

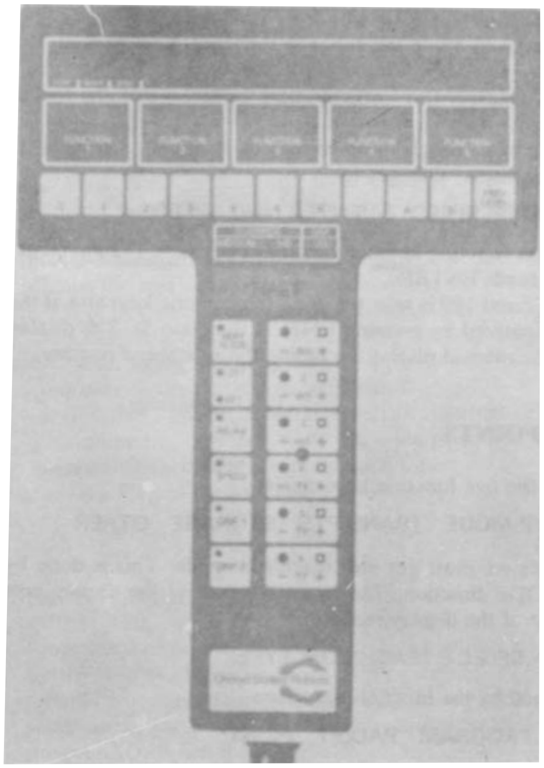


Fig. 9A.1 *Teach pendant for the MAKER robot system. (Photo courtesy: United States Robots, Inc.)*

9A.2 MOVING THE ROBOT

When the robot is in manual mode it is possible to move the robot using the teach pendant. The robot is in manual mode when the display reads MAN in the mode area of the display.

The robot can be moved in three ways: by individual joint, in world space, or in tool space.

In order to select joint space the JT W/T button should be pressed until the JT indicator is lit. When the JT indicator is lit the motion buttons will move the selected joint in the direction of the square or circle painted on the joints corresponding to the square or circle being selected on the teach pendant. For example, pushing the joint 3 button on the square will cause joint 3 on the robot to extend outward as long as the button is depressed.

If W/T is selected, the top three motions switches can be used to move the robot in the World X, Y and Z (WX, WY, WZ) directions and the bottom three switches can be used to move the robot in the Tool X, Y and Z (TX, TY, TZ) directions.

Pushing the VERT ALIGN (for vertical align) button aligns the tool Z axis with the world Z axis.

To adjust the speed with which the manipulator is moving the SPEED button is pushed and the display changes to:

MAN MONITOR SPEED: ⟨x⟩ ENTER NEW SPEED ⟨x⟩

4 reads CLEAR and 5 reads ENTER.

is correct, it is entered by pressing ENTER (function 5). The display then changes

9A.3 TEACHING POINTS

TCH MODE STP MODE TRANS PTS STORAGE OTHER

In order to teach points we must get into teach/edit mode. This is done by selecting the TCH MODE function. The mode segment of the display now reads TCH and the rest of the display reads

SELECT TEACH/EDIT TYPE

PROGRAM PALLET POINT

Since we are concerned with teaching points we would select FUNCTION 3 for teaching POINTs.

If the POINT selection is made, the teach pendant will request a number between 1 and 9999. This number is the 'name' of the point. If no point of that number has been previously taught then the display will read

POINT ⟨xxxx⟩ IS NOT TAUGHT GRP ⟨yyy⟩

ACCEPT STEP RECORD GROUP OTHER

Selecting OTHER presents the following additional options:

CONTROL COPY TRANS PTS DELETE OTHER

in the desired position and pushes the RECORD button. The point is now recorded as that number. If desired the position may have been copied from another point by using the COPY option.

If the point has already been taught and is being edited, the options are the same as for teaching a new point, but the display will read

POINT ⟨xxxx⟩ IS TAUGHT GRP ⟨yyy⟩

and then the ACCEPT function.

Each point is assigned to a GROUP. The group name is a number between 1 and

transformed as groups if necessary. The group number may be assigned or adjusted by using the GROUP function. We will not discuss the rest of the options in detail but we will describe their functions.

STEP is used to step the robot to different named points.

CONTROL is used to turn on and off the various system overides available during stepping. For example, it may disable the outputs from the controller to the peripheral equipment.

TRANS PTS allows the operator to transform a group of points as a unit.

maintaining their relationship to each other.

DELETE is used to delete points.

9A.4 TEACHING PROGRAMS

the display. In order to teach programs, the operator must select PROGRAM from the main TCH menu described earlier. When the PROGRAM selection is made the display will request a program number; this is the program ID. When the ID is selected, the display will read

⟨ID#⟩ TCH ⟨step#⟩ ⟨description of program step⟩

If no steps have been taught, the description will read, END OF PROGRAM.

MOVE STEP GRIPPER GOSUB OTHER

Pressing OTHER causes the function displays to read

REPLACE STEP INSERT DELETE OTHER

Pressing OTHER again, causes

MODIFY STEP COPY CONTROL OTHER

Obviously, there are a lot of options available for the types of steps to be programmed,

We will begin with a description of the most fundamental type of program step, a MOVE.

When MOVE is selected the following is displayed (in addition to the program and step numbers)

MOVE PT ⟨xxxx⟩ SPD ⟨yy⟩ ⟨ABC⟩

where xxxx represents the point number, yy the speed, and ABC the motion type. The functions display

ACCEPT POINT SPEED PTP/CP OTHER

Pushing OTHER presents

When all of the parameters are correct the operator pushes ACCEPT to teach that program step. The motion type ABC consists of three motion descriptors. These determine whether the motion will be

1. Point-to-point or continuous-path (PTP/CP)
2. Joint-interpolated or straight line (JI/ST)
3. Normal or touch-terminated (TOUCH)

signal is true. This feature allows the robot to move in a direction until an external signal is generated, such as by a touch sensor. The three motion descriptors are selected using the appropriate function. For example, a move step directing the robot to move to point 342, in a straight line, as a continuous-path motion, with a normal termination at speed 75 would appear on the teach pendant display as

MOVE PT 342 SPD 75 SCN

he wishes to teach the point.

MOVE is only one type of motion. Other types may be selected by pressing the REPLACE function, followed by the MOTION function. Figure 9A.2 illustrates the MOTION function and the resulting menu structure. The rest of

Fig. 9A.2 *Motion function menu.*

GRIPPER is used to teach a gripper operation step.

menu which allows choices such as unconditional subroutine calls, conditional subroutine calls (IF, THEN, ELSE), and loops.

OFFMOVE is used to move the joints or the tool relative to their present

APP is used to teach an approach to a point.

INP DLY is used for teaching various responses to input signals, such as

PLT CMD selects the special functions for operating on pallets.

SET REG allows the register operations such as setting, incrementing, and decrementing register values.

INSERT allows the insertion of steps between other steps.

DELETE is used to delete steps.

COPY allows the copying of steps.

CONTROL allows the operator to set certain system control parameters.

9A.5 TEACHING PALLETS

array is a regular geometric pattern and sometimes it is a random collection of points. The MAKER system allows some special functions for dealing with pallets. In this case we will consider only 'regular' pallets, that is, pallets whose points are arranged in a regular geometric pattern.

Consider a rectangular array of parts arranged in a 5 ×
the robot to unload the parts in order from the array and to load them into a single position, perhaps a machine tool. Rather than consider the intricacies of the loading

us is to unload the parts in the pallet and to place them in the machine by calling subroutine 43.

the pallet using the special features available. From the main TCH menu select the function PALLET. The display requests the pallet number (let us call this pallet 67) and then displays

67 TCH SELECT PALLET TYPE

and the functions read

REG PLT IRR PLT

Since we are dealing with an array of parts we will select a REG PLT (regular pallet). The display will now read

67 TCH REG PLT DIM1 ⟨x⟩ DIM2⟨y⟩ DIM3⟨z⟩

and the functions display

ORIGIN DIM1 DIM2 DIM3 OTHER

Pressing OTHER causes the functions to display

We will describe each function in sequence.

begin unloading the pallet. It must also be the intersection point of the pallet axes, in our case at one of the corners.

dimensional pallet).

Using the DIMn function causes the display to read

REG PLT DIM ⟨n⟩ SIZE ⟨x⟩ IND ⟨q⟩

and the functions to display

ACCEPT SIZE END POS MOVE END INDEX

END POS
moved to that point or the point can be (COPY) copied. MOVE END will move the arm to the endpoint if it has already been taught. INDEX is used to set the number

COUNT is used to set the pallet counter. When unloading a pallet it is set to the number of points in the array (in our case, 30), when loading it is set to 0. It may be set to some other number if a pallet must begin in the middle of a run.

GROUP is used to assign the pallet points to a group number.

CONTROL is used to set control parameters of the system.

Having taught the points in the array as a pallet, there are special motion functions available for operating on that pallet.

Selecting PLT CMD from the REPLACE menu will cause the display to read

SELECT PALLET STEP TYPE

and the functions will display

PMOVE PAPP PUT TAKE RESET

counter.

PUT adds 1 to the value of the pallet counter.

TAKE subtracts 1 from the pallet counter.

RESET resets the pallet counter to the empty or full value (as selected by the operator).

By comparing the value of the pallet counter to a value in a register we can tell if a pallet operation is complete and can stop or start again on a fresh pallet. Typically this would be done by comparing the pallet counter to a value in a register and using the IF-THEN-ELSE command.

9A.6 SUMMARY

The MAKER provides the same types of programming commands as many other systems but in a different environment. Rather than presenting the system as a programming language such as VAL and AML, the MAKER makes extensive use of menus in an attempt to simplify the man-machine interface.

References

1. *MAKER Robot System Operation Manual*, United States Robots, Inc., King of Prussia, PA, 1984.

Appendix

9B

Val II

This appendix presents a summary of the VAL II robot programming language, developed by Unimation, Inc., and used for its industrial robots. VAL II is an enhanced version of the VAL robot programming language, released by Unimation in 1979 for its PUMA series industrial robots. The appendix is based on the *User's Guide to VAL II*,[1] copyrighted by Unimation, Inc. Major portions of the appendix are reprinted from the manual by permission of Unimation Inc. Unimation makes no representation regarding the accuracy of the information presented in this appendix, and is not responsible for results obtained in using the information.

It is not the purpose of this appendix to provide a complete description of the VAL II programming language. Readers should not attempt to accomplish any actual programming of Unimation robot systems using this appendix as the sole reference. The available user manuals and other operating instructions should be consulted before progamming is attempted in VAL II. The objective of this summary is to acquaint the reader with some of the basic concepts and statements of the language, in order to build a general understanding of robot textual languages.

9B.1 GENERAL DESCRIPTION

VAL II is a computer-based control system and language designed for the Unimation industrial robots. It provides the capability to easily define the task a robot is to perform, since the tasks are defined by user-written programs. Other benefits include the ability to respond to information from sensor systems such as machine vision, improved performance in terms of arm trajectory generation, and working in unpredictable situations or using reference frames.

9B.2 MONITOR COMMANDS

The VAL II monitor mode is used for functions such as defining point locations in the work volume, editing programs, executing programs, calibrating the robot arm, and similar purposes. We will sample some of the commands in this section.

current location of the robot. In the following commands all distances and coordinate

command. The operator uses the teach pendant to move the robot manipulator to the

current robot arm location. A related command is

This command queries the system to display the current location of the robot in Cartesian world coordinates and wrist joint variables. It also displays the current gripper opening if the robot is equipped with a position-servoed hand.

is automatically assigned a new name. The assigned name is derived from the name

third is P3, and so forth. It is possible to teach a complete motion path by successively positioning the robot using powered leadthrough with the teach pendant and pressing the record button.

the command

sets the value of PA (a location variable) equal to the value of P1.

Commands for entering and exiting the program editing mode are the following (shown at the beginning and end of a program)

.
.
.
.

mode. A variety of commands are available during editing for deleting programming commands, inserting commands, and so on.

in the VAL II monitor mode is given prior to program execution in the following way

The command

Additional monitor commands are available in VAL II for storing programs onto

example,

causes the robot to carry out the particular programming instruction indicated. We will give examples of the use of this command in the following section.

9B.3 MOTION COMMANDS

VAL II. The basic instruction is

which causes the robot arm to move by a joint-interpolated motion to the point P1. A related motion command is

which causes the movement to be along a straight line path from the current location to the point P1.

VAL II has approach and depart commands which are written as follows

but offset from the point along the tool z
command moves the tool to point P
z axis. In this sequence, the tool z axis direction

statements, respectively.

command. The command applies to all subsequent motions until the speed is again

is to specify a numerical value without a units argument.

command, as explained above. The alternative way is to specify

which indicates that the subsequent motion commands are to be executed at a speed

A single joint can be changed by a certain amount with the statement

speed.

Another command in VAL II used for motion control is to align the tool or end effector for subsequent moves. The statement is simply

This causes the tool to be rotated so that its z axis is aligned parallel to the nearest axis of the world coordinate system. It is useful for lining up the tool before a series of locations are taught in which it is important that the tool be properly oriented. This

9B.4 HAND CONTROL

VAL II has provisions for controlling a pneumatically operated gripper or an electrically driven servoed hand. To operate the latter, certain conditions must be these commands cause the gripper to assume fully open and fully closed positions.

next motion.

.
.
.

of the robot, while the second statement causes the gripper to close immediately to

example,

provides a convenient method of grasping an object and testing to make sure that contact has been made.

The control of the gripper and simultaneous movement of the robot arm can be accomplished with a single statement rather than two separate commands as described above. The required statement is

This command generates a joint-interpolated motion from the previous position

corresponding straight line motion command is

These statements assume a servocontrolled gripper capable of responding to the

the value is otherwise.

9B.5 CONFIGURATION CONTROL

one or more of the commands in this section is to permit the robot to achieve a

respectively. The statements

9B.6 INTERI, OCK COMMANDS

The output and input signals of the robot controller can be determined in the VAL

possible commands are explained in this section.

known state (off).

signal numbers turn the corresponding signal on and negative numbers turn the

The WAIT command can be used to control program execution based on input

program execution due to external signals. The basic command is

variable VAR1 and looks for a transition of the signal from its current state to the alternative state. If the variable value is positive (as in the example above), VAL II

VAL II looks for a transition from on to off. The reactions are triggered by signal

for a transition from off to on as in the above command, and the signal is already on, the reaction does not occur until the signal goes off and then on again. Another

lower or equal priority. When a program control is returned to the previous location

invoked.

transition is detected, the reaction is queued and is not invoked until program priority has been reduced. Accordingly, depending on the relative priorities, there can be a considerable delay between when the signal transition is noticed by VAL II and when the corresponding subroutine is executed.

is invoked.

9B.7 INPUT/OUTPUT CONTROLS

Instructions are available in VAL II for communicating information to the operator and

used to obtain data from the operator. The construction of the command is

which the system waits for the operator to respond by typing in the value requested

Program execution then resumes.

statement, program execution normally waits for the output to be completed before

VAL II contains provisions for communicating with other digital devices
with

output an analog voltage signal from either of two channels provided with the analog

The analog output voltage is proportional to the value on the right-hand side in

9B.8 PROGRAM CONTROL

The VAL II language contains a number of instruction sequences that can be used to since the logical condition is tested after the statements or group of statements are executed.

among any number of groups. The following example illustrates the use of the

The program asks the operator to enter a test value. If the value is negative,

numbers from one to nine, and (3) all other positive numbers.

9B.9 SUMMARY

Additional statements are available in the VAL II language. The purpose of this appendix is to provide a sampling of the important features and statements that are [1] available from Unimation, Inc.

1. Unimatipn, Inc., a Westinghouse Company, *Programming Manual—User's Guide to VAL II*

Appendix

9C

Rail

This appendix presents a summary of the RAIL robot programming language, developed by Automatix, Inc., and used for its Robovision and Cybervision systems. The appendix is based on the *RAIL Software Reference Manual*,[1] copyrighted by Automatix, Inc. Major portions of the appendix are reprinted from the manual by permission of Automatix.

It is not the purpose of this appendix to provide a complete description of the RAIL language. Readers should not attempt to accomplish any actual programming of Automatix robot systems using this appendix as the sole reference. The available User Manuals and other operating instructions should he consulted before programming is attempted in RAIL. The objective of this summary is to acquaint the reader with some of the basic concepts and statements of the language, in order to build a general understanding of robot textual languages.

9C.l GENERAL DESCRIPTION

RAIL is Automatix's language for computer-aided manufacturing. It is designed to control the company's robovision, cybervision, and autovision systems, and general manufacturing equipment. Robovision is a system for robotic arc welding. There are special-purpose RAIL commands for interfacing between the robot and the welding equipment. Cybervision is the system for assembly. RAIL contains commands for controlling the robot in an assembly operation, and for interfacing with such external devices as parts feeders, conveyors, sensors, and material-handling equipment. RAIL also contains commands for interfacing with autovision (machine vision) systems which are designed for inspection and identification in a manufacturing operation. This appendix describes some of the features specific to robovision and cybervision systems.

9C.2 LANGUAGE FEATURES

The RAIL language provides the following robot programming features, most of which have been described in general terms in the text of Chap. 9:

1. Commands for moving the robot, including approaching or departing from locations.
2. Robot welding commands, and facilities for setting welding parameters, such as voltage and wire feed rate.

A simple method to access input or output lines connected to equipment such as

variables or programs, for editing programs, and for storing programs.

Data types, including integers, real numbers, character strings, logical data, arrays, points, paths, and reference frames.

User-selected variable names (up to 20 characters long) for referencing I/O channels, locations, weld paths, and so on.

Program control structures similar to those provided by the Pascal language, including IF..THEN..ELSE, WHILE..DO, WAIT, and other functions that can be inserted in an application program to make logical decisions at run-time, to change the sequence of execution of instructions, or to repeat the execution of any instructions.

Arithmetic, comparative, and logical expressions.

A built-in function library, including functions for square root, sine, cosine, arc-tangent, absolute value, time-of-day, and interval timing.

9C.3 LOCATIONS

The RAIL language has three types of data for robot locations: points, paths, and reference frames.

reference frame. The world space frame is commonly located at the robot base. The

P1 = (300.0, 200.0, 100.0, 50.0, 25.0, 35.0)

indicating the x, y, and z coordinates of the tool tip in the world space. The last three values are orientation angles, expressed in degrees, indicating the orientation of the tool tip relative to the world reference frame.

A path is a connected series of points. When the robot is commanded to move along a path in RAIL, it will move smoothly through each point in the path without stopping

SEAM1 = PATH (P1, P2, P3, P4)

it. For example, if a pallet has several holes within it (for part insertion), the holes can

location of the pallet in space, and not the location of each hole in the pallet. A frame
x axis, and a point lying
in the xy plane.

9C.4 ROBOT MOTION STATEMENTS

There are a variety of statements to control the motion of the robot in the RAIL language. The basic command is the MOVE statement. RAIL supports three types of

below), the user program can select the type of motion to be used for a given motion statement.

Straight line motion of the tool tip is accomplished by means of linear interpolation. This motion is the default condition in RAIL; if the user does not specify one of the other motion types, the robot will move in a straight line motion. The command for a straight line motion is

MOVE P1

This corresponds to joint-interpolated movement of the robot arm. The arm is moved from point to point using the least possible motion for each joint, and in a coordinated fashion so that all joints start and stop moving at the same time. The advantage of slew motion is that it usually permits faster execution of a move. The command for a

MOVE SLEW P1

along a circular arc formed by the points indicated in the path. The key word CIRCLE is used to specify a circular interpolation in a motion statement as follows:

MOVE CIRCLE PATH (P1, P2, P3, P4)

straight line motion. It will then move from point P1 to P3 using a path consisting of a circular arc through P2. For paths containing more than three points (as in the example statement above), a new circular path would be followed between points P3 combined with the next destination point in the sequence.

Other types of robot motion statements in RAIL include approach and depart statements, rotation of the gripper, actuation of the gripper, and actuation of a welding cycle.

These commands (similar to commands explained in the text of Chap. 9) permit the

z axis. The commands are

APPROACH 50 FROM Pl

.

.

.

DEPART 50

this case) from the point P1 along the tool *z* axis. The last statement causes the tool to depart from the point by 50 mm.

the roll axis. For example, the command

ROTATE HAND 180

from its current position.

These statements command the opening and closing of a gripper-type end effector. The RAIL statements are

OPEN and CLOSE

This is a RAIL command that permits a welding operation to be carried out at a point along a path. The motion is executed by the special WELD command, explained in Sec. 9C.6.

9C.5 LEARN STATEMENT

teach pendant. The statement can be used to program points in world space or relative statement

LEARN P1, P2, P3, P4

the four points are learned by moving the arm to the desired locations using the motion buttons on the teach pendant, and pressing the record button to store each point location in the proper sequence.

Paths are learned by depressing the path button on the teach pendant to indicate the start of a path, and then moving the robot through the desired path by means of the teach pendant, depressing the record button at each successive point in the path. Depressing the path button a second time signals the end of the learn routine.

Frames are learned by means of the statement

LEARN FRAME COORD1

The LEARN FRAME statement prompts the user to teach three locations. These locations correspond to the origin of the frame, a point along the *x* axis, and a point in the *xy* plane of the frame coordinate system.

9C.6 WELDING

RAIL provides a number of features for controlling a welding robot. These features include the capability to do the following: move the robot along a path while controlling the welding process parameters; stepping the robot through the program execution, with the option of adjusting variables or process parameters at each step; and modifying the welding parameters using the teach pendant or the alphanumeric keyboard.

The basic weld command in RAIL is the statement WELD, followed by an

used. The statement has the following format:

WELD SEAM1 WITH WELDSCHED[3]

WELDSCHED[3, WIREFEED) = 50
WELDSCHED[3, VOLTAGE) = 18
WELDSCHED[3, PREDWELL) = 1.5
WELDSCHED[3, FILLDELAY) = 0
WELDSCHED[3, PREFLOW) = 1.0
WELDSCHED[3, POSTFLOW) = 0
WELDSCHED[3, WEAVE) = OFF

RAIL contains provisions for controlling weave patterns in arc welding. The

and cycle time (seconds), as well as weave delay values. Each of these parameters is

9C.7 INPUT/OUTPUT

The controller unit used with the Automatix robots is the AI 32. This controller is equipped with the following I/O devices to control or communicate with other equipment:

48 solid state relays for I/O

4 digital potentiometers for output

16 12-bit A/D converters for input (optional)

RAIL statements can be used to associate user-selected variable names to interlock the controller with external equipment through these I/O devices. The user can then read from or write to the external equipment by using the variable name in the program. To declare an input or output port, the following type of statement is used:

INPUT PORT SWITCH1 7

input port number 7. The status of that variable can be checked in the program as

OUTPUT PORT MOTOR3 15

15. During the program this output can be turned on or off by means of the statements

MOTOR3 = ON
MOTOR3 = OFF

value 1 in RAIL). The second statement assigns the value of output port 15 to be off (which has the value of 0 in RAIL). In effect these two statements close and open the relay for port 15.

There are two WAIT commands in RAIL, one of which is used in conjunction with an interlock condition. This is the WAIT UNTIL command. It is used to synchronize the execution of a RAIL program with some condition. The test condition must be some relational expression, such as one that tests the status of an input port variable. When the WAIT UNTIL statement is encountered, program execution is stopped

WAIT UNTIL MOTOR3 = ON
WAIT UNTIL CONSPEED > 5.0

would require program execution to stop until the two input channel conditions (MOTOR3 = ON and CONSPEED > 5.0) are both true.

(MSEC). If not specified, the time units are assumed to be seconds. The following statements illustrate the WAIT command

```
WAIT 2 SEC
WAIT 2
WAIT 2000 MSEC
```

Each of the above three statements would result in an identical time delay.

9C.8 OPERATOR I/O AND FILE SYSTEM

The RAIL language has a variety of commands for the programmer or operator to communicate with the robot cell. The standard output device used with RAIL is the CRT, and the teach pendant used with RAIL is called the interactive command module, or ICM. This section will sample the sections of the RAIL manual pertaining to these types of instructions.

The READ statement reads data supplied from outside the program, and assigns the data values to RAIL variables in the program. There are two forms of READ statement: READ and READS. The READ statement is used to read numerical input from the operator. When a READ statement is encountered, execution of the program is halted until the operator has entered a number for each variable in the variable list. The numbers must be integers or real numbers. The READS statement is used to read character string input (i.e., text containing any ASCII character, upper or lower case alphabetic characters, numbers, punctuation, etc.) from the operator. The operator must enter a line of text for each variable in the variable list.

The WRITE statement takes RAIL expressions (e.g., text) in the program, formats them into an output line, and prints them externally. There are two forms of WRITE statement: WRITE and WRITEICM. The WRITE statement is used to write messages to the operator on the CRT. Each WRITE statement prints out a single line on the CRT. The following example will illustrate the READS, READ, and WRITE statements:

```
WRITE (‘Enter Part Name:’)
READS (PARTNAME)
WRITE (‘Enter part length:’)
READ (LENGTH)
```

The first WRITE statement causes the message to be printed on the CRT, and the READS statement causes the operator’s input to be stored in the program as the variable PARTNAME. The second WRITE statement requests the operator to input a numerical value which will be used in the program as the value of the variable called LENGTH.

The WRITEICM statement is a RAIL function that writes a character string to the ICM display. For example,

```
WRITEICM (‘OVER TEMP CONDITION’)
```

will print the message out to the ICM rather than the CRT. The limitation on the message is that it must be no longer than 24 characters in length, which is the length of the display on the interactive command module. If the character string is longer

cartridge tapes are available in the Automatix robot controller to store RAIL system

The SAVE command is used to save the current value of RAIL variables (and/

statement

SAVE 'PARTFILE' PARTCOUNT, GOODPARTS, BADPARTS

provides a means in the program to save the current values of the variables (PARTCOUNT, GOODPARTS, BADPARTS) in the variable list on tape storage. The

can be retrieved at any time using the LOAD statement. The statement

LOAD 'PARTFILE'

In addition to the SAVE and LOAD commands, there is another command in

When the FILER command is entered, the screen is cleared, and the following line is printed out at the top of the screen:

Filer: L)ist, D)elete, P)rint, M)ake, V)olume, I)nit tape, C)opy, Q)uit?

The line shows all of the available FILER commands. To run any of the commands,

names of the tapes currently in the two tape drives of the system. The I)nit tape

uit command is used to quit the FILER and return to RAIL.

9C.9 PROGRAM CONTROL

The RAIL language contains an assortment of the typical program control statements such as IF…THEN, IF…THEN…ELSE, and others.

The WHILE…DO statement permits the programmer to continue executing a

example illustrates its use:

```
WHILE CYCLESTOP = OFF DO
BEGIN
MOVE SLEW HOME
WAIT UNTIL SWITCH1 = ON
```

```
CLAMP = ON
APPROACH 50 FROM SEAM1
WELD SEAM1 WITH WELDSCHED[3]
DEPART 50
CLAMP = OFF
END
```

This example makes use of several statements discussed previously in the appendix. The keywords BEGIN and END are used here to enclose a compound statement (a group of statements to be executed in sequence).

9C.10 SUMMARY

Additional statements are available in the RAIL language. The purpose of this appendix is to provide a sampling of the important features and statements that are contained in RAIL. The interested reader is referred to the user's manual[1] available from Automatix, Inc.

References

1. Automatix Inc., *RAIL Software Reference Manual*, Document No. MN-RB-07, Rev. 5.00, October 1983.

Appendix

9D

Aml[1]

This appendix presents a summary of the AML robot programming language, developed by IBM Corporation, and used for its IBM 7565 Manufacturing System. The appendix is based on two AML publications cited in Refs. 1 and 2, copyrighted by IBM Corporation. Major portions of the appendix are reprinted from these manuals by permission of IBM.

It is not the purpose of this appendix to provide a complete description of the AML programming language. Readers should not attempt to accomplish any actual programming of IBM robot systems using this appendix as the sole reference. The available user manuals and other operating instructions should be consulted before programming is attempted in AML. The objective of this summary is to acquaint the reader with some of the basic concepts and statements of the language, in order to build a general understanding of robot textual languages.

9D.1 GENERAL DESCRIPTION

A Manufacturing Language (AML) is a high-level, interactive programming language designed for robotic programming. The language provides a variety of data types and operators, system subroutines, language control structures, display station and file I/O controls, interactive editing and debugging facilities, and system identifiers to help improve program readability.

AML is subroutine oriented. Programs are written by structuring the application as a set of calls to both user-written and system subroutines. The system makes no distinction between user-written subroutines and system subroutines, and the system treats both types similarly when they are called. Consistent with our terminology in the text and in the other appendixes, we shall refer to the AML system subroutines as commands.

---

[1] Portions of this appendix have been reprinted by permission from *A Manufacturing Language Concepts and User's Guide* (850912), © 1982, and *A Manufacturing Language Reference* (850915), © 1983, by International Business Machines Corporation.

9D.2 AML STATEMENTS

An AML statement is analogous to an English sentence. Its meaning is interpreted by first interpreting the words and expressions in it. There are three statement forms in the AML language:

1. Executable statement
2. Variable declaration statement
3. Subroutine declaration statement

With the exception of certain special cases, all AML statements end in a semicolon (;).

9D.2.1 Executable Statement

Executable statements perform calculations, comparisons, and other similar functions. They constitute the logic which the interpreter is to execute. Executable statements differ from the other two types because they do not reserve storage or provide names for variables or subroutines. An executable statement has the following form:

label: expression;

The statement consists of an optional label and the expression to be interpreted. A semicolon ends the statement. The label is separated from the expression by a colon. An example of an executable statement is

CALC: 5–1/2*10;

CALC is the user-selected label, and the expression consists of a simple arithmetic computation.

Compound executable statements are a sequence of statements contained on several lines. They are denoted by BEGIN…END words which enclose the sequence.

9D.2.2 Variable Declaration Statement

Variable declaration statements have the following form:

id: NEW expression;

or

id: STATIC expression;

The components of the variable declaration statement are the following: the name of the variable (called the 'id' in AML), a keyword (either NEW or STATIC) which identifies the statement as a declaration statement, and an expression which the interpreter evaluates to determine the variable type and initial value. A semicolon ends the statement. Two examples of variable declaration statements are given below:

X1: NEW 12.0;
X2: NEW (2.5*X1);

The first statement declares the variable X1 to have an initial value of 12.0. The second statement declares the variable X2 to be given by an expression which evaluates to the product of 2.5 times the current value of X1.

The keywords NEW and STATIC are used by the interpreter to determine how to treat a declared variable. For variables outside of a subroutine, there is no difference in the way the keyword is interpreted. The difference applies only for their use in a subroutine. NEW variables are interpreted each time the subroutine within which they reside is called. The interpreter interprets the expression on the right side of the colon every time the subroutine is called, and storage is reserved for the new variable each time. When the subroutine execution is completed, and control is returned to the main program, the storage for the new variable is cleared. STATIC variable declarations are only interpreted once after they are declared, and this occurs at the

executing. STATIC variables retain their storage and the values stored there even when the program is not executing. The last value saved for any STATIC variable will be available for further use in calculations in the main program.

9D.2.3 Subroutine Declaration Statement

A subroutine consists of a collection of AML statements. The subroutine declaration statements serve to reserve space in storage for the subroutine, and names the subroutine for later reference in the program. Subroutine declaration statements have the following form:

```
subrstatement1;
subrstatement2;
        .
        .
        .
subrstatementn;
END;
```

The components of the subroutine declaration are the following:

subroutine in AML. Parameters (pl, p2, p3, …pn) which are transferred from the

parenthesis.

The body of the subroutine, consisting of AML statements each ending with a semicolon:

```
subrstatement1;
subrstatement2;
        .
        .
        .
subrstatementn;
```

These statements have the same form and follow the same rules as other AML statements. The semicolon after the last statement (subrstatementn) and before the keyword END is optional.

The keyword END, indicating the end of the subroutine.

A semicolon follows the END word. Special rules regarding the semicolon apply for the END statement for subroutines nested within other subroutines.

9D.3 CONSTANTS AND VARIABLES

AML uses a wide assortment of constant and variable types. Constants and variables can be integer, real, or string. Aggregate constants and variables are also possible in AML, including aggregates containing a combination of real, integer, and string elements. Aggregates are enclosed within angle brackets, 〈 〉, and the elements are separated by commas.

AML permits the definition of two-dimensional aggregates in addition to the conventional one-dimensional aggregates. A two-dimensional aggregate contains both rows and columns of elements. A typical use of a two-dimensional aggregate would be to define an array of coordinates in a robot workspace. An example given in Ref. 1 involves the definition of six points (x, y, and z coordinates) for the IBM 7565 robot. The variable declaration statement is

R: NEW 〈〈–6.0, –25.0, 3.1〉, 〈–6.0, –2.2, 3.5〉,
〈1.3, .5, 3.2〉, 〈5.2, –25.1, 3.0〉,
〈8.1, –2.2, 3.3〉, 〈8.4, 15.5, 3.5〉〉;

The individual elements of the two-dimensional aggregate are contained within angle brackets and can be more than two dimensions. Hence, in this example, a two-dimensional aggregate is used to represent a three-dimensional array. The value of R can be depicted graphically for the six points as shown in Table 9D.l. The individual elements of the aggregate can be accessed in

Table 9D.1 *Coordinate values corresponding to the variable declaration of a series of points in space.*

Declaration statement			
R: NEW 〈〈–6.0, –25.0, 3.1〉, 〈–6.0, –2.2, 3.5〉, 〈1.3, .5, 3.2〉, 〈5.2, –25.1, 3.0〉, 〈8.1, –2.2, 3.3〉, 〈8.4, 15.5, 3.5〉〉;			
Corresponding coordinate values			
R	X	Y	Z
2	–6.0	–25.0	3.1
2	–6.0	–2.2	3.5
3	1.3	.5	3.2
4	5.2	–25.1	3.0
5	8.1	–2.2	3.3
6	8.4	15.5	3.5

AML by means of the symbol R(1), R(2), and so on. In other words, R(4) represents 〈5.2, –25.1, 3.0〉. This might be useful for idenditifying a point in space to which the robot would be commanded to move.

9D.4 PROGRAM CONTROL STATEMENTS

The AML language contains a number of program control statement for conditional execution of the program. These conditional expressions include IF…THEN, IF…

and IF…THEN…ELSE statement have the conventional interpretation.

The WHILE…DO statement allows the programmer too repetitively execute an

WHILE condition DO expression

An example of its use is the following:

```
X: NEW 1;
WHILE X LE 3 DO
      BEGIN
             DISPLAY (X, EOL);
             X = X + 1;
             END
DISPLAY ('DONE', EOL);
END;
```

The WHILE ...DO statement controls the iteration process. A compound statement is contained within the BEGIN...END words. The EOL word contained in the DISPLAY statements indicate 'end-of-line' to the device, corresponding to a carriage

the following output

```
1
2
3
DONE
```

that the order of condition and expression is reversed. Its basic form is therefore

For example, a subroutine similar to the above, but written using the REPEAT…

```
X: NEW 1;
REPEAT DISPLAY (X, EOL);

DISPLAY ('DONE', EOL);
END;
```

9D.5 MOTION COMMANDS

The AML language is designed for the IBM 7565 robot, which is a cartesian coordinate

9D.2 provides a listing of these joints together with information about their limits of travel. Joints 1, 2, and 3 are linear, while joints 4, 5, and 6 are rotational. The gripper

Fig. 9D.l

(Courtesy: International Business Machines Corporation)

Table 9D.2 *Joint details for the IBM 7565 Robot system.*

Number	*Joint name*	*Lower limit*	*Upper limit*
1	JX	–8.9 in.	+8.9 in.
2	JY	–29.4 in.	29.4 in.
3	JZ	–8.75 in.	8.75 in.
4	JR	–135°	+135°
5	JP	–90°	90°
6	JW	–135°	135°
7	JG	0 in.	3.25 in.

coordinates in the workspace to move to, or by specifying the incremental distance to be moved. The two associated commands are MOVE and DMOVE. For example, the command

MOVE(⟨JX, JY⟩, ⟨8.5, 1.5⟩);

indicates that the two joints JX and JY should be moved to their corresponding coordinate values of 8.5 and 1.5 in., respectively. It is also possible to specify the

statement

MOVE(⟨JX, JY, JZ⟩, R(1));

where R(1) might refer to point number 1 in the array presented in Table 9D.1.

The DMOVE command is used to move relative to the current position. For example, the statement

DMOVE(JZ, 3.0);

would move the robot wrist 3 in. up in a vertical direction.

The difference is that the MOVE statement completes the motion before proceeding with any subsequent statement(s) in the program. With AMOVE, the system can perform other work while the move is completing, such as data processing functions or calculations. Obviously, any subsequent moves must await completion of the current move. The format of the AMOVE statement is the same as the MOVE statement

AMOVE (⟨JX, JY⟩, ⟨8.5, 1.5⟩);

set of joints under powered leadthrough control. The IBM 7565 teach pendant has control buttons corresponding to particular joints and directions. For each joint, there is a positive and a negative direction button that can be depressed to achieve the desired motion for the joint. The longer the button is depressed, the faster the joint moves. When the button is released, the joint motion stops. Once the robot has been moved to the desired position, the 'End' button on the teach pendant is pressed and the set of axis positions is displayed on the monitor, and control is returned to the AML interpreter. For example, consider the following command:

⟨JX, JY, JZ⟩);

This would enable the x, y, and z axes for motion through the teach pendant. The user could then move those axes with the teach pendant buttons. At the completion of the manipulation, the user depresses the 'End' button, and the following type of information is displayed on the monitor

⟨2.03700.27.6000, 4.00300⟩

The execution speed of a programmed move is set with the SPEED command. The statement

SPEED(.5)

speed. Valid speeds range from greater than 0.0 to 1.0 of full speed of the robot.

9D.6 SENSOR COMMANDS

The IBM 7565 robot system provides a sensor I/O interface to connect with various

The system accommodates 1-bit digital input or output, as well as multiple-bit I/O

robot is equipped with a sensored gripper, additional sensor commands are possible to interface with its sensors.

9D.6.1 Sensor Input/Output

Three of the important commands in AML for sensor I/O are DEFIO, SENSIO, and

can be accessed by SENSIO or MONITOR. The form of the DEFIO command is

DEFIO(group, type, format, sbit, length, scale, offset)

are optional. The group is an integer or aggregate specifying the I/O channel. The type refers to the input or output device (0 for input and nonzero for output). The format is an integer or aggregate specifying the format of the sensor I/O data. A 1 means that a closed contact yields a –1 and an open contact yields a 0. A 2 means that the closed contact yields a 1 and the open contact value is again zero. The s–bit

Bit numbers range from 0 to 15, with bit 0 being the left-most bit. The length is an

16 bits.

The scale is an optional real number or aggregate specifying the scale factor that is to be used for scaling the I/O to make its size compatible with the system. The default value is one. Finally, the offset is an optional real number or aggregate specifying the constant to be used in offset the scaling. The default value is zero.

can be accessed by the SENSIO command. The return is referred to as the ionum.

output. The format of the SENSIO command is

SENSIO(ionum, template)

The ionum is the value (or values) that determines the input or output that is to be

10 I/Os to be executed from the same command. The template is an integer, real

used for output operations.

An example of the use of the DEFIO and SENSIO statements would be

```
DEFIO(21, 0, 1, 0, 1);
1016
SENSIO(1016, INT)
```

returns the ionum to be referenced by the SENSIO statement (1016). The INT is used to reference the values under the control of the AML program.

The MONITOR command allows the use of one or more of the I/O channels for reading of sensors at regular time intervals. This can be used to design an interrupt system for safety monitoring or other purposes. The MONITOR command differs from SENSIO since it permits quasicontinuous monitoring of incoming sensor

9D.6.2 Gripper-Sensing Capabilities

Additional sensing capabilities are available with the optional sensored gripper for the IBM 7565 system. These capabilities include force sensing with strain gauges and optical sensing with a light emitting diode. The gripper is shown in Fig. 9D.2, indicating the tactile sensing and optical sensing features.

Fig. 9D.2 *Sensored gripper used with IBM 7565 robot system: (a) Tactile sensing features, (b) Light emitting diode for optical sensing. (Courtesy: International Business Machines Corporation)*

The sensors can be used with SENSIO and MONITOR commands. The strain gauges are used to detect positive and negative forces in three directions as indicated in Fig. 9D.2. The strain gages corresponding to the three directions are

1. Tip sensors These indicate whether there is a force pressing up from the bottom of the gripper. This might occur, for instance, if the gripper is pressed against the worktable. It would also have a value when an object is being held in the gripper with

would cause a force to be registered in these sensors. The AML words SLT and SRT

MONITOR commands.

2. Pinch sensors If the gripper is closed on an object, these pinch sensors will indicate the pressure with positive measurement values. If the gripper is used to grasp the inside of a cup by opening against the sides of the cup, negative values will

SENSOR and MONITOR commands.

3. Side sensors If the gripper is moved against an obstacle from the side, these sensors will indicate the presence of the obstacle. The AML words SLS and SRS are

Two examples of the use of these gripper sensors will be used for illustration. The command

MONITOR(SLP,3, –1000.0, + 1000.0);

would cause the monitor to trip when the force exerted on the left pinch strain gauge deviates by 1000 grams from its current value. The command

SENSIO(⟨SLP, SLT, SRP, SRT, SLS, SRS⟩,6 of REAL);

would return the following aggregate which contains the force on each of the six

⟨236.432, 465.002, 21.5060, 17.8920, –15.0060, –23.7772⟩

These values are in grams of force.

The AML word LED can be used with the SENSIO and MONITOR commands to determine if an opaque object is blocking the light emitting diode. For example, the command

SENSOR(LED, INT)
–1

returns a 0 to indicate that no object is blocking the LED, whereas a value of –1 indicates that an object is blocking the light path. The word LED in the statement

9D.7 DATA PROCESSING

AML contains a number of data processing capabilities for interfacing with peripherals. The IBM 7565 robot includes the following data processing devices: display terminal (keyboard and CRT display), printer (optional), diskette, disk (optional), teach pendant (which incudes a 10-character display), and communications line. The communications line permits communication between different systems available for transferring data between the system and peripheral devices. Certain channels are reserved: channel 0 is the display terminal, channel 1 is the printer,

the system to the peripherals. These include the WRITE, PRINT, and DISPLAY

WRITE(0, 'Good Morning');

would write the word hello to the display terminal through channel 0. The maximum length of the record that is transmitted depends on the output channel. If a WRITE to channel 0 is issued (display terminal), the maximum length is 79 characters. If a WRITE to channel 1 is issued (printer), the maximum length is 132 characters. If a WRITE to channel 2 is issued (teach pendant display), the maximum length is 10 characters. If the data are longer than the maximum record length, an error message is returned.

The PRINT command writes the values of a list of expressions on a currently open data channel. PRINT is different from WRITE in that the data in the expression are converted to character format before being displayed, and in the fact that more than one record can be transmitted to the device. With WRITE, only one record is written to the device. For example, the following sequence accomplishes the same as the above:

X: NEW 'Good';
Y: NEW 'Morning';
PRINT(0,⟨X, Y, EOL⟩);

Another difference between PRINT and WRITE is that the output data is buffered until an EOL or EOP (end-of-print) character is detected, or the buffer is exceeded. Again, the channel number must be provided as one of the arguments in the command.

The DISPLAY command is the same as the PRINT directed to channel 0. It

DISPLAY('Good Morning', EOL);

The READ command is used to receive data from a peripheral device. It reads

format of the READ command:

READ(0, DATA1);

9D.8 SUMMARY

Additional statements are available in the AML language. The purpose of this appendix is to provide a sampling of the important features and statements that are contained in AML. The interested reader is referred to the user's manuals available from the IBM Corporation.[1,2]

References

1. IBM Corporation, A *Manufacturing Language Concepts and User's Guide*, Base Publication 850912, Boca Raton, FL, December 1982.
2. IBM Corporation, A *Manufacturing Language Reference*, Base Publication 850915, Boca Raton, FL, October 1983.
3. IBM Corporation, *A Manufacturing Language Screen Editor*, Base Publication 850916, Boca Raton, FL, December 1982.
4. Taylor, R. H., Summers, P. D., and Meyer, J. M., 'AML: A Manufacturing Language,' *The International Journal of Robotics Research* 1(3):19–41 (Fall 1982).

10 Artificial Intelligence

Introduction

Modern day robotics technology advanced through a number of stages that are sometimes called generations. The earliest robot was the 'master slave' telemanipulator developed to handle radioactive material during World War II. This was essentially a mechanical devise consisting of a master arm and an identical slave arm connected by wire rope and pulleys. The operator would move the master arm and the slave would follow the motion. This was a 'strictly' mechanical device. With the invention of the microprocessor and the development of NC technology, it became possible to reprogram mechanical machines. The early robots were simple pick and place devises with limited interaction with the environment. As computers got smaller, faster and cheaper, different areas like computer vision, interaction with the environment using force/touch sensors and the subject of Artificial Intelligence arose. Once robots ventured into outer space, it became inevitable that they have some intelligence to deal with 'uncertain or incomplete information'. Industrial manipulators soon followed with newer applications that required 'intelligence.' As an example, an AGV needed to move from one point to another without hitting an obstacle. An even more intelligent application is that of an AGV that has to find the goal point without any map of the area. Sometimes robots need to operate autonomously while some times they need to operate with a human in the loop. The general belief has been that 'humans' are better at making judgements, etc., or planning a task. However, in cases where exact information may not be available this may not be the case. For example a rover on mars may have better information about its surroundings than the human operators on earth. Hence, a new field of shared control has arisen, in which a human and a robot share the control. It is appropriate that we explore the subject of artificial intelligence and its opportunities in industrial robotics.

AI is a vast topic and this chapter will only serve as an introduction. Some of the goals of AI research and some of the general techniques used for attaining these goals will be discussed. We will then look at how these goals and techniques can be applied to robotics research. Finally, the programming language LISP which is uniquely suited for AI research will be introduced.

10.1 INTRODUCTION

this work is carried out as a branch of computer science, although it also contains elements of psychology, linguistics, and mathematics.

In the introduction to [1] the editors state:

10.2 GOALS OF AI RESEARCH

While the general objective of AI work is the development of machines which exhibit intelligent behavior, this statement is in itself too broad and ambiguous to be meaningful. The following paragraphs will describe some of the areas of AI which are presently being pursued as distinct areas of research.

1. Problem solving Examples of problem-solving systems are the chess-playing programs. Given a set of rules and strategies these systems are capable of playing

way to solving the problem of retrieving the paper the robot must identify and solve

robot must identify a possible path to reach the paper, then it must deal with obstacles along that path, such as opening the door. Finally, it must deal with grasping the paper and returning it. Almost all tasks with which we are confronted daily involve problem solving. Design of a system requires the breaking down of the problem into successively smaller problems until the solutions begin to become evident. At this point, we must 'show' robots all of the motions required to perform an assembly task. A robot endowed with a form of 'problem solver' may be able to develop the assembly strategy itself. We will discuss this example in greater detail in a later section.

The techniques of problem solving are the same whether developing a robotic assembly strategy solver or some other problem solver.

2. Natural language In spite of the proliferation of computers into department stores, many people are still uneasy or uncomfortable dealing with computers. This is due in part to the need to talk to a computer in its language rather than the user's 'natural' language. The problems facing natural language researchers are numerous. The computer must not only be able to understand the meanings of words, but how those meanings may differ in context with other words. The system must also be able to understand the syntax of the language so that the relationships of the words is

purely from a grammatical standpoint.

1

we associate with intelligence in human behavior–understanding language learning, reasoning, solving problems, and so on.

3. Expert systems This area of research is concerned with developing systems that

operator an expert system can recommend tests to be performed and ask appropriate questions until it arrives at a conclusion. At present, expert systems are being used to

of the issues involved with the design of expert systems include the problems of dealing with vast amounts of data, explaining the system's 'reasoning' on the way to reaching a conclusion, representing the data collected from the human experts, and improving the 'knowledge base' with experience.

4. Learning
experience. If machines were capable of learning then the task of endowing them

systems have been developed which have shown the ability to learn from experience, but to this date limited progress has been made.

5. Vision Most of the basic concepts employed in commercial vision systems are

permit the systems to perform scene analysis. That is, present the vision system with a scene and allow the system to identify objects within the scene.

Some of the other areas of AI research include: automatic programming, hardware development, and deductive reasoning.

6. Inference
with 'incomplete on uncertain informations.' The information may be incomplete, corrupted, or just noisy. As an example a vision system looking for a part may not have exact information due to varying lighting conditions the object being excluded, etc. There are several such examples such as, a dark region in a picture could be a shadow, or a hole in the surface. The robot will need to 'infer' in such cases based on other data. A dark region may not be a shadow if there are no objects near by in depending on the direction of light.

7. Search
representation of data. While playing a game there are several moves possible, but which is the best move?

10.3 AI TECHNIQUES

AI is concerned with the use of data or knowledge. Therefore techniques must be developed for two basic tasks: data representation and data manipulation. In this section, we will look at some of the approaches for representing data and using that

only the general techniques which might be employed by AI programs.

10.3.1 Knowledge Representation

When we discuss knowledge representation we are not concerned with the physical operation of the computer, that is, we are not discussing the storage of words as a series of Is and 0s. Rather we are discussing the relationships of facts with respect to each other, for example, the statement, 'Some birds have wings.'

The material in this section is summarized to a large extent from Ref. 1. For a more detailed account, the reader is directed to Ref. 1.

Before discussing the various representations of knowledge we must first describe the various types of knowledge which may require representation.

1. Objects More specifically facts about objects, such as 'robotics students drink heavily' or 'birds have wings.'

2. Events Not only the event itself, such as 'The robotics student broke his arm,' but perhaps the time or cause-effect relation of the event. 'The robotics student broke his arm yesterday and the nasty instructor made him pay for it.'

3. Performance If the AI system is one which is designed to control a robot then it must have data on the performance of the arm, that is, its kinematics, dynamics, what bits to manipulate in the hardware, and so on.

4. Metaknowledge This is the knowledge about our knowledge. This includes our knowledge of the origin of the information, its relative importance, its reliability, and so on. For example, one would give little weight to the following information 'The study of robotics is painless' if that information came from a history student.

At this point, let us look at some of the various techniques for representing knowledge.

1. Logic Formal logic was developed by mathematicians and philosophers to provide a method for making inferences from facts. Formal logic allows facts to be represented in a specific syntax, and by applying defined rules of inference to these facts, allows conclusions to be drawn. We are probably familiar with the following type of example:

Given the two statements

(a) All roboticists play games.

(b) Jack is a roboticist.

We can conclude by using a rule of inference that

(c) Jack plays games.

Conceivably, if a computer were endowed with all of the possible facts about a subject, and also all of the applicable rules of inference, then it should be able to develop any new facts about the subject that may be inferred from the original set. We will discuss later in the chapter the concept of 'combinatorial explosion' which may limit the usefulness of logic-based (or any other) representation schemes. As the number of facts become larger, the number of combinations of facts with the rules of inference which may apply also increases. This results in generating a problem too large for the computer to solve in a reasonable amount of time.

2. Procedural representations In the previous discussion, we examined a method for storing facts. It is also necessary to store information on how to use facts. In a logic-based system as above, every rule of inference must be applied to prove a new point. If it were possible to encode the information along with a way to use the computer could contain the following facts:

(a) Jack is a roboticist.
(b) All roboticists play games.

Additionally, the system could have stored the following procedure about Jack:

If need-prove plays ( )

show is-roboticist ( )

which states that if we need to prove some fact about , we can do so by showing that is a roboticist. With this information the system would be able to avoid any other information about Jack and jump instantly to the correct solution. This technique of storing information as a procedure is commonly used in computer programs in

3. Semantic networks Semantic networks are representations of information which consist of nodes and arcs. Nodes typically represent objects, concepts, or situations and the arcs represent the relationships between nodes. The nodes and links are labeled with simple language descriptions. Figure 10.1 illustrates the following information as a semantic network:

Fig. 10.1

Jack is a student.
Jack is a roboticist.
Roboticists play games.
Roboticists may be students.
Students may be roboticists.

Many inferences may be drawn about Jack, students, and roboticists by investigating the network.

Networks are used extensively in AI research as a means of knowledge representation. Unfortunately, they also have their drawbacks. They may be too simplistic; how does a network deal with ideas or large amounts of knowledge? As with all representation techniques, it cannot be pushed to an extreme.

4. Production systems Production systems store information in the form of items called 'productions.' Productions all have the form of IF some expression is true THEN some action. For example, the information we have about Jack and roboticists would be presented as:

(a) IF the person is a roboticist THEN he/she plays games.
(b) IF the person is reading this book THEN he/she is a roboticist.

If we were able to catch Jack reading this book then we would know that he is, in fact, a game player.

Production systems are used to break the problem down into successively more manageable tasks. They are typical of the construction of 'expert systems.' They provide for uniformly designed systems using the IF-THEN construction as well as a modularity, in that each rule has no direct effect on other rules. Unfortunately, among

5. Frames Another representation technique is the use of frames. Frames can be

following:

STUDENT Frame
Height: in inches
Weight: in pounds
Studies: Robotics or History
and for Jack the frame would look like
JACK Frame
Height: 70 inches
Weight: 150 pounds
Studies: Robotics

6. Other representation techniques In many cases choosing the correct representation technique can greatly simplify the problem. As an example let us investigate a

with each pixel representing a node and each arc representing the 'is-connected-to' property attached to each adjoining node. A simpler approach is to represent the picture as an array of values corresponding to the brightness of each individual pixel. The point is that the choice of a representation technique should be made only after giving consideration to the type of problem being solved. The above mentioned techniques are applied to different circumstances and in some cases different representation schemes should be found.

10.3.2 Problem Representation and Problem Solving

Before discussing problem-solving techniques, it will be useful to explain what is meant by 'problem solving.' Problem solving is the task of reaching some specified goal. Examples of these goals may be

1. Finding the proof to a mathematical theorem
2. Solving a puzzle, such as Rubik's cube
3. Determining a sequence of assembly steps
4. Choosing the next move in a chess game

In order to solve these problems, it is necessary to represent the problem in some way that is amenable to discussion and solution. Two possible schemes for problem representation are:

1. The state-space representation
2. The problem-reduction representation

1. State-space representation In this method we can visualize the problem as all of the possible states found in developing the solution configured as a tree. The tree is made of nodes, which represent the states of the system after certain actions have been taken. The actions are represented by the arcs that connect the nodes. This can be illustrated by using the 'traveling salesman' problem. A traveling salesman has to travel to four cities *A*, *B*, *C*, and *D*. The salesman wishes to travel to all four cities using the shortest possible path and by going to each city only once. He wishes to begin and end his trip at city *A*. The distances between the cities is given in Table 10.1. The state-space representation of the trip is shown in Fig. 10.2.

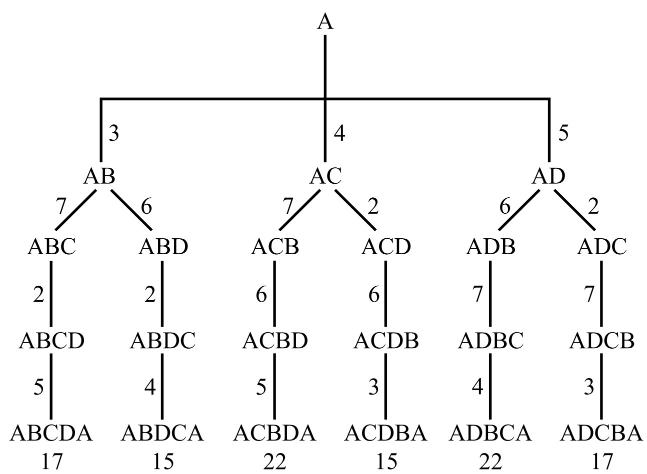


Fig. 10.2 *Tree for traveling salesman problem.*

Table 10.1 *Distance between cities.*

	A	*B*	*C*	*D*
A	0	3	4	5
B	3	0	7	6
C	4	7	0	2
D	5	6	2	0

By looking at the state-space representation, we can see that the problem becomes one of choosing the branch of the tree with the shortest sum of arc lengths. There are two paths which provide a solution of 15: *ABDCA* and *ACDBA*. Later in this section we will see how to search the state-space tree for this solution. This form of problem solving is called 'forward reasoning,' because we worked our way forward through all of the states until we found the solution.

2. Problem-reduction representation In this case, we can see an example of 'backward reasoning.' In the problem-reduction representation, we present the goal as the primary data item and then reduce the problem until we have a set of primitive problems; that is, simpler problems for which we have the data available. This simplification may involve breaking the problem down into a set or sets of smaller problems which must all be solved or into alternative problems, any one of which may be solved. The scheme is graphically represented as an 'and–or' graph. In an and–or graph arcs which are connected by a horizontal bar are 'anded' and arcs which are not connected are 'ored.' Figure 10.3 illustrates a simple and–or graph which states:

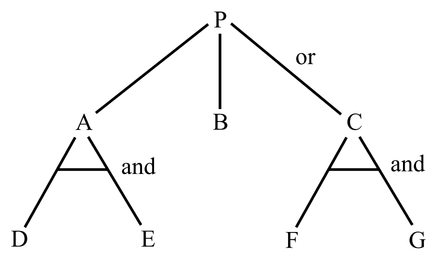


Fig. 10.3 *And-or graph.*

(a) *P* may be solved by solving *A* or *B* or *C*.

(b) *A* may be solved by solving *D* and *E*.

(c) *B* may be solved directly (*B* is a primitive).

(d) *C* may be solved by solving *F* and *G*.

(e) *D*, *E*, *F*, and *G* are primitives.

In this case, then, the simplest scheme for solving *A* is to solve *B*. In the next subsection, we will discuss the different techniques for searching for the solution.

10.3.3 Search Techniques in Problem Solving

Up to this point, we have considered techniques for representing knowledge and problems. If we employ the proper representation technique, then it is often only a matter of searching for a solution to reach the desired goal. In many cases, it is possible that the necessary search is too large to be successful in a reasonable amount of time. For example, consider the game of chess. Given that there is a known starting state and that the goal is to capture the opponent's king, it is possible to develop all of the possible move combinations in a state-space representation and then search for the best winning solution and follow that path. Unfortunately, the number of move combinations in a game of chess may be on the order of 10^{120}. Because of this type of 'combinatorial explosion,' techniques that seek to minimize the search effort have been developed. This does not mean that search is the only available technique. Just as the method of knowledge and problem representation must be carefully considered for each task, so should the solution technique. In any case some of the developed search techniques follow:

As the name implies, this search technique searches as deeply

In some systems, it is possible that the depth of the tree or certain branches of the tree are so deep that the solution is never found.

Fig. 10.4

The system evaluates all nodes at the same level in the tree before moving on to

and is not as easily trapped. In cases where the tree is relatively deep on all branches

Fig. 10.5

3. Hill climbing

Rather than moving in an arbitrary decision at each branch point, the hill-climbing algorithm attempts to make the best choice among the possible branches from the present node. This choice is based upon some selection technique. For example, in the traveling salesman problem, the next node chosen may be based on the total distance up to the next possible node. The risk here is that we are only looking at the next node, that is, at local information. While we may be making the best local choices, we may be missing a far better overall solution that had one bad arc.

This is a variation on hill climbing. In this case, rather than choosing the best next branch from a node, the system selects the best next node regardless of its position in the system. This generally provides the optimum solution, but does not guarantee it.

5. Branch and bound

next level (s). As the path ceases to be the optimum (it becomes too long) the system expands the new most promising path. In this way, it is always investigating the optimum path to the deepest level necessary. In order to ensure that the optimum path is found all partial paths must be extended until they become longer than the solution path, or until a shorter solution is found.

6. Constraints In some cases, it is not necessary to consider all of the possible options in developing the tree. Very often in the real world, constraints are placed upon information by its context. We know, for instance, that mice and cats are not likely to live in the same cage. Applying these constraints as we are traveling through

explored.

There are many other search techniques beyond those explained here but they are beyond the scope of this text. At this point it is worthwhile to review what has been covered so far.

Knowledge can be represented in a number of ways. Not only are the data important, but also the relation of data to other information and the rules for manipulating data are important. Problems may also be represented in a number of ways. Problems may be solved by forward or backward reasoning. By representing solutions as trees, optimal solutions may be found using search techniques.

10.4 LISP PROGRAMMING

The programming language traditionally used in AI research is called LISP, LISt Processing. LISP has many interesting features not commonly found in other programming languages. One of these is the fact that LISP data and 'LISP programs are both created out of lists.' Because data and programs are identical, LISP programs may modify or generate other LISP programs. We will introduce a number of LISP commands in this section so that simple programs may be developed.

LISP programs are lists constructed from elements called atoms. The following are lists:

(A, B, C, D)
(Item1 Item2)
((Iteml A)(ltem2 B))

The following are atoms:

A
B
Iteml
NIL

LISP attempts to evaluate all expressions and return the value of the evaluation. It uses prefix notation when evaluating the expressions. The fol1owing expression:

(PLUS 3 5)

returns

8.

LISP always tries to use the first element of a list as a command for evaluating the rest of the elements in the list which are the arguments. Since lists always take the form

(expression. expression).

where expression can itself be a list, we can write the fol1owing example:

(PLUS (PLUS 2 3)(PLUS 4 5))

LISP evaluates the two inner lists (expressions) first resulting in the equivalent of

(PLUS 5 9)

which returns

14.

The arithmetic functions of LISP are some of the easiest, so we will start with them. Some of the available functions can be demonstrated by examples. The value returned by LISP is to the right of the expression.

(PLUS 3 4)	7
(TIMES 5 7)	35
(DIFFERENCE 9 4)	5
(QUOTIENT 8 2)	4
(ADD1 8)	9
(SUB1 8)	7
(MAX 2 19 7)	19
(MIN 2197)	2
(EXPT 2 4)	16
(SQRT 9)	3

As we saw with the first example, the arguments do not have to be actual numbers. They may be other expressions.

Earlier we stated that one of the interesting features of LISP was that it was designed to manipulate lists. Two functions which are used for this purpose are CAR and CDR (pronounced could-er). CAR returns the first element in a list and CDR returns everything else. Again let us consider some examples.

(CAR ‘(A B C))	A
(CAR ‘((A B)(C D)))	(A B)
(CDR ‘(A B C))	(B C)
(CDR ‘((A B)(C D)))	((C D))

It should be noted that CAR may return an atom or a list and that CDR always returns a list. If CAR is asked to operate on an atom it will return an error. If CDR is asked to perform on a one-element list it will return NIL.

You may have noticed that we added some quote marks to the last set of examples. As stated earlier, LISP tries to evaluate expression by using the first element in each list as a function. The quote inhibits this evaluation so that LISP can tell the difference between the list and the function.

CAR and CDR can operate together in expressions. For example:

(CAR(CDR(CDR(CDR’(A BCD E)))))

returns

D

Of course, if we can take expressions apart there must be some way to put them back together. There are three functions that may be used for this purpose. They are APPEND, LIST, and CONS.

APPEND runs the elements of its lists together, for example,

(APPEND ‘(A B)’(C D)) (A BCD)

LIST makes a new list out of the arguments

(LIST ‘(A B)’(C D)) ((A B)(C D))

argument

If we always had to explicitly state the list we wished to manipulate it would

SETQ. SETQ is used to 'name' an expression. For example,

(CAR VW)   A

for this is DEFUN (DEFine FUNction). DEFUN uses the following format:

(DEFUN ⟨function name⟩

(⟨parameter 1⟩, ⟨parameter 2⟩ ... ⟨parameter n⟩)

⟨function description⟩)

A function to convert height in the form of feet and inches to just inches might look like

(PLUS (TIMES FT 12) IN)).

Typing in

would result in the response

70.

Earlier in the chapter, we saw that decisions have to be made in order to evaluate the success of certain operations. For this LISP provides predicates. Predicates are functions that provide only two possible responses: T (for true) and NIL. Some examples of predicates are:

(LESSP 2 3)	T
(LESSP 3 2)	NIL
(GREATERP 2 3)	NIL
(GREATERP 3 2)	T

which returns T if it has two arguments and they are equal.

is used with the following syntax:

⟨test1⟩ … ⟨resultl⟩)
(⟨test2⟩ … ⟨resultz⟩)
:
:
(⟨testn⟩ … (resultn⟩))

the last expression and returns that value.

Earlier in this chapter, we discussed information representation. In LISP, data are represented as lists and there are numerous commands available for manipulating

command with the following syntax:

(DEFSTRUCT(⟨name⟩)

(⟨property name 1⟩⟨default property⟩)

(⟨property name 2⟩⟨default property⟩)

(⟨property name n⟩⟨default property⟩))

Evaluation of DEFSTRUCT not only sets up the property list structure but also generates a function called MAKE-name. MAKE-name is used to assign values to a property list. DEFSTRUCT also generates 'selector' procedures called 'property name n.' The operation of DEFSTRUCT is probably best described using an example.

Example 10.1 Using **DEFSTRUCT** let us say that we want to set a property list that would be useful for describing dogs. We can assume that most dogs have four legs, so that would be a good default value. Let us also say that we wish to know the dog's color, but that it is variable so we will use NIL as the

begin by setting up the structure

(TYPE NIL))

BEAGLE. In this case, the three procedures would return the three default values

NIL, 4, and NIL. We can assign new values to the properties also by using SETQ and

We have left the number of legs alone. We can now evaluate the selector procedures

which returns

with the following syntax:

(SETF(⟨property⟩⟨name⟩)⟨new valuer⟩).

For example, we could change our Beagle to a purple dog by entering

We now have a way for associating properties with objects and for evaluating those properties. In the example at the end of the chapter we will see how this might be used.

At this point, we are not going to introduce any more LISP functions. The references listed at the end of the chapter can provide additional detail for those interested. Example problems are given at the end of the chapter.

10.5 AI AND ROBOTICS

While the material covered in this chapter so far may appear to be far removed from

in the future and in the function of sensory systems such as vision.

within the system. Rather than store pictures of the parts within the vision systems memory, certain features about the objects are stored, such as perimeter, area, number of holes, and so on. As systems become more complex and begin to be able to deal in crowded three-dimensional environments it will require greater intelligence on the part of the vision system. Techniques have been developed which allow vision

together. These systems rely heavily on search minimization methods to increase their processing speed.

Another promising area is research into task level programming. Rather than program the robot to make each required motion, it will be possible to make statements such as ‘Pick up the big red block.’ A program called SHRDLU, developed in 1971, allowed an operator to converse with a robot which lived in a world of blocks and pyramids. SHRDLU was able to respond and to plan out tasks. For example, if the red block was under a green block, SHRDLU would move the green block and set it down before getting the red block.

Factories are, in fact, limited environments. This constrains the number of problems which must be solved to make robot intelligence practical.

10.6 LISP IN THE FACTORY

In this section, we will develop an example LISP program which might be used to guide a robot in the performance of a simple task in the factory. The task which the robot must perform is the assembly of a gearbox consisting of three parts: a small gear, a large gear, and a plate with two shafts. The parts are presented to the robot by dumping the three parts out of a chute onto the worksurface in the view of a camera. Because of this feeding technique, it is possible for the parts to be improperly oriented and one on top of the other. In order to make the problem simpler, we will make some assumptions: the camera is able to recognize the parts and their location,

any orientation with its tool. Assume that the problem-solving procedure which we develop will not have to tell the robot how to perform simple tasks, only when to do them.

in place. If any part is in the way of another then the robot must clear that part before it can go on to complete the task. Let us also assume that the relationship among the

on top of another.

task as

```
(PLACE 'PLATE FIXTURE)
(PLACE 'GEAR1 PIN1)
(PLACE 'GEAR2 PIN2)
```

```
PIN1 abc
PIN2 lmn)
```

where xyz, abc, and lmn are the actual position values. Now all that must be done is

```
(GRAB (PART))
```

example, we will add some LISP primitives to make operation of our robot simpler. These are:

is uncovered by another part, and the system must know its location. If we set up a

```
(IS-UNDER NIL))
```

not they are covered then we can use

where the vision system supplies values for P1, P2, P3, U1, U2, and U3.

obstruction, clears it off, and grasps the part.

```
(DEFUN GRAB (PART)

(GRASP PART))
```

by GRASP; if not, GRASP is performed immediately.

```
(DEFUN GRASP (PART)


(SETQ GRIPPER PART))
```

GRASP moves to the location of the part, closes the gripper, and assigns the part name to the gripper.

```
(UNGRASP PART))
```

UNGRASP releases the part and clears the gripper

```
(DEFUN UNGRASP (PART)

(SETQ GRIPPER NIL))
```

In this procedure, FREESP ACE is a function which asks the vision system to return a value for an empty space where the system can drop off the obstruction

its new location and clears off the part which is to be grabbed.

The example demonstrates the use of LISP to perform a simple problem solving task as might be found in the factory. By considering the relationships of objects to each other we were able to perform the larger task by breaking it down into solvable primitive tasks.

10.6 Robotic Paradigms

AI in robotics mainly deals with interaction with the environment. Hence, a robot needs sensors to sense the environment. After obtaining enough relevant information it has to plan and then take correct action. Sometimes in emergency situations, it may not have the time to plan and needs to act immediately. Hence, we can say that intelligence in robotics deals with how we connect the three primitives of SENSE, PLAN and ACT.

The three primitives for organizing intelligence in robotics may be connected in three different ways as:

1. Hierarchical This is a serial connection of the three primitives.

Sense $\rightarrow$ Plan $\rightarrow$ Act

As an example we can think of a mobile robot that has sonars for distance measurement, trying to go to a goal point avoiding obstacles. The basic logic by this

of the obstacles around it. After the map is built and the goal point located it could plan its path towards the goal avoiding the obstacles. Hence there are three stages get

follow the path (ACT). All the three stages are connected in series.

2. Reactive In this case, there are only two primitives connected in series as shown below

PLAN → ACT

As an example we can again consider a mobile robot with sonar sensors. In this case, the robot just wanders around and if an obstacle comes in front of it, it just changes direction. This is also similar to the case of humans when we sense danger. When we touch a hot object we immediately react without really planing. Several insects also behave this way, if they sense danger they run in the opposite direction.

3. Hybrid This case is a combination between the earlier two in which some of the sensed information is used for planning while others are used for acting. As an example, we can again consider a mobile robot with sonar sensors for distance measurement. The robot could take the distance of objects on all sides and build a map. Then find a path and actuate the wheels as per the hierarchical parading. However, when it is following the path if an unknown object suddenly comes in front of it, it should either stop on turn immediately. Such cases are common as obstacles may be overlooked as the sonars may have missed data.

Problems

10.1 Describe three different search techniques.

10.2 Set up frames for the following:

(a) Dogs
(b) Cars
(c) Professors

10.3 Set up the and–or graph for the task performed in Sec. 10.6.

10.4 Evaluate the following expressions in the order given:

(SETO STUFF ‘(All THINGS ARE SILLY))

STUFF

(CAR STUFF)

(CDR STUFF)

(CAR(CDR STUFF))

10.5 Write a LISP program which might be able to determine if one object is on top of another. Assume you are given the locations of two objects in the form of a list (x, y, z) for each object. Assume that if x and y are equal for both parts and z is greater for one than the other, that one is on top. Use CAR and CDR to get the individual x, y, and z values and then perform the comparisons.

10.6 The solution in Prob. 10.5 could have been used to determine the OBJECT-IS-UNDER property in Sec. 10.6. Implement this feature.

10.7 Artificial Intelligence in robots deals with the connection between the three primitives of sense, plan and act. In each of the examples given below, explain how you would connect the primitives to design a controller.

(a) A mobile robot has to wander around and explore its environment without hitting obstacles.

(b) A mobile robot has to reach a goal point in the shortest time and also avoid hitting obstacles.

(c) A serial manipulator that has to detect a circular object on a moving coveyor and then grasp it.

References

1. A. Barr, P. Cohen, and E. Feigenbaum, *The Handbook of Artificial Intelligence*, William Kaufmann, Los Altos, CA, 1981.
2. R. Brown and P. Winston, *Artificial Intelligence: An MIT Perspective*, MIT Press, Cambridge, MA, 1980.
3. B. Horn and P. Winston, *LISP*, Addison-Wesley, Reading, MA, 1984.
4. P. McCorduck, *Machines Who Think*, Freeman, ‘San Francisco, CA, 1979.
5. K. Prendergast and P. Winston, *The AI Business*, MIT Press, Cambridge, MA, 1984.
6. P. Winston, *Artificial Intelligence*, Addison-Wesley, Reading, MA, 1984.

Applications Engineering for Manufacturing

PART FOUR

This part and the one that follows are concerned with the applications of robotics in manufacturing. Applications engineering deals with problems such as the design of the physical layout of the workcell, the control of the various components in the cell, evaluating the anticipated performance of the cell, and the economic analysis required to justify the robot project. To discuss these topics, this part of the book is divided into two chapters.

Chapter 11 addresses the physical design of the robot cell, and the coordination and control of the different pieces of equipment in the workcell. What are the various layout designs that can be used for the workcell? And what are the different control systems that can be used to coordinate the components of the cell? These systems are closely related to the robot programming methods discussed in Chaps. 8 and 9. One of the important techniques in workcell control involves the use of interlocks (WAIT, SIGNAL, and REACT) to tie the different activities in the cell together. In this chapter, techniques are also examined for simulating the workcell.

Chapter 12 deals with economics analyses used to justify robot projects. The analysis methods include the payback method, return on investment method, and equivalent uniform annual cost method. Robot and other programmable automation projects present certain unique problems in the economics justification of a project, and these issues are discussed in this chapter.

11

Robot Cell Design And Control

Introduction

Industrial robot generally work with other pieces of equipment. These pieces of equipment include conveyors, production machines, fixtures, and tools. The robot and the associated equipment form a workcell. The term workstation can also be used, but this term is generally limited to mean either (1) one workcell with a single robot or (2) one work location along a production line consisting of several robot workstations. Sometimes, human workers are included within the robot workcell to perform tasks that are not easily automated. These tasks might consist of inspection operations or operations that require judgment or a sense of touch that robots do not possess.

Two of the problems in robot applications engineering are the physical design of the workcell and the design of the control system which will coordinate the activities among the various components of the cell. In this chapter, these two problem areas are considered. In an important way, these topics bring together many of the technology and programming topics in the previous chapters of the book in order to apply robotics for productive purposes.

11.1 ROBOT CELL LAYOUTS

Robot workcells can be organized into various arrangements or layouts. These layouts can be classified into three basic type:

1. Robot-centered cell
2. In-line robot cell
3. Mobile robot cell

The following subsections describe these workcell configurations.

11.1.1 Robot-Centered Workcell

In the robot-centered cell, illustrated in Fig. 11.1, the robot is located at the approximate center of the cell and the equipment is arranged in a partial circle around it. The most elementary case is where one robot performs a single operation, either servicing a single production machine, or performing a single production operation. Initial installations of industrial robots in the 1960s were illustrative of this case. Die casting, one of the very first applications for a robot, required the robot to unload the part from the die after each casting cycle and dip it into a quenching bath. Other production machine applications required the robot to both load and unload the workpart. For some of these applications, the cycle times of the machine were relatively long compared to the part-handling time of the robot. Metal-machining operations are examples of this imbalance condition. This required the robot to be idle for a high proportion of the cycle, causing low utilization of the robot. To increase the robot utilization, the workcell concept was developed in which one robot serviced several machines as pictured in Fig. 11.1.

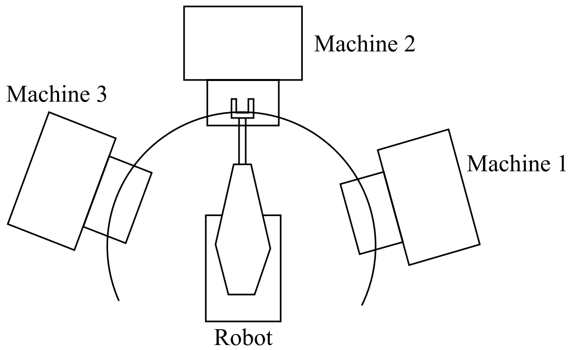


Fig. 11.1 *Robot-centered workcell layout.*

An application of the robot-centered cell in which the robot performs the process is arc welding. In this case, the root accomplishes the production operation itself, rather than servicing a production machine tool.

With these robot-centered cell arrangements, a method for delivering the workparts into and/or out of the cell must be provided. Conveyors, parts feeders with delivery chutes, and pallets are the means for accomplishing this function. Machining, die casting, plastic molding, and other similar production operations for discrete part production are examples of this case. These devices are used to present the parts to the robot in a known location and orientation for proper pick up. In arc welding, human operators are often used to accomplish the parts loading and unloading function for the robot. These kinds of cell layouts for arc welding are discussed in Chap. 14.

11.1.2 In-Line Robot Cell

With the in-line cell arrangement, pictured in Fig. 11.2, the robot is located along a moving conveyor or other handling system and performs a task on the product as it travels past on the conveyor. Many of the in-line cell layouts involve more than a single robot placed along the moving line. A common example of this cell type is found in car body assembly plants in the automobile industry. Robots are positioned along the assembly line to spot, and weld the car body frames and panels. The three categories of transfer systems that can be used with the in-line cell configuration[3] are:

1. Intermittent transfer
2. Continuous transfer
3. Non-synchronous transfer

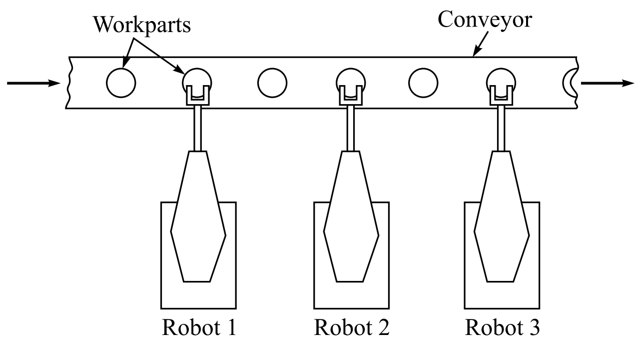


Fig. 11.2 *In-line robot workcell.*

1. Intermittent transfer system It moves the parts with a start-and-stop motion from one workstation along the line to the next. It is sometimes called a synchronous transfer system because all of the parts are moved simultaneously and then registered at their next respective stations. In a robot layout using intermittent transfer, the robot is in a stationary location and constitutes one position along the line at which a part or product stops for processing. The advantage possessed by the intermittent transfer system in robot applications is that the part can be registered in a fixed location and orientation with respect to the robot during the robot's work cycle.

2. Continuous transfer system This registration of the part relative to the robot becomes a problem when the continuous transfer system is used to move parts in the cell. With this type of transfer system, the workparts are moved continuously along the line at a constant speed. This means that the position and orientation of the part is continuously changing with respect to any fixed location along the line. The problem can be solved by using either of two means[2]:

(a) A moving baseline tracking system
(b) A stationary baseline tracking system

(a) The moving baseline tracking system It involves the use of some sort of transport system to move the robot along a path parallel to the line of travel of the workpart while the operation is performed on the part. In this way, the relative position of the part and the robot remain constant during the work cycle. The problem with this arrangement is that an additional degree of freedom must be provided for the robot to move along the conveyor. This additional degree of freedom is usually accomplished by mounting the robot on a cart which can be moved along a track or rail parallel to the conveyor. This solution involves considerable capital expense to construct the system to maintain accurate registration between the robot and the part. One of the operational problems that must be taken into account in the design of a production line with several robots is the potential interference and collision problem between robots at adjacent stations along the line. The easiest way to solve

to provide the wokcell with enough intelligence that it knows where each robot is at any moment, and can control the sequence so as to avoid collisions.

(b) Stationary baseline tracking system In this, the robot is located in a stationary position along the line but its manipulator is capable of tracking the moving workpart. 'Tracking' in this context means that the robot is able to maintain the positions of the programmed points, including the orientation of the end effector and the motion velocities, in relation to the workpart even though the part is moving along a conveyor. The engineering problems that must be solved to implement a stationary baseline tracking system are considerable although different from those encountered in the moving baseline system. First, the robot must

requires that the regular motion pattern of the manipulator be continuously translated in space in a direction parallel to the conveyor and at a speed equal to that of the conveyor. This allows the relative positions of the end effector and the part

cerned with the robot's 'tracking window.' The tracking window can be thought of as the intersection of the robot's work volume with the line of travel of the workpart along the conveyor. This concept is illustrated in Fig. 11.3. Allowances

reach. For a robot with tracking capability, the total motion cycle in a particular application must be consistent with the tracking window for that application. A third problem involves the sensing of the part on the conveyor.

Fig. 11.3 *Concept of 'tracking window'.*

For cells in which different product models will be processed, a sensor system must be used to identify which model is being delivered to the robot. A sensor is also required to determine that the part has entered the tracking window and that the robot can commence its work cycle. Other sensors are needed to track the position and velocity of the part during the cycle so as to coordinate with the robot tracking system. It is risky to presume that there will be no variations in the location and speed of the part as it is being processed.

3. Non-synchronous transfer The third type of transport system is non-synchronous transfer. It is also referred to by the name 'power-and-free' system. In this materials-handling system, each part moves independently along the conveyor in a stop-and-go fashion. When a particular workstation has completed its processing of a part, that part proceeds to move toward the next workstation in the line. Hence, at any given moment, some workparts are being processed while others are located between stations. The design and operation of this type of transfer system is more complicated than the other two because each part must be provided with its own independently operated, moving cart. However, the problem of designing and controlling the robot system used in conjunction with the power-and-free method is less complicated than for the continuous transfer method. For the irregular timing of arrivals on the non-synchronous transfer system, sensors must be provided to indicate to the robot when to begin its work cycle. The more complex problems of registration between the robot and the part that must be solved in the continuously moving conveyor systems are not encountered on either the intermittent transfer or the non-synchronous transfer.

11.1.3 Mobile Robot Cells

The third category of robot cell design is one in which the robot is capable of moving to the various pieces of equipment within the cell. This is typically accomplished by mounting the robot on a mobile base which can be transported on a rail system. The rail systems used in robot cells are either tracks fastened to the floor of the plant or overhead rail systems. Figure 11.4 illustrates the concept of the track-on-floor system, while the overhead rail system is shown in Fig. 11.5. The advantage of the overhead rail system compared to the floor-mounted track system is that less floor space is required. The disadvantage is the increased cost of constructing the overhead system.

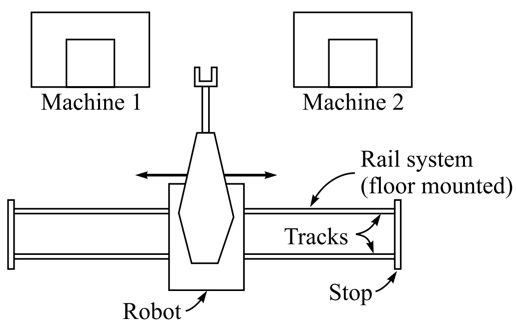


Fig. 11.4 *Mobile robot cell.*

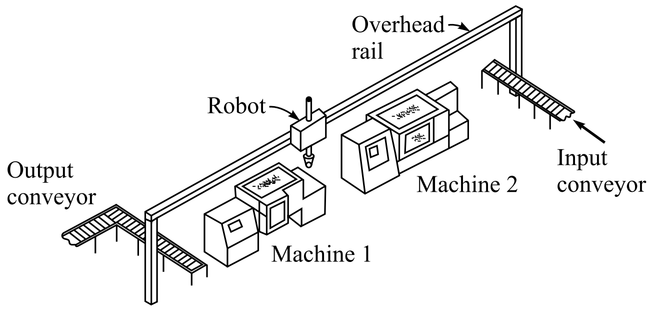


Fig. 11.5 *Overhead rail system for a mobile robot cell.*

A mobile robot cell would be appropriate when the robot is servicing several machine tools with long processing cycles. In this situation, the robot would be able to share its time among the machines without significant idle time for either itself or the machines it is servicing. If a separate robot were to service each of the machines, the utilization of the robots would be low because most of their time would be spent waiting for the machine cycles to complete. Accordingly, one of the problems in the design of a mobile robot cell is to find the optimum number of machines for the robot to service. The objective in this problem is to maximize the number of machines in the cell without causing idle time on any of the machines.

11.2 MULTIPLE ROBOTS AND MACHINE INTERFERENCE

In some robot cells, there will be more than one robot required to perform the application. The in-line robot cell is a common example of this situation. In other cases, one robot will work with more than one machine in the cell. Either the robot-centered cell or the mobile robot cell are illustrative of this possibility. In either of these situations, care must be taken to ensure that the different pieces of equipment do not interfere with one another. There are two ways in which this inferference can occur.

The first case involves physical interference of the robots, where the work volumes of two robots in the cell overlap each other. In this situation, the danger of collision exists between the robot arms. This is most easily prevented by separating the robots by an adequate distance to avoid the problem. However, there are some applications in which it is desirable for two robots to share the same space. An example would be where one robot places a workpart at a certain location, and the second robot picks the part up. The location must be in the work envelopes of both robots. Accordingly, an alternative approach is to coordinate the programmed motion cycles of the two robots so that the arms are never close enough to risk a collision.

The second type of interference is when there are two or more machines being serviced by one robot, and the machine cycles are timed in such a way that idle time is experienced by one or more machines while another machine is being serviced by the robot. This is called *machine interference* and it is a common problem encountered when a human worker is assigned to service multiple machines. The difference between machine interference with a human worker and machine interference with a robot is that the amount of interference in the human cell is affected by variations in worker cycle times and by the worker's level of effort. With greater variation in the cycle time and a lower effort level, the machine interference will tend to be greater. In a robot cell, the robot's cycle time will not be affected by effort level and the amount of cycle time variation will be significantly less than for human work.

Machine interference can be measured as the total idle time of all the machines in the cell as compared to the operator (or robot) cycle time. The measure is most commonly expressed as a percent. To illustrate the problem, an example of a three-machine cell is used, in which a robot is used to load and unload the machines.

Example 11.1 Each of the three machines in the cell are identical and they have identical cycle times of 50 s. This cycle time is divided between run time (30 s) and service time (load/unload) by the robot (20 s). The organization of the cycle time is shown in the robot and machine process chart of Fig. 11.6. It can be seen that each machine has idle time during its cycle of 10 s while the robot is fully occupied throughout its work cycle. Total machine idle time of all three machines is $3 \times 10 = 30$ s and the cycle time of the robot is $3 \times 20 = 60$ s. Accordingly, the machine interference is 30 s/60 s = 50%.

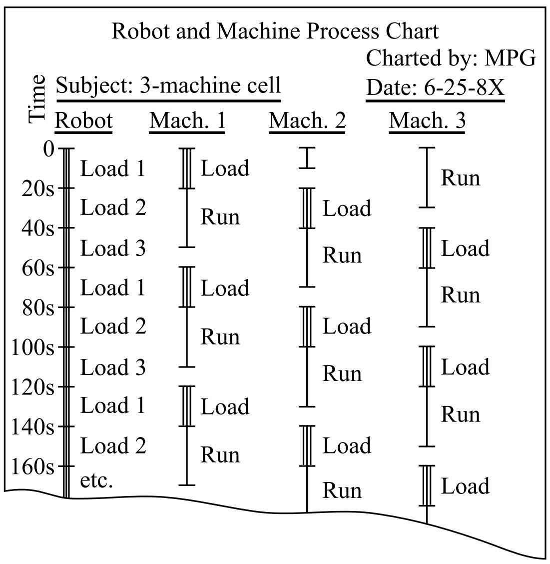


Fig. 11.6 *Robot and machine process chart for Example 11.1.*

In this example, the cycles of the three machines are the same. In this case, the question of whether or not machine interference will occur is determined by the relative values of machine cycle time and robot cycle time. The machine cycle time is the sum of service time and run time. The robot cycle time is equal to the number of machines multiplied by the service time. If the robot cycle time is greater than the machine cycle time, there will be resulting machine interference. If the machine cycle time is greater than the robot cycle time, there will be no machine interference, but the robot will be idle for part of the cycle.

In the case where the service and run times of the machines are different, the above relationships become complicated by the problem of determining the best sequence of servicing times for the machines into the robot cycle time. This problem will be explored in the exercises at the end of the chapter.

11.3 OTHER CONSIDERATIONS IN WORKCELL DESIGN

There are several other issues that must be considered in the design of the workcell. Among these considerations are the following:

1. Changes to other equipment in the cell To implement the workcell and interface the robot to the other equipment in the cell, alterations will often have to be made to the equipment. Special fixtures and control devices must be devised to permit the cell to operate as a single, integrated mechanism. Examples of these fixtures and controls include work-holding nests and conveyor stops to position and orient the parts for the robot, changes in the machines to allow the robot arm to gain access to the equipment, and limit switches and other devices to interface the various components in the cell.

2. Part position and orientation For raw workparts being delivered into the cell, it is important that the robot have a precise pick up location to get the parts from the conveyor or other work-handling system. At this pick up point, the parts must be in a known orientation to enable the robot to grasp and hold it consistently and accurately. During subsequent processing within the cell, this part orientation should not be lost. A method for achieving these objectives of part positioning and orientation must be designed into the workcell.

3. Part identification problem In cells where more than one type of part is processed or assembled, a method of identifying the particular part type must be determined. This can be done by any of a number of automated means, involving optical techniques or limit switches to sense differences in size or part geometry.

4. Protection of the robot from its environment In certain types of applicaltions (e.g., spray painting, hot metal-working operations), a means of protecting the robot from the adverse effects of its environment must be provided.

5. Utilities Providing the necessary utilities (e.g., electricity, air pressure, gas for furnaces, etc.) must be included among the factors considered in the design of the workcell layout.

6. Control of the workcell The activities of the robot must be coordinated with those of the other equipment in the cell. This subject is referred to by the term workcell control, and several sections are devoted to it and its related topics in this chapter.

7. Safety A means of protecting human personnel from harm in and around the robot workcell must be provided. This is generally accomplished by means of fences or other barriers, and by designing a safety monitoring system to interrupt the cell operation if unsafe conditions are encountered. These kinds of mechanisms and other issues related to safety will be discussed in Chap. 17.

11.4 WORKCELL CONTROL

In addition to the problem of designing the physical layout of the robot cell, another problem is concerned with coordinating the various activities that occur in the cell. Most of these activities occur sequentially, but simultaneous activities can also occur. There are other factors such as the safety of human personnel which must also be considered. Coordination of these various activities is accomplished by a device called the workcell controller or workstation controller. The functions performed by the workcell controller can be divided into three categories as suggested by Thomas[11]:

1. Sequence control
2. Operator interface
3. Safety monitoring

These functions are accomplished either by the robot controller itself or by a higher-level control device, such as a programmable controller. The robot controller usually has a limited input/output capability to permit interfacing with other pieces of equipment in the cell. If the control requirements to operate the cell become at all complicated, then a higher-level controller is needed. The use of programmable controllers shall be discussed in a subsequent section of this chapter. Also, the implementation of an effective workcell controller is dependent largely on the programming capabilities of the robots used.

11.4.1 Sequence control

Sequence control is the primary function of the workstation controller during regular automatic operation of the workcell. It includes the following kinds of control functions:

1. Control of the sequence of activities in the workcell.
2. Control of simultaneous activities.
3. Making decisions to proceed with the work cycle based on events that occur in the cell.
4. Making decisions to stop or delay the work cycle based on events that occur in the cell.

These functions occur during the normal operation of an average robot workcell. The following example illustrates these control requirements.

Example 11.2 This example is intended to demonstrate the importance of controlling the sequence of activities in the workcell during regular operation. The cell consists of a large robot, a numerically controlled machining center (which operates on an automatic cycle), and a belt conveyor for delivering raw workparts into the cell. Finished parts are placed on a pallet. The layout is shown in Fig. 11.7. The work cycle consists of the following sequence of

Fig. 11.7 *Workcell layout for Example 11.2.*

1. Robot picks up raw workpart from conveyor which has delivered the 'part to a known pick up position' (machine idle).

3. Machining center begins automatic machining cycle (robot idle).
4. Machine completes automatic machining cycle. Robot unloads machine and places part on the pallet (machine idle).
5. Robot moves back to pick up point (machine idle).

In this cell operation, nearly all of the activities occur sequentially. The only

to deliver raw parts into the cell during the work cycle. The purpose of the workstation controller in this example would be to make sure that the activities occurred in the correct sequence and that each step in the sequence has been completed before the next one is initiated. For example, the start of the automatic machining cycle must not occur until the robot has loaded the raw workpart into

Since all of the relevant activity in the preceding example is sequential, either the

of this work cell could be improved by using a double gripper. We have considered the different types of grippers in Chap. 5. A double gripper is a gripper with two grasping mechanisms for independent holding of two separate workparts. Use of a double gripper in our example would permit unloading and loading of the machine to be combined into a single (but more complicated) step. It would also allow the automatic machining cycle to occur at the same time that the robot was performing a portion of its motions. The following example illustrates how this would work.

Example 11.3 The same robot and equipment from Example 11.2 is used here except that a double gripper is employed for handling delivered the part to a known pick up position. Robot moves its double gripper into ready position in front of machining center (machine cycle in progress).

(machine idle).

part to pallet and places it in programmed location on pallet.

4. Robot moves back to pick up point (machine cycle in progress).

The feature which distinguishes this work cycle from the previous example is that several activities occur simultaneously. Because a double gripper is used, the workstation can be organized so that the machine is processing the part at the same time the robot is performing much of its work. This reduces the production cycle time, but it has the apparent effect of increasing the complexity of the control function.

Simultaneous activities in the workcell usually do not pose as much a problem as they might seem. Although the activities in Example 11.3 occur simultaneously, they are initiated sequentially. Hence, the purpose of the controller is basically the same as in Example 11.2. The robot, the conveyor, and the machining center each work as separate units in the cell. The conveyor operates continuously to deliver parts into the cell. The machine tool operates under numerical control to perform its automatic cycle. And the robot is regulated by its controller. The function of workcell control is to make sure that the various control cycles begin at the required times. To perform this function, the workstation controller must be capable of communicating back and forth with the various pieces of equipment in the cell. Signals must be sent by the workcell controller to the various components of the cell, and other signals must be received from the components. These signals are called interlocks, and shall be discussed in Sec. 11.5.

The workcell controller might be required to perform certain additional functions that are included within the general category of sequence control. These functions

1. Performing computations.
2. Dealing with exceptional events such as equipment breakdowns.
3. Perform irregular cycles, such as tool changing at periodic intervals.

The two preceding examples serve to illustrate the possible need for logic computations in the work cycle. The application included a palletizing sequence after the part was unloaded from the machining center. This palletizing sequence requires

The most typical pattern of locations is a rectangular matrix (e.g., 5 parts in one direction by 6 parts in the perpendicular direction on the pallet—a total of 30 parts). To place the parts in this arrangement, the coordinate position of each part placed on the pallet must be determined by the workcell controller based on parameters provided by the programmer (e.g., the programmer would specify a 5 by 6 pallet with 3 in. spacing of part centers in each direction). The workcell controller must calculate the new position on each successive cycle of operation. In addition, the palletizing operation creates a slightly different motion pattern on each cycle.

11.4.2 Operator Interface

The purpose of the operator interface in workstation control is to provide a means for human operators to interact with the operation of the cell. There are several situations where this would be required. Among the most important cases are the following:

1. The human is an integral part of the workcell.
2. Emergency stop conditions.
3. Program editing or data input by operator.

The first situation is where the human worker plays an integral role in the operation of the cell. The human performs a portion of the work cycle and the robot performs a portion of the cycle. In this situation, there is a need to allow for variations in the time required by the operator to perform the manual part of the cycle. In a robotic cell, this can be easily accomplished by means of stop/start controls placed conveniently for the operator. The operator uses these controls to regulate the robot cycle as required by the situation. The following example illustrates the need for using operator controls to allow for variability in operating pacing.

Example 11.4 In this example, a human operator is used with a robot to perform a hot working operation on a metal billet. Hot forging is a common operation in this category. The operator's task in the cycle is to remove hot billets from random positions in a furnace and place them in a nest from which the robot can retrieve the parts for the hot working operation. There would probably be variations in the time required for the worker to perform the manual portion of the cycle because of differences in the locations of properly heated parts in the furnace. At a particular time during the day, it might turn out that none of the parts in the furnace are heated to a sufficient temperature for the operation. This would be a reason for delaying the next operation cycle. For these and various other reasons, there is a need for the operator to be able to signal the robot when to start its portion of the cycle.

The second situation involves an emergency in which a human worker needs to prevent continued operation of the robot cycle. The reason for the emergency may be a safety problem or an irregularity in the work cycle that is potentially destructive to the robot or other equipment in the cell. A safety problem might arise when some person walking through the plant has unwittingly intruded into the robot's space. An irregular event in the work cycle might be that the robot has grasped the workpart improperly for loading into a machine and this would cause damage to the machine or the associated tooling during processing. In either case, the human worker located in the cell would have reason to interrupt the operation of the robot cycle until the emergency situation was corrected. Thomas[11] stresses the need for the operator controls to respond in a consistent and predictable manner to emergency stop conditions.

The third requirement for operator control is to perform editing of the program or other similar input functions. Some robot controllers require that the robot be in a non-operational mode when changes are made in the program. The more flexible controllers allow for editing to be accomplished while the robot is performing its regular cycle.

11.4.3 Safety Monitoring

In addition to the operator's ability to override the regular work cycle in the event of an observed safety hazard, the workcell controller should also be capable of monitoring its own operation for unsafe or potentially unsafe conditions in the cell. This function is called safety monitoring or hazard monitoring. The discussion of this topic shall be postponed until Chap. 17 when a more comprehensive treatment of robot safety is provided.

11.5 INTERLOCKS

An interlock in robotic workcell design is a method of preventing the work cycle

It is a feature of workcell control which plays an important role in regulating the sequence in which the various elements of the cycle are carried out. Referring back

1. To make sure that a raw workpart was at the pick up location on the conveyor before the robot tried to grasp the part.
2. To determine when the machining cycle was completed before the robot
3. To indicate that the part has been successfully loaded so that the automatic machining cycle can begin.

In each of these instances, it is critical that one element of the cycle has been completed before any attempt is made to begin the next element. The method of regulating the sequence of the elements would involve the use of interlocks.

output interlocks and *input interlocks*. Art output interlock involves the use of a signal sent from the workstation controller to one of the machines or other devices in the workcell. It corresponds to the SIGNAL programming statement used in Chaps. 8 and 9. For example, an output interlock would be used to signal the machining center in Examples 11.2 and 11.3 to commence the automatic cycle. The output signal originates from the workcell

the conditions would be that the workpart has been properly loaded and the robot gripper has been removed to a safe distance. These conditions are usually determined by means of input interlocks.

An input interlock makes use of a signal sent from one of the components in the cell to the workstation controller. It corresponds to the WAIT command used in Chaps. 8 and 9. It is employed to indicate that a certain condition or set of conditions have been met and that the programmed work cycle sequence can continue. As an illustration, an input interlock would be used in a machine-loading application to signal the workstation controller that the part has been properly loaded into the

Interlocks are essential in nearly all robotic workcells consisting of several operating pieces of equipment that must all work in a coordinated fashion. They serve to interface the various components of the workstation. Their use provides a synchronization and pacing of the activities in the cell which could not be accomplished through timing alone in the work cycle. Interlocks allow for variations in the times taken for certain elements in the cycle. They prevent work elements from starting before they should start. They also help to prevent the damage of the various components of the cell.

In the design of the workcell, consideration must be given not only to the regular sequence of events that will occur during normal operation of the cell, but also to the possible irregularities and malfunctions that might happen. In the regular cycle, the

For the potential malfunctions, the applications engineer must determine a method of identifying that the malfunction has occurred and what action must be taken to respond to the malfunction. Then, for both the regular and irregular events in the cycle, interlocks must be provided to accomplish the required sequence control and hazard monitoring that must occur during the work cycle. In some cases, the interlock signals can be generated by the electronic controllers for the machines and other devices used in the workcell. For example, numerically controlled machine tools would be capable of being interfaced to the workcell controller to signal completion of the automatic machining cycle. In other cases, the applications engineer must design the interlocks using sensors to generate the required signals.

Interlocks are often implemented by means of limit switches and other simple devices that serve as sensors. In some cases, the robot cell must make use of more advanced sensors in order to successfully perform the work cycle. Examples of the

position and orientation on a moving conveyor, and determining that a component had been properly assembled before proceeding with further assembly work on the product. Sensor systems for robotics were discussed in Chaps. 6 and 7.

11.6 ERROR DETECTION AND RECOVERY

operation of the robot cell. However, malfunctions and errors can occur during the cycle, for which some form of correction is needed to restore the workcell to regular automatic operation. In most robot cells, it is necessary to stop the workcell when errors occur, and to provide human assistance for the corrective action. This generally results in production delays before the maintenance crew arrives to diagnose the problem and make repairs. There is a trend in programmable automation technology to attempt to endow the robot (or other automated equipment) with the capability to sense errors and malfunctions when they occur, and to take the necessary compensating action to restore the system to normal operation. This capability is referred to as error detection and recovery.

error detection and *error recovery*. The detection problem is concerned with the use of the appropriate sensors to determine when an error has occurred. It also includes the associated intelligence to interpret the sensor signals so that errors can be properly random errors, systematic errors, and illegitimate errors. Random errors are those that result from stochastic phenomena and are usually characterized by their statistical nature. For example, part size in a machining operation would be expected to vary randomly about some mean value. Depending on the amount of the variation, this could cause problems in a subsequent manufacturing process. Systematic errors are not determined by chance but by some bias that exists in the process. For example,

systematic error in the product. The third class is the illegitimate error, typically resulting from an outright mistake, either by the equipment or by a human error. An

for a given process.

Example 11.5 In the context of an automated machining cell that is tended tooling,

This does not include the usual safety monitoring system that would probably be employed in the robot cell. In each of the categories, there are particular malfunctions

Error source category	Particular malfunction or error
1. Tooling	Tool wear-out Tool breakage Vibration (chatter) Tool not present Wrong tool loaded
2. Workpart	Workpart not present Wrong workpart Defective workpart Oversized or undersized part
3. Process	Wrong part program Wrong part Chip fouling No coolant when there should be vibration (chatter) Excessive force Cutting temperature too high

Contd.

4. Fixture	
	Part located properly (yes or no) clamps actuated
	Part dislodged during processing
	Part breakage
	Chips causing location errors
	Hydraulic or pneumatic failure
5. Machine tool	Vibration
	Loss of power
	Power overload
	Mechanical failure
	Hydraulic or electrical failure
6. Robot/end effector	Improper grasping of workpart.
	No part present at pick up Hydraulic or electrical failure
	Loss of positioning accuracy.
	Robot drops part during handling

Given that an error has occurred and that the error detection system has correctly

that can be employed by the robot to correct or compensate for the malfunction that

1. Adjustments at the end of the current cycle This recovery strategy would represent a relatively low level of urgency. At the end of the current cycle, the robot program would branch to a subroutine to make the required corrections, then branch back to the main program.

2. Adjustments during current cycle
corrective action must be taken during the current cycle of operation. However, it is not so urgent that the process must be stopped. The corrective action is typically accomplished by calling a special subroutine that has been designed to deal with the particular error.

The error in this case requires that the process be stopped, and that a subroutine be called to correct the error. At the end of the correction algorithm, the process can be resumed or restarted.

4. Stop process—call for help This action is usually taken either because the

human assistance is required to restore the system.

The following example continues from the previous example and show some

Example 11.6 Among the errors that can occur during the operation of the workcell, one of the possibilities is that the robot will drop the part. In terms of error recovery strategy, this would normally be classified as a category 1 situation, in which corrective action would be taken during the cycle. To search for the part on the factory floor would probably be a hopeless and time-consuming task for the robot. Instead, the logical recovery strategy would be to reach for the next part.

An example of a category 2 error recovery situation would be when a part is detected to be dimensionally oversized—too large for the normal machining sequence. The logical recovery strategy might be to invoke a subroutine to provide an additional machining pass to remove the extra material.

A tool failure during the machining operation would be a category 3 situation. The process would have to be stopped in order to replace the broken tool with a sharp tool before resuming the cut. The fact that the tool broke during the cut may result in damage to the work surface. If the likelihood of this is high, then it might be necessary to change the part.

An example of a category 4 error recovery situation might be a hydraulic failure of the robot drive system. An automatic recovery from this malfunction may not be possible, and this would necessitate calling for human assistance.

The error detection and recovery system is implemented by means of the sensors used in the workcell together with the robot programming system. In terms of the programming requirements, the textual languages have a distinct advantage in developing the sometimes complex logic that is often needed. For highly sophisticated robot cells, a majority of the programming may be required for the error detection and recovery system.

11.7 THE WORKCELL CONTROLLER

The control systems and components discussed in Chaps. 3 and 4 were concerned with controlling the motions of the robot's manipulator. The workcell control system is concerned with the coordination of the robot's activities with those of the other equipment in the cell.

A number of options are available to satisfy the requirements of the workcell controller. These options include the use of the robot controller itself, relays, programmable controllers, and small stand-alone computers (minicomputers or microcomputers). The decision of which option to select depends on the complexity of the cell (e.g., the number of separate pieces of equipment, the number of separate control actions that must be controlled, and the number of robots in the cell), and whether the robot controller alone is capable of handling all of the activities. In the subsections below, the various alternatives are compared.

11.7.1 The Robot Controller

There are various types of control technology used for robot controllers. These include the simpler limited sequence controllers, electronic controllers, and computer controls. The more sophisticated types usually have a limited input/output capability to interface with other equipment. This input/output interface is provided specifically for the incorporation of interlocks in the workcell. The robot controller has the capability to tie the incoming signals to the work cycle program, so that the proper sequencing of output signals and robot motions can take place. The number of input/output ports might range between 10 and 20 for playback robots. A typical arrangement for the input/output module of the robot controller would be as follows:

1. Input ports (perhaps 10 to 25 input lines) These would be used for incoming signals from external pieces of equipment. The signals would be binary (voltage on or off) and could be referenced as logical conditions in the robot program for purposes of interlocking. On newer controllers, the input ports would include the capacity to read in analog signals.

2. Output ports (perhaps 10 to 25 output lines) These ports would be used for output interlock signals to the external equipment. The signals would be initiated or terminated according to logical conditions in the robot program, thus resulting in some response by the external equipment. Again, some newer robots would have the capacity for analog outputs as well as binary signal outputs.

3. Input ports (perhaps five input lines) These would be reserved for safety interlocks. Upon receipt of a signal from the external safety sensor on one of these lines, the controller would immediately interrupt the program, thus stopping the robot. In some cases, these input ports might be used to simply turn the power off to the manipulator.

This represents a limited input/output capacity for a workcell with any degree of complexity. Also, today's robot controllers are generally limited to sequence control and, as the above list indicates, often do not possess the capability to incorporate any significant safety monitoring or operator interfacing into the workcell control system. With the growing use of computer controls and the need by competing robot manufacturers to increase the control capabilities of their products, it is expected that future robot controllers will be equipped with enhanced input/output capacity and the capability to control intelligent robots.

11.7.2 Electromechanical Relays

An electromechanical relay is a control device used to actuate electrical circuits in response to changes in incoming signals. They are commonly used in industrial applications to provide sequence control of electrically operated equipment although they are gradually being displaced by more modern devices such as programmable controllers.

Relays can be used to augment the capabilities of a robot controller in the design of a workcell control system. Their use would typically be reserved for simple robot cells, such as pick-and-place applications, and where the robot has very limited input/output capacity. With relays, it would be relatively-easy to include a simple safety monitoring scheme in the workcell. Such a scheme might consist of a fence surrounding the work place with a safety gate to gain access to the cell. Using the appropriate sensors (e.g., a limit switch to indicate closure of the safety gate) the relays could be set up to stop the robot, perhaps by interrupting its power source, as soon as a hazardous condition was sensed.

over to a new workstation control task, and the fact that they are susceptible to mechanical wear and are less reliable than computer-type controls. The functions of a relay panel can be accomplished by a programmable controller, which avoids the above problems.

11.7.3 Programmable Controllers

Programmable controllers were introduced in the late 1960s as a replacement for systems of electromechanical relays. Up until that time, relay panels constituted the standard technique for accomplishing sequence control in industrial operations. The

use could be readily learned by shop personnel who were familiar with the logic diagrams used for relay control panels.

with programmable memory that is capable of generating output signals according to logic operations and other functions performed on input signals. The program for a PC determines the sequence of operations and the generation of input and output signals. A PC is programmed by specifying the same kinds of logic diagrams, called ladder diagrams, used for years to set up relay control panels. Other programming methods are also possible on many programmable controllers, including the use of symbolic notation similar to computer programming. The functions that can be

1. Control relay functions The generation of an output signal based on logic rules applied to one or more input signals.

2. Timing functions
length of time.

3. Counting functions An internal counter in the PC is used to sum the number of contact closures and generate an output signal when the sum reaches a certain level.

4. Arithmetic functions Some PCs can perform the basic arithmetic operations such as addition, subtraction, multiplication, and division.

5. Analog control functions Another feature which is available on some PCs is the capability to simulate analog functions, such as proportional, integral, and derivative control.

These functions permit the PC to perform as a powerful robot workcell controller. Some PCs have the capacity to accept several hundred input/output connections,

a more complex workcell with more activities taking place in the cell. An automobile body spot-welding line, in which many robots perform various welding operations, would use a programmable controller as the overall cell control device. In addition to handling a greater number of input and output signals, the programmable controller also possesses other features that are beyond the capability of most robot controllers. These features include:

1. Maintenance and diagnostic functions The CRT terminal used to program the PC can also be used in some systems to monitor the operation of the workcell. Some PCs have sophisticated diagnostic capabilities to quickly determine the origin of a problem when it occurs.

2. Operator interface The use of the PC as the robot cell controller allows greater capacity and flexibility to implement the operator interface. Display terminals can be included in complex cells to provide operating performance information about production rates, tool usage, equipment breakdowns, and other data. Printers can be included at the control station to provide hard copy reports about the cell performance.

3. Safety monitoring More sophisticated hazard monitoring systems can be implemented with programmable controllers. A greater number of safety conditions can be observed while the cell is operating than is possible with the robot controller alone.

11.7.4 A Computer as the Workcell Controller

Some robot applications have requirements for which a digital computer is the most appropriate method of workcell control. The reference of the use of a stand-alone computer is used here (generally a minicomputer or microcomputer) rather than the computer which is used as the robot control unit. In cases where a computer is the workcell controller, it would be used either in series with a programmable controller or as a substitute for the PC. The computer might perform other functions in the plant. and so it would be implemented to control the robot cell in a time-sharing mode of operation. Also, the computer would probably form a component in a hierarchical computer network in the factory. connected down to the programmable controller(s) and/or robot controller(s) in the cell, and connected up to the next hierarchical level in the plant.

Programmable controllers are specialized devices that are designed to be interfaced with industrial processes. They are provided with input/output ports that can be directly wired to the plant equipment. This is an advantage over the digital computer, and special arrangements must be made to interface the computer to the industrial equipment in the cell. However, the PC has certain limitations in data processing and programming languages which gives the computer an advantage in applications requiring these capabilities. Some examples of the kinds of robot application features that might tend to favor the use of computers for workcell control would include the following:

1. Cases in which there are several cells whose operations must be coordinated, and significant amounts of data must be communicated between the cells.
2. Cells in which the error detection and recovery problem constitutes a significant portion of the coding that must be programmed into the workcell operation.

3. Where several different products are made on the same robot-automated production line, the operations at the different stations have to be coordinated and sequenced properly. Computers would be well suited to the data processing chores required in this type of application. In cases where the production lines are used for assembly operations, the various sizes and styles of the component parts must be sorted and matched to the particular model being assembled at each respective workstation along the line.

Situations in which a high level of production scheduling and inventory control are required in the operation of the cell. Again, this type of data processing function might require the use of a computer in addition to or as a substitute for a programmable controller.

The differences between digital computers and programmable controllers are principally differences in applications rather than differences in basic technology. The PC can in fact, be considered to be a specialized form of digital computer with dedicated features for input/output control of industrial equipment. The technologies of the two types of control devices are quite similar.

11.8 ROBOT CYCLE TIME ANALYSIS

The amount of time required for the work cycle is an important consideration in the planning of the workcell. The cycle time determines the production rate for the job, which is a significant factor in the economic success of the robot installation. In the case of work performed by a human operator, the time required to accomplish the cycle would be determined by one of several work measurement techniques. One of these work measurement techniques is called MTM (for Methods Time Measurement). With MTM, the work cycle is divided into its basic motion elements and standard time values are assigned to each of the elements to construct the time for the total cycle. The standard time values have been previously compiled by studying similar elements and analyzing the factors that determine the time required to perform the elements. For example, the time required for a human operator to transport an object from one place to another depends on such factors as the weight of the object, the distance the object is moved, and the precision with which the object is located at the end of the move.

An approach similar to MTM has been developed by Nof and Lechtman[8] at Purdue University for analyzing the cycle times of robot work. The method, called RTM (for Robot Time and Motion), is useful for estimating the amount of time required to accomplish a certain work cycle before setting up the workstation and programming the robot. This would allow an applications engineer to compare alternative methods of performing a particular robot task. It could even be utilized as an aid in selecting the best robot for a given application by comparing the performance of the different candidates on the given work cycle.

The methodology of RTM is similar to MTM. There are 10 general categories of robot work cycle elements as presented in Table 11.1. The 10 categories can be collected into four major groups:

1. Motion elements These are the manipulator movements, performed either with or without load.

2. Sensing elements These are sensory activities performed by robots equipped with sensing capabilities. Examples include vision sensing, force sensing, and position sensing.

3. End effector elements These elements relate to the action of the gripper or tool attached to the robot wrist as its end effector.

4. Delay elements These are delay times resulting from waiting and processing conditions in the work cycle.

To use Table 11.1, the robot work cycle must be divided into its corresponding elements, and each element is specified with its associated parameters such as distance, velocity, and so forth. Different models of robots will be capable of performing the various elements at different times. According to Nof and Lechtman, element time values must be determined for each available robot in order to use RTM. There are four possible approaches that can be used to determine the element times and analyze a robot cycle with RTM.[8] The first involves tables of elements, in which time values are determined for the different elements listed in Table 11.1. This is the basic approach used to analyze human work with MTM. The second approach is to develop regression equations for the more complicated elements whose values are functionally related to several factors. Once the equation is developed for a given element, the user simply plugs the factor values into the equation to calculate the element time. Both of the preceding approaches have been applied in work measurement of traditional human performance.

Table 11.1 *The 10 elements (and corresponding symbols) in RTMt.*

Element	*Symbol*	*Definition of element*	*Element parameters*
1	Rn	*n*-segment reach: Move unloaded manipulator along a path comprised of *n* segments	Displacement and velocity (or path geometry and velocity)
2	Mn	*n*-segment move: Move object along path comprised of *n* segments	Displacement and velocity (or path geometry and velocity)
3	ORn	*n*-segment orientation: Move manipulator mainly to reorient	Displacement and velocity (or path geometry and velocity)
4 4.1	SEi SE1	Stop on position error Bring the manipulator to rest immediately without waiting to null out joint errors	Error bound
4.2	SE2	Bring the manipulator to rest within a specified position error tolerance	
5 5.1	SFi SF1	Stop on force or moment Stop the manipulator when the force conditions are met	Force, torque, and touch

Contd.

5.2	SF2	Stop the manipulator when the torque conditions are met	
5.3	SF3	Stop the manipulator when either the force or torque conditions are met	
5.4	SF4	Stop the manipulator when the touch conditions are met	
6	VI	Vision operation	Time function
7	GRi	Grasp an object	Distance to open/close
7.1	GR1	Simple grasp object of by closing fingers	
7.2	GR2	Grasp object while centering. hand over it	
7.3	GR3	Grasp object by closing one finger at a time	
8	RE	Release object by opening fingers	
9	T	Process time delay when robot is part of the process	Time function
10	D	Time delay when robot is waiting for a process completion	Time function

† *Source*: (Reprinted with permission from *Industrial Engineering* Magazine, April, 1982. Copyright © Institute of Industrial Engineers, 25 Technology Park/Atlanta, Norcross, GA 30092.)

The third approach is called 'motion control,' and it can be applied to the group 1 elements involving robot motions. Motion control is concerned with the kinematic and dynamic analysis of the manipulator movement. It determines the element time values by considering the distances moved and the velocities to make the moves. It also considers acceleration and deceleration at the beginning and end of the moves. For example, if acceleration and deceleration are ignored for the moment, the time required to move the manipulator will be the distance S divided by the velocity V. For some robots, the acceleration and deceleration times can be approximated closely by a constant value.

The fourth modeling approach in RTM is called '*path geometry*' by Nof and Lechtman. This approach is similar to motion control and requires the specification of the motion path to be followed by the manipulator together with the robot joint and arm velocities. It turns out that most robot motions involve the simultaneous actuation of several joints, but one of the joints usually predominates because its relative move is the largest. This can be analyzed by one of the computer programs developed at Purdue to determine the time for the move.

Table 11.2 presents a listing of element time values and equations for calculating the times for selected elements given in Table 11.1. These values represent the element times for a hypothetical robot, and do not reflect actual values developed through the research of Lechtman, Nof, and others. We are using the listings in Table 11.2 to demonstrate the RTM method in a simplified form for the purposes of an example and exercises at the end of the chapter. Potential users of RTM should consult the original research reports[6,7,8] for guidance on applying the method in actual projects.

Table 11.2 *Hypothetical values for selected elements in RTM for a hypothetical robot model.†(Refer to Table 11.1 for element definitions.)*

Element	*Symbol*	*Element time, s*		*Parameters*
1	R1	$S/V + 0.40$	for $S > V/2.5$	S = distance moved (ft)
				V = velocity (ft/sec)
		0.40	for $S < V/2.5$	This is used for short moves
2	M1		For payloads of less than 1.0 lb	
		$S/V + 0.40$	for $S > V/2.5$	S = distance moved (ft)
				V = velocity (ft/sec)
		0.40	for $S < V/2.5$	This is used for short moves
			For payloads between 1 and 5 lb	
		$S/V + 0.60$	for $S > V/2.5$	S = distance moved (ft)
				V = velocity (ft/sec)
		0.60	for $S < V/2.5$	This is used for short moves
			For payloads between 5 and 15 lb	
		$S/V + 0.90$	for $S > V/2.5$	S = distance moved (ft)
				V = velocity (ft/sec)
		0.90	for $S < V/2.5$	This is used for short moves
4.1	SE1	0.1 V		V = previous velocity (ft/sec)
7.1	GR1	0.1		Assumed to be independent of any parameters
8	RE	0.1		Assumed to be independent of any parameters
9	T	T		T = robot delay time
10	D	D		D = process delay time

† The values and equations listed in this table are fictitious and are not intended to indicate performance values for any robot model. Also, although the listings are based on the RTM research at Purdue, the values derived from this table should not be interpreted to represent the results of the Purdue research.

Example 11.7 This example will illustrate the use of the RTM method. The work cycle consists of a simple task in which the robot must move parts weighing 3 lb from one conveyor to another conveyor. The sequence of the work cycle proceeds as follows:

1. Robot picks up part from first conveyor which has delivered the part to a known pick up position.
2. Robot transfers part to second conveyor and releases part.
3. Robot moves back to ready position at first conveyor.

A sketch of the workstation showing distances that the robot must move from one position to the next is shown in Fig. 11.8. The detailed sequence of elements to accomplish the work cycle is presented in Table 11.3. The conveyor delivers one part every 15 s, so the work cycle is limited by the conveyor feed rate. The RTM analysis in this problem would be useful for determining whether the time for the robot motion cycle is compatible with the conveyor feed rate. If the robot motion takes longer than 15 s, then the conveyor might have to be slowed down, or an alternative robot cycle developed. For example, the distances moved could be reduced to shorten the move element times. If the robot motion cycle takes less than 15 s, then the possibility of speeding up the parts delivery to the conveyor could be investigated.

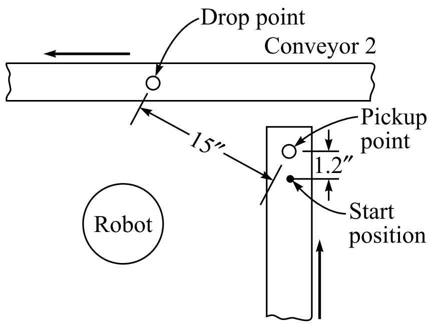


Fig. 11.8 *Workcell layout for Example 11.7.*

Table 11.3 *Detailed listing of work cycle elements for Example 11.7.*

Sequence	*Element description*
1	Conveyor delivers a part to a fixed position every 15 s. Robot in ready position above conveyor must await part delivery before executing its motion cycle.
2	Robot approaches part with gripper in open position. Speed setting is 0.1 ft/sec.
3	Robot gripper closes on part.
4	Robot lifts part 0.1 ft above conveyor. Speed is 0.1 ft/sec.
5	Robot moves part to a position 0.1 ft above second conveyor. Robot arm speed is 0.5 ft/sec. Distance traveled is 15 in.
6	Robot moves part to conveyor surface. Gripper would have to be oriented so that conveyor motion does not cause immediate tipping of the part when released. Speed is 0.1 ft/sec.
7	Robot gripper opens to release part on conveyor surface.
8	Robot moves empty gripper away from conveyor surface by 0.1 ft. Speed is 0.1 ft/sec.
9	Robot arm returns to ready position 0.1 ft above first conveyor surface.
	Distance traveled between conveyor is 15 in. Robot arm speed is 0.5 ft/sec.

Table 11.4 reduces the data contained in Table 11.3 to the RTM symbol notation, and presents the calculated element times as determined from the hypothetical robot values in Table 11.2. The total cycle time is 15 s. It turns out that the conveyor is the

limiting factor in the cycle, requiring 2.8 s more time than the robot motion cycle. It might be possible to reduce the feed rate on the conveyor down to one part every 12.2 s. This would provide a perfect match between the feed rate and the robot cycle.

Table 11.4 *Elements from Table 11.3 in RTM notation with resulting times.*

Sequence	RTM Symbol	Distance, ft	Velocity, ft/sec	Delay time, s	Element time, s	Description
1	D	—	—	15.0	15.0	Await delivery
2	RI	0.1	0.1	—	1.4	Approach part
3	GRI				0.1	Grasp part
4	MI	0.1	0.1	—	1.6	Lift part
5	M1	1.25	0.5	—	3.1	Move part
6	M1	0.1	0.1	—	1.6	Approach release point
7	RE	—	—	—	0.1	Release part
8	RI	0.1	0.1	—	1.4	Depart release point
9	R1	1.25	0.5	—	2.9	Reposition
Total					12.2	

11.9 GRAPHICAL SIMULATION OF ROBOTIC WORKCELLS

RTM can be considered a method of simulating, in terms of time, the activities in the robot workcell. Another method of simulation involves graphical modeling on a CAD/CAM system. Simulation based on computer graphics can be used not only to analyze cycle times, but to design the cell itself. It turns out that a substantial amount of time is spent in designing and laying out the cell, designing or selecting the equipment, and similar activities. One industry estimate[10] is that 60 to 80 percent of the total cell implementation time is spent on these design-related problems and cell fabrication. (The remaining 20 to 40 percent of the time is spent in programming and refining the cell.) With so much effort expended on the design of the robot cell, it is reasonable to utilize labor saving tools to make the process as efficient as possible.

This section will discuss the use of computer graphics to simulate the design and operation of the robot and the workcell. An example of university research in this area and an example of a commercial package for designing and simulating the robot cell are provided in the next section.

11.9.1 Research in Graphics Modeling for Robotics

Research in Lehigh University's Computer-Aided Design Laboratory in conjunction with the Institute for Robotics has led to the development of a graphics simulator of the PUMA 600 robot and the VAL language used to program the PUMA.[1] The simulator makes use of a FORTRAN callable graphics language to display the kinematic behavior of the PUMA in response to VAL motion statements. Algorithms for computing the positioning of the manipulator along segmented paths are used to simulate joint coordinate and straight line motions. Sequences of VAL commands can be entered interactively and their resulting motions shown on the graphics monitor.

The PUMA model was constructed by means of a series of extruded polyhedrons. To simulate motion, the vertices of each polyhedron are transformed and the model is redrawn. Since the PUMA consists of revolute joints, rotation transformations are used predominantly. The model can be scaled up or down, and other capabilities of the CAD/CAM system were exploited to facilitate viewing of the model. The research has explored both wire-frame and solid models, and the alternatives are illustrated in Figs. 11.9 and 11.10.

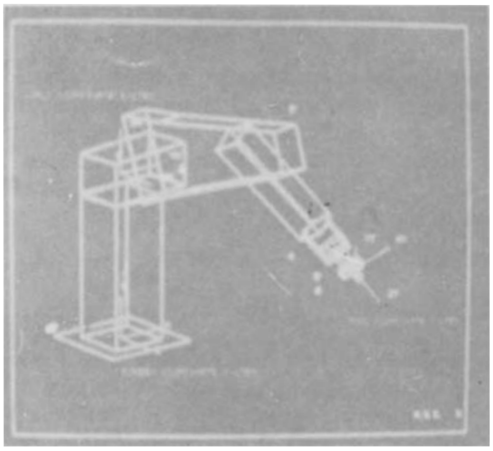


Fig. 11.9 *Computer graphics simulation of the PUMA robot with wire-frame model. (Photo courtesy: Computer-Aided Design Laboratory, Lehigh University)*

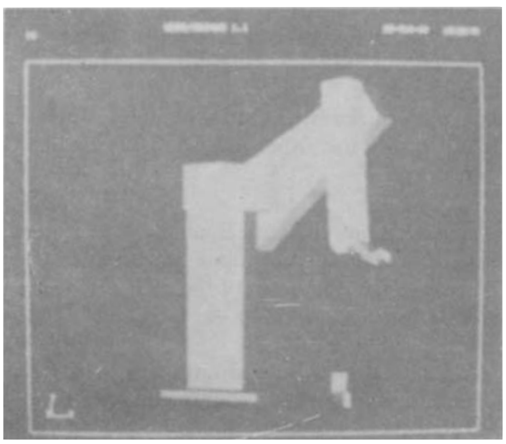


Fig. 11.10 *Computer graphics simulation of the PUMA robot with solids model. (Photo courtesy: Computer-Aided Design Laboratory, Lehigh University)*

The motivation for the graphics simulation research derives from our interest in several engineering issues related to robotics applications. These Collision detection between the robot and other objects in the workcell. This problem is difficult to check visually with a wire-frame model. An algorithm was developed to accomplish a coarse and fine check to determine interference.

Effects of acceleration, deceleration, arm member mass, payload mass, and other related factors on the dynamic performance of the robot.

Problems of off-line programming on the CAD/CAM system and then down-loading the program directly to the PUMA. This problem was discussed in the previous programming chapters.

Input/output to CAD/CAM systems for assembly simulation.

The simulator has also been found useful in teaching the principles of both computer graphics modeling and robotics to students at Lehigh. Even though the PUMA is not a large robot, there is a substantial safety problem involved in exposing students in significant numbers to the actual machine for 'handson' training. Simulation on the CAD/CAM system provides a safe trial run of the program, thus reducing hazards to the student and the robot.

11.9.2 The PLACE System

Several commercial packages are available for graphical simulation, and it is anticipated that these systems will grow in availability and use. At the time of this writing, the commercial simulation products include PLACE (McDonnell Douglas Manufacturing Industry Systems Company), Robographix (Computervision Corp.), and Robot-SIM (General Electric's Calma Co.). It seems appropriate to conclude the discussion of workcell design and control by describing the operation of these robotic simulation packages and the opportunities offered by them. We will use the PLACE system[5] as an example of these systems. Although the other commercial systems may not be organized in exactly the same way as the PLACE system, many of its features that are described are similar to those of other available systems.

The PLACE graphic simulation package consists of four modules. The first module to be released was PLACE, and the others were made available subsequently. The four modules are:

1. PLACE This stands for Positioner Layout and Cell Evaluation. It is used to construct a three-dimensional model of the robot workcell in the CAD/CAM data base and to evaluate the operation of the cell.
2. BUILD This module is used to construct models of the individual robots that might be used in a cell.
3. COMMAND This is used to create and debug programs off-line that would be downloaded to the robot to save on-line programming time.
4. ADJUST It must be expected that there will be a difference between the computer graphics model of the workcell and the actual workcell. The ADJUST module is used to calibrate the cell.

The configuration of the four software modules in relation to the design and programming of the workcell is presented in Fig. 11.11.

Fig. 11.11
the workcell.

PLACE is used to develop a computer graphics model of the robot and other workcell components in three dimensions. Figures 11.12 and 11.13 illustrate two models displayed on the McDonnell Douglas Unigraphics system. The system also permits the user to test out the motion sequence of the robot by means of a controlled

1. Model the cell components The user can enter the geometric data into the CAD

components. The robot models can be entered through the BUILD module and are available for use by PLACE.

2. Model the workcell The various components can be called from the CAD system and assembled into a workcell in various ways to examine alternative

alternatives in three dimensions, a better workcell design can be developed in less time compared to manual planning methods.

effector and commands the working point to move to individual target points in the cell. These target points represent positions in the cell which the end effector must visit in the execution of the work cycle. The user can verify the physical capacity of the robot to reach all of the desired points in the cell.

4. Build motion sequences This is accomplished using the animation capability.

combine these individual moves into motion sequences. The sequences can then be played back to permit viewing of the robot work cycle. The animation speed can be controlled and the user can utilize the zooming capability of the computer graphics system so as to visually verify that the robot arm clears the various obstacles that might be in the way during the motion sequences.

Analyze cycle times Cycle time analysis can be performed in PLACE to determine the time required to accomplish the work cycle. The basic computations for element times are similar to the RTM calculations from the previous section in this chapter.

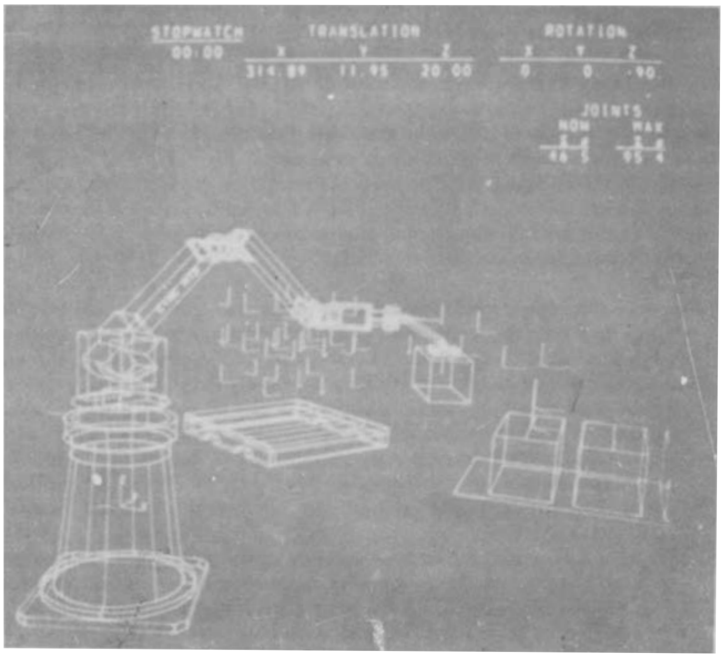


Fig. 11.12 *PLACE graphics simulation of robot cell for unloading cartons from conveyor onto pallet. (Photo courtesy: Computer-Aided Design Laboratory, Lehigh University)*

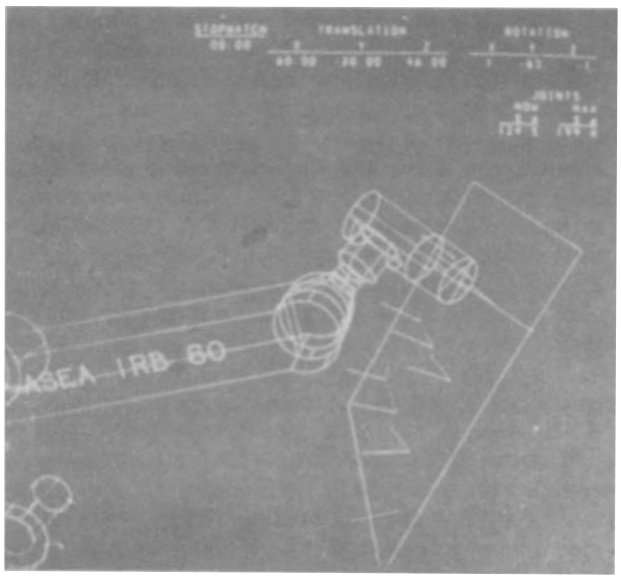


Fig. 11.13 *Computer graphics simulation of robot with tool as end effector showing close-up capability of PLACE. (Photo courtesy: Computer-Aided Design Laboratory, Lehigh University)*

The BUILD module can be used to construct three-dimensional computer graphics models of various robots that might be components of a workcell. The geometric characteristics of the robot (e.g., size, shape, etc.) are entered into the data base along with the design specifications of the robot. These specifications must include the number of degrees of freedom, the joint motions, joint travel limits, and other similar constraints on the robot's mechanical operation. Based on these parameters, the BUILD program creates the kinematic motion control equations that are used in PLACE. This relieves the workcell designer from the need to perform a detailed kinematic analysis for the robot motion cycle.

The COMMAND module is designed to permit off-line programming of robots. The limitations of current programming methods were discussed in Chaps. 8 and 9. Certainly one of the biggest limitations from a production viewpoint is that the robot itself must be employed during the programming procedure to teach it the locations of the various points in space, usually with a teach pendant. Even the textual languages require this definition of point locations using the teach pendant. It is a generally held belief among robotics engineers that off-line programming of robotics will probably require some form of three-dimensional computer graphics simulation in order to be practicable. COMMAND has been developed as step in the direction of off-line robot programming.

The COMMAND programming module permits certain portions of the robot program to be written in the textual language for the particular robot model. Opening and closing the gripper and other process-oriented commands are examples of the kinds of statements permitted. To program the motion cycle, the COMMAND module interacts with the PLACE module by allowing the user to call the various motion sequences that have been developed during cell design and evaluation. When the program has been developed, it is then translated into the language code for the particular robot controller to be used in the application. This step is similar to postprocessing in numerical control part programming. Its disadvantage is that a unique translator (postprocessor) must be available for each robot controller.

The final module in the PLACE package is ADJUST. This module is designed to address a problem area that relates to off-line programming. The problem is the likely existence of discrepancies between the actual robot workcell and the computer model of the cell that resides in the CAD/CAM data base. These discrepancies were mentioned previously in the discussion of world modeling in Chap. 9. There will undoubtedly be certain positional errors that exist between the actual physical objects in the workcell and the corresponding objects in the model. ADJUST provides the mechanism to calibrate the computer model for these errors by permitting positional data for the actual cell to be entered to update the PLACE model. This, of course, must be done after the robot cell has been installed in the factory. The original PLACE model was created during the design of the cell before its installation.

The calibration process involves the use of a special probe mounted in the position of the 'working point' (either attached to the wrist mounting plate or held by the gripper). A calibration program is written using the COMMAND module to move the robot through a series of specific test points in the cell. Corrections to the geometric model in PLACE are made according to the positional errors at each of the test points. These corrections are in turn made to the COMMAND program, thereby updating the robot application program.

roblems

11.1 For the machine cycle times from Example 11.1, determine the amount of machine interference and the amount of robot idle time (expressed as a percent) in a robot cell composed of two machines. Sketch the robot and machine process time chart similar to Fig. 11.6 to analyze the problem.

11.2 For the machine cycle times from Example 11.1, determine the amount of machine interference and the amount of robot idle time (expressed as a percent) in a robot cell composed of four machines. Sketch the robot and machine process time chart similar to Fig. 11.6 to analyze the problem.

11.3 Three machines will be organized in a machine cell using a robot to load and unload the machines. The cycle times of the three machines are given as

= 30 s, service time = 20 s
= 15 s, service time = 10 s
= 20 s, service time = 10 s

Determine the best sequencing of these activities using a robot and machine process time chart similar to Fig. 11.6 to analyze the problem. Determine the amount of machine interference and the amount of robot idle time (expressed as a percent) in the cell.

11.4 Make a list of the interlocks required for the workcell of Example 11.3 in the text. For each interlock indicate whether it is an input interlock or an output

sense before the signal is sent to the workcell controller. For each output

from the workcell controller.

11.5

appropriate means of implementing the interlock. Use the list of sensors in Table 6.2 of Chap. 6 for reference.

11.6 Make a list of the interlocks required for the workcell of Example 11.4 in the text. For each interlock indicate whether it is an input interlock or an output

sense before the signal is sent to the workcell controller. For each output

from the workcell controller.

11.7

appropriate means of implementing the interlock. Use the list of sensors in Table 6.2 of Chap. 6 for reference.

11.8 A robot workcell is to be installed for a plastic molding operation. The cell will consist of a large robot, the molding machine (which operates on an automatic cycle), and a belt conveyor for delivering the molded parts out of the cell. The

2. The

molding machine has overall dimensions of 50 in. in width by 120 in. in length. When the mold opens, the opening is 18 in. wide. The center of the mold is located at the center of the 50 in. machine width. The belt conveyor is 12 in. wide.

Note: For this problem the reader may want to refer ahead to Chap. 13 (Sec. 13.3) in which the plastic molding operation is discussed.

(a) Determine the likely sequence of activities in the work cycle. Use a list of steps similar to the format in Examples 11.2 and 11.3.

(b) What type of workcell layout is the most logical for this case?

(c) Make a sketch of the workcell, showing relative positions of the different pieces of equipment in the cell.

(d) Make a list of the interlocks required for the workcell. For each interlock indicate whether it is an input interlock or an output interlock. For

controller.

(e) *d*), determine an appropriate means of implementing the interlock. Use the list of sensors in Table 6.2 of Chap. 6 as a reference.

11.9 The sketch in Fig. P11.9 shows a proposed robot cell designed to process workparts through a certain industrial operation. The cell consists of the robot, a conveyor, and a processing machine. These components all operate under the robot cell controller. Parts (each part weighing 3 lb) arrive on the conveyor in a pallet (four parts to a pallet), are picked out of the pallet by the robot, loaded into the processing machine, processed, and unloaded by the robot into the same position on the pallet. When all four parts are processed in this manner, an interlock signal from the robot cell controller activates the conveyor to deliver the current pallet out of the cell and to deliver a new

Pallet in position with four raw workparts for start of cycle. Robot moves

moves to machine at 0.6 ft/sec, and loads into machine. The machine loading adds 2.0 s to the move. Robot arm moves to safe position 6 in. from part at a speed of 0.3 ft/sec.

Machine processes part for 20 s.

Robot arm moves from safe position at 0.3 ft/sec, unloads part, and moves it back to same pallet position at a speed of 0.6 ft/sec. Unloading requires 2.0 s. Part must be inserted back into pallet from 4 in. above. Speed of insertion is 0.2 ft/sec. Depart to 4 in. above pallet at 0.2 ft/sec.

Move to second pallet position (speed = 0.5 ft/sec) and repeat cycle.

Move to third pallet position (speed = 0.5 ft/sec) and repeat cycle.

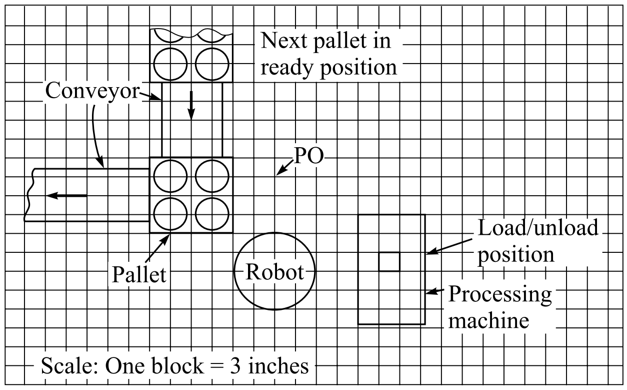


Fig. P11.9

References

1. M. B. Clifton and J. B. Ochs, 'An Interactive Computer Graphics Simulator of VAL Programming Language of the Unimation PUMA Robot,' *Proceedings*, IEEE Computer Society of the COMPCON Fall 1983 Conference, Arlington, VA, September 1983.
2. J. F. Engelberger, *Robotics in Practice,* AMACOM (American Management Association), New York, 1980, chaps. 4 and 5.
3. M. P. Groover, *Automation*, *Production Systems*, *and Computer-Aided Manufacturing*, Prentice-Hall, Englewood Cliffs, NJ, 1980, chaps. 4, 6, and 11.
4. M. P. Groover and E. W. Zimmers, Jr., *CAD/CAM: Computer-Aided Design and Manufacturing*, Prentice-Hall, Englewood Cliffs, NJ, 1984, chap. 10.
5. P. Howie, 'Graphic Simulation for Off-line Robot Programming,' *Robotics Today*, February 1984, pp. 63–66.
6. H. Lechtman and S. Y. Nof, 'A User's Guide to the RTM Analyzer,' *Technical Report*, Research Program on Advanced Industrial Robot Control, School of Industrial Engineering, Purdue University, 1981.
7. H. Lechtman and S. Y. Nof, 'Robot Performance Models Based on the RTM Method,' *Technical Report*, Research Program on Advanced Industrial Robot Control, School of Industrial Engineering, Purdue University, 1981.
8. S. Y. Nof and H. Lechtman, 'The RTM Method of Analyzing Robot Work,' *Industrial Engineering*, April 1982, pp. 38–48.
9. N. G. Odrey, 'Operational Strategies for Error Recovery Within a Manufacturing Work-station,' *Proposal for Research*, Lehigh University, May 1984.
10. R. N. Stauffer, 'Robot System Simulation,' *Robotics Today*, June 1984, pp. 81–90.
11. R. Thomas, 'Designing Controls for Robotics Work Cells,' *Industrial Engineering*, May 1983, pp. 34–39.

12. L. L. Toepperwein, M. T. Blackman, et al., 'ICAM Robotics Application Guide,' *Technical Report AFWAL-TR-80-4042*, Vol. II, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Ohio, April 1980.
13. Unimation, Inc., *Programming Manual—User's Guide to VAL II* (398Tl), Version 1.1, Danbury, CT, August 1984.

Economic Analysis for Robotics

Introduction

for a robotics project, there is also the economic issue. Will the robot justify itself economically? The economic analysis for any proposed engineering project is of considerable importance in most companies because management usually decides

we will consider the economic analysis of a robot project. We discuss the various

describe several methods for analyzing these factors to determine the economic merits of the project.

12.1 ECONOMIC ANALYSIS: BASIC DATA REQUIRED

To perform the economic analysis of a proposed robot project, certain basic information is needed about the project. This information includes the type of project being considered, the cost of the robot installation, the production cycle time, and the

12.1.1 Type of Robot Installation

There are two basic categories of robot installations that are commonly encountered.

there is a need for a new facility, and a robot installation represents one of the possible

are compared and the best alternative is selected, assuming it meets the company's investment criteria. The second situation is the robot installation to replace a current method of operation. The present method typically involves a production operation that is performed manually, and the robot would be used somehow to substitute for

absolute merits of the robot method.

perform the economic analysis. The following subsection discusses the kinds of cost and operating data that are used to analyze the alternative investment projects. The

12.1.2 Cost Data Required for the Analysis

The cost data required to perform the economic analysis of a robot project divide into two types: investment costs and operating costs. The investment costs include the purchase cost of the robot and the engineering costs associated with its installation

costs typically encountered in robot projects. The operating costs include the cost of

with the robot cell operation. The table lists most of the major operating costs for a

to identify the cost savings that will result from the use of a robot as compared

alternative methods. Material savings, scrap reductions, and advantages resulting

savings between the alternatives.

Table 12.1 *Direct costs associated with robot project.*

A. Investment cost

1. *Robot purchase cost*—The basic price of the robot equipped from the manufacturer with the proper options (excluding end effector) to perform the application.
2. *Engineering costs*—The costs of planning and design by the user company's engineering staff to install the robot.
3. *Installation costs*—This includes the labor and materials needed to prepare the installation site (provision for utilities, floor preparation, etc.).
4. *Special tooling*—This includes the cost of the end effector, parts positioners, and other fixtures and tools required to operate the work cell.
5. *Miscellaneous costs*—This covers the additional investment costs not included by any of the above categories (e.g., other equipment needed for the cell).

B. Operating costs and savings

6. *Direct labor cost*—The direct labor cost associated with the operation of the robot cell. Fringe benefits are usually included in the calculation of direct labor rate, but other overhead costs are excluded.

Contd.

7. *Indirect labor cost*—The indirect labor costs that can be directly allocated to the operation of the robot cell. These costs include supervision, setup, programming, and other personnel costs not included in category 6 above.
8. *Maintenance*—This covers the anticipated costs of maintenance and repair for the robot cell. These costs are included under this separate heading rather than in category 7 because the maintenance costs involve not only indirect labor (the maintenance crew) but also materials (replacement parts) and service calls by the robot manufacturer. A reasonable 'rule of thumb' in the absence of better data is that the annual maintenance cost for the robot will be approximately 10 percent of the purchase price (category 1).
9. *Utilities*—This includes the cost of utilities to operate the robot cell (e.g., electricity, air pressure, gas). These are usually minor costs compared to the above items.
10. *Training*—Training might be considered to be an investment cost because much of the training required for the installation will occur as a first cost of the installation. However, training should be a continuing activity, and so it is included as an operating cost.

The manner in which these investment costs and operating costs play out over

At the beginning of the project, the investment costs are being paid into the project with no immediate return. When the installation is completed and the project begins operation, the operating costs begin. However, there is also a compensating cash

operating cost. The difference between the revenues and the operating costs is the

the project.

Fig. 12.1

to recover its investment costs in the project in a relatively short period of time. The

period, and this payback period represents one of several methods for evaluating investment alternatives. The payback period method, as well as several other methods for analyzing the economics of robot projects are discussed in the following section.

12.2 METHODS OF ECONOMIC ANALYSIS

The three methods for analyzing investments and comparing investment alternatives that are in common use in industry are:

however, this is not always the case. We assume that the reader has some familiarity with the principles of engineering economy.

12.2.1 Payback Method

simple formula

$$= \frac{\text{IC}}{\text{NACF}}$$

where = the payback period

i is

used to identify the year in the following.

$$\sum_{i=1}^{n}$$

is determined so that the sum of the annual project

The reader should note that we have adopted the logical convention that costs are

since revenues derived from the robot project would be greater than the operating

beginning of the year or at the end of the year. Any investments are assumed to be

Most companies today require paybacks of no more than two or three years.

Example 12.1 Suppose that the total investment cost is estimated to be

$$= \frac{100,000}{45,000}$$

One of the disadvantages of the payback period method is that it ignores the time

minimum rate of return from its investments. The other two methods to be discussed do include this consideration.

12.2.2 Equivalent Uniform Annual Cost Method

associated with engineering economy calculations. We present a tabulation of these

greater than zero, this is interpreted to mean that the actual rate of return associated with the investment is greater than the MARR used by the company as the criterion.

Example 12.2

in determining the values for any interest factors required in our calculations.

be converted to its equivalent uniform annual cash value using the capital recovery

follows.

A/*P*

Since the resulting uniform annual cost value is positive, this robot project would be a good investment.

12.2.3 Return on Investment (ROI) Method

project based on the estimated costs and revenues. This rate of return is then compared with the company's minimum attractive rate of return to decide whether

Example 12.3 Again the same data are used from our previous two

A/*P*, *i*

A/*P*, *i*

A/*P* factor for

For *i* *A*/*P*

For *i* *A*/*P*

By interpolation, the rate of return for our problem turns out to be *i*

A/*P*, *i*

A/*P*

procedure because there is more than one interest factor that must be used in the are several complications that are encountered in the economic analysis of robot applications problems. These complications are not necessarily unique to robotics

12.3 SUBSEQUENT USE OF THE ROBOT

Many automation projects involve pieces of equipment that have service lives corresponding to the life cycle of the product that will be made on the equipment. The automated equipment is very specialized to manufacture the particular product

equipment often becomes obsolete and is of no further use or value to the company.

By contrast, robots represent programmable automation that can be used again after the current product life cycle is over. This is an attractive feature of an industrial

current production use. On the other hand, this feature tends to promote the use of

the investment cost of the robot for the application. One way of dealing with this problem is to recognize the opportunity for subsequent use of the robot by assigning it a salvage value at the end of the current project. The current project may have a

or four projects before either wearing out or becoming technologically obsolete.

The question that arises is this: How is the salvage value at the end of the current project determined? A reasonable procedure for assessing the salvage value at the end of the current application project is to use the straight line method of depreciation for

life. This is different from the service life of the current project on which the new

years. The service life for the robot would be longer than that.

Suppose the anticipated service life for the robot is eight years. At the end of that

To get the salvage value at the end of the project life, we multiply this annual depreciation by the project life and subtract the amount from the initial cost.

service life. Also, the company may elect to use a different depreciation method to

Example 12.4

will be reusable for another project when the life of the current project is over. The

from the current project.

follows:

A/*P*, *i* *A*/*F*, *i*

one unknown term.

Try *i*

Try *i*

presence of the salvage value, representing an evaluation of the robot's capacity to be reused on a subsequent application project, has increased the rate of return on the

12.4 DIFFERENCES IN PRODUCTION RATES

issue which often arises is the difference in production rates between the alternative methods. The automated method usually outproduces the manual method, and this advantage must be taken into account in the analysis. The same issue is relevant when a robot is used to automate an operation. The robot may be capable of working faster and with fewer rest breaks than the human operator. On the other hand, what sometimes happens in a robot application is that the robot cannot work quite as fast as a human operator, but the company may decide to use the robot for two or three shift operations, whereas it felt limited to one shift when the task was performed manually.

These changes create differences in daily production rates which would presumably affect revenues assignable to the alternatives. The easiest case to analyze is where the value added by the operation is known per product. This value added can then be used to determine revenues for the alternatives.

Example 12.5 Suppose two alternative methods are to be compared, one a manual method and the other a robot cell. The robot cell will

MARR is used by the company and the anticipated service life of either method is

period can be determined from

$$= \frac{29,000}{14,000}$$

The rate of return attributable to the manual operation would be determined by

$A/P, i$

$A/P, i$

attractive rate of return criterion for the company.

For the automated robot method, the annual revenues from the operation will be

therefore be

$$= \frac{100,000}{75,000 - 20,000}$$

The rate of return equation would be set up as follows:

$A/P, i$ $A/F, i$

The robot method seems preferable on the basis of both payback period and rate

calculation, whereas its effect is ignored in the payback method. One could conceive of a situation in which the payback method favored one alternative while the rate

method, the salvage value does not enter the analysis unless the calculated payback

the payback period becomes equal to the project life, so long as the total project

greater than the total revenues during the project life, then the project is clearly not worthwhile and the concept of payback period becomes blurred.

by a given operation because the operation is one of a sequence of processing steps

approach is to determine the unit cost per product made in the operation for the two

Example 12.6 costs for the manual method and the robot method. We will

determining the values of equivalent uniform annual cost for the two production

derived from the units sold, we have

A/P

The number of units produced annually in the operation would be

would be

A/P A/F

The number of units produced annually in the operation would be

The robot method would be favored.

revenues are not known. The assumption is that the operation must be performed by one method or another, and our problem is simply to determine the method whose cost is the lowest.

12.5 OTHER FACTORS MORE DIFFICULT TO QUANTIFY

in a robotics application project, there are other possible sources of costs and savings

costs and savings, and they include such factors as inventory savings, scrap savings, and reduced downtime. A comprehensive listing of these potential factors is presented

investment is much larger than the rate of return criterion used by the company. Meyer presents an approach for estimating some of these indirect factors, and we recommend this paper as a reference to the interested reader.

Table 12.2

1. *In-process inventories*—The savings in in-process inventory result from a reduced manufacturing lead time with a robot installation. Shorter operation cycle times, use of the second and third shifts, and the possibility for combining separate operations into one robot cell are reasons why the manufacturing lead time is reduced.
2. *Finished inventories*—The technical feasibility of using robots in flexible, adaptable manufacturing cells and assembly systems provides the opportunity for reducing the production batch size. Smaller lot sizes translate into lower final inventories.
3. *Materials savings*—In some applications, robots use the raw materials more efficiently in the production process. This leads to a lower usage rate of these materials. Robotic spray painting operations are an example of these savings; the consistency with which the paint is applied by a robot allows a reduction in the total amount of paint consumed as compared to a manual spray paint operation.
4. *Less scrap and rework*—The avoidance of human error in the operation, the consistency of the robot cycle (both in terms of timing and positional repeatability) are some of the factors that contribute to a more uniform product and a reduction in the scrap and rework rates when robots are used.
5. *Equipment utilization*—When robots are used to automate an operation, the utilization of the existing equipment generally increases. The reasons for the increase include the opportunity to convert to multishift operation of the equipment when robots are integrated into the operation, fewer breaks in the shift as compared to the requirements in a manual operation.
6. *Material handling*—When several operations are combined into a single robot cell, the amount of material handling in the plant is reduced.
7. *Floor space*—A well-designed robot cell typically reduces the amount of floor space required for the operation. This is especially true when several operations, previously accomplished at separate workstations, are combined into a single robot workcell.

Finally, there are considerations in deciding about a robot installation that are virtually impossible to quantify. These considerations include improvements in safety by removing human operators from the immediate dangers of the production operation, reduced dependence on direct labor and the associated personnel problems, better customer relations from improved delivery schedules and better quality, and the ability to use the automated factory as a showplace to impress customers and in a company's overall strategy for implementing robotics and automation. They are generally considerations which tend to favor the application of robotics.

12.6 ROBOT PROJECT ANALYSIS FORM

Many companies have developed a standard investment analysis form that they use to show the results of their economic evaluation of a proposed project. Several of the

analysis form that can be used to display the costs and savings for a proposed robot project according to the procedures we have developed in this chapter.

purpose in this chapter has been to present an introductory treatment of investment

Robot Economic Analysis From

Contd.

Fig. 12.2

roblems

12.1

which time the project will be retired. The estimated salvage value of the robot

(a)

(b)

per cent rate of return in the interest calculations.

(c)

investment.

12.2

which will prevent the full revenues from being realized.

12.3 Two production methods are to be compared, one a robot cell and the other a

service life for the project is three years.

(a)

method.

(b)

investment method.

(c)

would you place more reliance on and why?

12.4 A robot cell has been proposed for a certain industrial operation. The robot cell

payback period and the rate of return from the investment.

12.5

period and the rate of return from the investment.

12.6

into the problem.

12.7 Two manually operated workstations are currently used to produce a certain

A robot has been proposed to replace the two human operators. The robot is

12.8

robot was originally installed is now at the end of its life cycle, and the robot is being proposed for a new project which will last four years. The engineering

(a)

(b)
the robot?

(c) The robot is being proposed to replace a manual operation currently

over the manual operation? Support your answer.

References

,

Robot Applications in Manufacturing

PART FIVE

Current-day robot applications include a wide variety of production operations. For purposes of organization in this book, the operations will be classified into the following categories:

1. Material transfer and machine loading/unloading These are applications in which the robot grasps and moves a workpart from one location to another. This category includes applications in which the robot transfers parts into and out of a production machine. Examples of these load/unload operations include metal-machining operations, die casting, plastic molding, and certain forging operations. Material transfer and machine loading and unloading applications are covered in Chap. 13.

2. Processing operations These are operations in which the robot uses a tool as an end effector to accomplish some processing operation on a workpart that is positioned for the robot during the work cycle. Spot welding, arc welding, spray painting, and certain machining operations fall into this application category. Chapter 14 deals with these robot applications.

3. Assembly and inspection Assembly and inspection are relatively new applications for robots. (We are excluding welding operations from this group though they are assembly operations.) The robot is used to put components together into an assembly, or the robot is used to perform some form of automated inspection operation. More and more robots in the future will be equipped with vision capability to facilitate the performance of these operations. We discuss robot assembly and inspection in Chap. 15.

13 Material Transfer and Machine Loading/Unloading

Introduction

There are many robot applications in which the robot is required to move a workpart or other material from one location to another. The most basic of these applications is where the robot picks the part up from one position and transfers it to another position. In other applications, the robot is used to load and/or unload a production machine of some type. In this book we divide material-handling applications into two specific categories:

1. Material transfer applications
2. Machine loading/unloading applications

There are other robot applications which involve parts handling. These include assembly operations and holding parts during inspection. Assembly and inspection applications are treated in Chap. 15. Before discussing material transfer and machine loading/unloading applications, let us review some of the considerations that should be examined when robots are used for material handling.

13.1 GENERAL CONSIDERATIONS IN ROBOT MATERIAL HANDLING

In planning an application in which the robot will be used to transfer parts, load a machine, or other similar operation, there are several considerations that must be reviewed. Most of these considerations have been discussed in previous chapters of the book, and we itemize them below as a reference checklist.

1. Part positioning and orientation In most parts-handling applications the parts must be presented to the robot in a known position and orientation. Robots used in these applications do not generally possess highly sophisticated sensors (e.g., machine vision) that would enable them to seek out a part and identify its orientation before picking it up.

2. Gripper design Special end effectors must be designed for the robot to grasp and hold the workpart during the handling operation. Design considerations for these grippers were discussed in Chap. 5.

3. Minimum distances moved The material-handling application should be planned so as to minimize the distances that the parts must be moved. This can be accomplished by proper design of the workcell layout (e.g., keeping the equipment in the cell close together), by proper gripper design (e.g., using a double gripper in a machine loading/unloading operation), and by careful study of the robot motion cycle.

4. Robot work volume The cell layout must be designed with proper consideration given to the robot's capability to reach the required extreme locations in the cell and still allow room to maneuver the gripper.

5. Robot weight capacity There is an obvious limitation on the materialhandling operation that the load capacity of the robot must not be exceeded. A robot with sufficient weight-carrying capacity must be specified for the application.

6. Accuracy and repeatability Some applications require the materials to be handled with very high precision. Other applications are less demanding in this respect. The robot must be specified accordingly.

7. Robot configuration, degrees of freedom, and control Many parts transfer operations are simple enough that they can be accomplished by a robot with two to four joints of motion. Machine-loading applications often require more degrees of freedom. Robot control requirements are unsophisticated for most material-handling operations. Palletizing operations, and picking parts from a moving conveyor are examples where the control requirements are more demanding.

8. Machine utilization problems It is important for the application to effectively utilize all pieces of equipment in the cell. In a machine loading/unloading operation, it is common for the robot to be idle while the machine is working, and the machine to be idle while the robot is working. In cases where a long machine cycle is involved, the robot is idle a high proportion of the time. To increase the utilization of the robot, consideration should be given to the possibility for the robot to service more than a single machine. One of the problems arising in the multi machine cell is machine interference, discussed in Chap. 11.

We now proceed to deal with the specific cases of material transfer and machine loading/unloading applications in the following two sections.

13.2 MATERIAL TRANSFER APPLICATIONS

Material transfer applications are defined as operations in which the primary objective is to move a part from one location to another location. They are usually considered to be among the most straightforward of robot applications to implement. The applications usually require a relatively unsophisticated robot, and the interlocking requirements with other equipment are typically uncomplicated. These applications are sometimes called pick-and-place operations because the robot simply picks the part from one location and places it in another location. Some material transfer applications have motion patterns that change from cycle to cycle, thus requiring a more sophisticated robot. Palletizing and depalletizing operations are examples of this more complicated case. In this type of application, the robot must place each part in a different location on the pallet, thus forcing the robot to remember or compute a separate motion cycle until the pallet is fully loaded.

13.2.1 Pick-and-Place Operations

As defined above, pick-and-place operations involve tasks in which the robot picks up the part at one location and moves it to another location. In the simplest case, the part is presented to the robot by some mechanical feeding device or conveyor in a known location and orientation. The known location is a stationary location, achieved either by stopping the conveyor at the appropriate position, or by using a mechanical stop to hold the part at the stationary location. An input interlock (commonly based on using a simple limit switch) would be designed to indicate that the part is in position and ready for pickup. The robot would grasp the part, pick it up, move it, and position it at a desired location. The orientation of the part remains unchanged during the move. The desired location is usually at a position where there is the capability to move the part out of the way for the next delivery by the robot. This basic case is illustrated in Fig. 13.1. In this simple case, the robot needs only 2 degrees of freedom. As shown in the figure, 1 degree of freedom is needed to lift the part from the pick up point and put it down at the drop off point. The second degree of freedom is required to move the part between these two positions. In some pick-and-place operations, a reorientation of the workpart is accomplished during the move. This part reorientation requires a robot with one or more additional degrees of freedom.

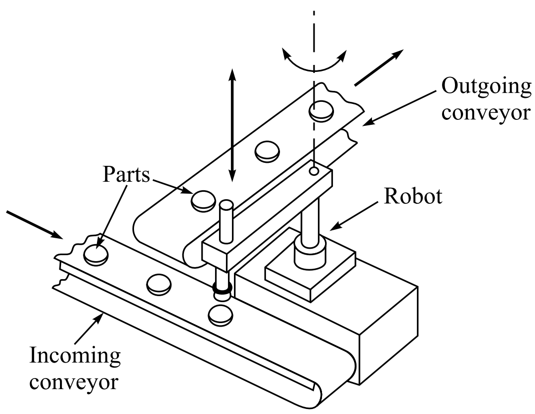


Fig. 13.1 *Simple pick-and-place operation.*

One complication encountered in material transfer operations is when the robot is required to track a moving pickup point. The general problems related to robot tracking systems were described in Chap. 11. In robotic materials handling, tracking arises when parts are carried along a continuously moving conveyor, typically an overhead hook conveyor, and the robot is required to pick part's from the conveyor. The opposite case is when the robot must put parts onto the moving conveyor. In either case a more sophisticated sensor-interlock system is required to determine the presence and location of the parts in the robot's tracking window.

Another complication arises in material transfer operations (as well as palletizing and other operations) when different objects are being handled by the same robot. In material transfer, a single conveyor might be used to move more than one type of part. The robot must be interfaced to some type of sensor system capable of distinguishing between the different parts so that the robot can execute the right program subroutine for the particular part. For example, there may be differences in the way the part must be retrieved from the conveyor due to part configuration, and the placement of the part by the robot may vary for different parts. In other cases where multiple items are handled, the information system which supports the workcell can be used to keep track of where each item is located in the workcell. The information support system would be used either in lieu of or in addition to sensors. The following example illustrates a material-handling situation which uses both sensors and a hierarchical information system to support the control of the robot cell.

Example 13.1 This is a good example of a complex material-handling robot application located at the Poughkeepsie plant of IBM Corporation. The application is described in an article by Giacobbe.[3] The purpose of the operation is to write and package 8-in. diskettes for the large computer systems produced at the Poughkeepsie plant. Prior to the robot cell, the operation was performed manually and was prone to error and damage to the diskettes. A robot cell was designed and installed to overcome these problems.

The robot cell includes: a Unimation PUMA 560 robot, two diskette writers, a box storage carousel, two envelope printers, and two label printers. The workcell controller is an IBM Series/1 computer. Figure 13.2 shows an overall view of the workcell. The individual tasks performed by the robot include getting the diskettes, labeling and packing them, setting up the boxes, closing them, labeling them, and storing them. The end effector is designed with mechanical fingers to handle the boxes and box lids, and vacuum pads to handle the flexible diskettes and diskette envelopes.

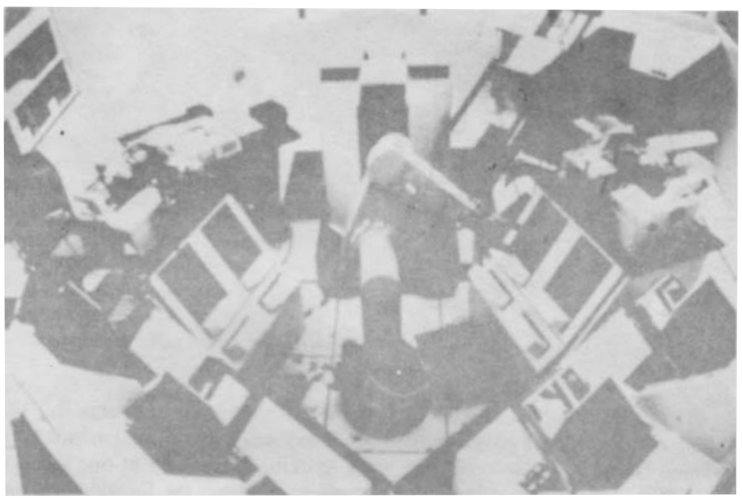


Fig. 13.2 *Robot workcell for writing diskettes. PUMA 560 robot is used for handling diskettes, envelopes, and boxes in the cell. (Photo courtesy: IBM Corporation)*

The Series/1 computer provides the overall coordination of the activities in the workcell. It is linked to a host computer in the plant which supplies diskette data and order information to the diskette writers. The host computer also receives data requests and status information from the robot cell controller. Giacobbe reports that about 90 percent of the coding for the system is for automatic error recovery. For example, if one of the diskettes is dropped during handling, sensors detect this error and the system requests that a duplicate diskette be written.

13.2.3 Palletizing and Related Operations

The use of pallets for materials handling and storage in industry is widespread. Instead of handling individual cartons or other containers, a large number of these containers are placed on a pallet, and the pallet is then handled. The pallets can be moved mechanically within the plant or warehouse by fork lift trucks or conveyors. Shipments of palletized product to the customer are very common because of the convenience in handling, both at the manufacturer's warehouse and the customer's receiving department.

The only handling of the individual cartons arises when the product is placed onto the pallet (palletizing) or when it is removed from the pallet (depalletizing). We will discuss palletizing as the generic operation. The loading of cartons onto pallets is typically heavy work, performed manually by unskilled labor. It is also repetitive work (picking cartons up at one location and putting them down at another location) except that the locations change from carton to carton. A typical pallet configuration is illustrated in Fig. 13.3, showing how each container must be placed at a different location on the pallet. The variation in carton location is in three dimensions, not simply two dimensions, since the pallets are usually stacked on top of each other in layers.

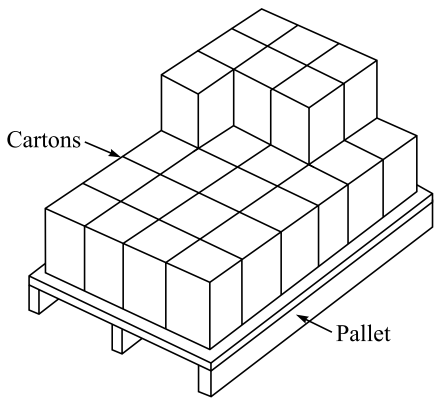


Fig. 13.3 *A typical pallet configuration.*

Robots can be programmed to perform this type of work. Because the motion pattern varies in the palletizing operation, a computer-controlled robot using a high-level programming language is convenient. This feature facilitates the mathematical computation of the different pallet locations required during the loading of a given pallet. The kinds of programming capabilities required in palletizing were discussed in our previous chapters on robot programming. A less sophisticated robot limited to leadthrough programming can also be used, but the programming becomes laborious because each individual carton location on the pallet must be individually taught. Another technical problem that must be addressed is that when humans perform palletizing, the cartons are often randomly located prior to loading. Unless some sensor scheme is used to identify these carton locations, they must be delivered to a known pick up point for the robot.

There are a number of variations on the palletizing operation, all of which use a similar work cycle and a robot with the same general features needed for the generic case. These other operations include:

Depalletizing operations, the reverse of palletizing, in which the robot removes cartons from a pallet and places them onto a conveyor or other location.

Inserting parts into cartons from a conveyor. This is similar to palletizing.

Figure 13.4 shows an example of this operation.

Removing parts from cartons. This is the reverse of the preceding situation.

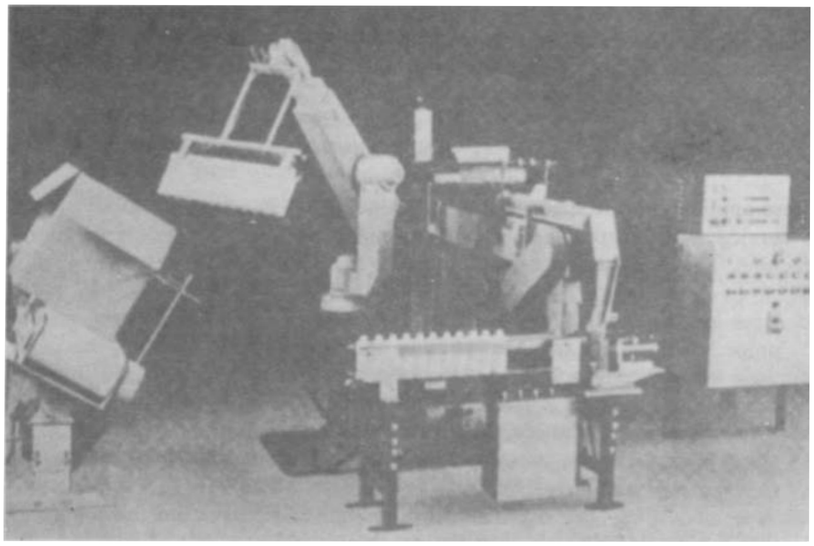


Fig. 13.4 *MAKER 100 Robot used to unload plastic bottles from a collector and to insert them into a cardboard carton. (Photo courtesy: United States Robots, subsidiary of Square D Company)*

Stacking and unstacking operations, in which objects (usually flat objects such as metal sheets or plates) are stacked on top of each other.

In palletizing and related operations, the robot may be called on to load different pallets differently. Reasons for these differences would include the following: The pallets may vary in size; different products may be loaded onto the pallets; and there may be differences in the numbers and combinations of cartons going to different customers. To deal with these variations, methods of identifying the cartons and/or pallets and the way in which they are to be loaded or unloaded must be devised.

problem. Differences in the loading or unloading of the pallets must be accomplished by means of program subroutines which can be called by the workcell controller. For depalletizing operations, the optical reader system would identify the pallet and the appropriate unloading subroutine would then be applied for that pallet. For palletizing operations, the systems problems can become more complicated because there may

robot were used to palletize cartons for different customer orders, it is conceivable that each customer order would be different. A method would have to be devised for delivering the correct combination of cartons to the palletizing workstation, and integrating that process with the robot loading procedure. In the future, robots may

proportions.

13.3 MACHINE LOADING AND UNLOADING

These applications are material-handling operations in which the robot is used to service a production machine by transferring parts to and/or from the machine. There

1. Machine load/unload The robot loads a raw work part into the process and

2. Machine loading The robot must load the raw work part or materials into the machine but the part is ejected from the machine by some other means. In a pressworking operation, the robot may be programmed to load sheet metal blanks

3. Machine unloading
that are loaded directly into the machine without robot assistance. The robot unloads the part from the machine. Examples in this category include die casting and plastic modeling applications.

production machine, the robot, and some form of parts delivery system. To increase the productivity of the cell and the utilization of the robot, the cell may include more than a single production machine. This is desirable when the automatic machine cycle is relatively long, hence causing the robot to be idle a high proportion of the time. Some cells are designed so that each machine performs the same identical

parts follow a different sequence of operations at different machines in the cell. In either case, the robot is used to perform the parts handling function for the machines in the cell.

Robots have been successfully applied to accomplish the loading and/or unloading

1. Die casting
2. Plastic molding

3. Forging and related operations
4. Machining operations
5. Stamping press operations

We will discuss these applications in the following subsections. For each application, a brief description of the manufacturing process will be given. More detailed descriptions of the processes are to be found in other references.[1,2,7,8]

13.3.1 Die Casting

Die casting is a manufacturing process in which molten metal is forced into the cavity of a mold under high pressure. The mold is called a die (hence the name,

die-casted parts include alloys of zinc, tin, lead, aluminum, magnesium, and copper.

The die consists of two halves that are opened and closed by a die casting machine. During operation the die is closed and molten metal is injected into the cavity by a

by pins which push the part away from the mold cavity. When the part is removed

is created during the casting process must be removed subsequently by a trimming operation which cuts around the periphery of the part. Thus, the typical die-casting production cycle consists of casting, removing the part from the machine, quenching, and trimming.

openings of the die per hour, depending on type of machine, the metal being cast, and the design of the part. For small parts, the die can be designed with more than one cavity, thus multiplying the number of parts made for each casting cycle. The die-casting machines have traditionally been tended by human operators. The work tends to be hot, repetitive, dirty, and generally unpleasant for humans.

2

The die-casting process represents a relatively straightforward application for industrial robots. The alterations required of the die-casting machine are minimal, and the interlocking of the robot cycle with the machine cycle can be accomplished by simple limit switches. Few problems are encountered in either the programming of the robot or the design of the gripper to remove the part from the machine when the die is opened. The process requires only that the robot unload the die-casting machine, since the metal is in the molten state before the part is formed. On some die-casting machines (called cold-chamber die-casting machines), the molten metal must be ladled from the melting container into the injection system. This part of the

13.3.2 Plastic Molding

Plastic molding is a batch-volume or high-volume manufacturing process used to

of processes, including compression molding, injection molding, thermoforming, blow molding, and extrusion. Injection molding is the most important commercially, and is the process in this group for which robots are most often used. The injection-molding operation is quite similar to die casting except for the differences in materials being processed. A thermoplastic material is introduced into the process in the form of small pellets or granules from a storage hopper. It is heated in a heating chamber

mold cavity under high pressure. The plastic travels from the heating chamber into the part cavity through a sprue-and-runner network that is designed into the mold.

of the mold come together. If too little material is injected into the cavity, sink holes and other defects are created in the part, rendering it unacceptable. When the plastic

the mold.

Injection molding is accomplished using an injection-molding machine, a highly sophisticated production machine capable of maintaining close control over the important process parameters such as temperature, pressure, and the amount of material injected into the mold cavity. Traditionally, injection-molding machines have been operated on a semi-automatic cycle, with human operators used to remove the parts from the mold. Many injection-molding operations can be fully automated so long as a method can be developed for removing the parts from the mold at the end of the molding cycle. If a part sticks in the mold, considerable damage to the mold can occur when it closes at the beginning of the next cycle. Methods of removing

directing an air stream to force the parts out of the mold, and the use of robots to reach into the mold and remove the parts. The selection of the method depends largely on the characteristics of the molding job (part size, weight, how many parts to be molded per shot).

Industrial robots are sometimes employed to unload injection-molding machines

One of the robot application problems in injection molding is that the production times are considerably longer than in die casting, hence causing the robot to be idle

time can be utilized to perform such tasks as cutting the parts from the sprue-and-

methods must be devised to accomplish these activities that do not rely on a human operator performing the unloading function. Cutting the parts from the sprue-and-runner system can be readily accomplished by the robot using a trimming apparatus

Another issue arising when long molding cycle times are involved is whether the robot should be used to tend one machine or two. If two molding machines are be different. This creates machine interference problems, in which one machine must wait for the robot because it is presently engaged in unloading the other machine. This waiting can lead to problems in over-heating of the plastic and upsetting of the delicate balance between the various process parameters in injection molding.

13.3.3 Forging and Related Operations

Forging is a metalworking process in which metal is pressed or hammered into the desired shape. It is one of the oldest processes and derived from the kinds of metalworking operations performed by blacksmiths in ancient times. It is most commonly performed as a hot working process in which the metal is heated to a high temperature prior to forging. It can also be done as a cold working process. Cold forging adds considerable strength to the metal and is used for high-quality products requiring this property such as hand tools (e.g., hammers and wrenches). Even in

formed part.

The term forging includes a variety of metalworking operations, some of which are candidates for automation using robots. These operations include die forging and upset forging. Other processes in the forging category include press forging and roll die forging. Generally, these processes do not lend themselves to the use of robots for parts loading and unloading of the machines.

Die forging is a process accomplished on a machine tool called a drop hammer in which the raw billet is hit one or more times between the upper and lower portions of a forging die. The die often has several cavities of different shapes which allow

shape. The drop hammer supplies mechanical energy to the operation by means of a heavy ram to which the upper portion of the forging die is attached. The ram is dropped onto the part, sometimes being accelerated by steam or air pressure. Die forging can be carried out either hot or cold.

the workpart (usually a cylindrical part) is increased by squeezing the material into the shape of a die. The formation of the head on a bolt is usually made by means of an upsetting operation. The process is performed by an upsetting machine, also called a *header*. The blank (unformed raw workpart) is clamped by the two halves of a die possessing the desired shape of the product. The die is open on one end, and a plunger is forced by the upsetting machine into the blank causing it to take the shape

automation to produce the parts. In other cases, where the production of parts is in medium-sized batches, automation can sometimes be accomplished using industrial robots.

Forging, especially hot forge operations, is one of the worst industrial jobs for humans. The environment is noisy and hot, with temperatures at the workplace well

furnace fumes, and lubricant mist. The operation itself is repetitive, often requiring considerable physical strength to move and manipulate the heavy parts during the operation. The human operator experiences the blows from the drop hammer directly through the grasping tongs used to hold the part and in the form of vibration through

The forging hammers and upsetting machines used for low- and medium-production runs are typically older machines, designed for manual operation, and do not lend themselves to the interfacing required for robotics automation.

justify the robot setup and programming effort for any single part.

The parts occasionally stick in the dies. This can be readily detected by a human operator but poses problems for the robot. To minimize the frequency of sticking, the human operator periodically sprays lubricant into the die openings. The robot would have to be equipped and programmed to do this also.

against these temperatures. Second, the gripper must be designed to withstand the shock from the hammer blows because the parts must typically be held in position by the robot during the process. Third, the gripper must be designed to accommodate substantial changes in the shape of the parts during successive hits in the forging cycle. Some aspects of the forging process require operator judgment. The part must

operation and the human operator often makes this judgment. A cold part would probably ruin the die. The raw workparts are generally placed in the heating furnace at random, and selecting the parts that are ready to be formed is an operator decision. Another problem is that different parts can require a different number of hammer

Each of these problem areas must be addressed in order for the robot forging application to be a success. Many of the problems are solved by making considerable use of interlocks and sensors. These devices permit the determination of such process variables as part temperature before processing, the presence of the workpart in the gripper, whether the part is stuck in the die, whether the robot arm is clear of the ram before operation of the drop hammer, and other factors.

13.3.4 Machining Operations

Machining is a metalworking process in which the shape of the part is changed by removing excess material with a cutting tool. It is considered to be a secondary

such as casting or forging has provided the basic shape of the part. There are a number of different categories of machining operations. The principal types include turning, drilling, milling, shaping, planing, and grinding. Commercially, machining is an important metalworking process and is widely used in many different products, ranging from those that are made in low quantities to those produced in very high numbers. In mid-volume and high-volume production, the operation is very repetitive with the same machining sequence being repeated on part after part.

The machine tools that perform machining operations have achieved a relatively high level of automation after many years of development. In particular, the use of computer control (e.g., computer numerical control and direct numerical control) permits this type of equipment to be interfaced with relative ease to similarly controlled equipment such as robots.

Robots have been successfully utilized to perform the loading and unloading functions in machining operations. The robot is typically used to load a raw workpart

part at the completion of the machining cycle. Figure 13.5 illustrates a machine tool

Fig. 13.5 *Unimate 2000 robot used to load and unload parts in a machine tool operation. (Photo courtesy: Unimation Inc.)*

The following robot features generally contribute to the success of the machine tool load/unload application[2]

1. Dual gripper The use of a dual gripper permits the robot to handle the raw

time to be reduced.

2. Up to six joint motions A large number of degrees of freedom of the arm and wrist are required to manipulate and position the part in the machine tool.

3. Good repeatability A relatively high level of precision is required to properly

4. Palletizing and depalletizing capability In midvolume production, the raw parts are sometimes most conveniently presented to the workcell and delivered away from the workcell on pallets. The robot's controller and programming capabilities must be sufficient to accommodate this requirement.

5. Programming features There are several desirable programming features that facilitate the use of robots in machining applications. In machine cells used for batch production of different parts, there is the need to perform some sort of changeover of the setup between batches. Part of this changeover procedure involves replacing the robot program for the previous batch with the program for the next batch. The robot should be able to accept disk, tape, or other storage medium for ease in changing programs. Another programming feature needed for machining is the capability to handle irregular elements, such as tool changes or pallet changes, in the program.

Example 13.2 This example illustrates some of the features of an automated machining cell consisting of a T3 robot, two turning centers, and an automated gaging station. Figure 13.6 shows the robot tending one of the turning centers. The raw workparts are castings which enter the workcell on a pallet

Fig. 13.6 *Cincinnati Milacron T3 robot performing a machine tool load/unload operation. See Example 13.2. (Photo courtesy: Cincinnati Milacron)*

right foreground. The robot picks up a raw workpart and exchanges the raw part in

gaging station, checked for the proper dimensions, and if determined to be within tolerance, is ready for loading into the second machine. The robot then exchanges the

That part is gaged in the automatic gaging station, and loaded onto the outgoing pallet if within tolerance.

The loading and unloading operations are accomplished from the rear of each turning center rather than the front of the machines. This means that the front of each machine is clear for operator access, tool replacement, and observation.

13.3.5 Stamping Press Operations

Stamping press operations are used to cut and form sheet metal parts. The process is performed by means of a die set held in a machine tool called a *press* (or stamping press). The sheet metal stock used as the raw material in the process comes in several

the press, the process can be made to operate in a highly automated manner at very

been performed by human workers, who must expose themselves to considerable jeopardy by placing their hands inside the press in order to load the blanks. During the last decade, the Occupational Safety and Health Act (OSHA) has required certain alterations in the press in order to make its operation safer. The economics of the OSHA requirements have persuaded many manufacturers to consider the use of robots for press loading as alternatives to human operators. Noise is another factor which makes pressworking an unfriendly environment for humans.

Robots are being used for handling parts in pressworking operations, largely as a

blanks into the press for the stamping operation. There are variations in the way this can be done. In forming operations, the robot can be used to hold the blank during the cycle so that the formed part is readily removed from the press. In the case of many cutting operations, the robot loads the blank into the press, and the parts fall through the die during the press cycle. Another robot application in pressworking involves the transfer of parts from one press to another to form an integrated pressworking cell. Figure 13.7 shows a *TLL*
pressworking operation.

One of the limiting factors in using industrial robots for press loading is the cycle time of the press. Cycle times of less than a second are not uncommon in pressworking. These cycle rates are too fast for currently available commercial robots. There is generally a direct relationship between the physical size of the part and the press cycle time required to make the part. Bigger presses are needed to stamp bigger parts and bigger presses are inherently slower. Accordingly, robots are typically used in pressworking for larger parts.

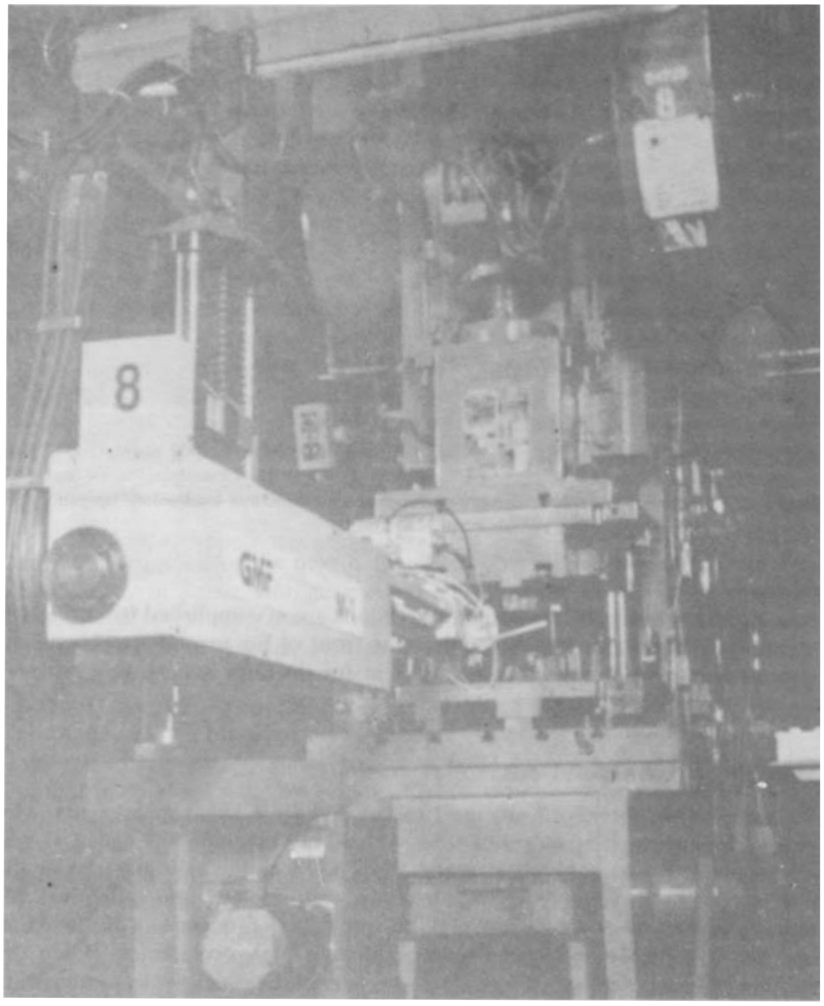


Fig. 13.7 *GMF robot performing machine loading/unloading task at a stamping press. (Photo courtesy: GMF Robotics)*

Problems

13.1 A robot is expected to pick up circular parts from a conveyor, check its diameter for correctness and then place it into pass or fail bins.

(a) How many degrees of freedom should the robot have?

(b) Draw the layout of the robotic system showing the position of the robot, conveyor, inspection system and bins.

13.2 Write a program in VAL for the designed robot in problem above.

13.3

(a) Spot welding
(b) Spray painting
(c) Inspection tasks

References

New York, 1984, chaps. 9, 11, 14, 15, and 17.
2. J. F. Engelberger, *Robotics in Practice*, AMACOM (Division of American

Robot Handling,' *Robotics Today*, April 1984, pp. 73–75.
4. M. P. Groover and E. W. Zimmers, Jr., *CAD/CAM: Computer-Aided Design and Manufacturing*, Prentice-Hall, Englewood Cliffs, NJ, 1984, Chap. 11.
Industrial Engineering, February 1984, pp. 21–24.
Industrial Engineering
7. R. A. Lindberg, *Processes and Materials of Manufacture*, 3rd ed., Allyn and Bacon, Boston, MA, 1983, Chaps. 3, 4, and 5.
8. B. W. Niebel and A. B. Draper, *Product Design and Process Engineering*, McGraw-Hill, New York, 1974, Chaps. 11, 12, 13, and 14.
Robotics Today,

Robotics Today,

Industrial Robots, Volume 2—Applications, Soc. Mfg. Engrs., Dearborn, MI,

Robotics Today, February 1984, pp. 38–41.

14

Processing Operations

Introduction

In addition to parts-handling applications, there is a large class of applications, in which the robot actually performs work on the part. This work almost always requires that the robot's end effector is a tool rather than a gripper. Accordingly, the use of a tool to perform work is a distinguishing characteristic of this group of applications. The type of tool depends on the processing operation that is performed. The processing operations that are performed by a robot are divided into the following categories for purposes of organizing this chapter:

1. Spot welding
2. Continuous arc-welding
3. Spray coating
4. Other processing operations

The two welding categories represent important application areas for robots. Spot welding is probably the single most common application for industrial robots in the United States today because they are widely used in automobile body assembly lines to weld the frames and panels together. Arc welding is an application that is expected to grow in use as the technology required for using robots in this process is developed. Spray coating usually means spray painting, an operation that is accompanied by an unhealthy work environment for humans, and therefore represents a good opportunity for robots. The term 'spray coating' is used to indicate that there are additional applications beyond painting for a robot to spray a substance onto a surface. The final category in the listing above is a miscellaneous applications area. It includes certain machining operations, polishing, deburring, and other processing operations. These operations are usually, but not always, characterized by the use of a rotating spindle by the robot. The four categories of operations and how robots are used to accomplish these operations are discussed in the following sections.

14.1 SPOT WELDING

As the term suggests, spot welding is a process in which two sheet metal parts are fused together at localized points by passing a large electric current through the parts where the weld is to be made. The fusion is accomplished at relatively low voltage levels by using two copper (or copper alloy) electrodes to squeeze the parts together at the contact points and apply the current to the weld area. The electric

producing the weld.

The two electrodes have the general shape of a pincer. With the two halves of the pincer open, the electrodes are positioned at the point where the parts are to be fused.

for the process. The two electrodes are squeezed together against the mating parts, and the current is applied to cause heating and welding of the contacting surfaces. Then the electrodes are opened and allowed to cool for the next weld. A water circulation system is often used to accelerate the cooling of the electrodes. The actual welding portion of the sequence typically requires less than a second. Therefore, the rates of production in spot welding are largely dependent on the time required for positioning of the welding electrodes and the parts relative to each other. Another factor that affects production rate is the wear of the electrodes. Because of the heat involved in the process, the tips of the electrodes gradually lose their shape and build up a carbon deposit which affects their electric resistance. Both of these effects reduce the quality of the welds made. Therefore, the electrode tips must periodically be dressed to remove the deposits and restore the desired shape.

Spot welding has traditionally been performed manually by either of two methods.

used for relatively small parts that can be easily handled.

The second method involves manipulating a portable spot-welding gun into position relative to the parts. This would be used for larger work such as automobile bodies. The word 'portable' is perhaps an exaggeration. The welding gun consists of the pair of electrodes and a frame to open and close the electrodes. In addition, large electrical cables are used to deliver the current to the electrodes from a control panel located near the workstation. The welding gun with cables attached is quite heavy and can easily exceed 100 lb in weight. To assist the operator in manipulating the gun, the apparatus is suspended from an overhead hoist system. Even with this assistance, the

worker at the high rates of production desired on a car body assembly line. There are often problems with the consistency of the welded products made on such a manual

14.1.1 Robots in Spot Welding

this type of production line to perform some or all of the spot-welding operations. A welding gun is attached as the end effector to each robot's wrist, and the robot is programmed to perform a sequence of welds on the product as it arrives at the workstation. Some robot spot-welding lines operate with several dozen robots all programmed to perform different welding cycles on the product. Today, the automobile manufacturers make extensive use of robots for spot welding. In 1980, it was reported[3] that there were 1200 robots used in this application. Figure 14.1 shows an overview of an automobile body assembly line in which robots are used to perform the spot-welding operations. Figure 14.2 shows a close-up of a spot-welding gun mounted on a Cincinnati-Milacron T3 robot performing its task inside the car body.

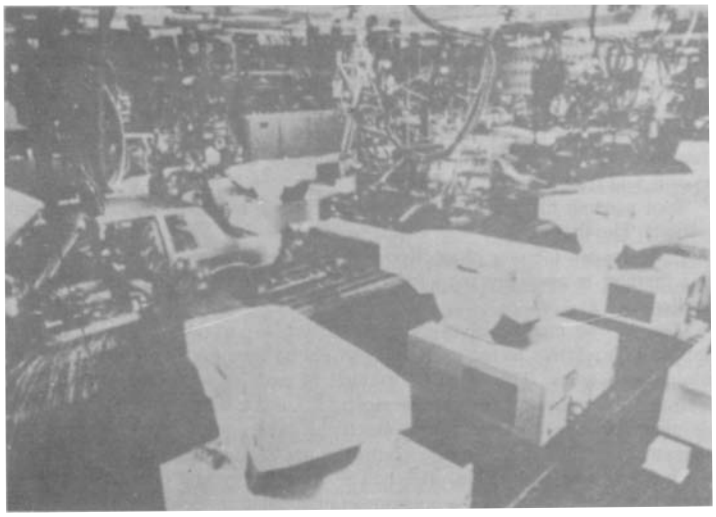


Fig. 14.1 *Robots performing spot welding operations on an automobile assembly line. (Photo courtesy: Unimation Inc., a Westinghouse Company)*

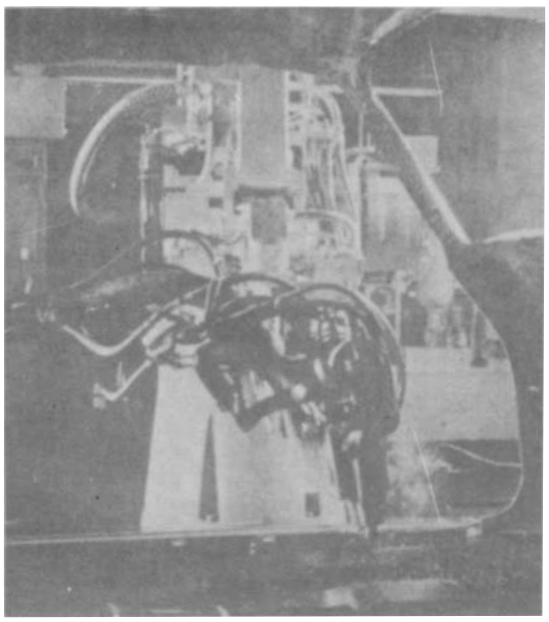


Fig. 14.2 *Close-up of a Cincinnati Milacron T3 robot performing a spot welding operation inside the car body. (Photo courtesy: Cincinnati Milacron)*

The robots used in spot welding must possess certain capabilities and features to

payload capacity to readily manipulate the welding gun for the application. The work volume must be adequate for the size of the product. The robot must be able to position

This might result in the need for an increased number of degrees of freedom. The controller memory must have enough capacity to accomplish the many positioning steps required for the spot-welding cycle. In some applications, the welding line is designed to produce several different models of the product. Accordingly, the robot must be able to switch from one programmed welding sequence to another as the models change. For welding lines in which there are multiple robots, programmable controllers are used to keep track of the different models at the various welding stations and to download the programs to the robots at individual workstations as needed.

of robots are improved product quality, operator safety, and better control over the production operation. Improved quality is in the form of more consistent welds and better repeatability in the location of the welds. Even robots with relatively

accurately than human operators. Improved safety results simply because the human is removed from a work environment where there are hazards from electrical shocks and burns. The use of robots to automate the spot-welding process should also result in improvements in areas such as production scheduling and in-process inventory control. The maintenance of the robots and the welding equipment becomes an important factor in the successful operation of an automated spot-welding production line.

14.2 CONTINUOUS ARC WELDING

Arc welding is a continuous welding process as opposed to spot welding which might be called a discontinuous process. Continuous arc welding is used to make long welded joints in which an airtight seal is often required between the two pieces of metal which are being joined. The process uses an electrode in the form of a rod or wire of metal to supply the high electric current needed for establishing the arc. Currents are typically 100 to 300 A at voltages of 10 to 30 V. The arc between the welding rod and the metal parts to be joined produces temperatures that are

electrode can also be used to contribute to the molten pool, depending on the type of welding process.

Arc welding is usually performed by a skilled human worker who is often assisted

unpleasant and hazardous. The arc from the welding process emits ultraviolet radiation which is injurious to human vision. As a result, welders are required to wear eye protection in the form of a welding helmet with a dark window. This dark

blind while wearing the helmet except when the arc is struck. Other aspects of the process are also hazardous. The high temperatures created in arc welding and the resulting molten metals are inherently dangerous. The high electrical current used to create the arc is also unsafe. Sparks and smoke are generated during the process and these are a potential threat to the operator.

There are a variety of arc-welding processes, and the reader is referred to the available manufacturing process texts for more details than it is possible to include here. For robot applications, two types of arc welding seem the most practical: gas metal arc welding (GMAW) and gas tungsten arc welding (GTAW). GMA welding (also called MIG welding for metal inert gas welding) involves the use of a welding wire made of the same or similar metals as the parts being joined. The welding wire serves as the electrode in the arc-welding process. The wire is continuously fed from a coil and contributes to the molten metal pool used in the fusion process. GMA welding is typically used for welding steel. In GTA welding (also called TIG welding for tungsten inert gas welding), a tungsten rod is used as the electrode to establish the arc. The melting point of tungsten is relatively high, and therefore the electrode

'it must be added separately from the electrode.' The GTA process is typically used for welding aluminum, copper, and stainless steel. In both GMA and GT A welding, inert gases such as helium or argon are used to surround the immediate vicinity of the welding arc to protect the fused surfaces from oxidation.

14.2.1 Problems for Robots in Arc Welding

Because of the hazards for human workers in continuous arc welding, it is logical

and economic problems encountered in applying robots to arc welding. Continuous arc welding is commonly used in the fabrication industries where products consisting

of any form in these circumstances. A related problem is that arc welding is often

pressure vessels, and ship hulls. Humans can position themselves into these areas more readily than robots.

variations in the components that are to be welded. These variations are manifested in two forms. One is the variation in the dimensions of the parts in a batch production job. This type of dimensional variation means that the arc-welding path to be followed will change slightly from part to part. The second variation is in the edges and surfaces to be welded together. Instead of being straight and regular, the edges are typically irregular. This causes variations in the gap between the parts and other problems in the way the pieces mate together prior to the welding process. Human welders are able to compensate for both of these variations by changing certain parameters in the welding process (e.g., adjusting the welding path, changing the

is large, etc.). Industrial robots do not possess the sensing capabilities, skills, and judgment of human welders to make these compensations.

There are two approaches to compensate for these variations and irregularities in robot welding applications:

1. Correct the stream production operations so that the variations are reduced to the point where they do not create a problem in the robot welding process.
2. Provide the robot with sensors to monitor the variations in the welding process and the control logic to compensate for part variations and weld gap irregularities.

Correction of the production operations that deliver parts to the arc-welding process is an attractive alternative because it tends to contribute to the overall

potential disadvantage of this approach is that it is likely to increase the cost of manufacturing the individual components because their dimensions must be held to closer tolerances. The second approach represents an area of intensive research and development activity in robotics. This area will be explored in a later subsection.

14.2.2 The Typical Robot Arc-Welding Application

Because of the technical and economic problems discussed above, the typical arc-welding application for a robot is one in which the quantities of production are medium or high.

them for welding. A part manipulator provides an additional capability in the form of 1 or 2 degrees of freedom to position and orient the components relative to the robot. The robot is equipped with a welding rod or wire-feed system, and the required power source to provide the electric current for the operation. The workcell controller is used to coordinate the robot motion, the welding current, the wire feed, the part manipulator, and any other activities in the cell.

One possible organization of a robot welding cell is illustrated in Fig. 14.3.

Fig. 14.3 *Robot arc-welding cell. (Reprinted by permission of Advanced Robotics Corp.)*

perform its welding cycle while the operator is unloading the previously welded assembly and loading the components into the holder for the next cycle. This arrangement permits higher utilization of the robot and greater productivity of the cell.

14.2.3 Features of the Welding Robot

An industrial robot that performs arc welding must possess certain features and capabilities. Some of the technical considerations in arc-welding applications are discussed in the following:

1. Work volume and degrees of freedom The robot's work volume must be large

for manipulation of the welding torch. Also, if two part holders are included in the workstation, the robot must have adequate reach to perform the motion cycle at both holders. Five or six degrees of freedom are generally required for arc-welding robots.

capabilities of the parts manipulator. If the parts manipulator has two degrees of freedom, this tends to reduce the requirement on the number of degrees of freedom possessed by the robot.

2. Motion control system Continuous-path control is required for arc welding. The robot must be capable of a smooth continuous motion in order to maintain uniformity of the welding seam. In addition, the welding cycle requires a dwell at the beginning of the movement in order to establish the welding puddle, and a dwell at the end of the movement to terminate the weld.

3. Precision of motion The accuracy and repeatability of the robot determines to a large extent the quality of the welding job. The precision requirements of welding jobs vary according to size and industry practice, and these requirements should be

4. Interface with other systems
output and control capabilities to work with the other equipment in the cell. These other pieces of equipment are the welding unit and the parts positioners. The cell controller must coordinate the speed and path of the robot with the operation of the parts manipulator and the welding parameters such as wire feed rate and power level.

5. Programming Programming the robot for continuous arc welding must be considered carefully, To facilitate the input of the program for welding paths with irregular shapes, it is convenient to use the walkthrough method in which the robot wrist is physically moved through its motion path. For straight welding paths, the robot should possess the capability for linear interpolation between two points in

path and the robot is capable of computing the straight line trajectory between the points.

Some welding applications require the robot to follow a weave pattern (back and forth motion across the welding seam) during the operation. Other applications require a series of passes along the same path, but each pass must be slightly offset from the previous one to allow for the welding bead that was laid down in the previous pass. Both of these requirements are generally associated with large welding jobs where the amount of material to be added is greater than what can be applied normally

provided with features to facilitate the programming of weave patterns and multiple

the parameters needed to accomplish the special pass. For the weave pattern, the parameters are the amplitude of the weave, the number of weaves per inch of travel, and the dwell on either side of the weave. For multipass operation, the magnitude

Chap. 9, the arc-welding programming features of the RAIL language were discussed, and the reader is invited to refer back to that section of the appendix.

The various programming features described above assume that the components to be welded possess regular edges along the intended weld path and that the components are uniform from part to part. Accordingly, the robot would be able to repeat the programmed motion path for each set of components to produce welded assemblies with great consistency. Unfortunately, some of the biggest technical

sensor systems to track the welding path during the process and compensate for the irregularities in the path.

14.2.4 Sensors in Robotic Arc Welding

At present, a wide variety of arc welding sensors are either commercially available or under development in various research and development laboratories. The systems that are commercially available needs concentration, leaving for the interested reader the task of exploring some of the research that is being done in this area. References 6 and 16 provide a more thorough treatment of these sensor systems than can be devoted to the subject here.

The robotic arc welding sensor systems considered here are all designed to track the welding seam and provide information to the robot controller to help guide the welding path. The approaches used for this purpose divide into two basic categories: contact and non-contact sensors.

Contact Arc-Welding Sensors Contact arc-welding sensors make use of a mechanical tactile probe (some of the probe systems would better be described as electromechanical) to touch the sides of the groove ahead of the welding torch and to feedback position data so that course corrections can be made by the robot controller. Some systems use a separate control unit designed to interpret the probe sensor measurements and transmit the data to the robot controller. To accomplish the position measurements, the probe must be oscillated from one side of the groove to the other by the sensor system. The nature of the operation of these sensor systems limits their application to certain weld geometries in which the side-to-side motion of the probe permits it to make contact with the edges or surfaces that are to be welded. Some of the weld geometries in this category include butt welds that have grooved joints, lap joints, and fillet welds. Figure 14.4 shows a diagram which depicts the operation of the contact arc-welding sensor and several of the types of joints in which it can be used.

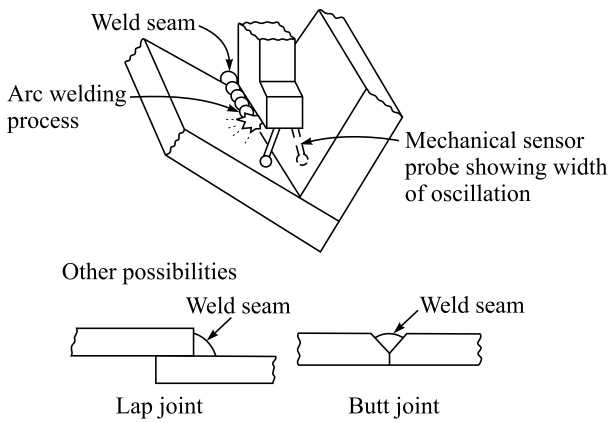


Fig. 14.4 *Diagram depicting the operation of the contact arc welding sensor and several of the types of joints in which it can be used.*

Another limitation of the contact arc welding sensor is that the probe must be maintained in the proper position ahead of the welding torch, and this makes these systems most effective on welds that are long and straight. These kinds of arc-welding applications do not make full use of most robots' capabilities for more complex path control.

Non-contact Arc Welding Sensors The second basic type of sensor system used to track the welding seam uses no tactile measurements. A variety of sensor schemes have been explored in this category, but the discussions will concentrate on arc sensing systems and vision-based systems since these are the approaches used more in today's commercial systems.

Arc sensing systems (sometimes called 'through-the-arc' systems) rely on measurements taken of the arc itself, in the form of either electric current (in constant-voltage welding) or voltage (in constant-current welding). In order to interpret these signals, they must be varied during the arc-welding process. This is accomplished by causing the arc to weave back and forth across the joint as it moves down the path. The side-to-side motion along the joint can be achieved by programming the robot to perform the weave pattern, or by means of a servo system that attaches to the robot wrist and determines the position of the torch, or by other mechanisms. The weaving motion permits the electrical signals to be interpreted in terms of vertical and cross-seam position of the torch. The controller performs an adaptive positioning of the torch as it moves forward along the joint centerline so that the proper path trajectory can be maintained. As irregular edges are encountered along the weld path, the control system compensates by regulating either the arc length (for constant-current systems) or the distance between the torch tip and the work surface (for constant-voltage systems). Operation of the typical through-the-arc seam tracking system is illustrated in Fig. 14.5.

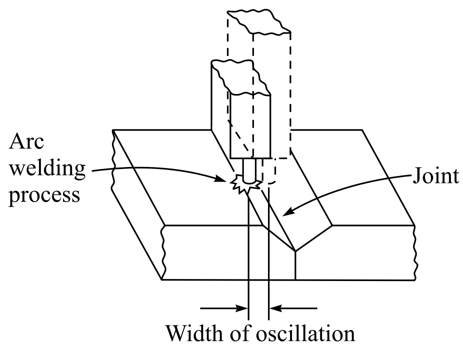


Fig. 14.5 *Operation of a typical 'through-the-arc' seam tracking sensor system, depicting oscillation from side to side in the welding groove.*

Limitations of the arc sensing sensors are similar to the limitations in the operation of the tactile systems described above. The welding joint must possess a grooved geometry which permits the weaving motion to be effective. In spite of this limitation, through-the-arc systems constitute the largest number of non-contacting seam trackers commercially available at the time of this writing. Companies offering these systems include CRC Welding Systems (Nashville, Tennessee), Advanced Robotics Corp. (Columbus, Ohio), and Hobart Brothers Co. (Troy, Ohio).

Vision-based systems represent a promising technology for tracking the seam in arc welding operations. These systems utilize a vision camera mounted on the robot near the welding torch to view the weld path. In some cases the camera is an integral component of the welding head. Highly structured light is usually required for the camera sensors to function reliably.

There are two approaches used with vision sensors for arc welding: two-pass systems and single-pass systems. In both types, the robot must be programmed for the welding path before the operation begins.

In the two-pass systems, the vision camera takes a preliminary pass over the seam before the welding operation begins. As indicated above, the robot must be programmed for the particular seam path before either pass is taken. Then the two passes are taken automatically by the robot. In the first pass, light is projected onto the seam and the camera scans the joint at high speed (speeds up to 1 m/s are claimed), checking for deviations from the anticipated seam path. These deviations are analyzed by the controller and remembered for the second pass. During the second pass, in which the welding process is performed, the controller makes adjustments in the seam path to correct for the deviations detected in the first pass. The first pass requires only about 10 percent of the time for the welding pass, and the advantage gained by using two passes is that the vision system can see a clean view of the welding path on the preliminary scanning pass, absent of the smoke and intense brightness encountered during actual welding. Examples of commercially available systems in this two-pass category include Unimation's Univision II system and the Robo Welder Series 1200 by Robotic Vision Systems Inc. (RVSI). Figure 14.6 shows the camera and the welding head of the RVSI seam tracking vision system, both components mounted as the end effector to the robot wrist.

In the single-pass system, the vision camera is aimed at the welding seam just ahead of the torch. Deviations from the programmed seam location are detected and corrections are made in the weld path. The obvious advantage of the single-pass systems, compared to the two-pass systems, is that time is saved by eliminating the need for a second pass along the weld path. Another advantage is that the single-pass systems are able to compensate for thermal distortions in the weld path caused by the welding process.

Examples of commercial vision systems in the single-pass category are the Robovision II from Automatix Inc., and WeldVision from General Electric. In the Automatix system, the camera is focused about 4 cm in front of the weld. The observed image is analyzed to extract the location of the center of the seam, the seam width, and the distance of the seam from the camera. In the general electric system, the vision sensor is incorporated into the design of the welding torch. The image observed by the camera includes the weld puddle and the seam ahead of it. By analyzing both the weld puddle and the seam, the controller is able to make adjustments in the process to automatically track the seam.

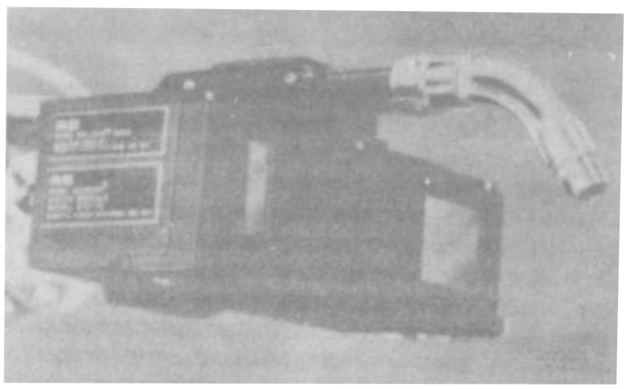


Fig. 14.6 *Robo Welder Series 1200 seam tracking vision system. (Photo courtesy: Robotic Vision Systems Inc.)*

14.2.5 Advantages and Benefits of Robot Arc Welding

A robot arc-welding cell for batch production has the potential for achieving a number of advantages over a similar manual operation. These advantages include the following:

1. Higher productivity
2. Improved safety and quality-of-work life
3. Greater quality of product
4. Process rationalization

The productivity of a manual arc-welding operation is characteristically quite low. The productivity is often measured by the 'arc-on' time. This gives the proportion of time during the shift that the welding process is occurring, and therefore production is taking place. Typical values of arc-on time range between 10 and 30 per cent. The lower value corresponds to one-of-a-kind welding jobs, and the higher value corresponds to batch type production. One of the reasons why the arc-on time is low in manual welding is the fatigue factor. The hand-eye coordination required and the generally uncomfortable working environment tend to be tiring to the human welder and frequent rest periods must be taken. With robot welding cells for batch production, a 50 to 70 per cent arc-on time can be realized. There are several factors that contribute to the increased arc-on time when robots are used in batch production. Certainly one factor is the elimination of the fatigue factor. Robots do not experience fatigue in the sense that human workers do. A robot can continue to operate during the entire shift without the need for periodic rest breaks. Another contributing factor is the presence of two parts positioners in the cell. The robot can be performing the welding operation at one position while the human operator is unloading the previous assembly and loading new components at the other position.

Improved safety and quality-of-work environment result from removing the human operator from an uncomfortable, fatiguing, and potentially dangerous work situation. As described above, the welding environment contains a number of serious hazards for human beings.

Greater product quality in robot arc-welding results from the capability of the robot to perform the welding cycle with greater accuracy and repeatability than its human counterpart. This translates into a more consistent welding seam, one that is

many welds accomplished by human welders.

The term *process rationalization* refers to the systematic organization of the work

cell forces the user company to consider such issues as the delivery of materials to the

and the problems of production and inventory control related to the operation of the cell. Typically, these issues are not adequately addressed when the company relies on human welding stations.

14.3 SPRAY COATING

Most products manufactured from metallic materials require some form of painted

varies in complexity from simple manual methods to highly sophisticated automatic techniques. The common industrial coating methods are divided into two categories:

2. Spray-coating methods

These are generally considered to be low-technology methods of applying paint to the product. Immersion involves simply dipping the part or product into a tank of liquid paint. When the object is removed, the excess paint drains back into the tank. The tanks used in the process can range in size from 1 or 2 gallons for small objects to thousands of gallons for large fabricated

dipping the parts into the tank, they are positioned above the tank and a stream of paint

coating are relatively simple processes, the methods for delivering the product to the painting operation may involve considerable mechanization. For example, conveyor systems are often used in high production to carry the parts down into the dipping tanks to apply the coating.

A more advanced immersion method is electrodeposition. This is a process in which a conductive object (the part or product) is given a negative electrical charge and dipped into a water suspension containing particles of paint. The paint particles are given a positive electrical charge, and consequently they are attracted to the negatively charged object (the cathode). The electrodeposition coating method is a highly sophisticated technique and requires close control over the process parameters (e.g., current, voltage, concentration of paint in suspension) in order to ensure the success of the operation. Its advantage is that it does not waste nearly as much paint as conventional immersion methods.

2. Spray coating methods The second major category of industrial painting is spray coating. This method involves the use of spray guns to apply the paint or other coating to the object. Spray painting is typically accomplished by human workers who manually direct the spray at the object so as to cover the desired areas. The paint spray systems come in various designs, including conventional air spray airless spray, and electrostatic spray. The conventional air spray uses compressed air mixed with the paint to atomize it into a high velocity stream. The stream of air and paint is directed through a nozzle at the object to be painted. The airless spray does not

decrease in pressure in front of the nozzle.

The electrostatic spray method makes use of either conventional air spray or airless spray guns. The feature which distinguishes the electrostatic method is that the object to be sprayed is electrically grounded and the paint droplets are given a negative electrical charge to cause the paint to adhere to the object better.

The spray-coating methods, when accomplished manually, result in many health hazards to the human operators. These hazards include[5]:

1. Fumes and mist in the air These result naturally from the spraying operation. Not all of the paint droplets become attached to the surface of the object. Some remain suspended in the atmosphere of the spray painting booth. To protect the human operators, ventilation systems must be installed in the booth and protective clothing and breathing masks must be worn. Even with this protection, the environment is uncomfortable and sometimes toxic for humans.

2. Noise from the nozzle The spray gun nozzle produces a loud shrill noise. Prolonged exposure by humans can result in hearing impairments.

3. Fire hazards Flammable paint, atomized into a fine mist and mixed with air, can result in flash fires in the spray painting booths.

4. Potential cancer hazards Certain of the ingredients used in modern paints are believed to be carcinogenic, with potentially unsafe health consequences to humans.

14.3.1 Robots in Spray Coating

Because of these hazards to humans, the use of industrial robots has developed as an alternative means of performing spray-coating operations. Spray-coating operations to which robots have been applied include painting of car bodies, engines, and other components in the automotive industry, spraying of paint and sound absorbing coatings on appliances, application of porcelain coatings in bathroom fixtures, and spray staining of wood products. Some of the applications have consisted of a stand-alone robot spraying a stationary workpart that has been positioned in a paint booth by a human worker. However, these applications are generally less successful because they rely heavily on the human worker and the utilization of the robot is relatively low.

In most robot spray-coating applications, the robots are usually part of a system that includes a conveyor for presenting the parts to the robot, and a spray booth for shielding the spraying operation from the factory environment. Figure 14.7 illustrates a robot spray painting a part. When a conveyor-robot system is used, the operation of the robot and the conveyor must be closely synchronized. In the case of an intermittent conveyor system, interlocks are used to coordinate the start and finish of the robot program with the movement of the conveyor. With a continuously moving conveyor, some form of baseline tracking system (discussed in Chap. 11) is required in order to synchronize the robot's motions with the movement of the conveyor.

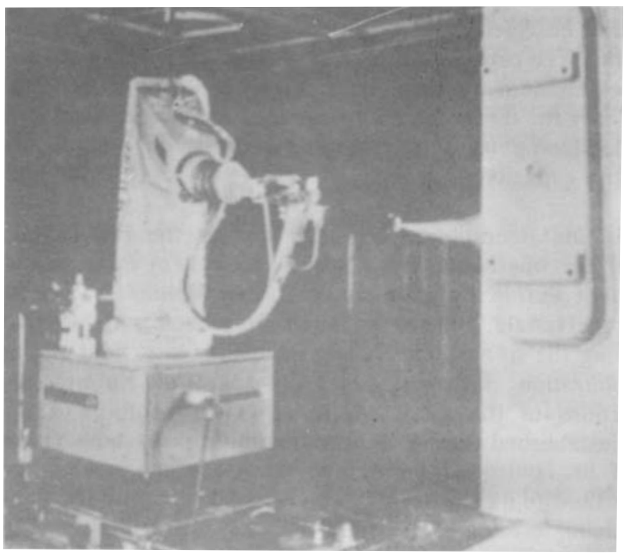


Fig. 14.7 *De Vilbiss/Trallfa robot performing a spray coating operation. (Photo courtesy: DeVilbiss Company)*

Another feature of many robot spray-coating applications is that the system must

correct spray cycle for that part.

In general, the requirements of the robot for spray-coating applications are the following:

In order to emulate the smooth movement of a human spray paint operator, the robot must possess many degrees of freedom in its manipulator and it must have continuous path capability.

2. Hydraulic drive Hydraulic drive is preferred over electric or pneumatic drive in spray-painting applications. In electric drive there is danger that a spark in the electric motor system may ignite the paint fumes in the spray booth environment. The motions generated in pneumatic drive are generally too jerky to be suitable for spray-coating applications.

3. Manual leadthrough programming In most spray-coating applications, the most convenient method of teaching the robot involves leadthrough programming in which the robot arm is manually pulled through the desired motion pattern by a human operator who is skilled in the techniques of spray painting. During the programming procedure, a 'teach arm' which is light and maneuverable, is often substituted for

4. Multiple program storage The need for multiple program storage arises in paint production lines in which more than one part style are presented to the robot for spraying. The capability to quickly access the program for the current part is a

the programs required or it must be interfaced to the cell controller (computer or programmable controller) for random access to this memory capacity.

In robot spray-coating operations, the spray gun is the robot's end effector. Control over the operation of the spray gun system must be accomplished by the robot during program execution. In addition to on–off control over the spray gun nozzle, some of the important process variables that must be regulated during the spray cycle

regulated through the output interlock functions of the robot controller. The operation of the interlock functions is established during the programming procedure. Other parameters that must be controlled during the spray-coating process are related to

loss of production time by programming a cleaning cycle into the workcell operation at regular intervals. The cleaning operation takes only a few seconds to complete and consists of the robot placing the spray nozzle under cleaning jets which spray solvent into the nozzle opening. Incorporating the cleaning operation into the work cycle should be planned to minimize the impact on the productive portion of the cycle.

advantages. These advantages include:

1. Removal of operators from hazardous environment
2. Lower energy consumption

4. Reduced coating material usage
5. Greater productivity

Removing the human workers from the kinds of hazards which characterize the

the ventilation requirements are reduced below the levels needed when humans are present. Therefore, less energy is needed to control the environment.

Other advantages include better quality and fewer rejects. Because the robot

paint required to coat the parts is typically reduced by 10 to 50 per cent when robots are used. These various features of the robot spray-coating cell result in substantial labor savings and improved productivity in the process.

Example 14.1 An example of a modern high-technology robotic spray-painting cell design is presented in a paper by Akeel.[1] The cell design is the result of the development efforts of the General Motors Manufacturing Staff directed at the problem of automating the paint shops in an automobile production plant. A typical paint cell for producing 60 jobs per hour consists of eight robots (four pairs, each pair servicing the two sides of the automobile) and is illustrated in Fig. 14.8. Other features of the cell include:

Fig. 14.8

Akeel [1].)

A machine vision system for identifying the body style so that the proper robot program can be used.

Backup robots so that if one of the production robots breaks down, they can be quickly replaced by the backups. An overhead crane system serves to replace the robots when needed.

Automatic two-axis door openers so that the internal surfaces of the car can be sprayed.

Supervisory computer control of the cell, including a backup computer that is operated in parallel with the primary system. The backup can be switched into service should the primary computer fail.

The individual painting robots (GM calls them paint spray machines) are seven-axis manipulators that are operated under computer control. The supervisory computer communicates the correct work cycles to the individual machines on the line. Each spray paint machine is equipped with two spray guns especially designed by GM engineers for automatic operation in the cell. Each spray gun is provided with its own paint supply system so that one system can be in the process of being changed over (purged, cleaned, and refilled with the next paint) while the other system is in production. The parallel paint supply lines are contained inside the manipulator arm.

14.4 OTHER PROCESSING OPERATIONS USING ROBOTS

In addition to spot welding, arc welding, and spray coating, there are a number of other robot applications which utilize some form of specialized tool as the end effector. Operations which are in this category include:

Drilling, routing, and other machining operations

Grinding, polishing, deburring, wire brushing, and similar operations

Riveting

Waterjet cutting

Laser drilling and cutting

This category excludes the applications in assembly, inspection, and non-manufacturing operations which might employ a tool as end effector. Assembly and inspection applications are discussed in the following chapter, and non-manufacturing applications are covered in Chap. 20. From the preceding list, it can be seen that a typical end effector in this category is a powered spindle attached to the robot's wrist. The spindle is used to rotate a tool such as a drill or grinding wheel. The purpose of the robot is to position the rotating tool against a stationary work part in order to accomplish the desired processing operation. In the other examples given in the above list (riveting, waterjet cutting, and laser operations), the end effector is not a powered spindle, but the job of the robot is still to position the tool relative to the part. Requirements of these applications vary, but one of the inherent disadvantages of robots in some of these operations is their relative lack of accuracy as compared to a regular machine tool. Small robots tend to be more accurate than large robots, but large robots are more likely to possess the strength and rigidity necessary to withstand the forces involved to hold the powered spindle against the part during the process. Figure 14.9 shows a PUMA 500 robot performing a wire brush deburring operation.

Fig. 14.9 *PUMA 500 robot performing a wire-brush operation to debur a workpart. (Photo courtesy: Unimation Inc.)*

Example 14.2 This example resulted from an Air Force ICAM (Integrated Computer-Aided Manufacturing) sponsored project,[4] and represents an important project in the development of processing applications. The application involves drilling and routing operations of different aircraft components. The components are various sheet metal fuselage panels for the F-16 fighter aircraft which is fabricated by General Dynamics Corporation in Fort Worth, Texas. A diagram of the cell concept is illustrated in Fig. 14.10. The principal components of the cell are the robot and a part fixture. The robot used in the application is a Cincinnati Milacron T3 robot which has a repeatability of + or –0.050 in. and a relatively large work envelope. The application requirement was that the robot should be able to reach over a 3 by 4 ft part area. The fixture has two positions so that the robot can be processing one part while loading and unloading is being accomplished at the opposite position. A human worker performs the loading and unloading tasks. The cell works in a batch production mode and was first placed in operation in October, 1979. Some of the important features about this robot application are:

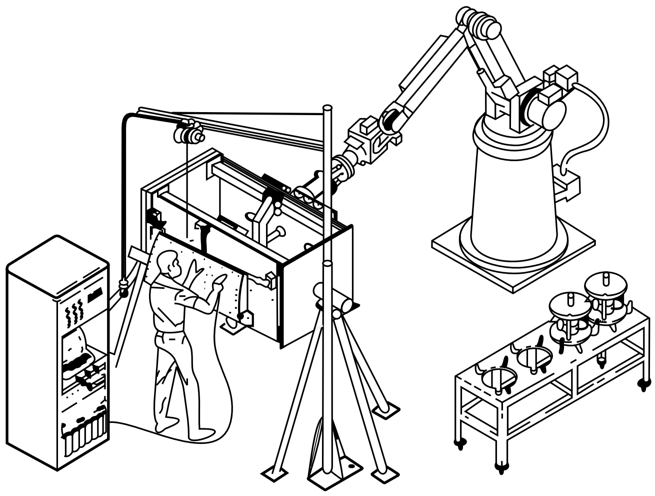


Fig. 14.10 *Diagram of the drilling and routing cell developed under Air Force ICAM sponsorship. (Reprinted with permission from Golden et al. [4].)*

1. Supervisory computer control to coordinate the activities of the robot and the part fixture.
2. Multiple program storage so that the robot executes the correct program on each different fuselage component.
3. Automatic workpart identification system employing an optical character reader to recognize which part is to be processed next. This allows the correct program to be called from multiple program storage.
4. Automatic tool changing of drills and routing tools. A tool rack is used to hold the different tools required in the sequence of processing operations. The robot is programmed to select the proper tool for the particular operation to be performed.
5. Templates are used to define hole patterns and router paths. Also, compliant end effectors are used to permit the tools to line up with the templates. These measures were adopted in order to overcome the inherent accuracy and repeatability limitations of the T3 robot.

Example 14.3 A more recent application, similar to the one described in Example 14.2, is the installation at Grumman Aerospace Corp. The robot cell is used to accomplish light machining of complex contour sheet metal parts for aircraft skins and other components. An ASEA IRB-60 robot performs routing, trimming, and drilling of the aircraft parts. The workcell consists of the robot mounted on a linear track system (this is a mobile robot cell), the part holders, tool-changing module, and cell controller. The cell layout is illustrated in Fig. 14.11. The track system provides approximately 20 ft of linear movement for the robot. This dramatically increases the robot's work volume.

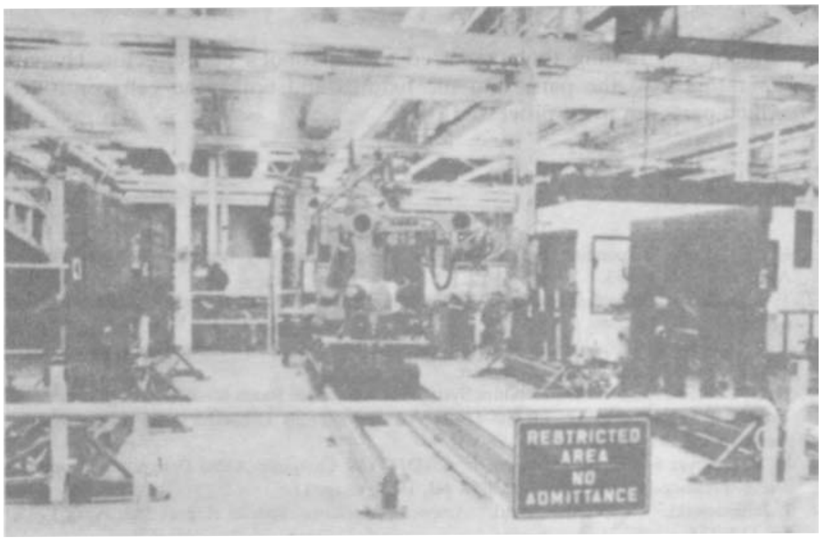


Fig. 14.11 *Cell layout for Example 14.3. (Photo courtesy: Grumman Aerospace Corp.)*

The tool-changing module is shown in Fig. 14.12. Several air-powered drilling and routing tools are used as the robot's end effector. Human operators load the parts into the fixtures and notify the cell controller which parts are to be processed.

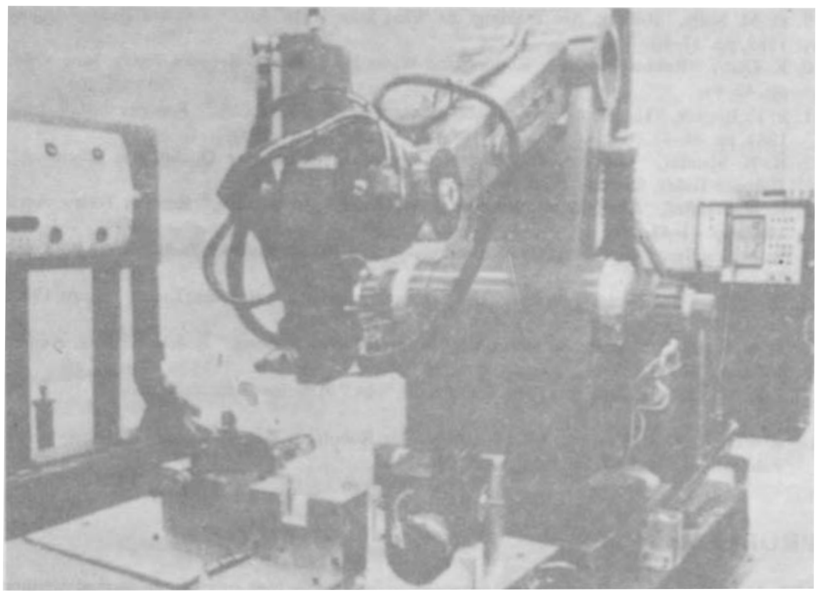


Fig. 14.12 *Tool-change set up for Example 14.3. (Photo courtesy: Grumman Aerospace Corp.)*

roblems

14.1 A robotic arc-welding cell is proposed to replace a high-production manual

cycle is 8.0 min, and the welding time is 9.0 min, which includes 4.0 min for repositioning the welding rod during the welding. The repositioning involves moving the welding rod to a new location on the parts without actually welding. It is therefore non-productive time. The two operators are paid for 8 hours per day (250 days per year), although 1 hour is lost per day for rest

not be considered in the analysis.

The proposed robot cell would replace the manual cell and would use one

two positions so that the human operator would load and unload components

time would remain the same because of technological process considerations, but the repositioning time (previously 4.0 min with the human welder) would be reduced by one-half with the robot welder. Time required for indexing

robot cell would operate for 7.5 hours per day although the operator would be paid for 8 hours.

to one-third their combined initial values at the end of the project. Annual

return criterion is 20 per cent.

(a) What is the hourly production rate for the current manual operation and what is the anticipated hourly production rate for the proposed robot cell?

(b) What is the arc-on time for the current manual operation and what is the anticipated arc on time for the proposed robot cell?

14.2 Should the proposed robot cell in Prob. 14.1 be installed? Support your answer using the economic analysis methods of Chap. 12.

References

1. H. A. Akeel, 'Expanding the Capabilities of Spray Painting Robots,' *Robotics Today*, April 1982, pp. 50–53.
2. T. J. Bublick, 'Guidelines for Applying Finishing Robots,' *Robotics Today*, April 1984, pp. 61–64.
3. J. F. Engelberger, *Robotics in Practice*, AMACOM (American Management Association), New York, 1980, Chaps. 11, 12, and 16.
4. H. D. Golden et al., 'ICAM Robotics System for Aerospace Batch Manufacturing—Task A,' *Technical Report AFWAL-TR-80-4142*, Vol. I, Materials Laboratory, Air Force Wright Aeronautical Laboratories, OH, 1980.
5. M. P. Groover and E. W. Zimmers, Jr., *CAD/CAM: Computer-Aided Design and Manufacturing*, Prentice-Hall, Englewood Cliffs, NJ, 1984, Chap. 11.
6. J. Jablonowski, 'Robots that Weld,' *American Machinist*, Special Report 753, April 1983, pp. 113–128.
7. R. J. McCluskey, 'Robotic System Cuts Airplane Parts,' *American Machinist*, August 1984, pp. 71–73.
8. S. Muller, 'Spot Welding: The Classic Case for the Quality Robot,' *Decade of Robotics*, IFS Publications, Bedford, England, 1983, pp. 34–39.
9. G. M. Nally, 'Robotic Arc Welding: At What State is the Art?' *Robotics Today*, August 1983, pp. 37–40.
10. K. Ostby, 'Robots Automate Routing and Water Jet Cutting,' *Robotics Today*, June 1984, pp. 42–43.
11. P. F. Rogers, 'The Economics of Robotic Arc Welding Workcells,' *Robotics Today*, June 1984, pp. 46–48.
12. R. N. Stauffer, 'Robogate and Unimates Team Up to Improve Quality and Efficiency,' *Robotics Today*, Summer 1980, pp. 24–30.
13. R. N. Stauffer, 'Anatomy of a Successful Arc Welding Installation,' *Robotics Today*, April 1982, pp. 41–42.
14. R. N. Stauffer, 'Automated Body Assembly—Circa 1972,' *Robotics Today*, April 1982, pp. 58–60.
15. R. N. Stauffer, 'Welding Robots: The Practical Approach,' *Robotics Today*, August 1983, pp. 43–44.
16. R. N. Stauffer, 'Update on Noncontact Seam Tracking Systems,' *Robotics Today*, August 1983, pp. 29–34.
17. J. A. Vaccari, 'Robots that Paint Can Create Jobs,' *American Machinist*, January 1982, pp. 131–134.
18. J. Weston, 'Arc Welding: A Difficult Path for Robots to Tread,' *Decade of Robotics*, IFS Publications, Bedford, England, 1983, pp. 40–43.

Assembly and Inspection

Introduction

15.1 ASSEMBLY AND ROBOTIC ASSEMBLY AUTOMATION

15.2 PARTS PRESENTATION METHODS

15.2.1 Bowl Feeders

Fig. 15.1 *Vibratory bowl feeder (Photo courtesy: FMC Corp., Material Handling Equipment Dev)* *selection and orientation*

passive orientation

Fig. 15.2 *Part selection and orientation methods used in vibratory bowl feeders: (a) Selection and orientation of cup-shaped parts, (b) Selection and orientation of screws. (Reprinted with permission from Boothroyd and Redford [3].)*

$u_f F_g < u_p[F_b + nW \quad \theta$

u_f

F_g

u_p

F_b

n

W

θ

Fig. 15.3 *Several types of escapement devices used in automated assembly: (a) Linear motion escapement device for disk-shaped parts, (b) Worm escapement device. (Reprinted with permission from Boothroyd and Redford [3].)*

15.2.2 Magazine Feeders

15.2.3 Trays and Pallets

15.3 ASSEMBLY OPERATIONS

15.3.1 Parts Mating

1. Peg-in-hole

2. Hole-on-peg

Fig. 15.4 *Two types of peg-in-hole assembly tasks: (a) Round peg-in-hole, (b) Square peg-in-hole—orientation about z-axis required.*

3. Multiple peg-in-hole

Fig. 15.5 *Multiple peg-in-hole assembly task: insertion of a semiconductor chip module into a circuit card.*

4. Stacking

15.3.2 Parts-Joining Tasks

1. Fastening screws

Fig. 15.6 *Operation of a power screwdriver. (Reprinted with permission from Boothroyd and Redford [3].)*

2. Retainers

Fig. 15.7

parts.

5. Welding and related joining method

6. Adhesives

7. Crimping

8. **Sewing**

15.4 COMPLIANCE AND THE REMOTE CENTER COMPLIANCE (RCC) DEVICE

Fig. 15.8 *Two possible errors for the peg-in-hole insertion task: (a) Lateral position error, (b) Angular error.*

Fig. 15.9 *Peg-in-hole insertion task: (a) Effect of small position error with chamfered parts, (b) Small angular error results during chamfer crossing as the peg rotates about the grip point at the top of the peg.*

Fig. 15.10 *Remote center compliance (RCC) device. (Photo courtesy: Lord Corporation)*

Fig. 15.11 *Mechanical gripper attached to the remote center compliance device. (Photo courtesy: Lord Corporation)* a b

a

c

d

Fig. 15.12 *Action of a remote center compliance device in peg-in-hole insertion task: (a) Action of RCC for lateral displacement, (b) Action of RCC for angular displacement. (Reprinted by permission courtesy of Lord Corporation [7].)*

Remote center distance

elastic center

1. Axial force capacity T

2. Compressive stiffness T

3. Lateral stiffness

4. Angular stiffness

5. Torsional stiffness

15.5 ASSEMBLY SYSTEM CONFIGURATIONS

15.5.1 Single-Workstaiton Assembly

Example 15.1

Fig. 15.13 *Single station robotic workcell for Example 15.1.*

Table 15.1 *Sequence of steps to assemble electric motor of Example 15.1.*

1. Place rear endbell into fixture
2. Set first bearing into endbell
3. Set rotor into bearing-endbell
4. Set stator around armature
5. Set second bearing on top of rotor shaft
6. Set front endbell over bearing-rotor-stator
7. Insert first screw
8. Insert second screw
9. Drive both screws
10. Insert first brush holder
11. Insert second brush holder
12. Press both brush holders
13. Off-load completed motor

Table 15.2 *Sequence of assembly steps for robot in Example 15.1.*

Step		*Time, s*
1.	Load first endbell (gripper 1)	8
2.	Change gripper	12
3.	Install bearing (gripper 2)	8
4.	Change gripper	12
5.	Install rotor (gripper 1)	8
6.	Install stator (gripper 1)	8
7.	Change gripper	12
8.	Install bearing (gripper 2)	8
9.	Change gripper	12
10.	Install endbell (gripper 1)	8
11.	Change gripper	12
12.	Install two screws (gripper 2)	16
13.	Change gripper	12
14.	Retrieve screwdriver (gripper 1)	12
15.	Drive two screws (gripper 1)	8
16.	Replace screwdriver (gripper 1)	12
17.	Reorient motor (gripper 1)	7
18.	Change ripper	12
19.	Install two brush holders (gripper 2)	16
20.	Change gripper	12
21.	Load motor into press (gripper 1)	8
22.	Cycle press	4
23.	Unload motor (gripper-1)	7
	Total	234

R

R

15.5.2 Series Assembly Systems

Fig. 15.14 *GMF robot loading a dial indexing machine. (Photo courtesy: GMF Robotics)*

Example 15.2

Fig. 15.15 *Series assembly cell in Example 15.2*

Example 15.3

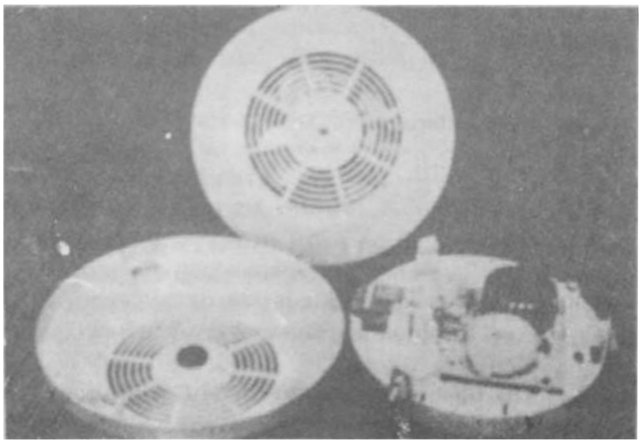


Fig. 15.16 *Smoke detector and compoments in Example 15.3. (Photo courtesy: United States Robots, subsidiary of Square D Company)*

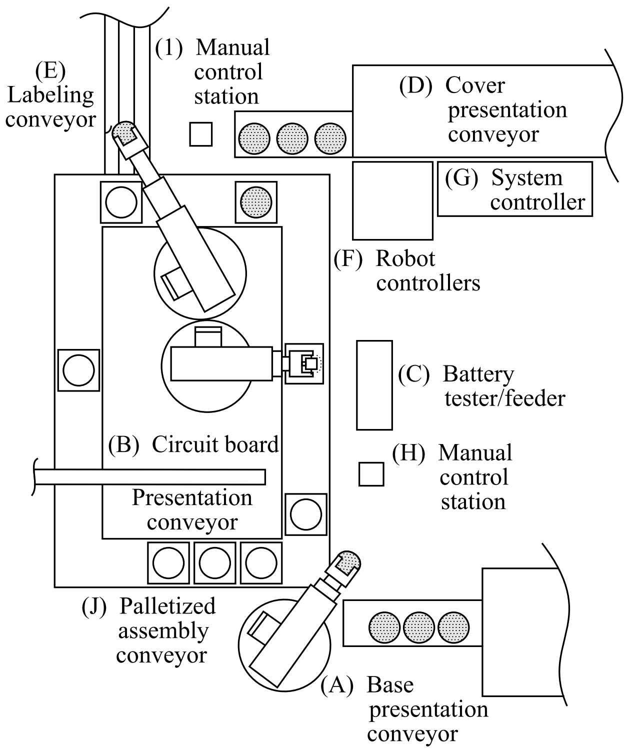


Fig. 15.17 *Layout of assembly system for Example 15.3.*

Fig. 15.18 *Overview of the smoke detector assembly cell in Example 15.3. (Photo courtesy: United States Robots, subsidiary of Square D Company)*

1. Station 1: Load Base

2. Station 2: Battery and Printed Circuit' Insertion

Fig. 15.19 *Special gripper used at station 2 in Example 15.3. (Photo courtesy: United States Robots, subsidiary of Square D Company)*

3. Station 3: Cover Assembly and Final Test

4. Workcell controller

15.5.3 Parallel Assembly Systems

Fig. 15.20 *Parallel workstations on an assembly system.*

Fig. 15.21 *Series layout with overlapping work envelopes.*

15.6 ADAPTABLE-PROGRAMMABLE ASSEMBLY SYSTEM

Fig. 15.22 *Overview of the Adaptable-Programming Assembly System (APAS) for assembling different styles of electric motors.*

Fig. 15.23 *View of one end of the APAS. (Photo courtesy: Unimation, Inc., Subsidiary of Westinghouse Corp.)*

15.7 DESIGNING FOR ROBOTIC ASSEMBLY

15.8 INSPECTION AUTOMATION

15.8.1 Vision Inspection Systems

Example 15.4

Example 15.5

Example 15.6

Fig. 15.24 *Automatic test cell of Example 15.6. (Photo courtesy: United States Robots, subsidiary of Square D Company)*

15.8.4 Integrating Inspection into the

T. Lund, Design for Assembly

IBM Technical Report

Mechanized Assembly

Final Report

Automation, Production Systems, and Computer-Aided Manufacturing

CAD/CAM: Computer-Aided Design and Manufacturing

Robowrist—Remote Center Compliance Devices RCC Series

Robotics Today

Robotics Today

American

Assembly Automation—A Management Handbook

Robotics Today

Robotics Today

Robotics Today

Journal of Dynamic Systems, Measurement, and Control

Implementation Principles and Issues

The preceding chapters on applications engineering for manufacturing were concerned with the technical problems that must be

In addition to the technical problems, there are other problem

issues are more management-oriented than the engineering is-

in an organization requires management involvement as well as

This part of the book is concerned with the management issues

16) provides a step-by-step approach that a company can apply

presumes a minimum starting knowledge about robotics on the

survey, selecting a proper application, and selecting the best

the installation of the robot cell, relying on analysis methods

operator safety, training, maintenance, and quality issues associ-

PART SIX

An Approach for Implementing Robotics

Introduction

Robotics is a sophisticated technology and the successful implementation of this technology in industry is a formidable management problem as well as a

Some of the steps described in the approach relate closely to the applications

of the approach go beyond the engineering techniques required to implement

The approach for implementing robotics is described in terms of a logical sequence of steps that a company would want to follow in order to

These seven implementation steps are discribed in the following seven sections

16.1 INITIAL FAMILIARIZATION WITH ROBOTICS TECHNOLOGY

expertise in robotics, but they believe that there are potential applications for robots of making rational decisions on robot projects, these personnel are faced with the

The sources of information on robotics include books, technical magazines and

In addition to these materials, there are also seminars, conferences, and trade

seminars are also offered by a number of organizations on topics such as end effector

Robotics International of SME, provides an opportunity for the robot industry to show its products and exchange ideas at the various technical paper presentations

time by turning on and off their support to their manufacturing staff functions as

successful, it is important that the manager responsible for using the system also be

makes sense to try to use their knowledge in developing and implementing the robot

developed a work force acceptance checklist to assess the potential for successfully plant, the problem of installing and operating a robot cell in the plant becomes much

Item	Points to be
points on similar issues for management-work	
management units shown the ability to establish rapport with workers or does	
concern for the dehumanizing aspects of	
workers who will supervise or perform	
penalized by new rates or robot downtime	
respect and regard for the talents, skills,	

Contd.

Range of Net Score *Probability of Acceptance*

Fig. 16.1 *General Electric work force acceptance checklist. (Adapted from General Electric source materials by permission.)*

16.2 PLANT SURVEY TO IDENTIFY POTENTIAL APPLICATIONS

the same general application selection criteria apply to both cases, the new facility

determine the best method for accomplishing the process using available robotics

is to substitute a robot in place of the human operator in an existing production

installation often requires that the robot be used in the same way that the human
In this second case, the potential applications for robots

In performing a plant survey, the objective is to determine those existing operations

characteristics will usually make a potential robot application technically practical
3

1. Hazardous or uncomfortable working conditions

The potential hazards include physical dangers and health hazards from heat, sparks,

not actually hazardous, but the workplace is considered uncomfortable, unpleasant,

these conditions, so they can more readily accept the automation of their jobs so long

applications in this category, including spot welding, arc welding, die casting, and

2. Repetitive operations Repetitive operations are very common in high-volume

suited to many operations in this category because of their capability to repeat a

requirements are that the robot must be provided with the proper end effector to

have been successfully used include pick-and-place operations, machine loading and

-

mechanical assistance to hold and manipulate these kinds of parts, such as a crane

4. Many manufacturing operations run two or three shifts in

temperature operations need startup periods that make it economical only to run the

using human workers, the labor cost is a variable cost that continues during the sec-

-

the operator needs some form of mechanical assistance in handling the workparts

16.3 SELECTION OF THE BEST APPLICATION

By surveying the plant operations, a number of potential robot applications can

appropriate to perform a preliminary economic analysis on the alternative potential

analysis, the existing operations would have to be studied to determine current production rates and costs, and a robot method would have to be proposed in order to

In addition to the economic criteria, the potential applications must be subjected

industrial operations during the survey[1]

given by experienced applications engineers is to start with a simple application, one that does not require a high level of sophistication in the workcell layout, workstation

to be a success so that the new technology will be accepted by the personnel in the

it is a good idea to accomplish the straightforward applications and minimize the risk

the technology of robotics, and this may cause the company to attempt an application

under pilot conditions in the laboratory, but its chances of success in the factory are

In the case of a new facility being planned but not yet in operation, the same

16.4 SELECTION OF THE ROBOT

Indeed, selection of the application must often be done with consideration of

The robot selected should possess an appropriate combination of technical features

when the current application is completed and another application is being sought for the robot, the new application may require greater technological capabilities than

table are considered to be representative of current robot application practice, but exceptions to the recommendations in the table can be found in successful installations

Table 16.1 *Technical features required of robots for selected applications.*

Application	*Typical technical features required*
Material transfer	Number of axes: 3 to 5 Control system: limited sequence or point-to-point playback Drive system: pneumatic or hydraulic (for heavy loads) Programming: manual, powered leadthrough
Machine loading	Anatomy: Polar, cylindrical, jointed arm Number of axes: 4 or 5 Drive system: electric or hydraulic (for heavy loads) Programming: powered leadthrough Control system: limited sequence or point-to-point playback
Spot welding	Anatomy: polar, jointed arm Number of axes: 5 or 6 Drive system: hydraulic or electric Programming: powered leadthrough Control system: point-to-point playback

Contd.

Arc welding	Anatomy: polar, jointed arm, cartesian Number of axes: 5 or 6 Drive system: electric or hydraulic Programming: manual or powered leadthrough Control system: continuous-path playback
Spracy coating	Anatomy: jointed arm Number of axes: 6 or more Drive system: hydraulic Programming: manual leadthrough Control system: continuous-path playback
Assembly	Anatomy: jointed arm, cartesian (box), SCARA Number of axes: 3 to 6 Drive system: electric Programming: powered leadthrough, textual language Control system: playback, point-to-point or continuous path Accuracy and repeatability: high

features for the particular application and then systematically comparing these

desirable features, and a rating score would be assigned to the candidate to indicate

relative importance among the various features, and this would be taken into account

score for the different robot models in each feature category would be a judgment call that the applications engineer would have to make based on the relative merits

Example 16.1 comparison of the application features against the available

The desirable features were each evaluated as to its relative priority by giving

The applications engineer made judgments to determine how each of the remaining

features listed in the form included non-technical considerations as well as technical

Technical Feature				
	Model A	*Model B*	*Model C*	*Model D*
'*Must Features*				
'*Desirabel Features*				
Total				

Conclusions

C

D

Fig. 16.2 *Sample form used to compare application features against robot technical speci-*

16.5 DETAILED ECONOMIC ANALYSIS AND CAPITAL AUTHORIZATION

appropriate robot application and has also decided which robot model would be best

The technical analysis would detail the engineering and technical feasibility of the required change to existing equipment, new equipment that must be acquired,

Based on the documentation of the economic and technical analysis, management

used to accomplish the detailed planning and engineering work and to purchase and

16.6 PLANNING AND ENGINEERING THE INSTALLATION

The planning and engineering of the installation involves many or the analysis and considerations in the approximate order in which the applications engineer would

Table 16.2 *Checklist of considerations and problem areas to be addressed during planning and engineering of the robot installation.*

1. Study of the operation method
   What is the basic purpose and function of the operation?
   What is the best method for a robot to perform the operation?
2. Design of robot workcell
   Which of the three basic types of cell layout should be used?
   (a) Robot-centered cell.
   (b) In-line robot cell.
   (c) Mobile robot cell.
   What changes to other equipment must be made to accommodate operation and control in the robot cell?
   Consideration of part positioning and orientation coming into the cell and leaving the cell.
   Consideration of part identification methods if more than one part style is processed through the cell.
   Protection of the robot from its environment.
   Provision for utilities and other services required for the cell.

Contd.

3. Workcell control
   What are the basic functions that must be performed by the work cell controller in this operation?
   What interfaces to the human operator must be included?
   What interlocks must be designed into the cell?
   What sensors must be used to accomplish the interlocks?
   Are there additional sensor requirements that must be satisfied?
   Type of workcell controller. Does the robot have sufficient control capacity or must an additional cell controller be incorporated into the cell?
4. Safety considerations designed into the cell.
5. End effector design.
6. Design of other tools and fixtures for the cell.

The initial item in the list is a careful study of the operation and the way it should

this issue during the selection of the application, the selection of the robot, and the

are likely to be differences between the most appropriate method for a robot and

robot, without some form of sensing capability, is unable to detect even the most

to inspect the parts before they are delivered to the workstation, or to incorporate a

applications engineer should attempt to develop a workplace layout that is best

For a stationary robot to perform the task, the cell would have to be designed so that

5 discussed the various types of end effectors and the calculations involved in the

16.7 INSTALLATION

Table 16.3 *Checklist of activities included in the installation phase.*

Purchase of the robot(s) and other equipment and supplies needed to install the workcell.
Preparation of the physical site in the plant where the robot cell is to be located. This might include altering the foundation to support heavy machine tools in the cell and to fix the relative positions of the robot and other equipment. Also included would be any provisions for protection of the robot from its environment (e.g., high temperatures, dangerous fumes or mist in the atmosphere, electrical noise, fire hazards, etc.)

1. Provision of electrical, pneumatic, and other utilities for the cell.
2. Adaptation of standard pieces of equipment for use in the cell.
3. Placement of robot and other equipment: installation of conveyors and other materials-handling systems for delivery of parts into and out of the cell.
4. Installation, checkout, and programming of the workcell controller.
5. Installation of interlocks and sensors, and integration with the workcell controller.
6. Installation of safety systems.
7. Fabrication of end effectors and other tooling.

might include software bugs, sensor problems, improperly located components in the

The time required to complete the installation is typically from three months to a

There are several additional issues related to installation that should be consid-

concerned with activities for which some planning must take place in advance of

References

Robotics Today,

Robotics in Practice,

CAD/CAM: Computer-Aided Design and Manufacturing,

Robotics Today,

Industrial Robots, Volume I-Fundamentals,

Robotics Today,

Safety, Training, Maintenance, and Quality

Introduction

17.1 SAFETY IN ROBOTICS

[6] These are:

17.1.1 Workplace Design Considerations for Safety

Fig. 17.1 *Cell layout using part manipulator to separate human worker from robot for safety*

17.1.2 Safety Sensors and Safety Monitoring

in robotics[4]:

of the robot.

Fig. 17.2 *Three levels of safety sensor systems: Level 1—perimeter penetration; Level 2—intruder detection in the workcell; Level 3 intruder detection inside the robot work*

17.1.3 Other Safety Measures

17.2 TRAINING

2

1. Awareness *Awareness*

applications

operation and maintenance

5. Safety *safety*

17.3 MAINTENANCE

17.3.1 Maintenance Staff

for robotics.

remedial maintenance

cost.

17.3.2 Preventive Maintenance

reports that one of

MTBF – MTTR
MTBF

Example 17.1

200 – 8
200

300 – 6
300

17.3.3 Spare Parts Policy

5

either as spare parts or when the robot breaks.

Example 17.2

×

×

year.

E $\times$ $\times$

E

17.4 QUALITY IMPROVEMENT

the operation.

roblems

17.1

17.2

Robot

5.2
6.6
2

6.2

3 3.4

5.5

4 45 3.3
5.5
236

3.2

(a)

(b)

(a).

(c)

the fact that there are

17.3

(a)

(b) **(a)**

ot.

17.4

References

Robotics in Practice

Industrial
Engineering
Industrial
Engineering
Technical Paper
MS82-221
Industrial
Engineering
Industrial
Engineering

Social Issues and the Future of Robotics

PART SEVEN

Social and Labor Issues

Introduction

Table 18.1 *Time allocations in three basic types of discrete parts manufacturing.†*

Plant activity	*High production, %*	*Batch production, %*	*Job shop production, %*
Plant shutdown	27	28	34
Losses from second and third shifts	—	40	44
Equipment failures	7	6	—
Inadequate storage	7	—	—
Tool changes	7	7	—
Load/unload, non cutting, etc.	14	4	
Setups, gauging, etc.		7	
Setups, gauging, loading, etc.			12
Losses from non optimal cutting			2
Other idle time			2
Work standards, allowances, etc.	16		
Productive cutting	22	8	6
	100	100	100

18.1 PRODUCTIVITY AND CAPITAL FORMATION

Units of output
Units of input

total
factor productivity

18.2 ROBOTICS AND LABOR

18.2.1 The Effect of Robotics on Direct Labor

6

Fig. 18.1 *Projected number of human workers displaced or not hired as a result of the installation of robots in manufacturing. These projections are determined by multiplying the installed base in Fig. 1.8 times a rate of substitution of three workers per robot. Since three is considered to be a relatively high substitution rate, the projected values are probably high.*

18.2.2 Effect of Robotics on Labor Unions

18.2.3 Quality of Working Environment

18.2.4 Impact on Professional Staff

18.3 EDUCATION AND TRAINING

OTAReports

18.4 INTERNATIONAL IMPACTS

Table 18.2 *Robot installations operating at end of 1982* **.**

Country	*Number*	*Per cent of total*
Japan	31,900	66
United States	6,301	13
West Germany	4,300	9
Sweden	1,450	3
Italy	1,100	2
France	993	2
United Kingdom	977	2
Belgium	305	<1
Poland	285	<1

Contd.

Canada	273	<1
Czechoslovakia	154	<1
Finland	98	<1
Switzerland	73	<1
Netherlands	71	<1
Denmark	63	<1
Austria	50	<1
Singapore	25	<1
Korea	10	<1
Total	48,428	

Source: *Worldwide Robotics Survey and Directory, 1983*

18.5 OTHER APPLICATIONS

roblems

18.1

(a)

(b)

18.2

References

American Machinist

Robotics Today

Robotics: Applications and Social Implications

Decade of Robotics

Industrial Engineering

U. S. New s & World Report

Decade of Robotics

Industry Week

OTA Report

OTA Report

Robotics Today

Robotics Today

19

Robotics Technology of the Future

Introduction

The approach adopted in the preceding chapters of this book has been to describe the state of the art in the technology, programming, and applications of industrial robots. However, the state of the art in this field is changing rapidly, and it is difficult for us to be satisfied with merely providing a description of the present status. The purpose of this chapter is to describe some of the research and development that is currently taking place and to forecast some of the future advances in robotics technology that will result from these efforts. We include robot programming and related software developments within the scope of robotics technology. The implications of these technological developments in terms of future robot applications will be discussed in Chap. 20.

As explained in chapter 1, robotics technology has progressed through three generation, since the advent of the microprocessor, cheap electronic and smart materials technology. Todays robots are intelligent, have multitude of sensing capabilities and are truly autonomous. Some aspects of present days robots and future robots may be listed as:

1. Intelligence The future robot will be an intelligent robot, capable of making decisions about the tasks it performs based on high-level programming commands and feedback data from its environment. Robots used for robot human interaction will be able to respond to human commands (etc. voice, gesture etc.) Such robots could also exhibit human behavior such as happiness and sadness etc.

2. Sensor capabilities The robot will have a wide array of sensor capabilities, including vision, tactile sensing, and others. With sensors becoming cheaper and computing power increasing rapidly, most robot will have a wide array of sensors for interacting with the environment. Sensors such as vision, GPS for positioning and localization, touch etc will be extensively used.

3. Telepresence It will possess a telepresence capability, the ability to communicate information about its environment (which may be unsafe for humans) back to a remote 'safe' location where humans will be able to make judgments and decisions about actions that should be taken by the robot.

4. Mechanical design The basic design of the robot manipulator will be mechanically more efficient, more reliable, and with improved power and actuation systems compared to present day robots. Some robots will have multiple arms with advanced control systems to coordinate the actions of the arms working together. The design of the robot is also likely to be modularized, so that robots for different purposes can be constructed out of components that are fairly standard.

5. Mobility and navigation Future robots will be mobile, able to move under their own power and navigation systems. Future robots are likely to go into areas like undersea, unmanned arial vehicles, into nuclear plants etc.

6. Universal gripper Robot gripper design will be more sophisticated, and universal hands-capable of multiple tasks will be available. The advent of smart materials would lead to the development of grippers using elastic or active fingers. Such gripper would be used for handling micro parts in micro assembly etc.

7. Systems integration and networking Robots of the future will be 'user friendly' and capable of being interaced and networked with other systems in the factory to achieve a very high level of integration. Also as the internet gets faster and more accessible the control of robots would be possible over the internet.

The following sections will present a review of these areas of robotics development.

19.1 ROBOT INTELLIGENCE

The concept of an intelligent robot is not a new concept. In Chap. 2, the 'intelligent robot' was defined as one of four basic control systems categories for industrial robots. Although this type of robot must be considered relatively primitive at the present time, the level of sophistication of the intelligent robot will evolve in the future as the prerequisite technological components (e.g., sensors, mobility, programming languages, networking, etc.) are developed. The concept of the intelligent robot includes the capability to receive high level instructions that are expressed as commands to do a general task, and to translate those instructions into a set of actions that must be followed to accomplish the task. The future intelligent robot will also be aware of its environment and will be able to make decisions about its actions based in part on an interpretation of its environment. We discussed these issues in the context of future generation robot languages in Chap. 9, and we discussed the field of artificial intelligence in Chap. 10. In this chapter we will present a summary of an alternative approach to robot control developed at the National Bureau of Standards (NBS).

The Industrial Systems Division of NBS[1,2,3,7] has designed the framework for a real-time hierarchical control system that will be capable of responding to goal-oriented instructions and will receive sensor data regarding its environment. The following summary of the NBS system will provide a narrative of the likely way in which the future intelligent robot will operate.

In the NBS Real-Time Control System (RCS), a hierarchical control structure is used. The overall scheme of the system as it would be applied in an automated factory is illustrated in Fig. 19.1. In the operation of the RCS, goals expressed as commands

Fig. 19.1 *Hierarchical control structure used in the Real-Time Control System at the National Bureau of Standards. (Reprinted by permission from Albus.[1])*

at the highest level are decomposed at each successively lower level into simpler commands. The levels at the top of the hierarchical pyramid would correspond to factory shop orders, and these orders would be decomposed respectively into workcell instructions, and workstation commands. In turn, these commands would

information at these lower levels of the RCS are shown in Fig. 19.2. At the lowest level, the signals to drive the robot, gripper, and other equipment are generated.

Fig. 19.2

System.

each control module is the same, regardless of its level in the hierarchical structure.

receives status information from feedback sensors and from the control module at the level immediately below. It also receives commands from the next higher level in the control structure. The module preprocesses the input data, combining and converting it into the proper format for the processing which is to be accomplished at that level. The processing is done by means of a state-table, consisting of a list of output procedures which are determined according to the status conditions represented by the input data. A postprocessor in the control module is utilized to convert the output procedures and relevant other data from the state-tables into formats that are required by components external to the immediate control module. These components would consist of other control levels or pieces of equipment that are controlled by the module.

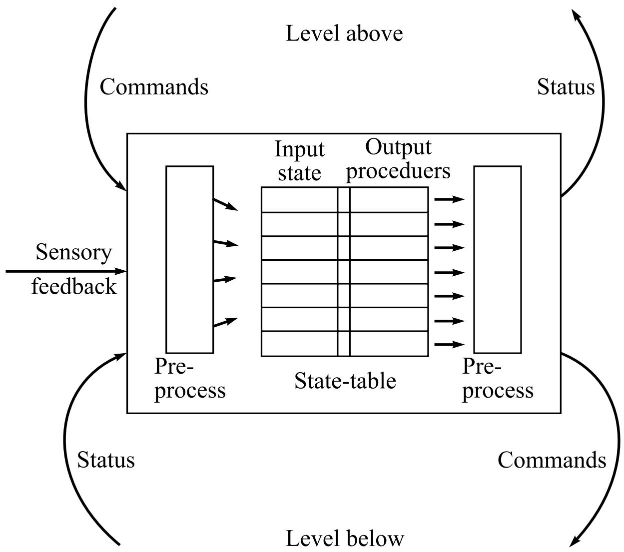


Fig. 19.3 *Configuration of each control module, irrespective of the level in the control hierarchy.*

19.2 ADVANCED SENSOR CAPABILITIES

The sensors used in robotics have previously been discussed in Chaps. 6 and 7. It is anticipated that the typical robot of the future will be much more richly endowed with sensor capabilities than today's machines. These sensor capabilities would permit the robot to be more aware of its environment, to communicate with human operators more readily, and to make use of the higher level of intelligence described in the previous section. Included among the advanced sensor capabilities will be three-dimensional vision and tactile sensing. The following paragraphs will address some of the research developments currently in progress in these areas.

19.2.1 Three-Dimensional Machine Vision

Although three-dimensional vision has been demonstrated to be technologically feasible (for example, the development work of Robotic Vision Systems, Inc.), there are a limited number of economically feasible systems being applied in industry today. It is expected that high-resolution three-dimensional vision sensors currently under development will be used in future applications to permit the robot to recognize objects more readily, to measure the distances between itself and objects using range-finding techniques, and to avoid obstacles in its path. Three-dimensional vision sensors would also facilitate safety monitoring by enabling the robot to easily determine the position of a human in the workcell, and to have its manipulator avoid contact with the human. The development of high-speed microprocessors will assist in the development of three-dimensional machine vision. One approach to three-dimensional vision is described. in Ref. 15. Stereo vision has been used on several space missions and also for 3D reconstruction. Using this technique a 3D virtual reality world can be developed for manipulation, path planning etc.

19.2.2 Tactile Sensing

Compared to human touch-sensing capabilities, current tactile sensors are quite rudimentary. Sophisticated tactile sensing would permit a variety of characteristics to be determined about the robot's environment or the object it is handling. These characteristics include the roughness of the contacting surface, the elasticity of the material, the weight of the object, its shape, and other physical features. Togai[20] reports that the Japanese are developing three types of tactile sensors in their work on the next generation robot:

1. Shear sensors This is the capability to sense slip between an object and the gripper finger surfaces. Applications of this sensor would be in handling both rigid objects as well as materials that are soft and limp, such as textiles and garments.

2. Contact sensors Contact sensors with multiple contact pads, similar to the Lord Corporation sensors described in Chap. 6, would be useful in recognizing objects and in positioning objects optimally in the gripper.

3. Force sensors These would be applied for detecting an object's elasticity, for holding an object with the proper gripping force, and for following contours with a certain specified force level.

19.3 TELEPRESENCE AND RELATED TECHNOLOGIES

As defined in Chap. 1, telecherics is a technology that involves the use of remote control manipulators, or teleoperators, to perform handling tasks. The technology is used for hazardous handling tasks (e.g., handling radioactive materials) inside an unsafe environment. It is anticipated that some robots of the future may be employed as sophisticated teleoperators, combining their own intelligence with the intelligence and judgment of humans located in some remote position. (Chapter 20 will explore a number of possible applications where the future robot would work in an environment that is hazardous for humans.) For example, the tasks performed may be so complicated that human assistance is required to deal with the complexities. When used in this way, the robot will need a two-way channel of communication with the humans: first, it will need to acquire information (e.g., sensor data) about its task and its environment and to transmit this information back to the humans; and, second, it will need to receive complex instructions and commands from the humans.

Telepresence is concerned with the first portion of this channel of communication. The functions of telepresence are: information acquisition using advanced sensor technologies, feedback of this information to a remote location, and display of this information in a manner that facilitates interpretation by humans.

The objective of information acquisition and feedback is to gather sensor data about the robot's task and its environment in a manner that simulates the human sensory functions. Three-dimensional vision and tactile sensing, as described in the previous section, would be two of the more important sensory capabilities involved. Other capabilities would also be involved, including the possibility that the robot, through its own sensors and intelligence, would be able to make observations and draw logical conclusions about its situation, so that these can be communicated to the remote location.

The objective in the display function is to make the information as complete and realistic as possible so that the humans almost believe that they are in the place of the robot. They will therefore be able to relate to the task situation more realistically. In effect, the humans will project their presence into the situation. The presumption is that this mode of operation will improve the performance of the remote robot by making its actions and reactions more like those of the humans who are guiding it.

The capability for the robot to use speech synthesis to communicate information about its task and environment is also a possible means for the future robot to augment the display function in telepresence. Speech synthesis is a technology that exists today. There are presently two commonly used techniques for speech synthesis by machine. The first relies on the fact that spoken English (or other human language) makes use of a limited number of sounds called 'phonemes.' Phonemes are the smallest units of speech that distinguish one utterance from another in the language. By stringing together the appropriate phonemes, whole words can be generated. A typical system based on this technique would consist of a microprocessor that selects the appropriate phoneme codes out of memory according to a prescribed set of rules. The codes would then be processed by a speech synthesis device so that the output would emulate the human voice. The problem that is encountered with this first speech synthesis technique is the inability of the system to make speech sound 'natural.'

The second common technique for speech synthesis uses spoken words that have been recorded by a human. Since the possibility exists for very large vocabularies to be required with the accompanying demands on computer memory, an approach known as linear predictive coding (LPC) is often used to compress the amount of information that must be stored in memory. LPC allows the system to record the human speech, and to compress the data and store it in memory. When used for speech synthesis, the LPC data are expanded and the system sounds almost exactly like the original speaker. Consequently, the LPC approach provides a more natural sounding synthesized speech than the phoneme-based technique. Of course, the limitation on the LPC system is that only those words that were recorded are available for use. In today's industrial environment, a limited vocabulary is usually adequate for most applications.

The use of speech synthesis to enhance the telepresence functions is a likely technology in the future of robotics. What is presently missing is the sophisticated artificial intelligence required to synthesize thoughts and observations into sentences that can then be communicated by means of speech.

The opposite direction in the channel of communication between the robot and remotely located humans is voice programming, the oral communication of instructions to the robot. Speech, as an input to robots, computers, and other machines, has been an area of research and development since the 1960s. The most natural way for a human to communicate is orally.

There are a number of issues and research problems involved with the use of voice programming. First, there is the problem of getting the voice data into a form that the computer can comprehend. The solution usually involves digitizing the amplitude

spoken word by comparing its frequency spectrum against a model stored in computer memory. The problem is complicated by the fact that different persons enunciate their words differently and possess different voice tones and speech patterns.

Another problem area is the separation of spoken words from each other. Most people speak in what is described as continuous speech, running the words together.

next to make the interpretation of each separate word. This problem is usually solved by speaking discretely, one word at a time.

restricting the number of operators who use the system. For example, a robot control system might only need to make use of the words, 'up,' 'down,' 'right,' 'left,' 'fast,' 'slow,' 'go to (point A),' and 'stop.' In the training of the system, each operator would repeat these words into the system a number of times so that the controller could learn the pattern of the user's voice in enunciating each vocabulary word. In the subsequent use of the system, the operators would be able to command the robot by speaking the words so that the speech recognition system could understand each word in sequence. For each word or set of words, the robot would accomplish some corresponding action or task. The assumption behind voice programming is that the human operator would be able to command and communicate with the robot more quickly and in a way that is more natural to the operator. Uses for voice programming in robotics would include applications in hazardous environments, and one-of-a-kind jobs where programming time is critical. Another area where this technology holds great promise is in the design of robotic aids for handicapped persons.

Both voice programming and speech synthesis are available today for implementation in robotics. However, they have seen very little use outside of the research

-

day's applications. It is anticipated that certain future applications, those requiring closer communication between the robots and their human masters, will use either or both of these methods.

19.4 MECHANICAL DESIGN FEATURES

improved power and actuation systems compared to current robots. Some future robots may have more than a single arm and will require advanced control systems to coordinate the actions of the arms working together. The design of the robot is also likely to be modularized, so that robots for different purposes can be constructed out of components that are fairly standard. One of the improvements in the area of mechanical design is the direct-drive robot.

19.4.1 The Direct-Drive Robot

The effects of backlash and compliance on the performance of a robot manipulator were discussed in Chap. 3. One way to reduce these effects is to couple the actuator directly to the joint. This robot design has been called the 'direct-drive mechanical arm' (or direct-drive robot, for short).[4,5]. Direct drive involves locating the motor or actuator for a given joint contiguous to that joint, thus eliminating the need for a power transmission mechanism between the joint and the motor. In the typical robot of today, the motor is usually positioned remotely in the robot base or body.

force sensing (see Chap. 6) or selectively applied joint compliance to facilitate assembly (see Chap. 15). Additionally, the direct-drive robot would require less maintenance due to fewer components in its fabrication. Finally, because fewer parts are needed to assemble the robot, this would presumably produce a favorable cost

task for a manipulator is to hold a load steady against gravity. When electric motors are used, this results in high currents running through the motor at stall. In order to support the load without the help of a power transmission to multiply the motor's stall torque, a very high current must be used to generate the necessary holding torque. If a

In a dc motor, the power in the armature is a function of the square of the current. The temperature rise of the armature is proportional to the power. On the other hand, if the motor is relatively large, it becomes too heavy to be practical for locating a motor

these two problems.

There are a number of ways in which the direct-drive technology is being made

If the joint is oriented so that the motor does not have to support the load against gravity, the motor is only required to supply the torque necessary to overcome friction and to accelerate and decelerate the mass of the total payload (arm and load). When the payload is at rest, the motor requires relatively little power, thus allowing the opportunity to dissipate heat. The SCARA design (Selectively Compliant Arm for Robot Assembly) has the joints oriented so that the axes of rotation of the principal joints are vertical, hence the load is carried by the joint frame rather than the motor. Figure 2.8 (Chap. 2) shows the SCARA design.

Another approach that facilitates the direct-drive robot design is in the design of the motors themselves. Developments are being made in the materials being used in motor design. One of the design considerations is that the torque that a motor

motor. New magnetic materials such as samarium-cobalt and neodynium-iron hold the promise of motors with higher torque capacities for the same current input.

19.4.2 Multiple-Arm Coordination

are typical examples of this case. In walking machines (described in the following section), there is a need to coordinate the actions of the legs in order to provide balance and propulsion for the machine. At the present time, it is possible to provide only a crude level of coordination of the motion of the two arms through the use of interlocks. No machine is available that is capable of the hand-to-hand coordination

is the capability of a robot equipped with machine vision to coordinate its actions with a moving conveyor to pick up a part in motion. While attacks on this type of

manipulators with each other is much greater. One aspect of the problem which is in need of solution is in the area of arm dynamics. The robot controller must be able to precisely determine the robot's present as well as its future locations quickly enough to achieve the level of control required in hand-to-hand coordination.

19.4.3 Human Centered Robotics

Recently Robots have started entering society in a variety of ways such as, helpers, rehabilitation devise, pets, security guards etc. Robots interacting with a human need to be designed keeping in mind safely as well as ergonomics. The three main designs for human friendly robots are:

3. Design of human friendly control systems.

Robots designed for human interaction should also be ergonomically designed, keeping in mind the basic structure of a human.

19.5 MOBILITY, LOCOMOTION, AND NAVIGATION

Today's industrial robots tend to be planted at one location. By contrast, materials and people in a factory generally move about. Providing robots with the capacity to move under their own power would greatly increase their potential utilization. Robot mobility in the factory environment can be used either to move materials from one

work volume of today's stationary robots. Outside of the factory, there are many possible applications for a mobile robot capable of self-navigation. There are two basic ways in which robots can be made mobile: wheeled vehicles and pedded vehicles (walking machines). The following subsections will examine some of the possibilities within these two categories.

19.5.1 Wheeled Vehicles

The current state-of-the-art in self-propelled machine locomotion is the automated guided vehicle (AGY). AGYs are typically battery-powered, three-or four-wheeled

systems, instructions to start/stop or change routes are communicated to each vehicle electronically over radio frequencies. Automated guided vehicle systems are typically used in today's applications for moving materials in warehouses and factories.

Some development work is now being done to add manipulators to the AGYs. In this way the guided vehicle would be able to transport materials between workcells and to load and unload the machines in the workcells. In effect, this type of guided vehicle-and-manipulator would represent a possible method of implementing a mobile robot cell, described in Chap. 11. Gantry robots, an example of which is pictured in Fig. 2.6 (Chap. 2), Two), represent one means of providing mobility for the manipulator.

In order for robot vehicles to be able to be used in uncontrolled environments, they would need to navigate autonomously without the use of tracks or guide wires in the

several 'intelligent' functions. These would include: 'scene analysis' (most probably implemented by means of machine vision), 'trajectory planning' (the ability to plan alternative routes from starting point to destination point and to select the best route among them), and 'obstacle avoidance' (the ability to develop strategies to navigate around obstacles along the planned vehicle path based on the scene analysis). These

intelligence. Prototype vehicles have been demonstrated at Stanford University[11] and at Carnegie-Mellon University which were able to view the scene in front of the vehicle and navigate around obstacles. However, the vehicles required impractically long times to move a few feet. The navigation problem is an active research area.[9,19]

With the exploration of planets and the moon getting more focus, several countries

settlement. The Mars missions have developed six wheeled rovers that can function autonomously and can also be telemanipulated from earth. Such rovers are still limited in moving over moderately rough terrain with stones, boulders etc. While future missions hope to explore craters or drill deep below the lunar surface for water. Such advanced rovers need to be designed for better traction control and stability, while minimizing power requirements and mass.

19.5.2 Walking Machines

Wheeled vehicles have limitations; they can only travel over relatively smooth surfaces. Vehicles with tank tracks would be an improvement for rough terrain. Walking machines[12,14,21] offer the greatest versatility for dealing with a variety of surfaces and obstacles. However, walking machines must over-come all of the same technological hurdles as autonomous locomotive wheeled vehicles, with the additional problem of coordinating the motions of the legs. In addition, since it is assumed that such vehicles will be used over rough terrains, they must be highly adaptive to the irregularities of the terrain.

There are a number of factors that must be considered in the design and control of walking machines. These factors include the number of legs, gait selection, balance, and coordination of the legs, Research has been done on one-legged, two-legged, four-legged, and six-legged machines. A one-legged machine would have no coordination problems, but would be much more difficult to balance than a six-legged machine. The gait of a walking device describes the sequence in which the legs are brought into use during the walking motion. It generally refers to the duration of the stance phase and swing phase of each leg. The stance phase occurs while the leg is on the ground providing support or propulsive force. The swing phase occurs while the leg is in the air preparing for the next stance. Some gaits are more stable than others. Six-legged walkers (such as cockroaches) employ a gait known as the 'alternating tripod' gait. In this gait, the pairs of three legs alternate between the stance phase and the swing phase. This enhances the stability of the walking machine by providing that there are three legs on the ground at a time. Another gait, called the 'crawl' gait, has only one leg in the air at a time; while a gallop has no legs in the air at certain moments during the gait. Figure 19.4 illustrates some of the possible gaits that might be used by a walking machine.

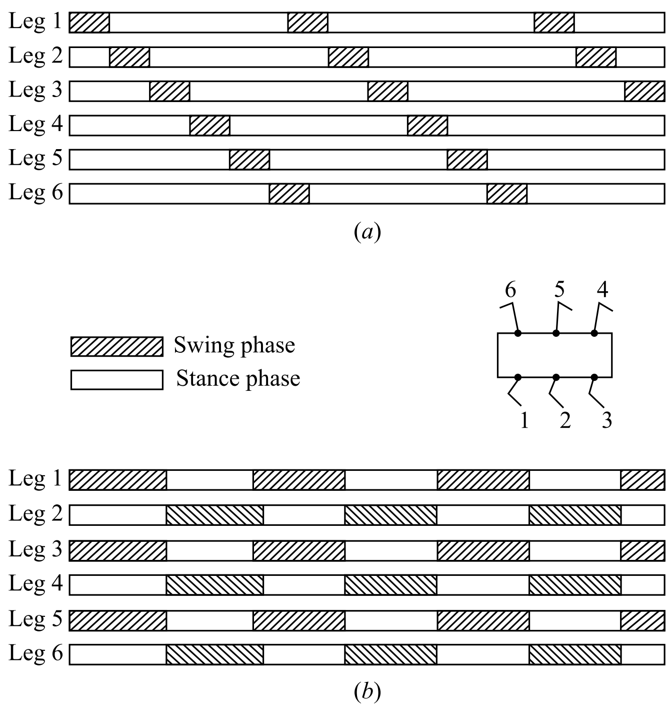


Fig. 19.4 *Some of the possible gaits that might be used by a walking machine.*

During the stance phase, a leg is subject not only to the forces required to support the machine, but also the forces generated by the motion of the body of the walking machine. Figure 19.5 illustrates the forces on the leg of a six-legged pedipulator. The leg must be able to control the motion of the foot to ensure that the body maintains motion along the desired path (usually a straight line path).

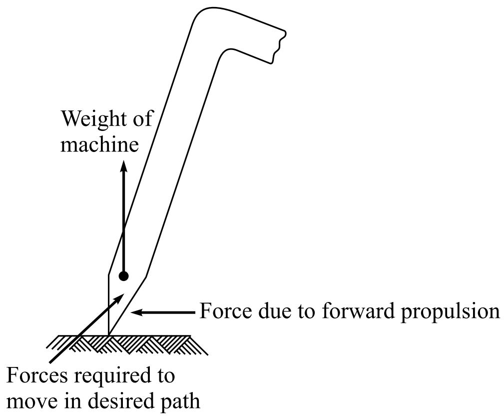


Fig. 19.5 *Diagram showing the forces on the end of the leg of a six-legged pedipulator.*

Walking machines present a significant challenge to the robot designer since the problems in their design include force sensing (in the foot), multileg coordination, balance, navigation, obstacle avoidance, and others. Figure 19.6 shows one commercial design of a six-legged walking robot, called the ODE X 1.[17]

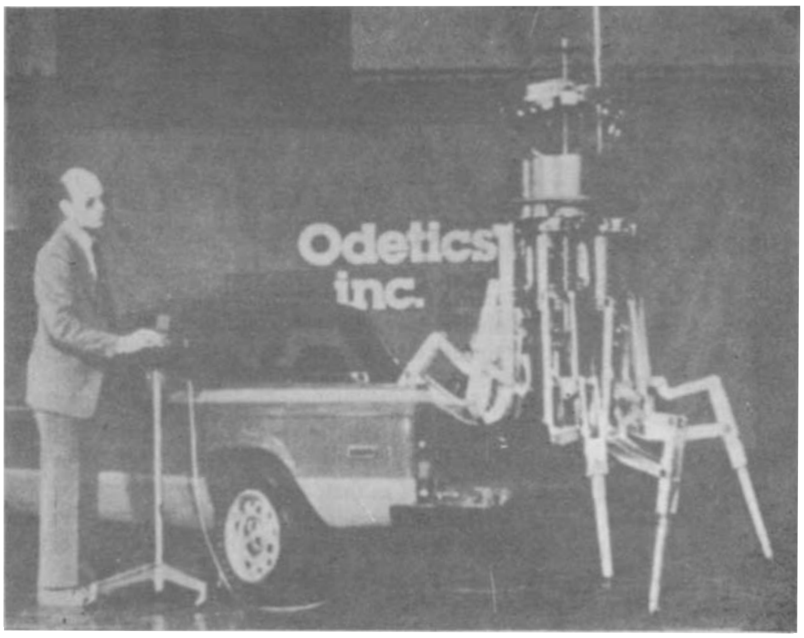


Fig. 19.6 *ODEX 1, a six-legged walking machine. (Photo courtesy: Odetics, Inc., Anaheim, CA)*

Several biped walking robots have been developed in the last two decades. The most famous amongst them being ASIMO from Honda. Biped robots walk by controlling the zero moment point such that it is inside the foot perimeter when the robot is on a single leg stance. The zero moment point is the point about which the resultant torque on the system is zero.

19.6 THE UNIVERSAL HAND

The design of the end effector is a critical consideration in the application of robotics

application. Our discussion in this section will deal exclusively with gripper design. By comparison to the human hand, a robot's gripper is very limited in terms of its mechanical complexity, practical utility, and general versatility. In order to realize the full potential of future robotics technology, grippers must be designed more like the human hand, both in their sensory and control capabilities as well as their

The servoed gripper represents one step in the direction toward increased versatility and utility. It is a technology which exists today. The servoed gripper is

with sensors and controlled by a servomechanism such as a dc servomotor. By

tactile and proximity sensing. An example of a gripper equipped with tactile-sensing capabilities is shown in Fig. 6.3 (Chap. 6). This gripper is capable of sensing forces being applied to a part or to some other object.

Another example of a servoed gripper is a prototype hand equipped with tactile and position sensing that was developed at the Massachusetts Institute of Technology. This prototype was demonstrated in an application in which the gripper would close

robot was able to determine whether it had picked up one cup or two cups, since

such as this would be capable of handling a large assortment of parts in assembly applications and other tasks.

The anthropomorphic hand is another approach in the pursuit of the universal gripper. Development work is proceeding at a number of research centers to develop an articulated hand with attributes similar to those of the human hand.[10], N13, 18

thumb, is a most universal tool. It can do much more than simply hold an object. For

in their hands without putting the object down. Most of the work in articulated hand

of designing a three-armed robot with tactile and force sensing and a control system that is sophisticated enough to coordinate the actions of the three arms. Aside from the control problem, the designer must be able to package about nine motors or actuators into a very small volume (perhaps slightly larger than the human hand). One approach to solve this packaging problem has been to use tendons (small cords and pulley systems) and to locate the motors in the base of the hand or in the arm

shown in Fig. 5.17 (Chap. 5).

19.7 SYSTEMS INTEGRATION AND NETWORKING

developed are the Salisbury hand, MIT/UTAH hand etc. Most of these hands are actuated via wire rope mechanisms with the actuators located above the wrist. With the new trend in smart materials there has been several attempts of developing micro and nano hands for handling small parts and tasks.

A technological area which goes beyond robotics is in systems integration and computer networking. Robotics is only one of many computer-oriented technologies that will be used in the future automated factory. These technologies include computer-aided design, computer-aided process planning, manufacturing resource planning, manufacturing information systems, expert systems (and other applications of

and other examples of production automation. The problem facing designers of the products in these technologies is to make them compatible with each other so that they can be integrated into a single factory control system. A related problem is to design the robots and other systems so that they can be easily installed and connected to the existing factory network.

In the factory of the future, all of the production equipment will share a common data base, perhaps in the general arrangement diagrammed in Fig. 19.1. In order for robots to participate in the factory network system, they must be able to communicate with the network. A good deal of effort is currently being exerted in the United States

able to share some of the data with host computers, such as motion programs or machine status information. Eventually, more data will need to be communicated between the various stations in the network in order to realize the full capabilities of the factory wide real-time control system described in Sec. 19.1.

References

1. J. S. Albus, , BYTE Books, Peterborough, NH,

2. J. S. Albus, A. J. Barbera, and R. N. Nagel, 'Theory and Practice of Hierarchical Control,' , Twenty-Third IEEE Computer Society International

3. J. S. Albus, C. R. McLean, A. J. Barbera, and M. L. Fitzgerald, 'Hierarchical Control for Robots in an Automated Factory,' , 13th International Symposium on Industrial Robots and Robots 7, Chicago, IL, April

4. H. Asada and T. Kanade, 'Design of Direct-Drive Mechanical Arms,' *Technical Report CMU-R1-81-1*, Carnegie-Mellon University, April 1981.
5. H. Asada, K. Youcef-Toumi, and R: Ramirez, 'M.IT. Direct-Drive Arm Project,'
6. R. Ayres and S. Miller, 'Industrial Robots on the Line,' ,
7. A. J. Barbera, M. L. Fitzgerald, J. S. Albus, and L. S. Haynes, 'RCS: The NBS Real-Time Control System,' National Bureau of Standards, Working Paper (undated).
8. J. F. Engelberger, , AMACOM, New York, 1980, chap. 9.
9. M. Julliere, L. Marce, and H. Place, 'A Guidance System for a Mobile Robot,' , 13th International Symposium on Industrial Robots and
10. D. Lian, S. Peterson, and M. Donath, 'A Three-Fingered, Articulated, Robotics Hand,' , 13th International Symposium on Industrial
11. H. P. Moravec, 'Obstacle Avoidance and Navigation in the Real World of the Seeing Robot Rover,' PhD Thesis, Stanford University, Palo Alto, CA, 1981.
12. K. Pearson, 'The Control of Walking,'
13. D. A. Petersen, 'Development of a 16-Axis End Effector,' ,
14. M. H. Raibert and I. E. Sutherland, 'Machines that Walk,'
15. M. Rioux, '3-D Camera Based on Synchronized Scanning,' *Conference* , 13th Jnternational Symposium on Industrial Robots and Robots 7,
16. , American Society of Mechanical Engineers, New York, 1982.
17. M. Russell, 'ODEX I, The First Functionoid,' *Unmanned Systems*, Fall 1983.
18. J. Salisbury and J, Craig, 'Articulated Hands: Force Control and Kinematic Issues,' 1(1), (1982).
19. M. Takano and G. Odawara, 'Development of New Type of Mobile Robot TO-ROYER,' , 13th International Symposium on Industrial
20. M. Togai, 'Japan's Next Generation of Robots,' *Computer*
21. M. Weiss, *Bachelor's Thesis*, Massachusetts Institute of Technology, Cambridge, MA, 1977.

20 Future Applications

Introduction

The kinds of technological advances described in Chap. 19 will permit robots to be used in new applications for the future. It is not possible to predict all of the future opportunities for the use of robots or to forecast the order in which the applications will occur. A combination of economic and technical factors will determine how these applications will be introduced. It seems clear that future robot applications will include not only manufacturing operations but also non-manufacturing operations as well.

In the early 1970 robots were mainly confined to industrial applications. The main users were the automobile companies for spot welding, spray painting or simple pick and place operations. As robotics advanced, commensurate with the development of computer and electronics technology, it entered a new phase of non-industrial use of robots. Such robots were engaged in diverse applications such as defense, rehabilitation, medical surgery, physical therapy, house hold use etc. Hence, today we have two classes of robots (a) industrial robots and (b) service robots. Industrial robots as of today are still engaged mainly in the automobile industries or other manufacturing industries to do repetitive tasks such as welding, painting, or in hazardous applications. Service robots (defense, rehabilitation etc.) are increasing by the day and it is expected that one day these type of robots will be more in number than their industrial counterparts. However there are still many types of tasks that robots find difficult to do. Most of such tasks are still performed by humans in industries etc. Such task may deal with incomplete or uncertain information, flexible material handling etc. A few of such difficult tasks are given in the next section.

This final chapter will explore some of the possible applications, with greater emphasis on the non-manufacturing uses of robots in the future. Considering that employment in manufacturing constitutes only about 18 per cent of the total work force (19 million workers out of a total employment of 103 million as indicated in Chap. 18), greater opportunities for robot applications may ultimately be found in the non-manufacturing sector of the economy. We will divide our presentation of the future applications into three areas: (i) manufacturing, (ii) hazardous and inaccessible environments, and (iii) service industries. Before discussing these three application areas, let us examine the kinds of common chatacteristics that these applications will possess, and the capabilities of the robots that will perform them.

20.1 CHARACTERISTICS OF FUTURE ROBOT TASKS

The most prominent characteristic about present-day robot applications is that they require the robot to perform a repetitive motion pattern. Although the motion pattern is sometimes complicated, variations in the motion pattern are minimum. Also, the level of sensor technology required in the application is fairly low.

The enhancements in the technological capabilities of future robots will permit the applications to evolve in new directions. Some of the important characteristics of future robot tasks that will distinguish them from typical present-day applications are the following:

1. The tasks will be increasingly complicated. In addition to repetitive tasks, robots will perform semirepetitive and even non-repetitive operations.
2. The tasks will require higher levels of intelligence and decision-making

   be incorporated into the design of robot controllers.
3. Some of the tasks will require 'robust mobility,' the capability to move about the work area without relying on rails or moving platforms to execute the move.
4. The robot tasks of the future will commonly make use of a variety of sensor capabilities, including vision, tactile sensing, and voice communication.

   effector technology. The requirements for hand articulation and tactile sensing capabilities will be far in advance of today's gripper devices. The concept of the universal hand will be much closer to reality.
6. The greater variety of robot applications will require that robot anatomy become more specialized and differentiated according to the applications. The

   that the robot is supposed to serve. The economics of this specialization will be

   construction, and standardization of components.

   maintaining, and repairing the machine. The reliability improvements will also be incorporated into robot designs that are not used in these kinds of environments.
8. The inaccessible environments may require the use of a telepresence capability, so that humans can instruct the robot during the task.

20.2 FUTURE MANUFACTURING APPLICATIONS OF ROBOTS

Robots are extensively used in manufacturing as we have seen in earlier chapters that the cost of labor is going up while the cost of robots are going down. Some examples of robotic use in industries is as given in next.

And by the year 2000, if the projected trends continue, the installed base will increase to approximately one million units. Not all of these units will be employed in manufacturing operations, but many of them will. The question is: what will they be doing? Will they perform the same kinds of tasks that robots perform today, or will future robot applications in manufacturing be different?

Let us attempt to estimate the application trends to see how robots are likely to be used in manufacturing operations in the future. We will also consider the various problem areas that are presented by these operations and how future robots might be capable of dealing with the problems.

Non-manufacturing uses of robots are very limited in numbers and dollar value at the time of this writing (July 1984). Examples of these current uses include research and development applications and teaching robots in colleges and universities. Teaching robots constitute a growing share of the market in terms of numbers of units; however, the price per unit for these robots is low compared to the price of an industrial grade robot. By 1990, the non-manufacturing applications will still constitute a relatively small proportion of the total robot installations. Sometime after the turn of the century, this proportion is projected to grow into a majority.

20.2.1 Assembly Applications

The assembly process represents an important future application for robots, and we have discussed this application in Chap. 15. The area of assembly in which robots are expected to be used is in batch production operations. In the mass production of relatively simple products (e.g., flashlights, pens, and other mechanical products with fewer than 10 components), robots will probably never be able to compete with fixed automation in terms of speed and throughput rates. Even with lower-cost robots in the future, the economics will favor the use of high-speed specialized machines to accomplish the assembly tasks for these products. It is in the batch assembly of medium and small lots (e.g., electric motors, pumps, and many other industrial products) and in the high production of more complex assembled products (e.g., automobiles, televisions, radios, clocks), that robots are most likely to be utilized. However, these kinds of operations are currently the domain of human workers who possess the intelligence, dexterity, and adaptability needed for the tasks that go far beyond the capabilities of present-day robots.

This general area of assembly automation is sometimes referred to by the name programmable assembly. The present state of the art in programmable assembly is such that relatively few robots are employed in this technology. An estimated five per cent of the current systems use robotics technology, but this proportion is expected to grow to 30 per cent in 1990.[12] This suggests that advances in robot technology directed at the assembly process, combined with a better understanding of programmable assembly techniques, will occur during the period between 1985 and 1990. Some of the technological improvements needed to introduce robots in greater numbers into the assembly process include:

1. Improvements in sensor technology (especially machine vision)
2. Higher accuracy and repeatability
3. Higher speeds
4. Changes in design concepts and fastening methods for products to permit easier assembly by robots

6. Improved off-line programming methods that will permit complex robot programs to be developed from design data with the aid of advanced CAD/

   the required assembly tasks

rechnology is electronic assembly. The tremendous growth potential in the electronics industry over the next two decades provides a substantial impetus for the robotics industry to develop new robotic assembly systems.

20.2.2 Arc-Welding Application

arc-welding operations are accomplished manually today. Present state-of-the-art robot arc-welding installations almost invariably involve the production of medium or high quantities of items. In this situation, the robot must be programmed to perform the required welding cycle and the parts to be welded must be placed in

cell as explained in Chap. 14. The productivity of these semi-automated welding cells can be two or three times as high as the corresponding manual cell in which

encountered in manual welding operations. The economics of the application require

unit of product can overcome the initial cost of the programming time and the special

One of the technical problems described in Chap. 14 that arises in using robots for arc welding is the variation in the part edges that are to be welded. Human welders are capable of compensating for these variations during the welding operation, but the conventional playback robot cannot. This inability of playback robots to follow the variations in the welding gap has inhibited their use in the arc-welding process. Several sensor technologies are being developed to deal with the problem of part edge variations. It is anticipated that the widespread adoption of these sensor technologies will be an important factor in the expanded use of robots for arc welding in 1990 and beyond.

20.2.3 Parts Handling and New Robot Applications

Parts handling and machine loading represent a third large area of future robot applications, although the proportion of these applications compared to others will probably decline modestly. Perhaps the biggest limitation in using robots for these functions is the problem of locating and orienting the part so that the robot can fetch it at the beginning of the work cycle. In the past, there has been only one solution to this problem and that is to present the parts to the robot in a known position and orientation. This solution requires the parts to be prepositioned and preoriented for the robot application by means of some form of parts-handling device. Additional expense is involved to engineer the parts-handling capability and sometimes an extra manual operation is required to load workparts into the device. The problem of part position and orientation in robot applications provides reinforcement to a general argument among factory automation specialists that part orientation must be established when processing of the part initially begins and should never be lost during subsequent manufacturing and assembly operations. This is certainly not the common practice in today's factories geared largely toward manual operations in which parts are usually stored in random arrangements in tote pans, bins, and boxes.

Running counter to this part orientation argument are the recently introduced commercial systems capable of retrieving workparts that are all mixed together in a tote pan. These systems are called 'bin-picking' systems. They are based on the use of machine vision and we have previously described their operation in Chap. 7. It is anticipated that this bin-picking capability will be an important factor which will allow robots to be used in an increasing number of parts-handling, machine-loading/unloading, and many other factory operations. According to the Delphi study by the [12] the proportion

of the total market in 1990 and 10 per cent in 1995. Robots equipped with machine vision to accomplish the bin-picking problem might simultaneously perform visual inspection operations on the parts that are being retrieved.

In Table 20.2, there is an entry listed as 'other manufacturing operations, not covered by the above categories.' We expect that several new robot applications in manufacturing will be developed that are not discussed in Chaps. 13 through 15, and

of the total. The advances that are being made in robotics technology and related computer software will allow new uses of industrial robots that are today only found in the laboratory or not yet even seriously considered. These possibilities include wire harness assembly, garment manufacturing, shoemaking, product packaging, food-processing operations, dipping cycles in electrochemical plating, and a host of other unanticipated operations.

20.2.4 Flexible Manufacturing Systems

Finally, another technology that will spur future robot applications in manufactur-
-

numerically controlled machines) that are interconnected by means of a materials handling and storage system, and which operates as an integrated system under computer control. Although these systems started appearing approximately 15 years ago,

and operate as improvements are made in the technology and as we learn more about them. The improving cost advantage compared to other forms of production will

2

systems and machining cells to perform the materials handling function. Industrial robots with properly designed grippers have been found to be ideal for handling rotational workparts in this type of application. Figure 20.1 shows a possible layout for a machining cell that uses a combination of robots and conveyors to handle parts. The conveyors are used to bring parts into and out of the cell and the robots are used to handle parts between machines in the cell.

Fig. 20.1 *Layout of a machining cell that uses two robots to handle parts between machines. Note the in-process storage for transferring parts between robots. Parts are delivered into and out of the cell by conveyors.*

20.3 HAZARDOUS AND INACCESSIBLE NON-MANUFACTURING ENVIRONMENTS

Manual operations in manufacturing that are characterized as unsafe, hazardous, uncomfortable, or unpleasant for the human workers who perform them have traditionally been ideal candidates for robot applications. Examples include die casting, hot forging, spray painting, and arc welding. The workers who are displaced by robots in these operations are usually relieved to be out of the workplace as long as they are given alternative jobs that are better. It is anticipated that the same reaction will apply to hazardous manual tasks that are performed in non-manufacturing situations.

The desire to remove a human worker from an unsafe environment is a worthy ambition and will undoubtedly lead to the development of new applications for robots. Additional robot applications will be developed for environments that are either inaccessible or altogether inhospitable for human beings. Examples of potential non-manufacturing robot applications that are in hazardous or inaccessible environments include the following:

1. Construction trades
2. Coal mining
3. Hazardous utility company operations
4. Military applications
5. Fire fighting
6. Undersea operations
7. Robots in space

We will explore some of these possibilities, but not in great detail. For the interested reader, several of the references provide more comprehensive discussions of the topics.[1,2,6,8,13]

20.3.1 Robots in the Construction Trades

As indicated in Table 20.2, nearly all of the present applications of robots are in manufacturing, where they substitute for human workers in operations that are manual labor intensive. The construction industry represents an interesting opportunity for applying robotics technology because it is also based largely on the use of manual labor. There are three features about the manufacturing operations that have made robots relatively easy to apply. First, many (though certainly not all) of the operations where robots are substituted for human labor are hazardous. This feature has promoted the acceptance of robots in manufacturing by the workers. Second, the production operations can be performed at a single work location. And third, the tasks are highly repetitive. These three features apply only in varying degrees in the construction trades. First, some construction work must be considered hazardous since it is performed at high elevations. In spite of safety percautions that can be taken, there is no doubt that fabricating the steel frame of tall buildings or constructing high bridges is dangerous for the construction workers. Second, in the building trades, robots would typically be required to move about the construction site rather than remain at a single location. Providing the robot with mobility is

that the construction site usually consists of dirt piles, ruts, ditches, and debris, and the robot would not only have to negotiate around these obstacles, but it would also have to climb stairs and squeeze through doorways as well. The third feature of many manufacturing jobs that make them ideal for robots to be used is the repetitive nature of the work. In high production and even in batch manufacturing, the robot performs either the same or a limited number of motion patterns over and over again. Some construction jobs can almost be considered repetitive. Examples of these jobs would include ditchdigging, bricklaying, applying roof shingles, painting, laying ceramic tile, installing insulation, and other similar tasks. These jobs require a repetition of motion cycles that are very similar, but the location of the work is always changing and this requires a translation of the motion pattern in space with each new cycle.

Another reason why construction work is an interesting possibility for the application of robotics is that many of the common construction machines use mechanisms similar to the mechanisms used in robotics. Backhoes, front-end

their hydraulically operated manipulators and digging claws. All of these machines currently work under the control of human operators who direct the machines to accomplish their digging or lifting functions.

Solving the robot mobility problem might be done by borrowing from the

large wheels could be used to move about the construction grounds, with hydraulically operated legs to stabilize the robot at each location. For digging long trenches, the robot would move through a series of locations forming the line of the trench, with a similar motion cycle repeated at each location to dig the section of the trench. Getting the robot to follow the correct trench line and to accomplish the proper amount of excavation at each section, without any human assistance, would require a

that goes slightly beyond the current state-of-the-art in robotics.

For performing work internal to a building during construction, an entirely different anatomy would be required in the robot design. Perhaps in this case, the optimum method of solving the mobility problem would be to use a pedipulator. A walking robot might be designed which could climb a staircase as well as move

would be to provide it with the ability to adapt to the variety of different tools (drills, hammers, chisels, paint brushes, sanders, caulkers, etc.) that are used in the building trades. For a technologically sophisticated construction robot, this might involve a simple change of end effectors. The ability to effectively utilize all of these tools is

intelligence.

These technical problems will be solved probably within the coming decade. When this happens, the development of construction robots will be reduced to a problem of economics. The prospect of being able to keep a robot working nearly 24 hours per day without having to pay time-and-a-half or double-time will probably be appealing enough to motivate the introduction of robotics in some form into the construction trades before the next century.

20.3.2 Underground Coal Mining

Among industrial occupations, underground coal mining is one of the most dangerous

conditions have dramatically improved since the early 1900s, deaths in mining accidents still run at a much higher rate than the average of other work accidents. Around 1900, fatalities in mining accidents were running around 450 per 100,000

since 1969 with the enactment by Congress of more stringent mine safety laws, to around 80 deaths per 100,000 per year. This compares with an average rate of about 10 fatalities for all industrial work accidents. Even if the mine worker survives the accident statistics illustrated in these numbers, there is still the problem of 'black lung' disease that is so common among those who have been exposed to the environment in underground mines for many years. Yet coal is perceived as a major source of

the coal from the earth in a manner that does not place humans at undue risk for their lives is one that might be addressed through the use of robotics technology.

Highly mechanized systems are currently being used in the mining industry but their operation requires the attention of human workers for guidance and control. Examples of these systems include power undercutters and rotating head machines for digging and excavating at the mine face, conveyors for bringing the coal up from the excavation site to the surface, and loading machines for transferring the freshly dug coal to the conveyors. Full automation in mining operations would involve the implementation of machinery which could accomplish one or more of these functions without humans in attendance.

One possible 'robotic' machine for doing the tunneling and retrieval of the coal from the cut face of the mine is pictured in Fig. 20.2. The mechanical complexity of the tasks to be accomplished are suited to robotics technology; however, once again the programming, sensory, and intelligence requirements go somewhat beyond the available state-of-the art. To be fully automatic, the tunneling robot would have to be capable of moving forward into the mine as the excavation proceeds. Accordingly, mobility would have to be designed into the machine, probably by means of tank

surface and the shape of the opening to be tunnelled) could be placed in the robot's memory. A vision system would be the likely sensor system to determine how the cutting is progressing and to guide the robot in its digging operations. The dirty, dusty environment in the mine as a result of the excavation process presents a problem in mechanical reliability, and the design of the machine would have to take this problem into account. Also, the large forces encountered during the drilling process would add to the wear and tear on the machine. Periodical1y, as the tunnel is carved out, roof support would have to be constructed and the conveyor systems would have to be extended forward into the mine opening. These tasks might still be accomplished by human workers, or alternatively by sophisticated construction robots equipped for such work.

Fig. 20.2 *Robotic coal mining machine for tunneling and scooping coal from the mine face.*

The technical complexity of the coal mining operation described above poses

independently of human attendance. However, given that there will be a commercial demand for coal well into the next century and that there are approximately 150,000

machine is great.

20.3.4 Utilities, Military, and Fire Fighting Operations

There are many other non-manufacturing work situations that present hazards or potential hazards to those employed in the work. These include certain utility

technology might be utilized to reduce some of the risks to humans.

Some of the activities associated with utility company operations are unsafe for humans and represent possible applications for robots. The most prominent examples are maintenance and repair operations in radioactive boilers and the handling of nuclear fuels and other radioactive substances. Other examples include utility pole construction and repair and other high wire activities, coal pile grooming and

liners and ducts during planned boiler outages. Additional utility company activities might include construction work (e.g., construction robots) for power stations and security monitoring of plant and surrounding areas.

Contemplating the various possible uses for advanced humanoid robots in military

of sending robots on a suicide mission deep into enemy territory without risking the lives of friendly soldiers must surely be a source of great interest to military

operations and the logistics support for these operations might be performed by

or ship cannons, working in the engine room on board ship, and construction work for pontoon bridges and other temporary structures.

such as explosive mine disposal.

20.3.5 Undersea Robots

The ocean represents a rather hostile environment for human beings due principally to extreme pressures and currents. Even when humans venture into the deep, they are limited in terms of mobility and the length of time they can remain underwater. It seems much safer and more comfortable to assign aquatic robots to perform whatever tasks must be done underwater. Among the possible uses of undersea robots are the following: exploring for minerals, gathering geological samples, underwater mining and drilling operations, retrieving lost objects, underwater construction, and undersea

extended periods of time at virtually any depth. They would be largely self-contained, equipped with on-board power supply, sensors, computer, and several manipulators for various functions. The power supply would probably consist of batteries which would not be nearly so heavy underwater as they are for vehicles on land. Other forms of power using fuel cells might also be adapted for underwater operation over extended periods of time. Hydraulic actuators and motors could operate using water

water jets combined with controlled buoyancy. The robots could be designed to be virtually weightless underwater, achieving the appropriate buoyancy by means of air

would be changed to permit the robot to regulate its underwater depth. The on-board sensors and computer would permit the robot to perform many of its normal functions independent of any human interference. However, radio communication would allow instructions to be given from surface ships and to provide sensor readings from the robot. Video cameras mounted on the robot would permit those on the surface to see underwater and perhaps to give commands accordingly.

The U.S. Navy has been using underwater vehicles since the mid-1960s with features similar to some of the ones described above. The term used by the Navy for these 'robots' is remotely operated vehicle or ROV. Instead of being self-contained, these vehicles usually have an umbilical cord to the surface for power and control. Applications for these undersea vehicles have consisted mainly of recovering military ordnance lost near the coast, using a gripper mounted at the end of a manipulator arm to grasp the items. Fig. 20.3 illustrates the Navy's Cable-controlled Underwater Recovery Vehicle.

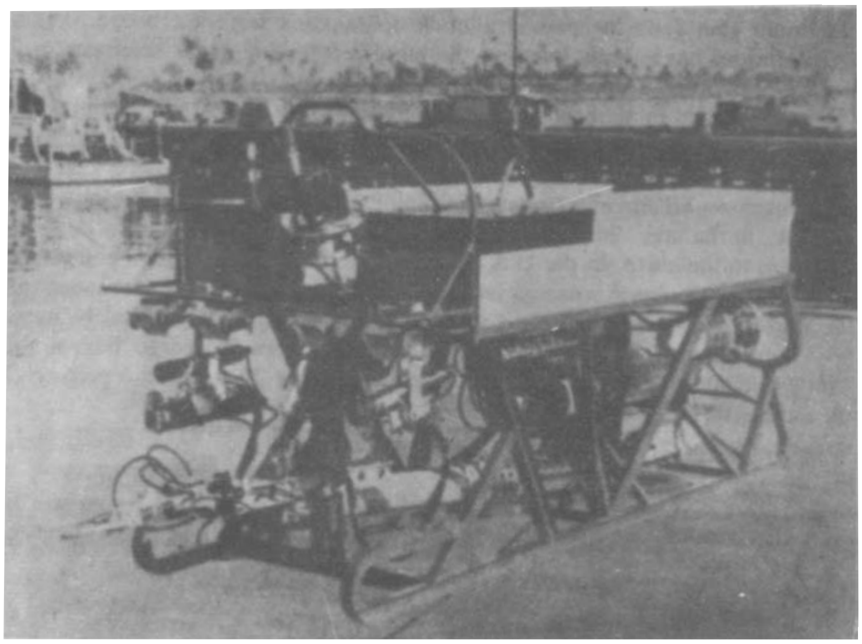


Fig. 20.3 *U.S. Navy's CURV III (Cable-controlled Underwater Recovery Vehicle). Note gripper arm in front of vehicle in lower left of picture. (Official Photograph: U.S. Navy)*

20.3.6 Robots in Space

Space is another inhospitable environment for humans, in some respects the opposite of the ocean. Instead of extremely high pressures in deep waters, there is virtually no pressure in outer space. In order to permit humans to survive the extreme conditions, they must be contained in some form of life-support system that provides pressure, air, and other requirements. In future space travel to faraway planets, the sheer enormity of the distances involved compared with the limitations on rocket velocity means that humans would be required to spend, perhaps, years away from earth in order to accomplish a space voyage within our own solar system. (Travel outside of the solar system would require more time than humans have available.) The safety issues involved in space travel would be considerable. Reliability of the equipment over extended time periods would pose a significant risk for human space travelers.

Robots would not need the elaborate support systems required for humans, and the time factor in space travel would have no emotional or psychological effects on robots. Equipment reliability would still be a problem but it would be only a reliability problem. There would be no threat to human life from equipment that fails in space travel if no humans were on board. These considerations have surely been on the minds of the engineers, scientists, and managers involved in the space program.

Technologies related to robotics have been used in the space program in several

manipulator was used to dig a trench on the moon's surface and to perform other

Space Shuttle started to use a 48-ft-long manipulator arm to remove payloads from the cargo bay of the shuttle and to handle various items in space. Figure 20.4 shows a picture of the shuttle arm.

Fig. 20.4 *Remote manipulator arm on-board the U.S. Space Shuttle for handling cargo and other chores in space. Note radius of earth between shuttle bay and manipulator*

The functions that would be performed by future robots and manipulators in space include exploration, construction in space, rescue missions, maintenance and repair, space transportation, materials processing, and other industrial operations in space.

could be programmed to roam the surface of the planet, gather samples, take measurements, perform experiments, analyze the data, and send the results back

software would be able to make decisions on where to explore, what samples to gather, and which samples to bring back to earth if a return trip is contemplated.

Robots could also be used in the construction of space stations, factories, and large cargo vehicles that are built in outer space. The robots could be used to move materials, help in docking maneuvers for sections of the construction, and perform other functions that would assist the human workers who are supervising the project. These applications would allow the number of humans required to accomplish the project to be reduced, thereby reducing the need for more life-support systems in space. Rescue missions for astronauts or construction workers stranded in space could be carried out by robots. Other uses of space robots would include maintenance and repair operations on the equipment, and space travel involving the transportation of humans and/or cargo through space. In each of these applications, humans would control the robots using high-level commands and the robots would have adequate intelligence to carry out the instructions.

space. Examples of these operations include containerless processing of liquid

metals without convection or sedimentation. Some biotechnology processes could

offered by space which are advantageous in these processes are zero gravity and zero atmospheric pressure (close to a perfect vacuum). In addition to other sophisticated forms of automation, the use of robots to accomplish these manufacturing processes in space would reduce the need for human attendants and their associated life support systems, and would probably lead to lower production costs for the resulting materials.

20.4 SERVICE INDUSTRY AND SIMILAR APPLICATIONS

In addition to non-manufacturing robot applications that are considered hazardous, there are also opportunities for applying robots to the so-called service industries. The possibilities cover a wide spectrum of jobs that are generally non-hazardous. We present the following subsections to illustrate the potential applications.

20.4.1 Teaching Robots

The concept of 'teaching robots' may extend beyond the use of small safe machines in college classrooms and laboratories. Such robots are widely used today for teaching the principles of programming (as well as limited applications) to undergraduates and two-year technical school students. In the future, teaching robots might be useful in elementary school systems. Children would be likely to consider a small robot (close to the size of a child) to be a friendly machine and would be willing to 'play' with the machine in an interactive mode to learn basic skills and concepts, much in the same way that personal computers are used today in many elementary schools. Robotic 'teachers' helpers' would multiply the capabilities of human teachers, perhaps increasing the permissible student–teacher ratio.

20.4.2 Retail Robots

Intelligent robots might be used in certain repetitive functions in retail establishments, such as cleaning, straightening the merchandise, checkout at cash registers, and merchandise restocking.

20.4.3 Fast Food Restaurants

Engelberger[4]

tasks required in a typical fast food restaurant. Fast food store operations are very labor intensive, especially in stores that stay open 24 hours per day. The skill levels required of the employees are very modest and many of the tasks are quite repetitive. With certain changes in the organization of the work in these restaurants, it is not

the food, dispensing beverages and ice cream, and making up orders based on instructions from a human order-taker.

20.4.4 Bank Tellers

Automatic tellers are used today for simple transactions such as deposits and withdrawals. Telephone checking is just beginning to be used as this chapter is being written. There will no doubt be a continuation of the trends in banking automation into the future, with the possibility that friendly teller robots may some day perform nearly all of the common customer-related transactions in a bank. Such a robot would have to be able to communicate in a manner which is unintimidating and convenient to the customer (voice recognition and speech synthesis technologies would have to be advanced beyond today's state-of-the-art). It would also have to add, subtract,

customer's account status.

20.4.5 Garbage Collection and Waste Disposal Operations

Collecting garbage is another operation performed by humans today which is mostly routine. There have been a number of attempts to mechanize garbage collection operations involving the use of large fabricated steel containers that could be readily

operations today still rely on one truck driver and one or two workers who must collect the garbage cans and empty them into the back of the garbage truck. These latter functions could surely be perfomed in the future by mobile robots specially designed for lifting the garbage cans.

20.4.6 Cargo Handling, Loading, and Distribution Operations

boxcars, typically require a combination of clerical and physical labor that is routine and prone to mistakes when done manually. For large distribution centers, automated storage and retrieval systems (AS/RS) are used to computerize and mechanize these clerical and manual functions. Installation of an AS/RS facility is usually a multimillion dollar investment. For the smaller warehouse that either cannot afford to install an automated storage and retrieval system or whose volume of operations does not warrant a large system, robots or robotic-type devices may become useful for some of the order picking and loading functions. As these functions are currently organized around the use of manual labor, the robots would require mobility and the capacity to handle variations in the shape and size of the items and containers used in warehouse operations. Although order picking is repetitive in a general sense, the locations of the items to be collected are different, and the robot would require

picking cycle. The robots would also need to be able to receive ordering instructions

20.4.7 Security Guards

Security guards lead a lonely existence, periodically roaming through the building to check for intruders and other irregularities. The duties also include sitting in front of closed-circuit TV monitors whose cameras are trained on entrances, exits, and other areas of the building and surrounding grounds.

robots, equipped with sensors to detect the presence of human intruders, could wander through the building on a random schedule designed to foil the intentions of burglars who might rely on a regular timetable to carry out their sinister activities. Sensing the presence of humans in unauthorized building space, the robot would communicate its observations to a central station manned by human security guards who are prepared to take appropriate action.

20.4.8 Medical Care and Hospital Duties

aides, orderlies, and technicians is clerical and routine. Robots are likely to perform some of this work in the future. Some of the hospital functions that might be automated include delivering linens, making beds, clerical duties such as entering

hospital pharmacy and central supply, and transporting patients for different services in the building. Some of the duties might even include aspects of patient care such as monitoring vital signs, and passing water and food to the patients.

A related medical care activity that might be performed by robots or robotic devices at some point in the future is assistance for paraplegics and other physically handicapped persons. Providing handicapped persons with full-time robot servants is a meritorious social objective that might eventually be realized.

20.4.9 Agricultural Robots

Although the labor content required to operator a farm has been drastically reduced over the last 60 years by mechanized equipment, there still remain opportunities for further automation. The Japanese[10] have identified a variety of tasks that might be accomplished with the help of future robots in the agricultural and related industries. These tasks include harvesting, soil cultivation, fertilizer spreading, and application of insecticides. Related areas of potential robot applications might be found in forestry and livestock care and management. The possibility of using robotic devices to shear sheep in Australia has been explored, and some of this work is illustrated in Fig. 20.5.

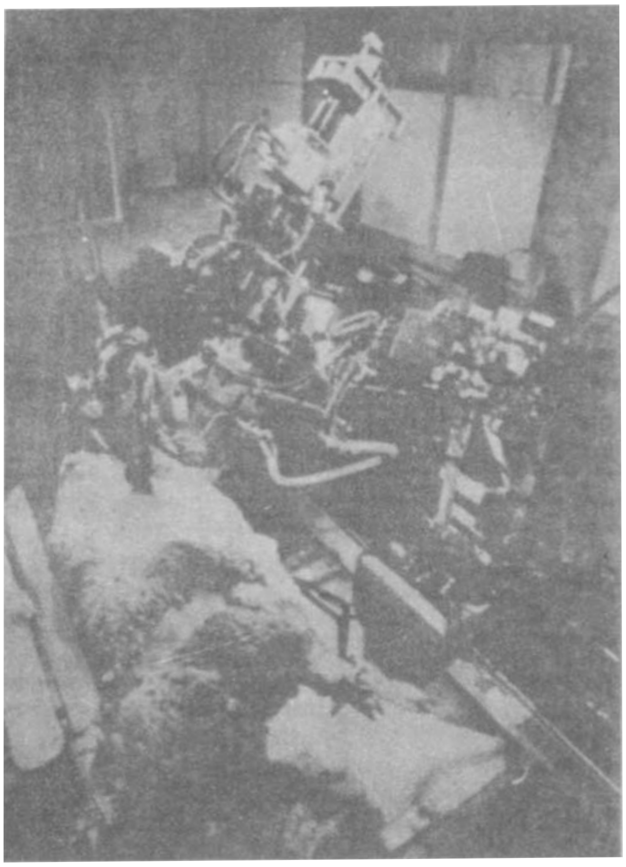


Fig. 20.5 *Research on sheep-shearing robots is of interest in New Zealand and Australia. (Photo courtesy: Cary Wolinsky/Stock Boston)*

20.4.10 Household Robots

The prospect of a domestic robot in nearly every home provides a tremendous market potential and a tremendous commercial opportunity for the company that captures that market. Chores that might be accomplished by a household robot include dishwashing, rug vacuuming, making beds, furniture dusting, window washing, and

in the design of a construction robot (discussed in Sec. 20.3) also arise in the case of a household robot. The robot would need to be capable of mobility and obstacle

high-level oral commands (e.g., 'wash the dishes,' 'clean the rug,' 'make the beds,' etc.) and to reduce those commands to a detailed set of actions that must be carried out one by one in order to perform the given chore. In addition to its regular duties during the day, the household robot could be on duty at night, per forming monitoring functions with its sensors to make sure the house is secure against burglars, and to act

robots.' The cost of a highly functional household robot would be limited not by

Fig. 20.6

the intelligence requirements for the machine, but by its mechanical and sensor requirements. It is anticipated that advances in microprocessor technology will permit powerful computers (relative to today's standards) to be mounted on-board future robots and that the cost of these computers will be a minor portion of the total robot price. The development costs for software used in the household robot will be spread over many thousands (perhaps millions) of units, thus allowing the software portion of the price to be minimized. It is probable that various software packages will be commercially available for the household robot, just as different software is available for today's personal computers. New software introduced to the market would permit an existing household robot to be upgraded every year or so, allowing it to accomplish increasingly complex tasks.

The mechanical structure of the robot and its sensor systems would probably establish a lower limit on the price of a household robot. Even if manufacturing costs have been significantly reduced by the economies of mass production, the material costs of a robot large enough to perform useful household chores would be substantial. Albus has estimated that the price of such a robot would be in the range $4000 to $6000 (in 1980 dollars). The choice for an average household might be between buying a new car or a new household robot. And if the decision is based on how much of the family's time is affected by each of the two alternatives, it would probably turn out that the robot would have a bigger impact on the family's lifestyle.

Specially designed robots might be capable of performing lawn and garden work. The possibilities include mowing the lawn, spreading fertilizer and other chemicals, grass trimming, and clipping a hedge or bush. These robots could be powered by gasoline engines, similar to today's tractor-type mowers. A simple instruction, such as 'mow the lawn' would engage the robot to accomplish several hours work, requiring it to reduce that macro-level command into a complex sequence of travel motions to finish the job.

The use of domestic robots in hotels for cleaning and making over the guest rooms would add an extra dimension to the market for this class of robot. These machines could be kept busy a high proportion of the day and their worth to the hotel would be measured in terms of the work they could accomplish compared to a human maid employed by the hotel. The investment criteria for the hotel would be similar to that used for current industrial robots in manufacturing applications.

20.4.11 Rehabilitation Robots

With a rapidly ageing population and a decreasing birth rate in most countries health care for the elderly is becoming an important issue. This recent trend has led to a new area called human–centred robotics. The main focus of this area are:

1. Exoskeletons for human support or augmenting human physical capabilities.
2. Personal assistants for the elderly.
3. Helpers in activities of daily living.
4. Nurses and medical assistants.
5. Robots for the disabled or for physical therapy.

Many elderly persons living alone need constant supervision for medical emergencies. Several robots have been developed whose main function is to live with the elderly persons and ‘keep an eye’ on them in case they require urgent medical attention, e.g., in case an elderly person falls down the robot is programmed to immediately call for medical emergency. Several of these robots are designed like toys or pets and don’t resemble industrial manipulators in any way.

Helper robots can perform several of the house hold activities autonomously. Today, we have a large member of different robots performing tasks such as vacuum cleaning, dishwashing, and some of these can also assist humans by operating a micro wave oven etc. In hospitals and old age homes there is a large shortage of trained persons such as nurses or therapists. The elderly persons sometimes require assistance for daily activities such as bathing, eating etc. Different types of robots have been developed for helping the elderly perform their daily tasks. With the shortage of medical attendants like nurses there is also a large shortage of physical therapists. Therapists robots have been developed that can help a disabled person perform exercise such as moving their hands or large. In the case of gait rehabilitation, such robots help a person to learn how to walk, etc.

20.5 SUMMARY

In the preceding chapters of the book, we have discussed the technology, programming, and applications of robots: how they work, how to work them, and what work they can do. In the present chapter, we have examined the prospects for smarter, mobile robots in the future to manufacture products more cheaply, build bridges more safely, explore outer space, search under the sea, help doctors in patient care, and assist homemakers with domestic chores.

A substantial opportunity exists in the technology of robotics to relieve people from the boring, repetitive, hazardous, and unpleasant work in all forms of human labor. There is a social value as well as a commercial value in pursuing this opportunity. The commercial value of robotics is obvious. Properly applied, robots can accomplish routine, undesirable work better than humans and at lower cost. As the technology advances, and more people learn how to use robots, the robotics market will grow at a rate that will approach the growth of the computer market over the past 30 years. One might even consider robotics to be a mechanical extension of computer technology.

The social value of robotics is that these wonderfully subservient machines will permit humans more time to do work that is more challenging, creative, conceptual, constructive, and cooperative than at present. There is every reason to believe that the automation of work through robotics will lead to substantial increases in productivity, and that these productivity increases year by year will permit humans to engage in activities that are more cultural and recreational. Not only will robotics improve our standard of living; it will also improve our standard of life.

In the first chapter of the book, we mentioned Carel Capek’s science fiction play about sinister robots which ultimately brought great harm to humans. It seems appropriate in this final chapter to express our belief that the field of robotics, in contrast to Capek’s play, offers the promise of great commercial and social benefit to humankind.

roblems

20.1 Robots still cannot replace humans in several industrial applications. List the applications where a robot still cannot be applied and why?

20.2

applications? What do you think are future application of robots?

20.3 Look around your institute laboratory as in other areas such as shopping malls, hospitals and see if you can see potential areas where robots can be used or are being used.

References

1. J. S. Albus,
Chap. 11.

,

3. L. Conigliari, 'Trends in the Robotics Industry,' *Technical Paper MS82-122*,

4. J. F. Engelberger, *Robotics in Practice*

5. J. F. Engelberger, 'The Household Robot: by 1993,' *Decade of Robotics*, IFS Publications, Bedford, England, 1983, pp. 12–13.

Decade of Robotics, IFS Publications, Bedford, England, 1983, pp. 102–103.

7. W. B. Gevarter, 'Robotics: An Overview.' *Computers in Mechanical Engineering*. August 1982, pp, 43–49.
8. E. Heer, 'Robots in Space,' *Decade of Robotics*, IFS Publications. Bedford, England, 1983, pp. 104–107.
9. V. D. Hunt, . Industrial Press Inc., New York, 1983, Chap. 14.
10. Japan Industrial Robot Association,
Tomorrow, Fuji Corporation, Tokyo, Japan. 1982.

Robotics, Washington, D.C., February 1982.

12. D. N. Smith and R. C. Wilson, Industrial Robots,
and Technology

Decade of Robotics. IFS Publications. Bedford, England, 1983, pp. 100–101.

Index