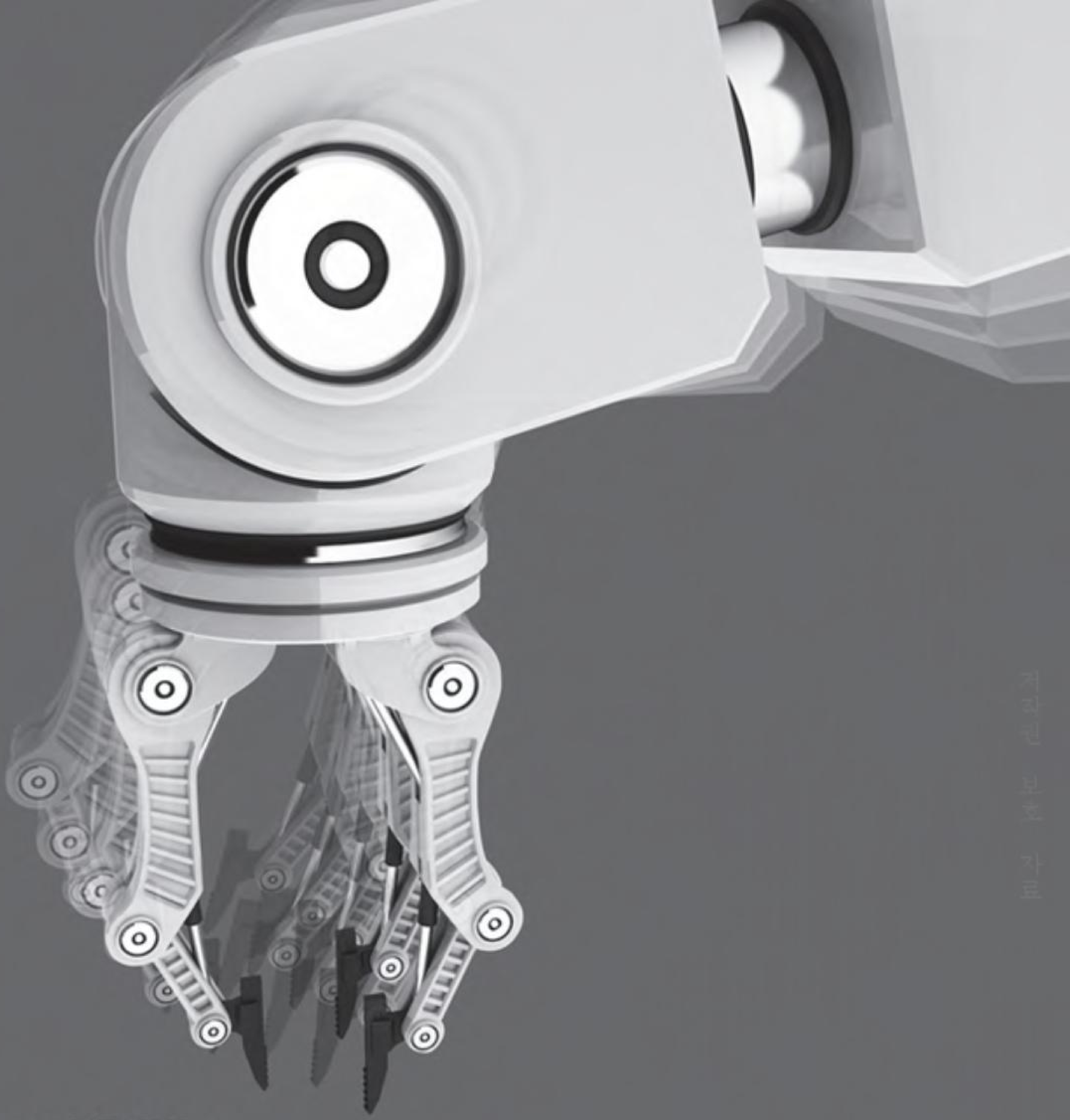


Second Edition

Introduction to ROBOTICS

Analysis, Control, Applications

SAEED B. NIKU

VP & Publisher	Don Fowley
Executive Editor	Linda Ratts
Editorial Assistant	Renata Marchione
Marketing Manager	Christopher Ruel
Media Editor	Lauren Sapira
Production Manager	Janis Soo
Assistant Production Editor	Yee Lyn Song
Cover Designer	RDC Publishing Group Sdn Bhd
Cover Photo Credit	© Alexey Dudoladov/iStockphoto

This book was set in 11/12 Bembo by Thomson Digital, and printed and bound by Courier Westford. The cover was printed by Courier Westford.

This book is printed on acid free paper. ♾

This book was previously published by: Pearson Education, Inc.

Library of Congress Cataloging-in-Publication Data

Niku, Saeed B. (Saeed Benjamin)
An introduction to robotics : analysis, control, applications / Saeed Niku.—2nd ed.
p. cm.
Includes index.
ISBN 978-0-470-60446-5 (hardback)
1. Robotics. I. Title. II. Title: Robotics analysis, control, applications.
TJ211.N547 2010
629.8'92—dc22

2010024723

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

VP & Publisher	Don Fowley
Executive Editor	Linda Ratts
Editorial Assistant	Renata Marchione
Marketing Manager	Christopher Ruel
Media Editor	Lauren Sapira
Production Manager	Janis Soo
Assistant Production Editor	Yee Lyn Song
Cover Designer	RDC Publishing Group Sdn Bhd
Cover Photo Credit	© Alexey Dudoladov/iStockphoto

This book was set in 11/12 Bembo by Thomson Digital, and printed and bound by Courier Westford. The cover was printed by Courier Westford.

This book is printed on acid free paper. ♾

This book was previously published by: Pearson Education, Inc.

Library of Congress Cataloging-in-Publication Data

Niku, Saeed B. (Saeed Benjamin)

An introduction to robotics : analysis, control, applications / Saeed Niku.—2nd ed.

p. cm.

Includes index.

ISBN 978-0-470-60446-5 (hardback)

1. Robotics. I. Title. II. Title: Robotics analysis, control, applications.

TJ211.N547 2010

629.8'92—dc22

2010024723

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Brief Contents

Contents

Brief Contents

CHAPTER 1

Fundamentals

1.1 Introduction

Robotics, in different forms, has been on humans' minds since the time we could build things. You may have seen machines that artisans made that try to mimic human motions and behavior. Examples include the statues in Venice's San Marcos clock tower that hit the clock on the hour and figurines that tell a story in the fifteenth-century Astronomical Clock on the side of the Old Town Hall Tower in Prague (Figure 1.1). Toys, from simple types to sophisticated machines with repeating movements, are other examples. In Hollywood, movies have even portrayed robots and humanoids as superior to humans.

Although in principle humanoids are robots and are designed and governed by the same basics, in this book, we will primarily study industrial manipulator type robots. This book covers some basic introductory material that familiarizes you with the subject; it presents an analysis of the mechanics of robots including kinematics, dynamics, and trajectory planning; and it discusses the elements used in robots and in robotics, such as actuators, sensors, vision systems, and so on. Robot rovers are no different, although they usually have fewer degrees of freedom and generally move in a plane. Exoskeletal and humanoid robots, walking machines, and robots that mimic animals and insects have many degrees of freedom (DOF) and may possess unique capabilities. However, the same principles we learn about manipulators apply to robot rovers too, whether kinematics, differential motions, dynamics, or control.

Robots are very powerful elements of today's industry. They are capable of performing many different tasks and operations, are accurate, and do not require common safety and comfort elements humans need. However, it takes much effort and many resources to make a robot function properly. Most companies of the mid-1980s that made robots are gone, and with few exceptions, only companies that make real industrial robots have remained in the market (such as Adept, Staubli, Fanuc, Kuka, Epson, Motoman, Denso, Fuji, and IS Robotics as well as specialty robotic companies such as Mako Surgical Corp. and Intuitive Surgical). Early industrialist predictions about the possible number of robots

1.6 Advantages and Disadvantages of Robots

- Robotics and automation can, in many situations, increase productivity, safety, efficiency, quality, and consistency of products.
- Robots can work in hazardous environments such as radiation, darkness, hot and cold, ocean bottoms, space, and so on without the need for life support, comfort, or concern for safety.
- Robots need no environmental comfort like lighting, air conditioning, ventilation, and noise protection.
- Robots work continuously without tiring or fatigue or boredom. They do not get mad, do not have hangovers, and need no medical insurance or vacation.
- Robots have repeatable precision at all times unless something happens to them or unless they wear out.
- Robots can be much more accurate than humans. Typical linear accuracies are a few ten-thousandths of an inch. New wafer-handling robots have micro-inch accuracies.
- Robots and their accessories and sensors can have capabilities beyond those of humans.
- Robots can process multiple stimuli or tasks simultaneously. Humans can only process one active stimulus.
- Robots replace human workers, causing economic hardship, worker dissatisfaction and resentment, and the need for retraining the replaced workforce.
- Robots lack capability to respond in emergencies, unless the situation is predicted and the response is included in the system. Safety measures are needed to ensure that they do not injure operators and other machines that are working with them.[3] This includes:
  - Inappropriate or wrong responses
  - Lack of decision-making power
  - Loss of power
  - Damage to the robot and other devices
  - Injuries to humans
- Robots, although superior in certain senses, have limited capabilities in:
  - Cognition, creativity, decision-making, and understanding
  - Degrees of freedom and dexterity
  - Sensors and vision systems
  - Real-time response
- Robots are costly due to:
  - Initial cost of equipment and installation
  - Need for peripherals
  - Need for training
  - Need for programming

1.7 Robot Components

A robot, as a system, consists of the following elements, which are integrated together to form a whole:

Manipulator or the rover: This is the main body of the robot which consists of the links, the joints, and other structural elements of the robot. Without other elements, the manipulator alone is not a robot (Figure 1.3).

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

mounted on a conveyor belt or is otherwise mobile, the location of the base of the robot relative to the belt or other reference frame is known. Since this location does not need to be defined by the controller, the remaining number of degrees of freedom is still six, and consequently, unique. So long as the location of the base of the robot on the belt or the location of the mobile platform is known (or selected by the user), there is no need to find it by solving the set of equations of robot motions, and as a result, the system can be solved.

Can you determine how many degrees of freedom the human arm has? This should exclude the hand (palm and the fingers), but should include the wrist. Before you go on, try to see if you can determine it.

The human arm has three joint clusters: the shoulder, the elbow, and the wrist. The shoulder has 3 degrees of freedom, since the upper arm (humerus) can rotate in the sagittal plane, which is parallel to the mid-plane of the body; the coronal plane (a plane from shoulder to shoulder); and about the humerus (please verify this by rotating your arm about the three different axes). The elbow has just 1 degree of freedom; it can only flex and extend about the elbow joint. The wrist also has 3 degrees of freedom. It can abduct and adduct, flex and extend, and, since the radius bone can roll over the ulna, it can rotate longitudinally (pronate and supinate). Consequently, the human arm has a total of 7 degrees of freedom, even if the ranges of some movements are small. Since a 7-DOF system does not have a unique solution, how do you think we can use our arms?

Please note that the end effector of the robot is never considered as one of the degrees of freedom. All robots have this additional capability, which may appear to be similar to a degree of freedom. However, none of the movements in the end effector are counted toward the robot's degrees of freedom.

There are cases where a joint may have the ability to move, but its movement is not fully controlled. For example, consider a linear joint actuated by a pneumatic cylinder, where the arm is fully extended or fully retracted, but no controlled position can be achieved between the two extremes. In this case, the convention is to assign only a ½-degree of freedom to the joint. This means that the joint can only be at specified locations within its limits of movement. Another possibility for a ½-degree of freedom is to assign only particular values to the joint. For example, suppose a joint is made to be only at 0, 30, 60, and 90 degrees. Then, as before, the joint is limited to only a few possibilities, and therefore, has a partial degree of freedom.

Many industrial robots possess fewer than 6 degrees of freedom. Robots with 3.5, 4, and 5 degrees of freedom are in fact very common. So long as there is no need for the additional degrees of freedom, these robots perform very well. For example, suppose you intend to insert electronic components into a circuit board. The circuit board is always laid flat on a known work surface, and consequently, its height (z value) relative to the base of the robot is known. Therefore, there is only a need for 2 degrees of freedom along the x- and y-axes to specify any location on the board for insertion. Additionally, suppose that the components are to be inserted in any direction on the board, but the board is always flat. In that case, there is a need for 1 degree of freedom to rotate about the vertical axis (z) in order to orientate the component above the surface. Since there is also need for a ½-degree of freedom to fully extend the end effector to insert the part or to fully retract it to lift the robot before moving, only 3.5 degrees of freedom are needed: two to move over the board, one to rotate the component, and 1/2 to insert or retract. Insertion robots are very common and are extensively used in electronic industry. Their advantage is that

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

positive n-axis of the local Tool frame will move the hand along the n-axis of the Tool frame. If the arm were pointed elsewhere, the same motion along the local n-axis of the Tool frame would be completely different from the first motion. The same $+n$-axis movement would be upward if the n-axis were pointed upward, and it would be downward if the n-axis were pointed downward. As a result, the Tool reference frame is a moving frame that changes continuously as the robot moves; therefore, the ensuing motions relative to it are also different depending on where the arm is and what direction the tool frame has. All joints of the robot must move simultaneously to create coordinated motions about the Tool frame. The Tool reference frame is an extremely useful frame in robotic programming where the robot is to approach and depart from other objects or to assemble parts.

1.12 Programming Modes

Robots may be programmed in a number of different modes, depending on the robot and how sophisticated it is. The following programming modes are common:

Physical Set-up: In this mode, an operator sets up switches and hard stops that control the motions of the robot. This mode is usually used along with other devices such as Programmable Logic Controllers (PLC).

Lead Through or Teach Mode: In this mode, the robot's joints are moved with a teach pendant. When the desired location and orientation is achieved, the location is entered (taught) into the controller. During playback, the controller moves the joints to the same locations and orientations. This mode is usually point-to-point; as such, the motion between points is not specified or controlled. Only the points that are taught are guaranteed to reach.

Continuous Walk-Through Mode: In this mode, all robot joints are moved simultaneously, while the motion is continuously sampled and recorded by the controller. During playback, the exact motion that was recorded is executed. The motions are taught by an operator, either through a model, by physically moving the end-effector, or by "wearing" the robot arm and moving it through its workspace. Painting robots, for example, may be programmed by skilled painters through this mode.

Software Mode: In this mode of programming the robot, a program is written offline or online and is executed by the controller to control the motions. The programming mode is the most sophisticated and versatile mode and can include sensory information, conditional statements (such as if . . . then statements), and branching. However, it requires a working knowledge of the programming syntax of the robot before any program is written. Most industrial robots can be programmed in more than one mode.

1.13 Robot Characteristics

The following definitions are used to characterize robot specifications:

Payload: Payload is the weight a robot can carry and still remain within its other specifications. As an example, a robot's maximum load capacity may be much larger than its specified payload, but at these levels, it may become less accurate, may not follow its

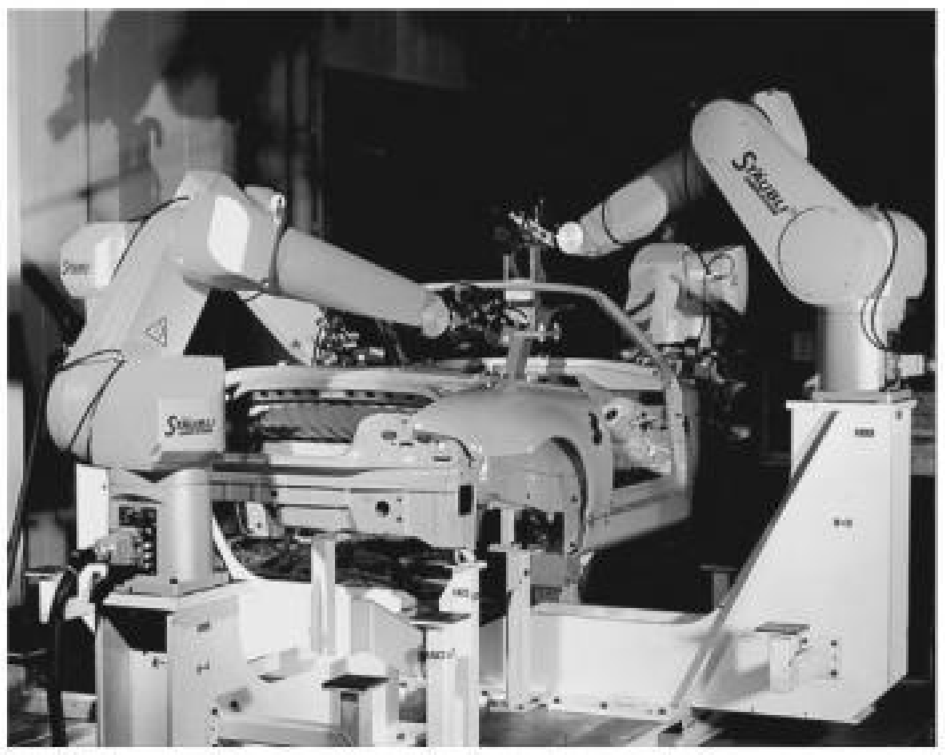


Figure 1.13 Staubli RX FRAMS (Flexible Robotic Absolute Measuring System) robots in a BMW manufacturing facility. (Reprinted with permission from Staubli Robotics.)

application, a robot would search for and find the location of each rivet, detect and mark the rivets with fatigue cracks, would drill them out, and move on. The technicians would insert and install new rivets. Robots have also been extensively used for circuit board and chip inspection. In most applications like this, including part identification, the characteristics of the part (such as the circuit diagram of a board, the nameplate of a part, and so on) are stored in the system in a data library. The system uses this information to match the part with the stored data. Based on the result of the inspection, the part is either accepted or rejected.

Sampling with robots is used in the agriculture industry as well as in many other industries. Sampling can be similar to pick and place and inspection, except that it is performed only on a certain number of products.

Assembly tasks usually involve many operations. For example, the parts must be located and identified, they must be carried in a particular order with many obstacles around the set-up, they must be fit together, and then assembled. Many of the fitting and assembling tasks are complicated and may require pushing, turning, bending, wiggling, pressing, snapping the tabs to connect the parts, and other operations. Slight variations in parts and their dimensions due to larger tolerances also complicate the process since the robot has to know the difference between variations in parts and wrong parts.

Manufacturing by robots may include many different operations such as material removal (Figure 1.14), drilling, de-burring, laying glue, cutting, and so on. It also includes insertion of parts such as electronic components into circuit boards, installation of boards into electronic devices, and other similar operations. Insertion robots are very common and are extensively used in the electronic industry.

Medical applications are also becoming increasingly common. As an example, Curexo Technology Corporation's Robodoc® was designed to assist a surgeon in total joint replacement operations. Since many of the functions performed during this procedure—such as cutting the head of the bone, drilling a hole in the bone's body, reaming the hole for precise dimension, and installation of the manufactured implant joint—can be performed with better precision by a robot, the mechanical parts of the operation are assigned to the robot. This is also important because the orientation and the shape of the bone can be determined by a CAT scan and downloaded to the robot controller, where it is used to direct the motions of the robot for a best fit with the implant.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

components, and telepresence capability.[26] And finally, in another application, a telerobot was used for microsurgery.[27] In this case, the location of the telerobot is of secondary concern. The primary intention is to have the telerobot repeat the surgeon's hand movements at a smaller scale for reduced tremor during microsurgery.

1.17 Other Robots and Applications

Since the first edition of this book was published, new robots and issues have appeared. Such is the nature of this active subject. Therefore, you should expect that there will be applications and robots that are not included in this edition either. However, the following are just a sample of some systems that show a trend and future possibilities.

Roomba™, a robot vacuum cleaner, commercially available for years, autonomously and randomly moves throughout an area and vacuums the dust. It also finds its own docking station to recharge. All its intelligence is based on a few simple rules: randomly move around, turn left or right when hitting an obstacle, back up and turn around when in a corner, and find the docking station.[28]

Robots such as Honda's *ASIMO*, Bluebotic's *Gilbert*, Nestle's *Nesbot*, Anybots's *Monty*, and many others are intelligent humanoid robots with humanlike features and behavior. ASIMO walks, runs, goes up and down staircases, and interacts with people. Nesbot brings coffee to workers who have ordered it online.[28] Monty loads a dishwasher and does other chores, while Robomower mows your lawn while you read.[29] Figure 1.19 shows a picture of Nao robot.[30] Like others, Nao is a fully programmable robot that can behave autonomously–it communicates with humans and it walks, dances, and performs tasks.

A number of different robots have also been designed and used for emergency services during natural and human-caused disasters. These robots, equipped with special sensors, are capable of looking for live humans and animals buried under rubble and reporting their locations to rescuers. Similar robots are also used for diffusing bombs and other explosive devices. SDA10 dual-arm robot by Motoman, Inc. (Figure 1.20), has 15 axes of motion. The two arms can move independently or in a coordinated manner. It can transfer a part from one gripper to the other without the need to set it down.

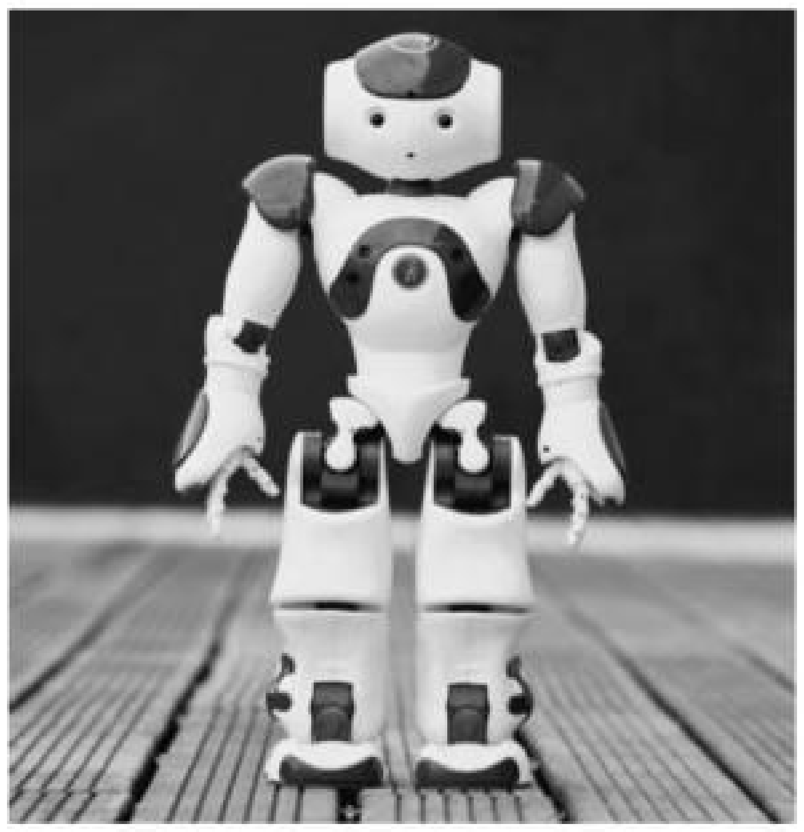


Figure 1.19 Nao humanoid robot. (Reprinted with permission from Aldebaran Robotics (Picture by C. De Torquat).)

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

directive that requires the military to replace up to one-third of its logistics vehicles with these "robots" by the mid 2010s.[46]

Animatronics refers to the design and development of systems used in animated robotic figures and machines that look and behave like humans and animals. Examples include animatronic lips, eyes, and hands.[47,48] As more sophisticated animatronic components become available, the action figures they replace become increasingly real.

Another area somewhat related to robotics and its applications is Micro-Electro-Mechanical-Systems (MEMS). These are micro-level devices designed to perform functions within a system that include medical, mechanical, electrical, and physical tasks. As an example, a micro-level robotic device may be sent through major veins to the heart for exploratory or surgical functions, a MEMS sensor may be used to measure the levels of various elements in blood, or a MEMS actuator may be used to deploy automobile airbags in a collision.

1.18 Social Issues

We must always remember the social consequences of using robots. Although there are many applications of robots where they are used because no workers can do the same job, there are many other applications in which a robot replaces a human worker. The worker who is replaced by a robot will lose his or her income. If the trend continues without consideration, it is conceivable that most products will be made by robots, without the need for any human workers. The result will be fewer workers with jobs who have the money to buy the products the robots make. Of importance is the issue of social problems that arise as increasingly more workers are out of jobs as well as its social and economic consequences. One of the important points of negotiations between the automobile manufacturers and the United Auto Workers (UAW) is how many human jobs may be replaced by robots, and at what rate.

Although no solution is presented in this book, many references are available for further study of the problem.[49,50] However, as engineers who strive to make better products at lower costs and who may consider using robots to replace human workers, we must always remember the consequences of this choice. Our academic and professional interest in robotics must always be intertwined with its social and economic considerations.

Summary

Many people who are interested in robotics have background information about robots and may even have interacted with robots too. However, it is necessary that certain ideas are understood by everyone. In this chapter, we discussed some fundamental ideas about robotics that enable us to better understand what they are for, how they can be used, and what they can do. Robots can be used for many purposes, including industrial applications, entertainment, and other specific and unique applications such as in space and underwater exploration and in hazardous environments. Obviously, as time goes by, robots will be used for other unique applications. The remainder of this book will discuss the kinematics and kinetics of robots, their components such as actuators, sensors and vision systems, and robot applications.

References

1. Capek, Karel, "Rossum's Universal Robots," translated by Paul Selver, Doubleday, NY, 1923.
2. Valenti, Michael, "A Robot Is Born," *Mechanical Engineering*, June 1996, pp. 50–57.
3. "Robot Safety," Bonney, M. C., Y. F. Yong, Editors, IFS Publications Ltd., UK, 1985.
4. Stein, David, Gregory S. Chirikjian, "Experiments in the Commutation and Motion Planning of a Spherical Stepper Motor," Proceedings of DETC'00, ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Baltimore, Maryland, September 2000, pp. 1–7.
5. Wiitala, Jared M., B. J., Rister, J. P. Schmiedler, "A More Flexible Robotic Wrist," *Mechanical Engineering*, July 1997, pp. 78–80.
6. W. A. Gruver, B. I. Soroka, J. J. Craig, and T. L. Turner, "Industrial Robot Programming Languages: A Comparative Evaluation." *IEEE Transactions on Systems, Man, and Cybernetics* SMC–14(4), July/August 1984.
7. Bonner, Susan, K. G. Shin, "A Comprehensive Study of Robot Languages," *IEEE Computer*, December 1982, pp. 82–96.
8. Kusiak, Andrew, "Programming, Off-Line Languages,"International Encyclopedia of Robotics: Applications and Automation, Richard C. Dorf, Editor, John Wiley and Sons, NY, 1988, pp. 1235–1250.
9. Gruver, William, B. I. Soroka, "Programming, High Level Languages," International Encyclopedia of Robotics: Applications and Automation, Richard C. Dorf, Editor, John Wiley and Sons, NY, 1988, pp. 1203–1234.
10. VAL-II Programming Manual, Version 4, Unimation, Inc., Pittsburgh, 1988.
11. McGuinn, Jack, Senior Editor, "These Bots are Cutting-Edge," *Power Transmission Engineering*, October 2008, pp. 32–36.
12. Salisbury, Kenneth, Jr., "The Heart of Microsurgery," *Mechanical Engineering*, December 1998, pp. 46–51.
13. "Stanford Rehabilitation Center," Stanford University, California.
14. Garcia, Mario, Saeed Niku,"Finger-Spelling Hand," California Polytechnic State University masters thesis, San Luis Obispo, 2009.
15. Leary, Warren, "Robot Named Dante to Explore Inferno of Antarctic Volcano," *The New York Times*, December 8, 1992, p. B7.
16. "Ultrasonic Detector and Identifier of Land Mines," *NASA Motion Control Tech Briefs*, 2001, p. 8b.
17. Chalmers, Peggy, "Lobster Special," *Mechanical Engineering*, September 2000, pp. 82–84.
18. "Snakelike Robots Would Maneuver in Tight Spaces," *NASA Tech Briefs*, August 1998, pp. 36–37.
19. "Lobster Special," *Mechanical Engineering*, September 2000, pp. 82–84.
20. "Robot Population Explosion," *Mechanical Engineering*, February 2009, p. 64.
21. http://www.jpl.nasa.gov/pictures/
22. Wernli, Robert L., "Robotics Undersea," *Mechanical Engineering*, August 1982, pp. 24–31.
23. Asker, James, "Canada Gives Station Partners a Hand—And an Arm," *Aviation Week & Space Technology*, December 1997, pp. 71–73.
24. Puttre, Michael, "Space-Age Robots Come Down to Earth," *Mechanical Engineering*, January 1995, pp. 88–89.
25. Trovato, Stephen A., "Robot Hunts Sludge and Hoses It Away," *Mechanical Engineering*, May 1988, pp. 66–69.
26. "The Future of Robotics Is Now," *NASA Tech Briefs*, October 2002, p. 27.
27. "Telerobot Control for Microsurgery," *NASA Tech Briefs*, October 1997, p. 46.
28. "From Simple Rules, Complex Behavior," *Mechanical Engineering*, July 2009, pp. 22–27.
29. Drummond, Mike, "Rise of the Machines," *Inventors Digest*, February 2008, pp. 16–23.
30. http://www.aldebaran-robotics.com/eng/PressFiles.php
31. www.berkeleybionics.com
32. "Lightweight Exoskeleton with Controllable Actuators," *NASA Tech Briefs*, October 2004, pp. 54–55.

1.2. Draw the approximate workspace for the following robot. Assume the dimensions of the base and other parts of the structure of the robot are as shown.

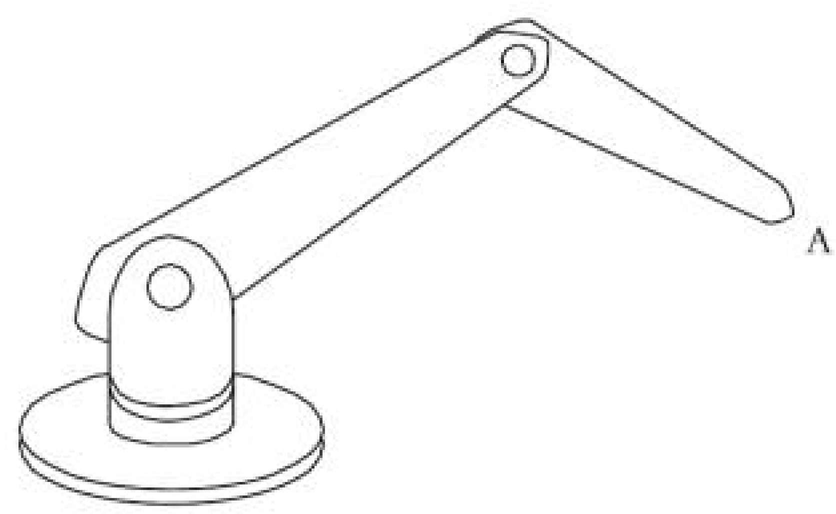


Figure P.1.2

1.3. Draw the approximate workspace for the following robot. Assume the dimensions of the base and other parts of the structure of the robot are as shown.

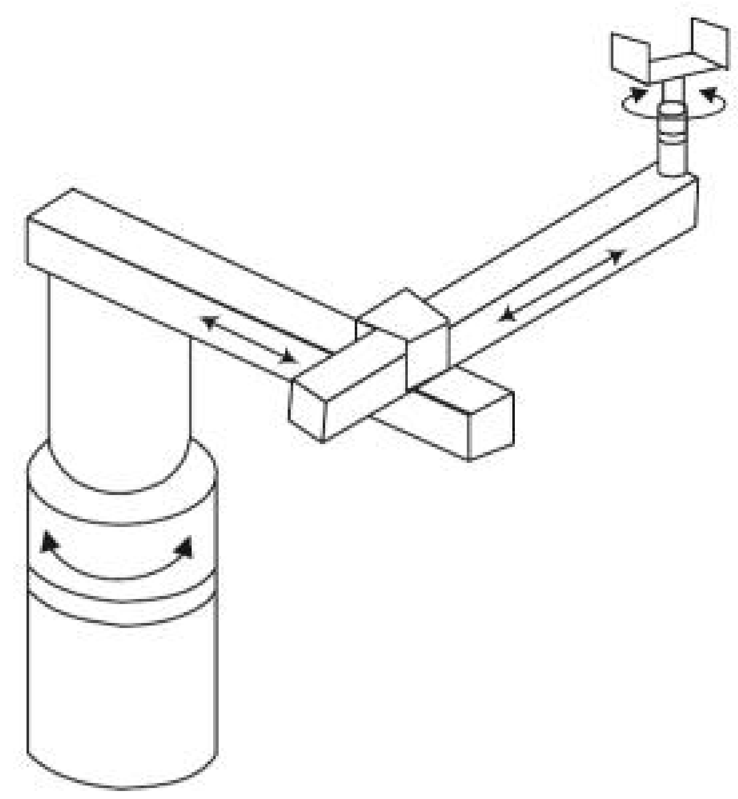


Figure P.1.3

CHAPTER 1

Fundamentals

1.1 Introduction

Robotics, in different forms, has been on humans' minds since the time we could build things. You may have seen machines that artisans made that try to mimic human motions and behavior. Examples include the statues in Venice's San Marcos clock tower that hit the clock on the hour and figurines that tell a story in the fifteenth-century Astronomical Clock on the side of the Old Town Hall Tower in Prague (Figure 1.1). Toys, from simple types to sophisticated machines with repeating movements, are other examples. In Hollywood, movies have even portrayed robots and humanoids as superior to humans.

Although in principle humanoids are robots and are designed and governed by the same basics, in this book, we will primarily study industrial manipulator type robots. This book covers some basic introductory material that familiarizes you with the subject; it presents an analysis of the mechanics of robots including kinematics, dynamics, and trajectory planning; and it discusses the elements used in robots and in robotics, such as actuators, sensors, vision systems, and so on. Robot rovers are no different, although they usually have fewer degrees of freedom and generally move in a plane. Exoskeletal and humanoid robots, walking machines, and robots that mimic animals and insects have many degrees of freedom (DOF) and may possess unique capabilities. However, the same principles we learn about manipulators apply to robot rovers too, whether kinematics, differential motions, dynamics, or control.

Robots are very powerful elements of today's industry. They are capable of performing many different tasks and operations, are accurate, and do not require common safety and comfort elements humans need. However, it takes much effort and many resources to make a robot function properly. Most companies of the mid-1980s that made robots are gone, and with few exceptions, only companies that make real industrial robots have remained in the market (such as Adept, Staubli, Fanuc, Kuka, Epson, Motoman, Denso, Fuji, and IS Robotics as well as specialty robotic companies such as Mako Surgical Corp. and Intuitive Surgical). Early industrialist predictions about the possible number of robots

Figure 1.1 Centuries-old figurines and statues that mimic human motions.

in industry never materialized because high expectations could not be satisfied with the present robots. As a result, although there are many thousands of robots in industry working tirelessly and satisfactorily for the intended jobs, robots have not overwhelmingly replaced workers. They are used where they are useful. Like humans, robots can do certain things, but not others. As long as they are designed properly for the intended purposes, they are very useful and continue to be used.

The subject of robotics covers many different areas. Robots alone are hardly ever useful. They are used together with other devices, peripherals, and other manufacturing machines. They are generally integrated into a system, which as a whole, is designed to perform a task or do an operation. In this book, we will refer to some of these other devices and systems used with robots.

1.2 What Is a Robot?

If you compare a conventional robot manipulator with a crane attached to, say, a utility or towing vehicle, you will notice that the robot manipulator is very similar to the crane. Both possess a number of links attached serially to each other with joints, where each joint can be moved by some type of actuator. In both systems, the ''hand'' of the manipulator can be moved in space and placed in any desired location within the workspace of the system. Each one can carry a certain load and is controlled by a central controller that controls the actuators. However, one is called a robot and one is called a manipulator (or, in this case, a crane). Similarly, material handling manipulators that move heavy objects in manufacturing plants look just like robots, but they are not robots. The fundamental difference between the two is that the crane and the manipulator are controlled by a human who operates and controls the actuators, whereas the robot manipulator is controlled by a computer that runs

- Class 2: *Fixed Sequence Robot*: a device that performs the successive stages of a task according to a predetermined, unchanging method, which is hard to modify
- Class 3: *Variable Sequence Robot*: same as in class 2, but easy to modify
- Class 4: *Playback Robot*: a human operator performs the task manually by leading the robot, which records the motions for later playback; the robot repeats the same motions according to the recorded information
- Class 5: *Numerical Control Robot*: the operator supplies the robot with a movement program rather than teaching it the task manually
- Class 6: *Intelligent Robot*: a robot with the means to understand its environment and the ability to successfully complete a task despite changes in the surrounding conditions under which it is to be performed

The Robotics Institute of America (RIA) only considers classes 3–6 of the above as robots. The Association Francaise de Robotique (AFR) has the following classification:

- Type A: handling devices with manual control to telerobotics
- Type B: automatic handling devices with predetermined cycles
- Type C: programmable, servo controlled robots with continuous or point-to-point trajectories
- Type D: same as C but with capability to acquire information from its environment

1.4 What Is Robotics?

Robotics is the art, knowledge base, and the know-how of designing, applying, and using robots in human endeavors. Robotic systems consist of not just robots, but also other devices and systems used together with the robots. Robots may be used in manufacturing environments, in underwater and space exploration, for aiding the disabled, or even for fun. In any capacity, robots can be useful, but they need to be programmed and controlled. Robotics is an interdisciplinary subject that benefits from mechanical engineering, electrical and electronic engineering, computer science, cognitive sciences, biology, and many other disciplines.

1.5 History of Robotics

Disregarding the early machines that were made to mimic humans and their actions and concentrating on the recent history, one can see a close relationship between the state of industry, the revolution in numeric and computer control of machinery, space exploration, and the vivid imagination of creative people. Starting with Karel Capek and his book, *Rossum's Universal Robots*,[1] and later, movies like *Flash Gordon, Metropolis, Lost in Space, The Day The Earth Stood Still*, and *The Forbidden Planet*,[2] the stage was set for a machine to be built to do a human's job (and, of course, R2D2, C3PO, Robocop, and others continued the trend).

Capek dreamed of a scenario where a bioprocess could create human-like machines, devoid of emotions and souls, who were strong, obeyed their masters, and could be produced quickly and cheaply. Soon, the market grew tremendously when all major countries wanted to ''equip'' their armies with hundreds of thousands of slave robotic soldiers, who would fight with dedication, but whose death would not matter. Eventually, the robots decided that they were actually superior to the humans, took over the whole world, and killed everyone. In this story, the word ''rabota,'' or worker, was coined and is used even today. After World War II, automatic machines were designed to increase productivity, and machine-tool manufacturers made numerically controlled (NC) machines to enable manufacturers to produce better products. At the same time, multi-degree-of-freedom manipulators were developed for work on nuclear materials. Integration between the NC capability of machine tools and the manipulators created a simple robot. The first robots were controlled by strips of paper with holes, which electric eyes could detect and which controlled the robot's movements. As industry improved, the strip of paper gave way to magnetic tapes, to memory devices, and personal computers. The following is a summary of events that have marked changes in the direction of this industry.

1922	Czech author Karel Capek wrote a story called *Rossum's Universal Robots* and introduced the word rabota (worker).
1946	George Devol developed the magnetic controller, a playback device. Eckert and Mauchley built the ENIAC computer at the University of Pennsylvania.
1952	The first numerically controlled machine was built at MIT.
1954	George Devol developed the first programmable robot.
1955	Denavit and Hartenberg developed homogeneous transformation matrices.
1961	U.S. patent 2,988,237 was issued to George Devol for ''Programmed Article Transfer,'' a basis for Unimate™ robots.
1962	Unimation™ was formed, the first industrial robots appeared, and GM installed its first robot from Unimation™.
1967	Unimate™ introduced MarkII™ robot. The first robot was imported to Japan for paint spraying applications.
1968	An intelligent robot called Shakey was built at the Stanford Research Institute (SRI).
1972	IBM worked on a rectangular coordinate robot for internal use. It eventually developed the IBM 7565 for sale.
1973	Cincinnati Milacron™ introduced T3 model robot which became very popular in industry.
1978	The first PUMA robot was shipped to GM by Unimation™.
1982	GM and Fanuc of Japan signed an agreement to build GMFanuc robots.
1983	Robotics became a very popular subject, both in industry as well as academia. Many programs in the nation started teaching robotic courses.
1983	Unimation™ was sold to Westinghouse Corporation, who subsequently sold it to the Staubli of Switzerland in 1988.
1986	Honda introduced its first humanoid robot called H0. First Asimo was introduced in 2000.
2005	Between January and March, over 5,300 robots were ordered by the North American manufacturing companies at a value of $302 million.

Figure 1.3 A Fanuc M-410iWW palletizing robotic manipulator with its end effector. (Reprinted by permission from Fanuc Robotics, North America, Inc.)

End effector: This part is connected to the last joint (hand) of a manipulator that generally handles objects, makes connections to other machines, or performs the required tasks (Figure 1.3). Robot manufacturers generally do not design or sell end effectors. In most cases, all they supply is a simple gripper. Generally, the hand of a robot has provisions for connecting specialty end effectors specifically designed for a purpose. This is the job of a company's engineers or outside consultants to design and install the end effector on the robot, and to make it work for the given situation. A welding torch, a paint spray gun, a glue laying device, or a parts handler are but a few possibilities. In most cases, the action of the end effector is either controlled by the robot's controller, or the controller communicates with the end effector's controlling device (such as a PLC).

Actuators: Actuators are the "muscles" of the manipulators. The controller sends signals to the actuators, which, in turn, move the robot joints and links. Common types are servomotors, stepper motors, pneumatic actuators, and hydraulic actuators. Other novel actuators are used in specific situations (this will be discussed later in Chapter 7). Actuators are under the control of the controller.

Sensors: Sensors are used to collect information about the internal state of the robot or to communicate with the outside environment. As in humans, the robot controller needs to know the location of each link of the robot in order to know the robot's configuration. When you wake up in the morning, even without opening your eyes, or when it is completely dark, you still know where your arms and legs are. This is because feedback sensors in your central nervous system embedded in muscle tendons send information to the brain. The brain uses this information to determine the length of your muscles and, consequently, the state of your arms, legs, and so on. The same is true for robots, where sensors integrated into the robot send information about each joint or link to the controller that determines the configuration of the robot. Still similar to your major senses of sight, touch, hearing, taste, and speech, robots are equipped with external sensory devices such as a vision system, touch and tactile sensors, speech synthesizer, and the like that enable the robot to communicate with the outside world.

Controller: The controller is rather similar to your cerebellum; although it does not have the power of the brain, it still controls your motions. The controller receives its data from the computer (the brain of the system), controls the motions of the actuators, and coordinates the motions with the sensory feedback information. Suppose that in order for the robot to pick up a part from a bin, it is necessary that its first joint be at 35°. If the joint is not already at this magnitude, the controller will send a signal to the actuator—a current to an electric motor, air to a pneumatic cylinder, or a signal to a hydraulic servo valve—causing it to move. It will then measure the change in the joint angle through the feedback sensor attached to the joint (a potentiometer, an encoder, etc.). When the joint reaches the desired value, the signal is stopped. In more sophisticated robots, the velocity and the force exerted by the robot are also controlled by the controller.

Processor: The processor is the brain of the robot. It calculates the motions of the robot's joints, determines how much and how fast each joint must move to achieve the desired location and speeds, and oversees the coordinated actions of the controller and the sensors. The processor is generally a computer, which works like all other computers, but is dedicated to this purpose. It requires an operating system, programs, peripheral equipment like a monitor, and has the same limitations and capabilities. In some systems, the controller and the processor are integrated together into one unit. In others, they are separate units, and in some, although the controller is provided by the manufacturer, the processor is not; they expect the user to provide his or her processor.

Software: Three groups of software programs are used in a robot. One is the operating system that operates the processor. The second is the robotic software that calculates the necessary motions of each joint based on the kinematic equations of the robot. This information is sent to the controller. This software may be at many different levels, from machine language to sophisticated languages used by modern robots. The third group is the collection of application-oriented routines and programs developed to use the robot or its peripherals for specific tasks such as assembly, machine loading, material handling, and vision routines.

1.8 Robot Degrees of Freedom

As you may remember from your engineering mechanics courses, in order to locate a point in space, one needs to specify three coordinates (such as the x-, y-, z-coordinates along the three Cartesian axes). Three coordinates are necessary and enough to completely define the location of the point. Although different coordinate systems may be used to express this information, they are always necessary. However, neither two nor four will be possible; two is inadequate to locate a point in space, and four is impossible. There is simply too much information. Similarly, if you consider a three-dimensional device that has 3 degrees of freedom within the workspace of the device, you should be able to place the device at any desired location. For example, a gantry (x,y,z) crane can place a ball at any location within its workspace as specified by the operator.

Similarly, to locate a rigid body (a three-dimensional object rather than a point) in space, we need to specify the location of a selected point on it; therefore, it requires three pieces of information to be located as desired. However, although the location of the object is specified, there are infinite possible ways to orientate the object about the selected point. To fully specify the object in space, in addition to the location of a selected point on it, we need to specify the orientation of the object as well. This means that six pieces of information are needed to fully specify the location and orientation of a rigid body. By the same token, there need to be 6 degrees of freedom available to fully place the object in space and orientate it as desired.

For this reason, robots need to have 6 degrees of freedom to freely place and orientate objects within their workspace. A robot that has 6 degrees of freedom can be requested to place objects at any desired location and orientation. If a robot has fewer degrees of freedom, we cannot arbitrarily specify any location and orientation for the robot; it can only go to places and to orientations that the fewer joints allow. To demonstrate this, consider a robot with 3 degrees of freedom, where it can only move along the x-, y-, and z-axes. In this case, no orientation can be specified; all the robot can do is to pick up the part and move it in space parallel to the reference axes. The orientation always remains the same. Now consider another robot with 5 degrees of freedom, capable of rotating about the three axes, but only moving along the x- and y-axes. Although you may specify any orientation desired, the positioning of the part is only possible along the x- and y-, but not z-axes. The same is true for any other robot configurations.

A system with 7 degrees of freedom would not have a unique solution. This means that if a robot has 7 degrees of freedom, there are infinite ways it can position a part and orientate it at the desired location. In order for the controller to know what to do, there must be some additional decision-making routine that allows it to pick only one of the infinite solutions. As an example, we may use an optimization routine to pick the fastest or the shortest path to the desired destination. Then the computer has to check all solutions to find the shortest or fastest response and perform it. Due to this additional requirement, which can take much computing power and time, no 7-degree of freedom robot is used in industry. A similar issue arises when a manipulator robot is mounted on a moving base such as a mobile platform or a conveyor belt (Figure 1.4). In either case, the robot has an additional degree of freedom, which, based on the above discussion, is impossible to control. The robot can be at a desired location and orientation from infinite distinct positions on the conveyor belt or the mobile platform. However, in this case, although there are too many degrees of freedom, the additional degrees of freedom are known and there is no need to solve for them. In other words, generally, when a robot is

Figure 1.4 A Fanuc P-15 robot. (Reprinted with permission from Fanuc Robotics, North America, Inc.)

they are simple to program, less expensive, smaller, and faster. Their disadvantage is that, although they may be programmed to insert components on any size board in any direction, they cannot perform other jobs. They are limited to what 3.5 degrees of freedom can achieve, but they can perform a variety of functions within this design limit.

1.9 Robot Joints

Robots may have different types of joints, such as linear, rotary, sliding, or spherical. Spherical joints are common in many systems but they possess multiple degrees of freedom, and therefore, are difficult to control. Consequently, they are not common in robotics except in research.[4] Most robots have either a linear (prismatic) joint or a rotary (revolute) joint. Prismatic joints are linear; there is no rotation involved. They are either hydraulic or pneumatic cylinders or linear electric actuators. These joints are used in gantry, cylindrical, or spherical robot variations. Revolute joints are rotary, and although hydraulic and pneumatic rotary joints are common, most rotary joints are electrically driven, either by stepper motors or, more commonly, by servomotors.

1.10 Robot Coordinates

Robot configurations generally follow the coordinate frames with which they are defined, as shown in Figure 1.5. Prismatic joints are denoted by P, revolute joints are denoted by R, and spherical joints are denoted by S. Robot configurations are

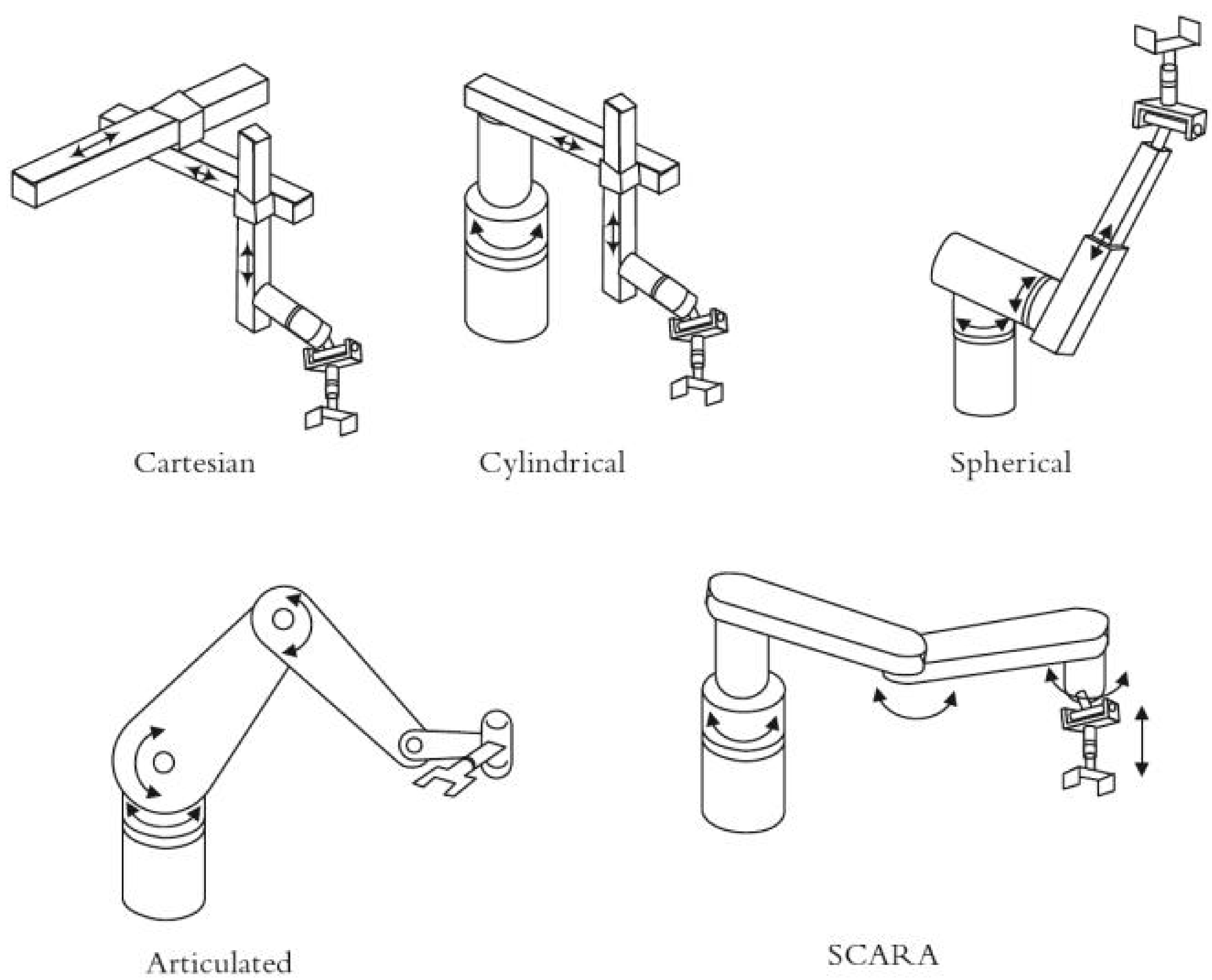
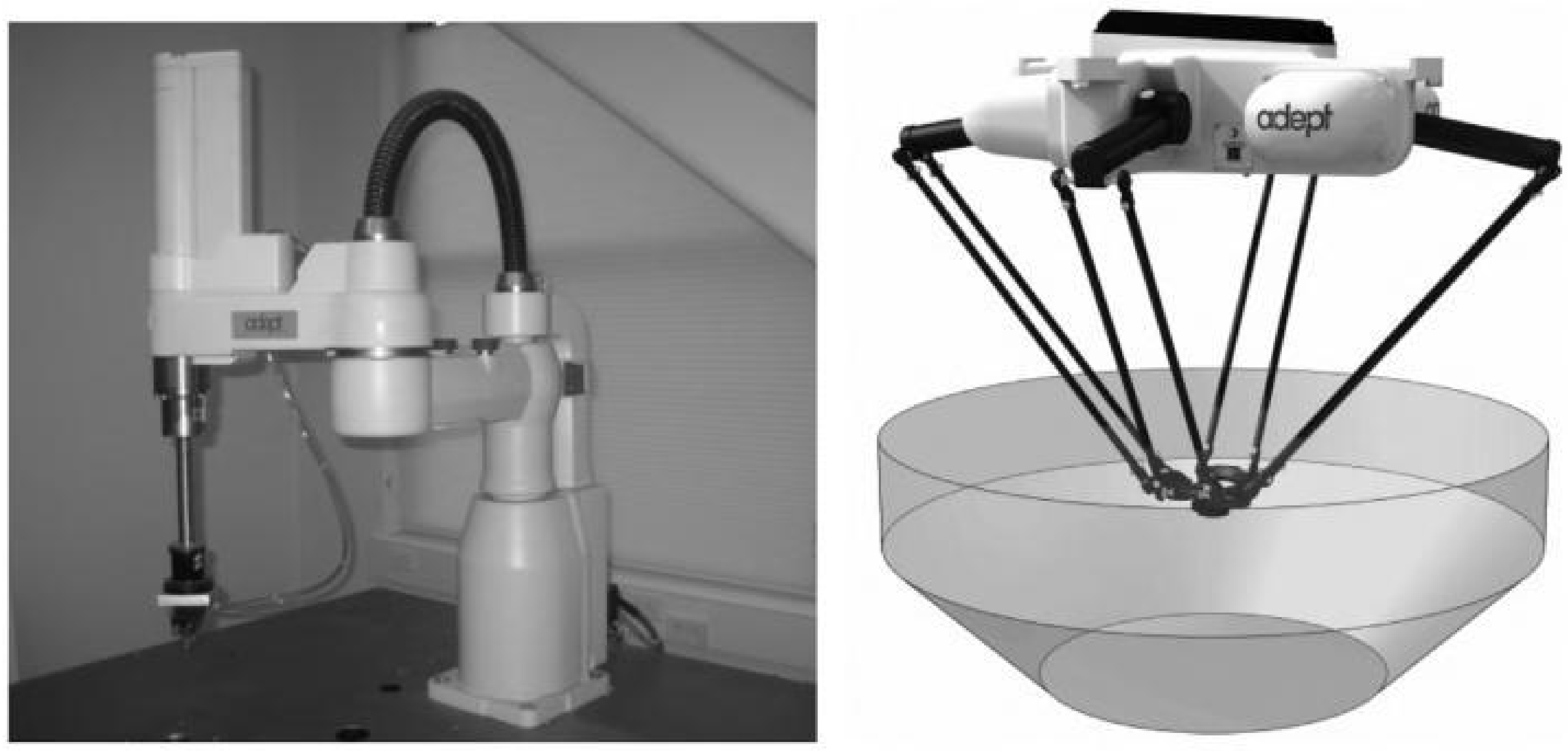


Figure 1.5 Some possible robot coordinate frames.



(a) (b)

Figure 1.6 (a) An Adept SCARA robot. (b) The Adept Quattro™ s650H robot. (Printed with permission from Adept Technology, Inc.)

specified by a succession of P, R, or S designations. For example, a robot with three prismatic and three revolute joints is specified by 3P3R. The following configurations are common for positioning the hand of the robot:

Cartesian/rectangular/gantry (3P): These robots are made of three linear joints that position the end effector, which are usually followed by additional revolute joints that orientate the end effector.

Cylindrical (PRP): Cylindrical coordinate robots have two prismatic joints and one revolute joint for positioning the part, plus revolute joints for orientating the part.

Spherical (P2R): Spherical coordinate robots follow a spherical coordinate system, which has one prismatic and two revolute joints for positioning the part, plus additional revolute joints for orientation.

Articulated/anthropomorphic (3R): An articulated robot's joints are all revolute, similar to a human's arm. They are the most common configuration for industrial robots.

Selective Compliance Assembly Robot Arm (SCARA): SCARA robots have two (or three) revolute joints that are parallel and allow the robot to move in a horizontal plane, plus an additional prismatic joint that moves vertically (Figure 1.6). SCARA robots are very common in assembly operations. Their specific characteristic is that they are more compliant in the $x-y$ plane but are very stiff along the z-axis, therefore providing selective compliance. This is an important issue in assembly, and will be discussed in Chapter 8.

1.11 Robot Reference Frames

Robots may be moved relative to different coordinate frames. In each type of coordinate frame, the motions will be different. Robot motions are usually accomplished in the following three coordinate frames (Figure 1.7):

World Reference Frame: This is a universal coordinate frame, as defined by the x-, y-, and z-axes. In this case, the joints of the robot move simultaneously in a coordinated

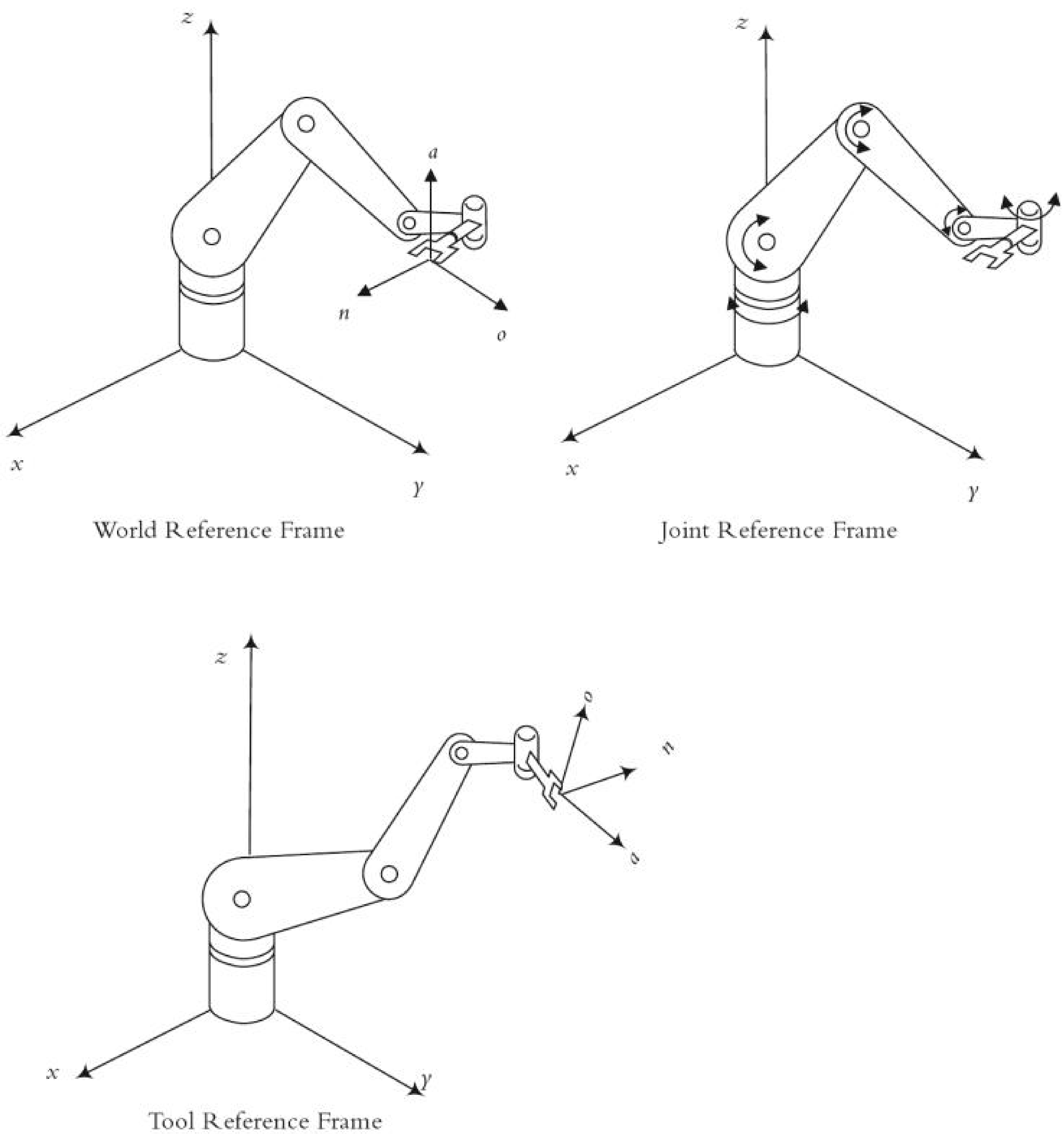


Figure 1.7 A robot's World, Joint, and Tool reference frames. Most robots may be programmed to move relative to any of these reference frames.

manner to create motions along the three major axes. In this frame, no matter where the arm, a positive movement along the x-axis is always in the plus direction of the x-axis, etc. The World reference frame is used to define the motions of the robot relative to other objects, define other parts and machines with which the robot communicates, and define motion trajectories.

Joint Reference Frame: This is used to specify movements of individual joints of the robot. In this case, each joint is accessed and moved individually; therefore, only one joint moves at a time. Depending on the type of joint used (prismatic, revolute, or spherical), the motion of the robot hand will be different. For instance, if a revolute joint is moved, the hand will move on a circle defined by the joint axis.

Tool Reference Frame: This specifies movements of the robot's hand relative to a frame attached to the hand, and consequently, all motions are relative to this local n,o,a-frame. Unlike the universal World frame, the local Tool frame moves with the robot. Suppose the hand is pointed as shown in Figure 1.7. Moving the hand relative to the positive n-axis of the local Tool frame will move the hand along the n-axis of the Tool frame. If the arm were pointed elsewhere, the same motion along the local n-axis of the Tool frame would be completely different from the first motion. The same $+n$-axis movement would be upward if the n-axis were pointed upward, and it would be downward if the n-axis were pointed downward. As a result, the Tool reference frame is a moving frame that changes continuously as the robot moves; therefore, the ensuing motions relative to it are also different depending on where the arm is and what direction the tool frame has. All joints of the robot must move simultaneously to create coordinated motions about the Tool frame. The Tool reference frame is an extremely useful frame in robotic programming where the robot is to approach and depart from other objects or to assemble parts.

1.12 Programming Modes

Robots may be programmed in a number of different modes, depending on the robot and how sophisticated it is. The following programming modes are common:

Physical Set-up: In this mode, an operator sets up switches and hard stops that control the motions of the robot. This mode is usually used along with other devices such as Programmable Logic Controllers (PLC).

Lead Through or Teach Mode: In this mode, the robot's joints are moved with a teach pendant. When the desired location and orientation is achieved, the location is entered (taught) into the controller. During playback, the controller moves the joints to the same locations and orientations. This mode is usually point-to-point; as such, the motion between points is not specified or controlled. Only the points that are taught are guaranteed to reach.

Continuous Walk-Through Mode: In this mode, all robot joints are moved simultaneously, while the motion is continuously sampled and recorded by the controller. During playback, the exact motion that was recorded is executed. The motions are taught by an operator, either through a model, by physically moving the end-effector, or by "wearing" the robot arm and moving it through its workspace. Painting robots, for example, may be programmed by skilled painters through this mode.

Software Mode: In this mode of programming the robot, a program is written offline or online and is executed by the controller to control the motions. The programming mode is the most sophisticated and versatile mode and can include sensory information, conditional statements (such as if . . . then statements), and branching. However, it requires a working knowledge of the programming syntax of the robot before any program is written. Most industrial robots can be programmed in more than one mode.

1.13 Robot Characteristics

The following definitions are used to characterize robot specifications:

Payload: Payload is the weight a robot can carry and still remain within its other specifications. As an example, a robot's maximum load capacity may be much larger than its specified payload, but at these levels, it may become less accurate, may not follow its intended trajectory accurately, or may have excessive deflections. The payload of robots compared to their own weight is usually very small. For example, Fanuc Robotics LR Mate™ robot has a mechanical weight of 86 lb and a payload of 6.6 lb, and the M-16i ™ robot has a mechanical weight of 594 lb and a payload of 35 lb.

Reach: Reach is the maximum distance a robot can reach within its work envelope. As will be seen later, many points within the work envelope of the robot may be reached with any desired orientation (called dexterous). However, for other points close to the limit of robot's reach capability, orientation cannot be specified as desired (called nondexterous point). Reach is a function of the robot's joints and lengths and its configuration. This is an important specification for industrial robots and must be considered before a robot is selected and installed.

Precision (validity): Precision is defined as how accurately a specified point can be reached. This is a function of the resolution of the actuators as well as the robot's feedback devices. Most industrial robots can have precision in the range of 0.001 inches or better. The precision is a function of how many positions and orientations were used to test the robot, with what load, and at what speed. When the precision is an important specification, it is crucial to investigate these issues.

Repeatability (variability): Repeatability is how accurately the same position can be reached if the motion is repeated many times. Suppose a robot is driven to the same point 100 times. Since many factors may affect the accuracy of the position, the robot may not reach the same point every time but will be within a certain radius from the desired point. The radius of a circle formed by the repeated motions is called repeatability. Repeatability is much more important than precision. If a robot is not precise, it will generally show a consistent error, which can be predicted, and therefore, corrected through programming. For example, suppose a robot is consistently off by 0.05 inches to the right. In that case, all desired points can be specified at 0.05 inches to the left and thereby eliminate the error. However, if the error is random, it cannot be predicted and consequently cannot be eliminated. Repeatability defines the extent of this random error. Repeatability is usually specified for a certain number of runs. Larger numbers of tests yield larger (bad for manufacturers) results, but more realistic (good for the users) results. Manufacturers must specify repeatability in conjunction with the number of tests, the applied payload during the tests, and the orientation of the arm. For example, the repeatability of an arm in a vertical direction will be different from when the arm is tested in a horizontal configuration. Most industrial robots have repeatability in the 0.001 inch range. It is crucial to find out about the details of repeatability if it is an important specification for the application.

1.14 Robot Workspace

Depending on their configuration and the size of their links and wrist joints, robots can reach a collection of points around them that constitute a workspace. The shape of the workspace for each robot is uniquely related to its design. The workspace may be found mathematically by writing equations that define the robot's links and joints and that include their limitations such as ranges of motions for each joint.[5] Alternately, the workspace may be found empirically by virtually moving each joint through its range of motions, combining all the space it can reach, and subtracting what it cannot reach.

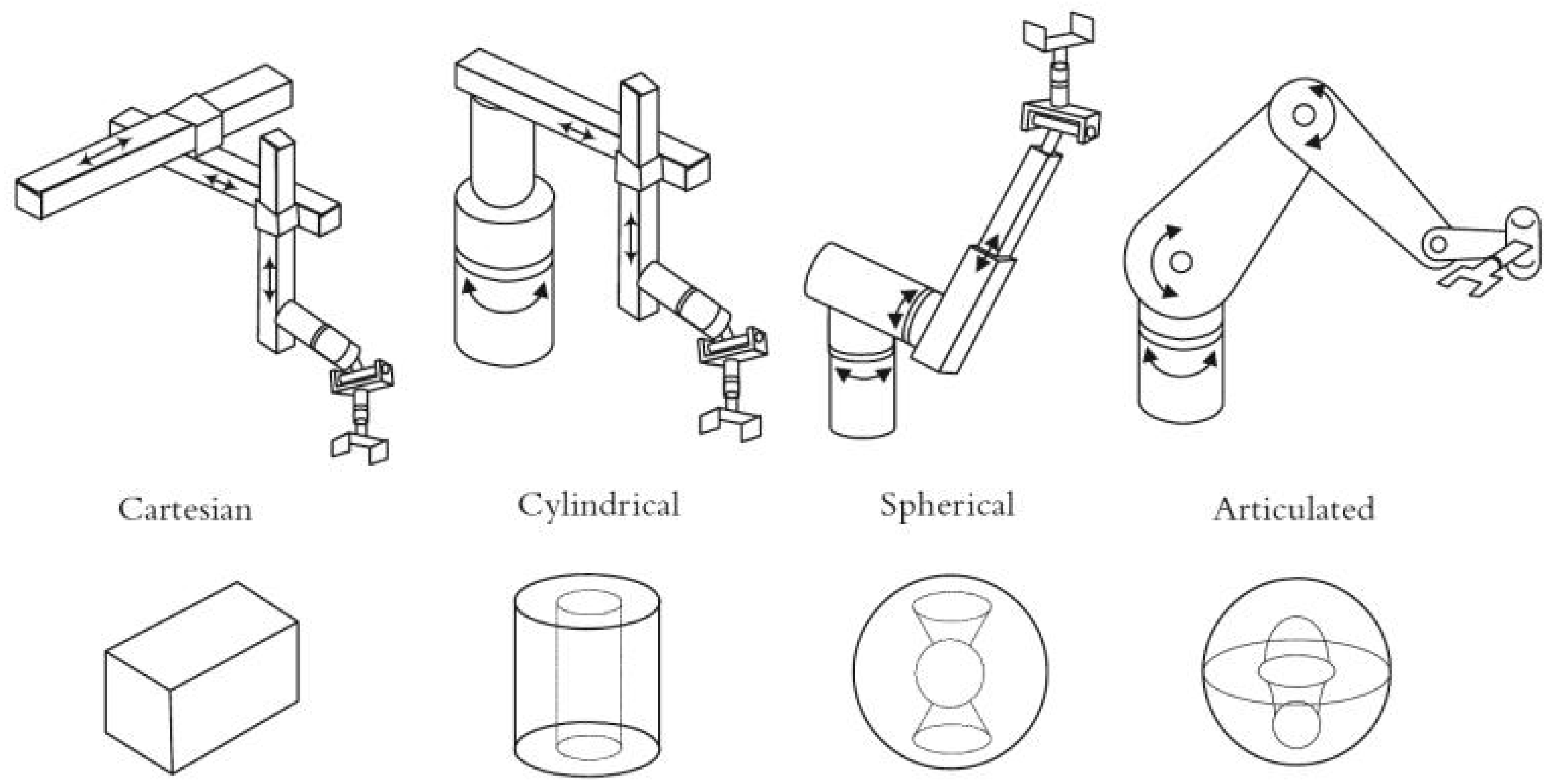


Figure 1.8 Typical approximate workspaces for common robot configurations.

Figure 1.8 shows the approximate workspace for some common configurations. When a robot is considered for a particular application, its workspace must be studied to ensure that the robot will be able to reach the desired points. For accurate workspace determination, refer to manufacturers' data sheets.

1.15 Robot Languages

There are perhaps as many robot languages as there are robot manufacturers. Each manufacturer designs its own robotic language; therefore, in order to use any particular robot, its brand of the programming language must be learned. Many robot languages are based on some other common language such as Cobol, Basic, C, and Fortran. Other languages are stand-alone and do not relate to any other common language.

Robotic languages are at different levels of sophistication, depending on their design and application. This ranges from machine level to a proposed human intelligence level.[6–9] High-level languages are either interpreter-based or compiler-based.

Interpreter-based languages execute one line of the program at a time. Each line of the program has a line number. The interpreter interprets the line every time it is encountered (it converts the robot program to a machine language program that the processor can understand and execute) and executes each line sequentially. The execution continues until the last line is encountered or until an error is detected, at which time execution stops. The advantage of an interpreter-based language is in its ability to continue execution until an error is detected, which allows the user to run and debug the program, portion by portion. As a result, debugging programs is much faster and easier. However, because each line is interpreted every time, execution is slower and not very efficient. Many robot languages such as Unimation™ VAL®, Adept's V+®, and IBM's AML® (A Manufacturing Language) are interpreter based.[9,10]

Compiler-based languages use a compiler to translate the whole program into machine language (which creates an object code) before it is executed. Since the processor executes the object code, these programs are much faster and more efficient. However, since the whole program must first be compiled, it will be impossible to run any part of the program if there are any syntax errors present, even before the logic of the program is tested. As a result, debugging these programs is much more difficult. Certain languages such as AL© are more flexible. They allow the user to debug the program in interpreter mode, while the actual execution is in compiler mode. The following is a general description of different levels of robotic languages.[7]

Micro-Computer Machine Language Level: In this level, the programs are written in machine language. This level of programming is the most basic and is very efficient, but it is difficult to understand and difficult for others to follow. All languages will eventually be interpreted or compiled to this level. However, in the case of higher level programs, the user writes the programs in a higher level language that is easier to follow and understand.

Point-to-Point Level: In this level (such as in Funky and Cincinnati Milacron's T3), the coordinates of the points are entered sequentially, and the robot follows the points as specified. This is a very primitive and simple type of program, and it is easy to use, but not very powerful. It also lacks branching, sensory information, and conditional statements.

Primitive Motion Level: In these languages, it is possible to develop more sophisticated programs, including sensory information, branching, and conditional statements (such as VAL by Unimation, V^+ by Adept, and so on). Most languages in this level are interpreter-based.

Structured Programming Level: Most languages in this level are compiler-based, are powerful, and allow more sophisticated programming. However, they are also more difficult to learn.

Task-Oriented Level: There are no actual languages in existence in this level—yet. Autopass, proposed by IBM in the 1980s, never materialized. Autopass was supposed to be task-oriented. This means that instead of programming a robot to perform a task by programming each and every step necessary to complete it, the user was to only mention the task, while the controller would create the necessary sequence. Imagine that a robot is to sort three boxes by size. In all existing languages, the programmer will have to tell the robot exactly what to do; therefore, every step must be programmed. The robot must be told how to go to the largest box, how to pick up the box, where to place it, where to go to the next box, and so on. In Autopass, the user would only indicate "sort," while the robot controller would create this sequence automatically. This never happened.

Example 1.1

The following is an example of a program written in V^+, which is used with Adept robots, is interpreter-based, and allows for branching, sensory input and output communication, straight-line movements, and many other features. As an example, the user may define a distance "height" along the z-axis of the end effector, which can be used with commands called APPRO (for approach) and DEPART in order to approach an object or depart from an object without collision. A command called MOVE will allow the robot to move from its present location to the next specified location. However, MOVES will do the same in a straight line. The difference is discussed in detail in Chapter 5. In the following listing, a number of different commands are described in order to show some of the capabilities of V^+.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Figure 1.14 A Fanuc LR Mate 200i robot is used in a material removal operation on a piece of jewelry. (Reprinted with permission from Fanuc Robotics, North America, Inc.)

Other surgical robots such as Mako Surgical Corporation's robot system and Intuitive Surgical's da Vinci system are used in a variety of surgical procedures, including orthopedic and internal surgery operations. For instance, da Vinci possesses four arms, three that hold instruments (one more than a surgeon could) and one to hold a 3D imaging scope that displays the surgical area to a surgeon behind a monitor (Figure 1.15). The surgeon directs the robotic movements with the help of haptic guidance systems, even remotely.[11] Similarly, other robots have been used to assist surgeons during microsurgery, including operations on heart valves in Paris and Liepzig.[12]

Assisting disabled individuals has also been tried with interesting results. Much can be done to help the disabled in their daily lives. In one study, a small tabletop robot was programmed to communicate with a disabled person and to perform simple tasks such as placing a food plate into the microwave oven or placing it in front of the disabled person

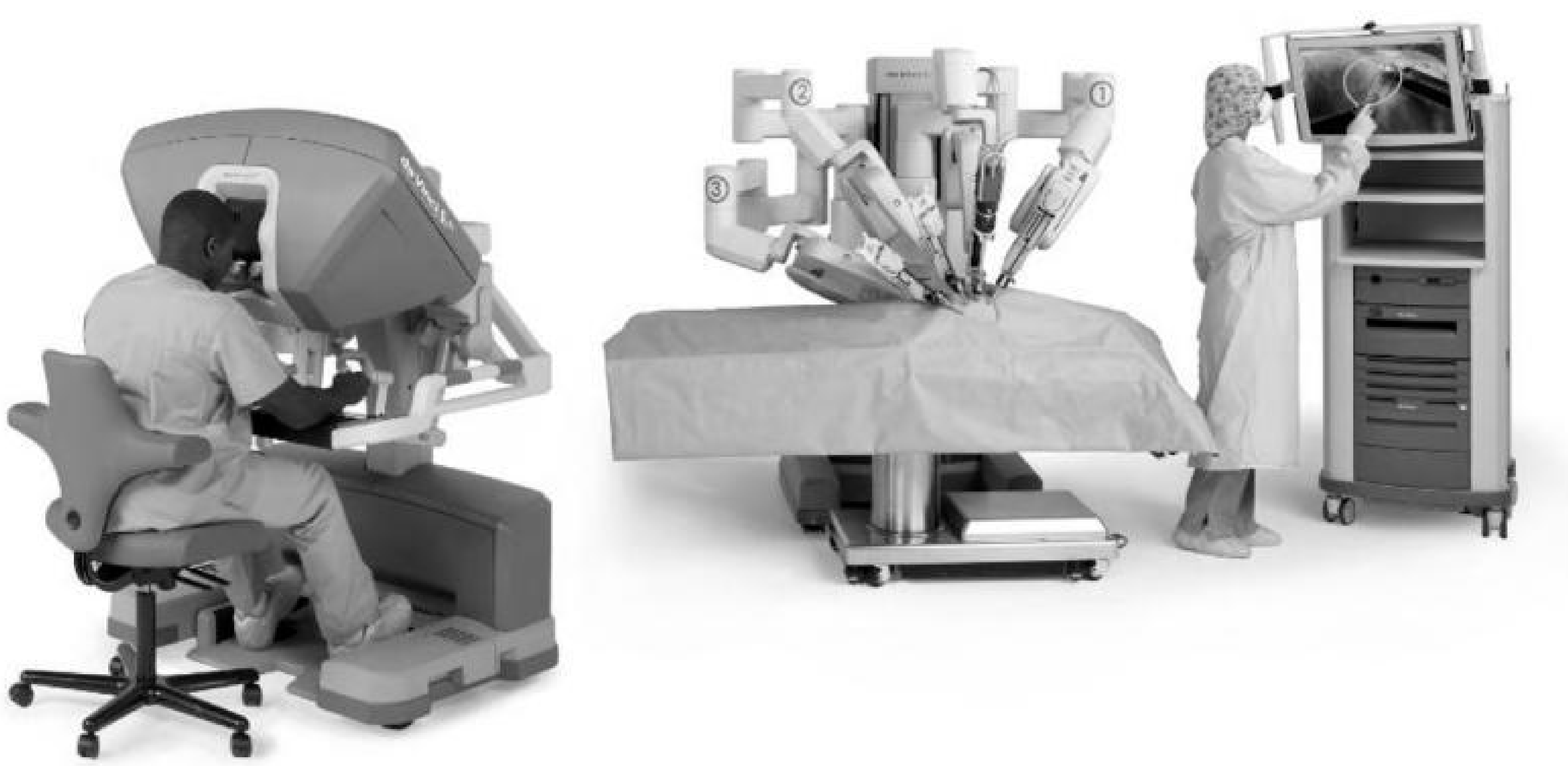


Figure 1.15 The da Vinci surgical system. (Image courtesy of Intuitive Surgical, Inc. (2010).)

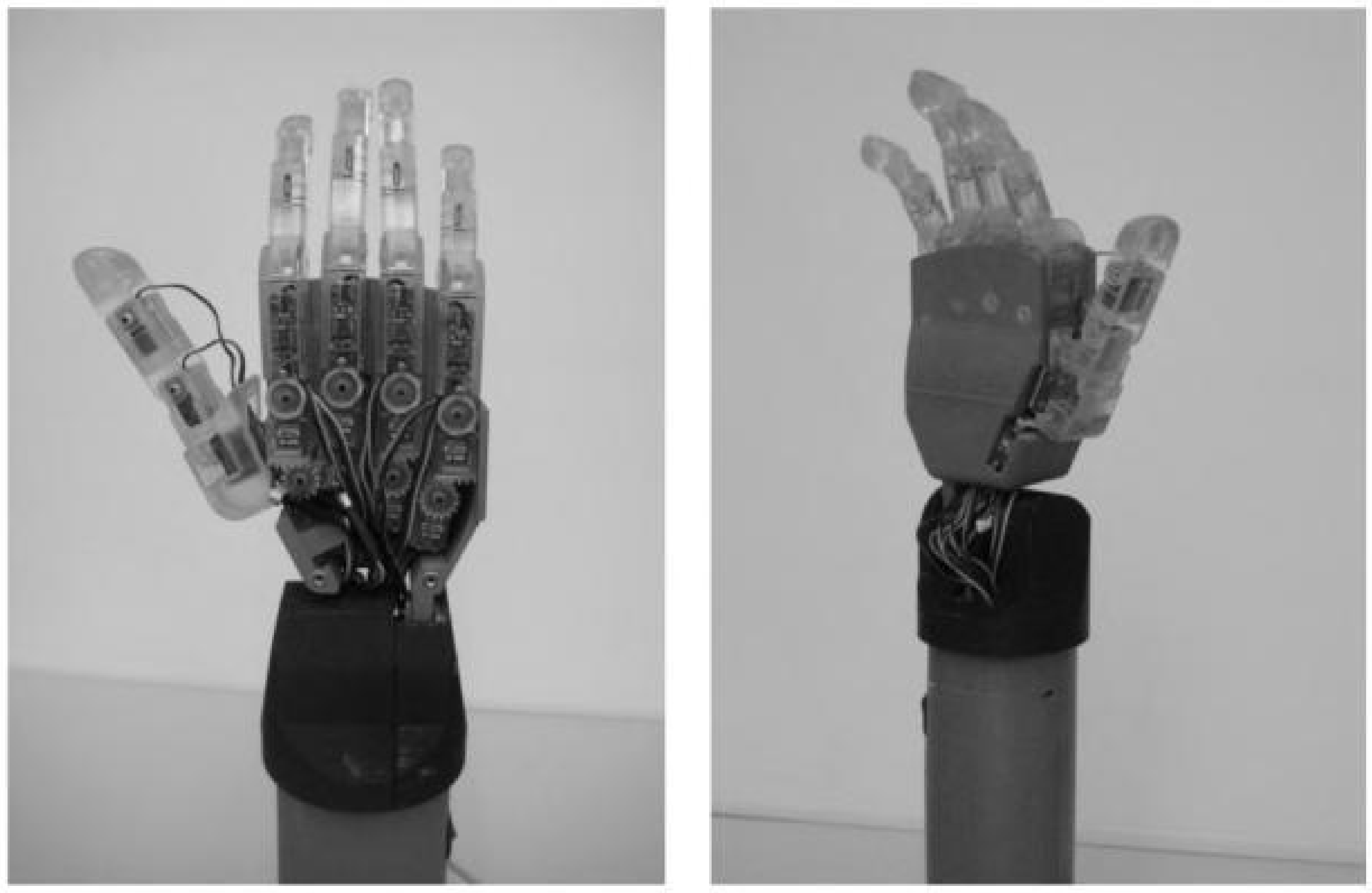


Figure 1.16 Finger-spelling hand for communication with blind-and-deaf individuals. (Supported by the Smith Kettlewell Eye Research Institute, San Francisco).

to eat.[13] Many other tasks were also programmed in the same fashion. The finger-spelling hand (Figure 1.16), designed for communication with the blind-and-deaf individuals, is capable of making gestures that spell all letters of the alphabet. With its 17 servomotors, the hand is mounted on an arm and can be held by one hand while the other hand reads the letters. The letters typed in a computer are coded and sent to the hand.[14]

Hazardous environments are well-suited for robotics. Because of their inherent danger, in these environments humans must be well-protected. However, robots can access, traverse, maintain, and explore these areas without the same level of concern. Servicing a radioactive environment, for example, is easier with a robot than a human. In 1993, an eight-legged robot called Dante was to reach the lava lake of constantly erupting volcano Mount Erebus in Antarctica and study its gases.[15] A variety of mine-detecting robots have also been put to use with the idea that a robot may be expendable whereas a human is not. One such robot uses vibrating ultrasonic pods to identify underground mines, therefore eliminating the need for human searches.[16] Another minesweeping robot, running on two all-terrain spiral tubes, has a bare-bones design and is meant to be expendable. Therefore, it is used to move in suspected areas and explode the mine.[17] A snakelike robot with articulating serial sections can maneuver through tight spaces. It is made of a series of sections, each composed of a pair of plates with fixed-length struts holding them together while linear actuators move one plate relative to the other, creating a snakelike motion.[18] Similarly, a crustacean-looking "lobster" robot was developed for searching ocean bottoms for mines and other weapons.[19] Another robot called Talon by QinetiQ™, designed for dangerous duty, can run alongside a soldier, cross broken terrain, and clear mines.[20]

Underwater, space, and inaccessible locations can also be serviced or explored by robots. So far, it is still impractical to send a human to other planets, even Mars, but there have been a number of rovers (Figure 1.17) that have already landed and explored it.[21] The same is true for other space and underwater applications.[22–24] Until recently for example, very few sunken ships were explored in deep oceans because no one could

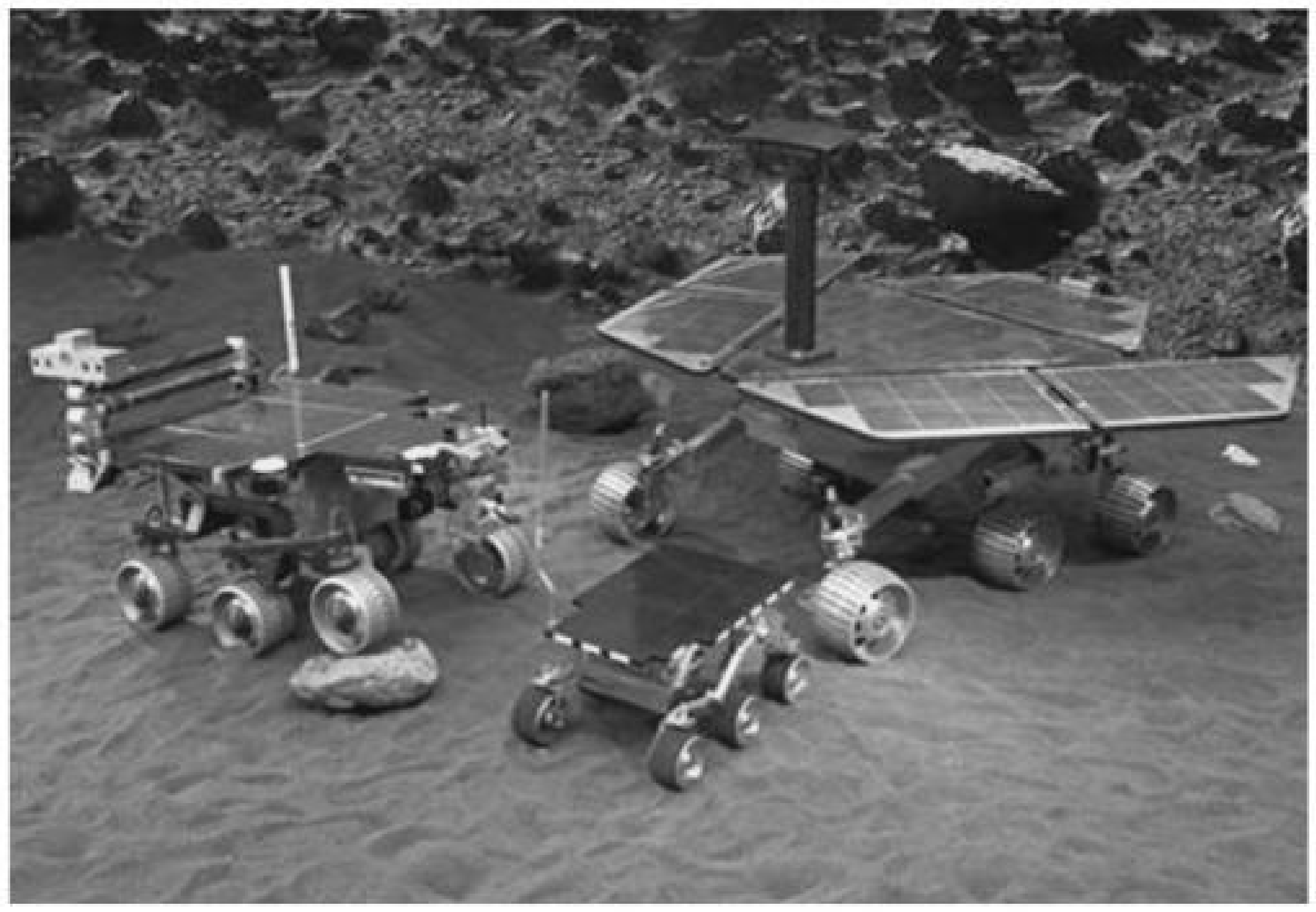


Figure 1.17 NASA Mars rovers in a lab test area, showing the FIDO rover (left) next to models of the Sojourner and MER rovers (center and right, respectively). (Courtesy NASA/JPL-Caltech.)

access those depths. Many crashed airplanes as well as sunken ships and submarines are nowadays recovered quickly by underwater robots.

In an attempt to clean the sludge from inside of a steam generator blowdown pipe, a teleoperated robot called Cecil was designed to crawl down the pipe and wash away the sludge with a stream of water at 5000 psi.[25] Figure 1.18 shows The Arm, a 6-DOF, bilateral force-feedback manipulator, used primarily on manned submersibles and remotely operated vehicles. The Arm is controlled via a remote master that also "feels" everything the slave arm feels. The system can also perform preprogrammed motions through a "teach and repeat" system.

NASA has developed a Robonaut, a humanoid anthropomorphic robot that functions as an astronaut. It has two five-fingered, tool-handling end effectors, modular robotic

Figure 1.18 The Arm, a 6-DOF, bilateral force-feedback manipulator, used primarily on manned submersibles and remotely operated vehicles. (Reprinted with permission from Western Space and Marine, Inc.)

components, and telepresence capability.[26] And finally, in another application, a telerobot was used for microsurgery.[27] In this case, the location of the telerobot is of secondary concern. The primary intention is to have the telerobot repeat the surgeon's hand movements at a smaller scale for reduced tremor during microsurgery.

1.17 Other Robots and Applications

Since the first edition of this book was published, new robots and issues have appeared. Such is the nature of this active subject. Therefore, you should expect that there will be applications and robots that are not included in this edition either. However, the following are just a sample of some systems that show a trend and future possibilities.

Roomba™, a robot vacuum cleaner, commercially available for years, autonomously and randomly moves throughout an area and vacuums the dust. It also finds its own docking station to recharge. All its intelligence is based on a few simple rules: randomly move around, turn left or right when hitting an obstacle, back up and turn around when in a corner, and find the docking station.[28]

Robots such as Honda's *ASIMO*, Bluebotic's *Gilbert*, Nestle's *Nesbot*, Anybots's *Monty*, and many others are intelligent humanoid robots with humanlike features and behavior. ASIMO walks, runs, goes up and down staircases, and interacts with people. Nesbot brings coffee to workers who have ordered it online.[28] Monty loads a dishwasher and does other chores, while Robomower mows your lawn while you read.[29] Figure 1.19 shows a picture of Nao robot.[30] Like others, Nao is a fully programmable robot that can behave autonomously–it communicates with humans and it walks, dances, and performs tasks.

A number of different robots have also been designed and used for emergency services during natural and human-caused disasters. These robots, equipped with special sensors, are capable of looking for live humans and animals buried under rubble and reporting their locations to rescuers. Similar robots are also used for diffusing bombs and other explosive devices. SDA10 dual-arm robot by Motoman, Inc. (Figure 1.20), has 15 axes of motion. The two arms can move independently or in a coordinated manner. It can transfer a part from one gripper to the other without the need to set it down.

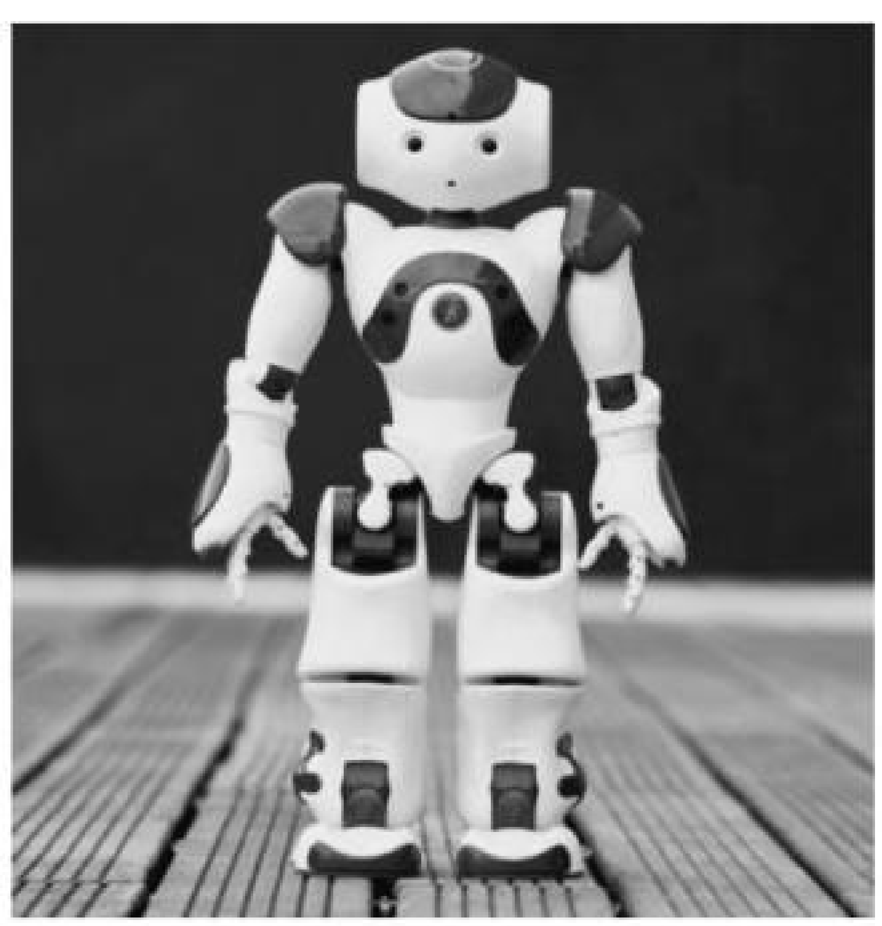


Figure 1.19 Nao humanoid robot. (Reprinted with permission from Aldebaran Robotics (Picture by C. De Torquat).)

Figure 1.20 SDA10 dual arm robot. (Reprinted with permission from Motoman, Inc.)

Exoskeletal assistive devices, although not robots, follow the same logic and allow a human to carry large loads for extended periods of time. In fact, these devices can conceivably be used to aid the disabled in different forms, including helping a wheel-chair-bound person walk. One lightweight exoskeleton device called Human Universal Load Carrier (HULC) assists people in carrying heavy loads (as of the date of this writing, over 200 lbs. for 10 hours). As shown in Figure 1.21, the skeleton is worn by the person who directs the motions of the frame, but the frame carries the weight and is actuated by a battery driven hydraulic pump.[31,32]

Figure 1.21 The Human Universal Load Carrier (HULC) is an un-tethered, hydraulic-powered anthropomorphic exoskeleton that provides users with the ability to carry loads of up to 200 lb for extended periods of time and over all terrains. (Reprinted by permission from Berkeley Bionics.)

References

1. Capek, Karel, "Rossum's Universal Robots," translated by Paul Selver, Doubleday, NY, 1923.
2. Valenti, Michael, "A Robot Is Born," *Mechanical Engineering*, June 1996, pp. 50–57.
3. "Robot Safety," Bonney, M. C., Y. F. Yong, Editors, IFS Publications Ltd., UK, 1985.
4. Stein, David, Gregory S. Chirikjian, "Experiments in the Commutation and Motion Planning of a Spherical Stepper Motor," Proceedings of DETC'00, ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Baltimore, Maryland, September 2000, pp. 1–7.
5. Wiitala, Jared M., B. J., Rister, J. P. Schmiedler, "A More Flexible Robotic Wrist," *Mechanical Engineering*, July 1997, pp. 78–80.
6. W. A. Gruver, B. I. Soroka, J. J. Craig, and T. L. Turner, "Industrial Robot Programming Languages: A Comparative Evaluation." *IEEE Transactions on Systems, Man, and Cybernetics* SMC–14(4), July/August 1984.
7. Bonner, Susan, K. G. Shin, "A Comprehensive Study of Robot Languages," *IEEE Computer*, December 1982, pp. 82–96.
8. Kusiak, Andrew, "Programming, Off-Line Languages,"International Encyclopedia of Robotics: Applications and Automation, Richard C. Dorf, Editor, John Wiley and Sons, NY, 1988, pp. 1235–1250.
9. Gruver, William, B. I. Soroka, "Programming, High Level Languages," International Encyclopedia of Robotics: Applications and Automation, Richard C. Dorf, Editor, John Wiley and Sons, NY, 1988, pp. 1203–1234.
10. VAL-II Programming Manual, Version 4, Unimation, Inc., Pittsburgh, 1988.
11. McGuinn, Jack, Senior Editor, "These Bots are Cutting-Edge," *Power Transmission Engineering*, October 2008, pp. 32–36.
12. Salisbury, Kenneth, Jr., "The Heart of Microsurgery," *Mechanical Engineering*, December 1998, pp. 46–51.
13. "Stanford Rehabilitation Center," Stanford University, California.
14. Garcia, Mario, Saeed Niku,"Finger-Spelling Hand," California Polytechnic State University masters thesis, San Luis Obispo, 2009.
15. Leary, Warren, "Robot Named Dante to Explore Inferno of Antarctic Volcano," *The New York Times*, December 8, 1992, p. B7.
16. "Ultrasonic Detector and Identifier of Land Mines," *NASA Motion Control Tech Briefs*, 2001, p. 8b.
17. Chalmers, Peggy, "Lobster Special," *Mechanical Engineering*, September 2000, pp. 82–84.
18. "Snakelike Robots Would Maneuver in Tight Spaces," *NASA Tech Briefs*, August 1998, pp. 36–37.
19. "Lobster Special," *Mechanical Engineering*, September 2000, pp. 82–84.
20. "Robot Population Explosion," *Mechanical Engineering*, February 2009, p. 64.
21. http://www.jpl.nasa.gov/pictures/
22. Wernli, Robert L., "Robotics Undersea," *Mechanical Engineering*, August 1982, pp. 24–31.
23. Asker, James, "Canada Gives Station Partners a Hand—And an Arm," *Aviation Week & Space Technology*, December 1997, pp. 71–73.
24. Puttre, Michael, "Space-Age Robots Come Down to Earth," *Mechanical Engineering*, January 1995, pp. 88–89.
25. Trovato, Stephen A., "Robot Hunts Sludge and Hoses It Away," *Mechanical Engineering*, May 1988, pp. 66–69.
26. "The Future of Robotics Is Now," *NASA Tech Briefs*, October 2002, p. 27.
27. "Telerobot Control for Microsurgery," *NASA Tech Briefs*, October 1997, p. 46.
28. "From Simple Rules, Complex Behavior," *Mechanical Engineering*, July 2009, pp. 22–27.
29. Drummond, Mike, "Rise of the Machines," *Inventors Digest*, February 2008, pp. 16–23.
30. http://www.aldebaran-robotics.com/eng/PressFiles.php
31. www.berkeleybionics.com
32. "Lightweight Exoskeleton with Controllable Actuators," *NASA Tech Briefs*, October 2004, pp. 54–55.

33. Chang, Kenneth, John Scwartz, "Led by Robots, Roaches Abandon Instincts," *The New York Times*, November 15, 2007.
34. Yeaple, Judith A., "Robot Insects," *Popular Science*, March 1991, pp. 52–55 and 86.
35. Freedman, David, "Invasion of the Insect Robots," *Discover*, March 1991, pp. 42–50.
36. Thakoor, Sarita, B. Kennedy, A. Thakoor, "Insectile and Vemiform Exploratory Robots," *NASA Tech Briefs*, November 1999, pp. 61–63.
37. http://www.youtube.com/watch?v=uBikHgnt16E
38. Terry, Bryan,"The Robo-Snake," Senior Project report, Cal Poly, San Luis Obispo, California, June 2000.
39. O'Conner, Leo, "Robotic Fish Gotta Swim, Too," *Mechanical Engineering*, January 1995, p. 122.
40. Lipson, Hod, J. B. Pollack, "The Golem Project: Automatic Design and Manufacture of Robotic Lifeforms," http://golem03.cs-i.brandies.edu/index.html.
41. Corrado, Joseph K., "Military Robots," *Design News*, October 83, pp. 45–66.
42. "Low-Cost Minesweeping," *Mechanical Engineering*, April 1996, p. 66.
43. IS Robotics, Somerville, Massachusetts.
44. "Biomorphic Gliders," *NASA Tech Briefs*, April 2001, pp. 65–66.
45. Baumgartner, Henry, "When Bugs Are the Machines," *Mechanical Engineering*, April 2001, p. 108.
46. Markoff, John, "Crashes and Traffic Jams in Military Test of Robotic Vehicles," *The New York Times*, November 5, 2007.
47. Jones, Adam, S. B., Niku, "Animatronic Lips with Speech Synthesis (AliSS)," Proceedings of the 8th Mechatronics Forum and International Conference, University of Twente, The Netherlands, June 2002.
48. Sanders, John K., S. B. Shooter,"The Design and Development of an Animatronic Eye," Proceedings of DETC 98/MECH: 25th ASME Biennial Mechanisms Conference, September 1998.
49. Coates, V. T., "The Potential Impacts of Robotics," Paper number 83-WA/TS-9, American Society of Mechanical Engineers, 1983.
50. Albus, James, "Brains, Behavior, and Robotics," Byte Books, McGraw-Hill, 1981.

Problems

1.1. Draw the approximate workspace for the following robot. Assume the dimensions of the base and other parts of the structure of the robot are as shown.

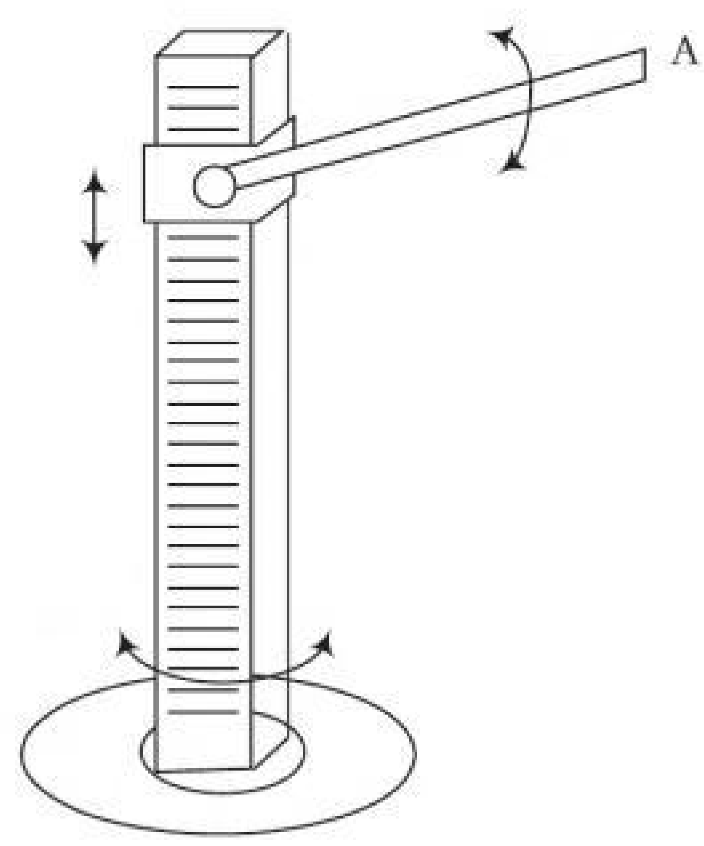


Figure P.1.1

1.2. Draw the approximate workspace for the following robot. Assume the dimensions of the base and other parts of the structure of the robot are as shown.

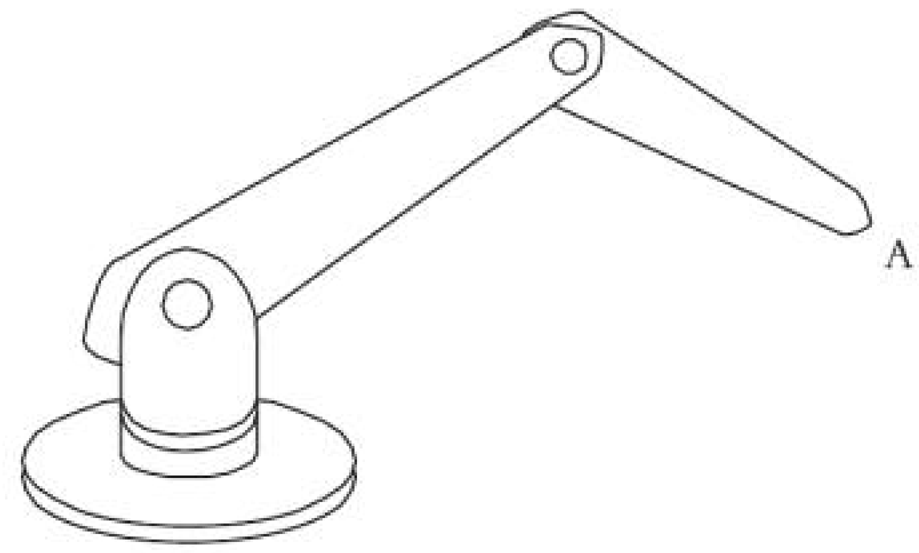


Figure P.1.2

1.3. Draw the approximate workspace for the following robot. Assume the dimensions of the base and other parts of the structure of the robot are as shown.

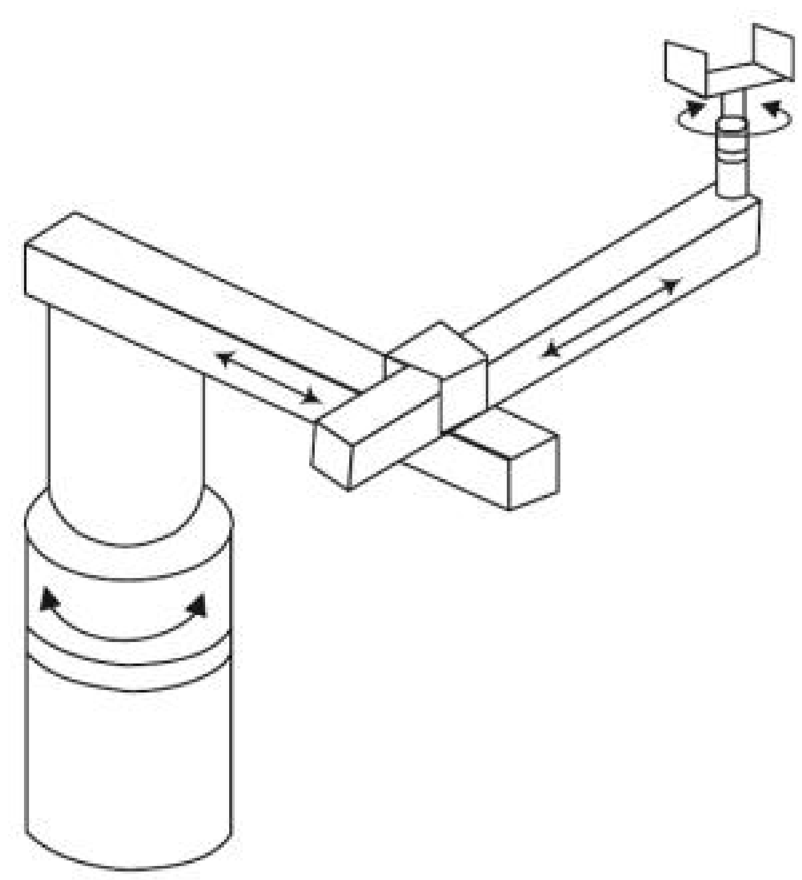


Figure P.1.3

CHAPTER 2

Kinematics of Robots: Position Analysis

2.1 Introduction

In this chapter, we will study forward and inverse kinematics of robots. With forward kinematic equations, we can determine where the robot's end (hand) will be if all joint variables are known. Inverse kinematics enables us to calculate what each joint variable must be in order to locate the hand at a particular point and a particular orientation. Using matrices, we will first establish a method of describing objects, locations, orientations, and movements. Then we will study the forward and inverse kinematics of different robot configurations such as Cartesian, cylindrical, and spherical coordinates. Finally, we will use the Denavit–Hartenberg representation to derive forward and inverse kinematic equations of all possible configurations of robots—regardless of number of joints, order of joints, and presence (or lack) of offsets and twists.

It is important to realize that in practice, manipulator-type robots are delivered with no end effector. In most cases, there may be a gripper attached to the robot; however, depending on the actual application, different end effectors are attached to the robot by the user. Obviously, the end effector's size and length determine where the end of the robot will be. For a short end effector, the end will be at a different location compared to a long end effector. In this chapter, we will assume that the end of the robot is a plate to which the end effector can be attached, as necessary. We will call this the ''hand'' or the ''end plate'' of the robot. If necessary, we can always add the length of the end effector to the robot for determining the location and orientation of the end effector. It should be mentioned here that a real robot manipulator, for which the length of the end effector is not defined, will calculate its joint values based on the end plate location and orientation, which may be different from the position and orientation perceived by the user.

2.2 Robots as Mechanisms

Manipulator-type robots are multi-degree-of-freedom (DOF), three-dimensional, open loop, chain mechanisms, and are discussed in this section.

Multi-degree-of-freedom means that robots possess many joints, allowing them to move freely within their envelope. In a 1-DOF system, when the variable is set to a particular value, the mechanism is totally set and all its other variables are known. For example, in the 1-DOF 4-bar mechanism of Figure 2.1, when the crank is set to 120°, the angles of the coupler link and the rocker arm are also known, whereas in a multi-DOF mechanism, all input variables must be individually defined in order to know the remaining parameters. Robots are multi-DOF machines, where each joint variable must be known in order to determine the location of the robot's hand.

Robots are three-dimensional machines if they are to move in space. Although it is possible to have a two-dimensional multi-DOF robot, they are not common (or useful).

Robots are open-loop mechanisms. Unlike mechanisms that are closed-loop (e.g., 4-bar mechanisms), even if all joint variables are set to particular values, there is no guarantee that the hand will be at the given location. This is because deflections in any joint or link will change the location of all subsequent links without feedback. For example, in the 4-bar mechanism of Figure 2.2, when link AB deflects as a result of load *F*, link BO_2 will also move; therefore, the deflection can be detected. In an open-loop system such as the robot, the deflections will move all succeeding members without any

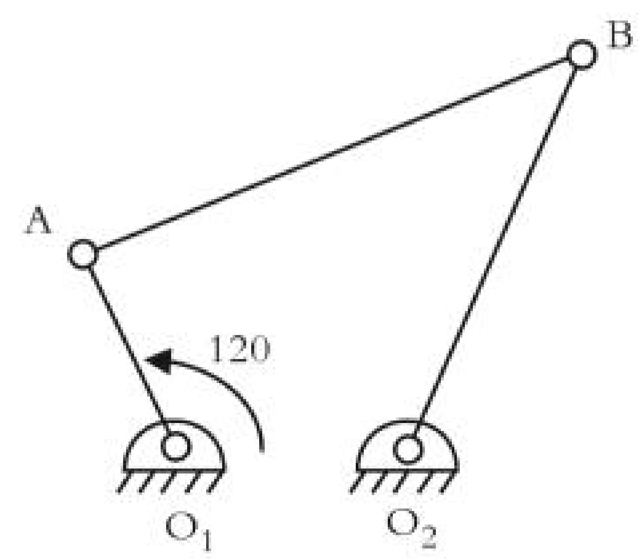


Figure 2.1 A 1-DOF closed-loop 4-bar mechanism.

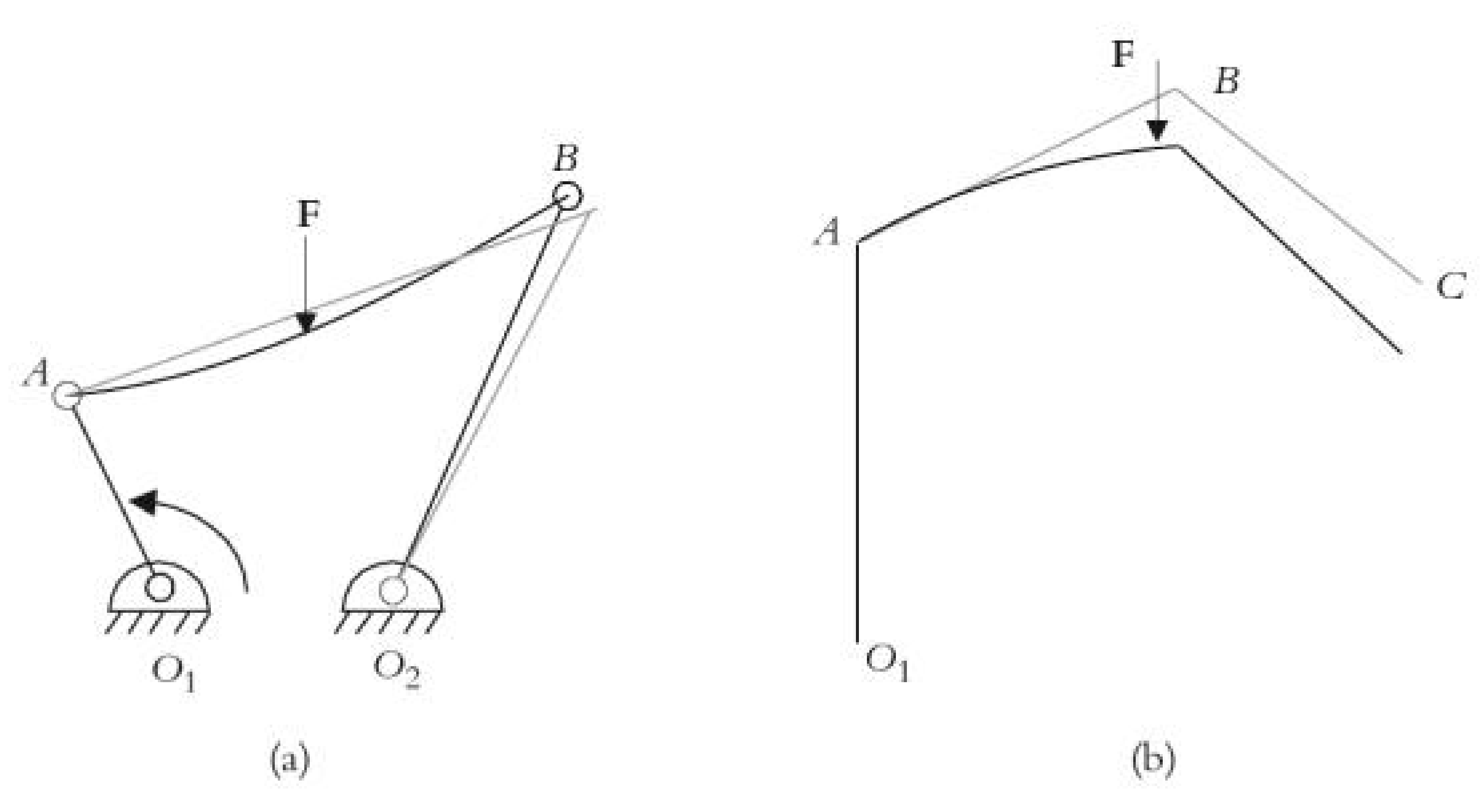


Figure 2.2 Closed-loop (a) versus open-loop (b) mechanisms.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

then:

$$\mathbf{P} = a_x\mathbf{i} + b_y\mathbf{j} + c_z\mathbf{k} \tag{2.4}$$

where a_x, b_y, and c_z are the three components of the vector in the reference frame. In fact, point P in the previous section is in reality represented by a vector connected to it at point P and expressed by the three components of the vector.

The three components of the vector can also be written in matrix form, as in Equation (2.5). This format will be used throughout this book to represent all kinematic elements:

$$\mathbf{P} = \begin{bmatrix} a_x \\ b_y \\ c_z \end{bmatrix} \tag{2.5}$$

This representation can be slightly modified to also include a scale factor w such that if P_x, P_y, and P_z are divided by w, they will yield a_x, b_y, and c_z. Therefore the vector can be written as:

$$\mathbf{P} = \begin{bmatrix} P_x \\ P_y \\ P_z \\ w \end{bmatrix} \quad \text{where } a_x = \frac{P_x}{w}, b_y = \frac{P_y}{w}, \text{ etc.} \tag{2.6}$$

w may be any number and, as it changes, it can change the overall size of the vector. This is similar to the zooming function in computer graphics. As the value of w changes, the size of the vector changes accordingly. If w is bigger than 1, all vector components enlarge; if w is smaller than 1, all vector components become smaller.

When w is 1, the size of these components remains unchanged. However, if $w = 0$, then a_x, b_y, and c_z will be infinity. In this case, P_x, P_y, and P_z (as well as a_x, b_y, and c_z) will represent a vector whose length is infinite but nonetheless is in the direction represented by the vector. This means that a *direction vector* can be represented by a scale factor of $w = 0$, where the length is not important, but the direction is represented by the three components of the vector. This will be used throughout the book to represent direction vectors.

In computer graphics applications, the addition of a scale factor allows the user to zoom in or out simply by changing this value. Since the scale factor increases or decreases all vector dimensions accordingly, the size of a vector (or drawing) can be easily changed without the need to redraw it. However, our reason for this inclusion is different, and it will become apparent shortly.

Example 2.1

A vector is described as $\mathbf{P} = 3\mathbf{i} + 5\mathbf{j} + 2\mathbf{k}$. Express the vector in matrix form:

(a) With a scale factor of 2.
(b) If it were to describe a direction as a unit vector.

Solution: The vector can be expressed in matrix form with a scale factor of 2 as well as 0 for direction as:

$$\mathbf{P} = \begin{bmatrix} 6 \\ 10 \\ 4 \\ 2 \end{bmatrix} \quad \text{and} \quad \mathbf{P} = \begin{bmatrix} 3 \\ 5 \\ 2 \\ 0 \end{bmatrix}$$

However, in order to make the vector into a unit vector, we normalize the length to be equal to 1. To do this, each component of the vector is divided by the square root of the sum of the squares of the three components:

$$\lambda = \sqrt{P_x^2 + P_y^2 + P_z^2} = 6.16 \text{ and } P_x = {}^{3}/_{6.16} = 0.487, \text{ etc. Therefore,}$$

$$\mathbf{P}_{unit} = \begin{bmatrix} 0.487 \\ 0.811 \\ 0.324 \\ 0 \end{bmatrix}$$

Note that $\sqrt{0.487^2 + 0.811^2 + 0.324^2} = 1.$ ■

Example 2.2

A vector $\mathbf{p}$ is 5 units long and is in the direction of a unit vector $\mathbf{q}$ described below. Express the vector in matrix form.

$$\mathbf{q}_{unit} = \begin{bmatrix} 0.371 \\ 0.557 \\ q_z \\ 0 \end{bmatrix}$$

Solution: The unit vector's length must be 1. Therefore,

$$\lambda = \sqrt{q_x^2 + q_y^2 + q_z^2} = \sqrt{0.138 + 0.310 + q_z^2} = 1 \quad \rightarrow \quad q_z = 0.743$$

$$\mathbf{q}_{unit} = \begin{bmatrix} 0.371 \\ 0.557 \\ 0.743 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{p} = \mathbf{q}_{unit} \times 5 = \begin{bmatrix} 1.855 \\ 2.785 \\ 3.715 \\ 1 \end{bmatrix}$$

■

2.4.3 Representation of a Frame at the Origin of a Fixed Reference Frame

A frame is generally represented by three mutually orthogonal axes (such as x, y, and z). Since we may have more than one frame at any given time, we will use axes x, y, and z to represent the fixed Universe reference frame $F_{x,y,z}$ and a set of axes n, o, and a to represent another (moving) frame $F_{n,o,a}$ relative to the reference frame. This way, there should be no confusion about which frame is referenced.

The letters *n*, *o*, and *a* are derived from the words *normal*, *orientation*, and *approach*. Referring to Figure 2.6, it should be clear that in order to avoid hitting the part while trying to pick it up, the robot would have to approach it along the *z*-axis of the gripper. In robotic nomenclature, this axis is called *approach-axis* and is referred to as the *a*-axis. The orientation with which the gripper frame approaches the part is called *orientation-axis*, and it is referred to as the *o*-axis. Since the *x*-axis is normal to both, it is referred to as *n*-axis. Throughout this book, we will refer to a moving frame as $F_{n,o,a}$ with *normal*, *orientation*, and *approach* axes.

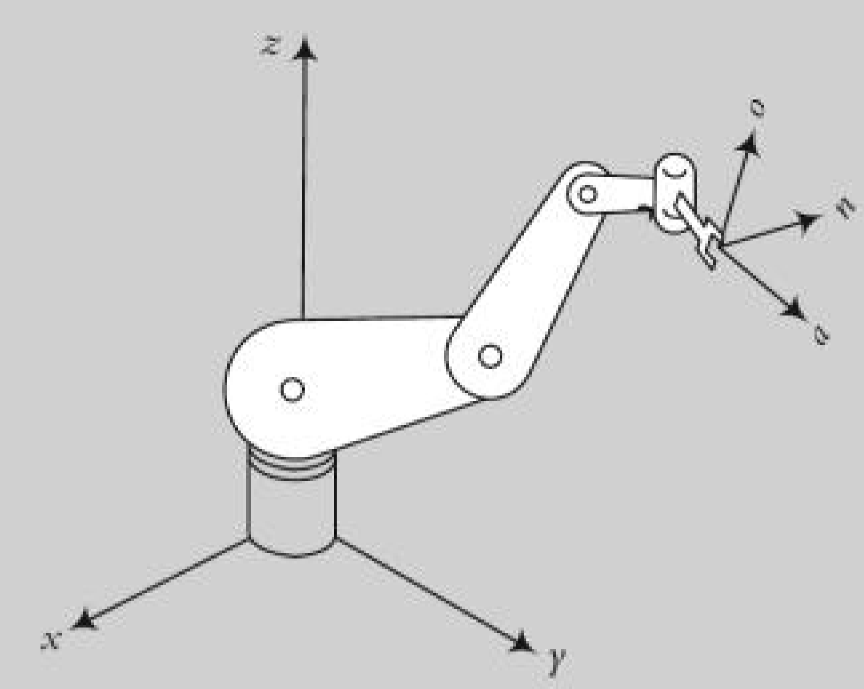


Figure 2.6 The normal-, orientation-, and approach-axis of a moving frame.

Each direction of each axis of a frame $F_{n,o,a}$ located at the origin of a reference frame $F_{x,y,z}$ (Figure 2.7) is represented by its three directional cosines relative to the reference frame as in section 2.4.2. Consequently, the three axes of the frame can be represented by three vectors in matrix form as:

$$F = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \tag{2.7}$$

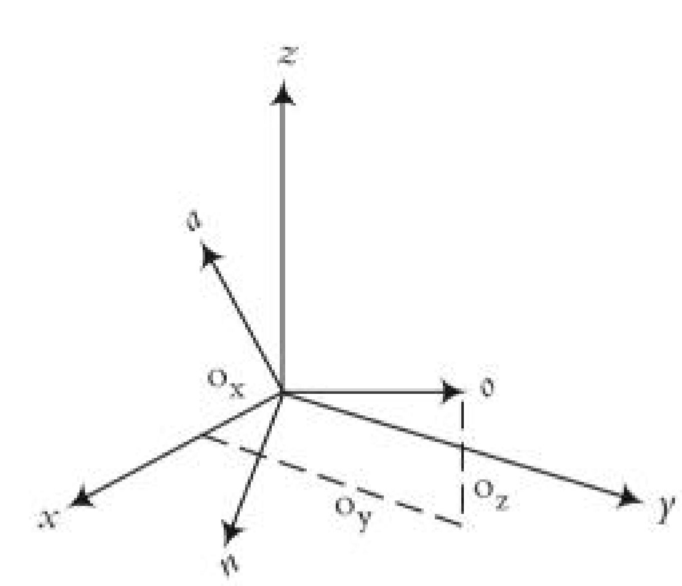


Figure 2.7 Representation of a frame at the origin of the reference frame.

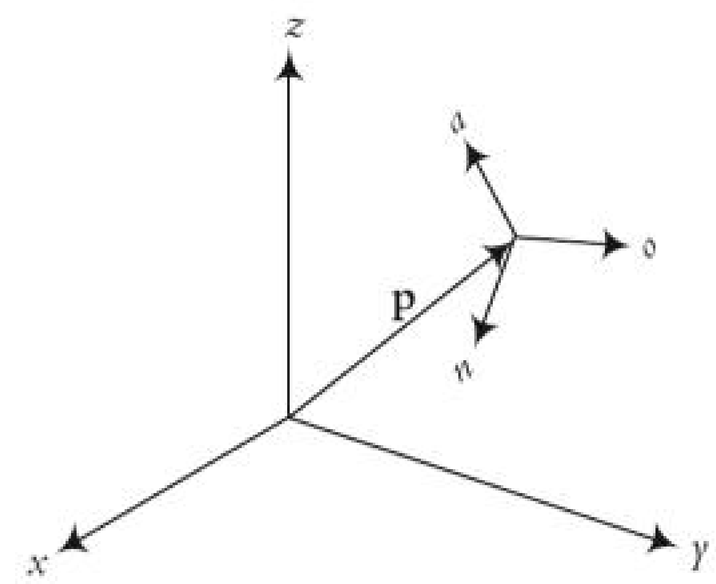


Figure 2.8 Representation of a frame in a frame.

2.4.4 Representation of a Frame Relative to a Fixed Reference Frame

To fully describe a frame relative to another frame, both the location of its origin and the directions of its axes must be specified. If a frame is not at the origin (or, in fact, even if it is at the origin) of the reference frame, its location relative to the reference frame is described by a vector between the origin of the frame and the origin of the reference frame (Figure 2.8). Similarly, this vector is expressed by its components relative to the reference frame. Therefore, the frame can be expressed by three vectors describing its directional unit vectors and a fourth vector describing its location as:

$$F = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

As shown in Equation (2.8), the first three vectors are directional vectors with $w = 0$, representing the directions of the three unit vectors of the frame $F_{n,o,a}$, while the fourth vector with $w = 1$ represents the location of the origin of the frame relative to the reference frame. Unlike the unit vectors, the length of vector p is important. Consequently, we use a scale factor of 1.

A frame may also be represented by a 3×4 matrix without the scale factors, but it is not common. Adding the fourth row of scale factors to the matrix makes it a 4×4 or *homogeneous* matrix.

Example 2.3

The frame F shown in Figure 2.9 is located at 3,5,7 units, with its n-axis parallel to x, its o-axis at 45° relative to the y-axis, and its a-axis at 45° relative to the z-axis. The frame can be described by:

$$F = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 0.707 & -0.707 & 5 \\ 0 & 0.707 & 0.707 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

■

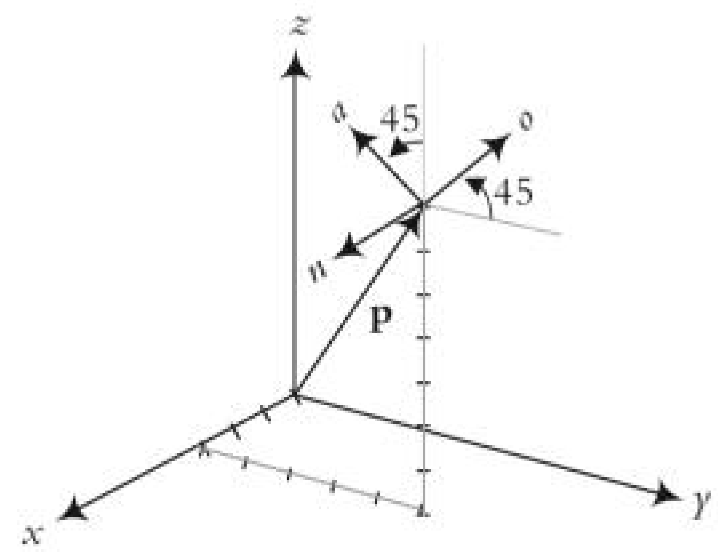


Figure 2.9 An example of representation of a frame.

2.4.5 Representation of a Rigid Body

An object can be represented in space by attaching a frame to it and representing the frame. Since the object is permanently attached to this frame, its position and orientation relative to the frame is always known. As a result, so long as the frame can be described in space, the object's location and orientation relative to the fixed frame will be known (Figure 2.10). As before, a frame can be represented by a matrix, where the origin of the frame and the three vectors representing its orientation relative to the reference frame are expressed. Therefore,

$$F_{object} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

As we discussed in Chapter 1, a point in space has only three degrees of freedom; it can only move along the three reference axes. However, a rigid body in space has six degrees of freedom, meaning that not only can it move along x-, y-, and z-axes, it can also rotate about these three axes. Consequently, all that is needed to completely define an object in space is six pieces of information describing the location of the origin of the object in the reference frame and its orientation about the three axes. However, as can be seen in Equation (2.9), twelve pieces of information are given: nine for orientation, and three for position (this excludes the scale factors on the last row of the matrix because they do not add to this information). Obviously, there must be some constraints present in this

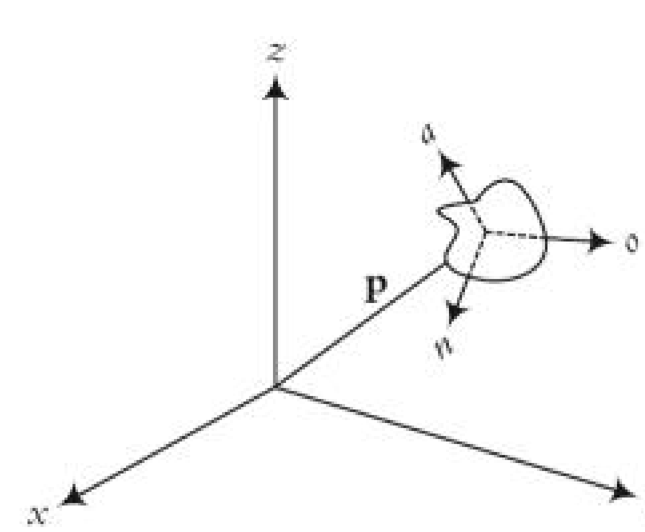


Figure 2.10 Representation of an object in space.

representation to limit the above to six. Therefore, we need 6 constraint equations to reduce the above from twelve to six. The constraints come from the known characteristics of a frame that have not been used yet, that:

- the three unit vectors **n**, **o**, **a** are mutually perpendicular, and
- each unit vector's length, represented by its directional cosines, must be equal to 1.

These constraints translate into the following six constraint equations:

$$\begin{aligned}
&1.\ \mathbf{n}\cdot\mathbf{o} = 0 \text{ (the dot-product of } \mathbf{n} \text{ and } \mathbf{o} \text{ vectors must be zero)}\\
&2.\ \mathbf{n}\cdot\mathbf{a} = 0\\
&3.\ \mathbf{a}\cdot\mathbf{o} = 0\\
&4.\ |\mathbf{n}| = 1 \text{ (the magnitude of the length of the vector must be 1)}\\
&5.\ |\mathbf{o}| = 1\\
&6.\ |\mathbf{a}| = 1
\end{aligned} \tag{2.10}$$

As a result, the values representing a frame in a matrix must be such that the above equations remain true. Otherwise, the frame will not be correct. Alternatively, the first three equations in Equation (2.10) can be replaced by a cross product of the three vectors as:

$$\mathbf{n} \times \mathbf{o} = \mathbf{a} \tag{2.11}$$

Since Equation (2.11) includes the correct right-hand-rule relationship too, it is recommended that this equation be used to determine the correct relationship between the three vectors.

Example 2.4

For the following frame, find the values of the missing elements and complete the matrix representation of the frame:

$$F = \begin{bmatrix} ? & 0 & ? & 5 \\ 0.707 & ? & ? & 3 \\ ? & ? & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: Obviously, the 5,3,2 values representing the position of the origin of the frame do not affect the constraint equations. Please notice that only 3 values for directional vectors are given. This is all that is needed. Using Equation (2.10), we will get:

$$\begin{aligned}
&n_x o_x + n_y o_y + n_z o_z = 0 && \text{or} \quad n_x(0) + 0.707(o_y) + n_z(o_z) = 0\\
&n_x a_x + n_y a_y + n_z a_z = 0 && \text{or} \quad n_x(a_x) + 0.707(a_y) + n_z(0) = 0\\
&a_x o_x + a_y o_y + a_z o_z = 0 && \text{or} \quad a_x(0) + a_y(o_y) + 0(o_z) = 0\\
&n_x^2 + n_y^2 + n_z^2 = 1 && \text{or} \quad n_x^2 + 0.707^2 + n_z^2 = 1\\
&o_x^2 + o_y^2 + o_z^2 = 1 && \text{or} \quad 0^2 + o_y^2 + o_z^2 = 1\\
&a_x^2 + a_y^2 + a_z^2 = 1 && \text{or} \quad a_x^2 + a_y^2 + 0^2 = 1
\end{aligned}$$

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

2.6 Representation of Transformations

A transformation is defined as making a movement in space. When a frame (a vector, an object, or a moving frame) moves in space relative to a fixed reference frame, we represent this motion in a form similar to a frame representation. This is because a transformation is a change in the state of a frame (representing the change in its location and orientation); therefore, it can be represented like a frame. A transformation may be in one of the following forms:

- A pure translation
- A pure rotation about an axis
- A combination of translations and/or rotations

In order to see how these can be represented, we will study each one separately.

2.6.1 Representation of a Pure Translation

If a frame (that may also be representing an object) moves in space without any change in its orientation, the transformation is a pure translation. In this case, the directional unit vectors remain in the same direction, and therefore, do not change. The only thing that changes is the location of the origin of the frame relative to the reference frame, as shown in Figure 2.11. The new location of the frame relative to the fixed reference frame can be found by adding the vector representing the translation to the vector representing the original location of the origin of the frame. In matrix form, the new frame representation may be found by pre-multiplying the frame with a matrix representing the transformation. Since the directional vectors do not change in a pure translation, the transformation T will simply be:

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

where d_x, d_y, and d_z are the three components of a pure translation vector $\mathbf{d}$ relative to the x-, y-, and z-axes of the reference frame. The first three columns represent no rotational movement (equivalent of a 1), while the last column represents the translation. The new

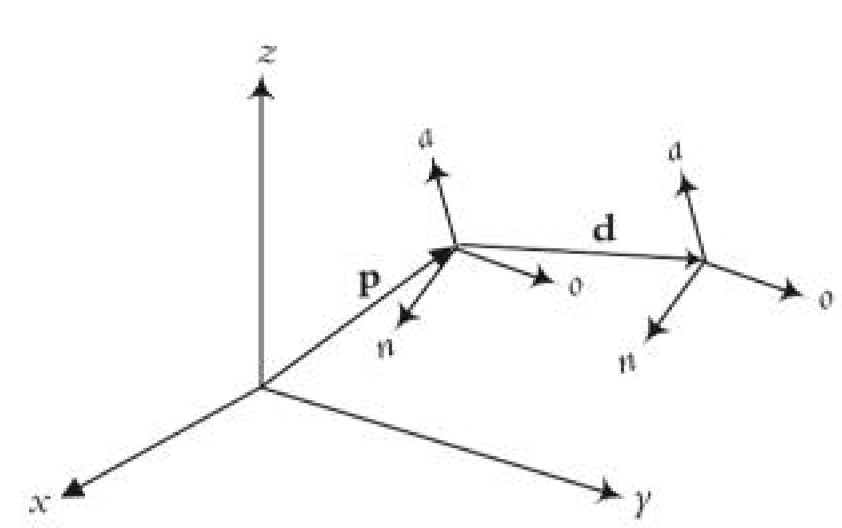


Figure 2.11 Representation of a pure translation in space.

location of the frame will be:

$$F_{new} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x + d_x \\ n_y & o_y & a_y & p_y + d_y \\ n_z & o_z & a_z & p_z + d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

This equation is also symbolically written as:

$$F_{new} = Trans(d_x, d_y, d_z) \times F_{old} \quad (2.16)$$

First, as you can see, pre-multiplying the frame matrix by the transformation matrix will yield the new location of the frame. Second, notice that the directional vectors remain the same after a pure translation, but the new location of the frame is at **d**+**p**. Third, notice how homogeneous transformation matrices facilitate the multiplication of matrices, resulting in the same dimensions as before.

Example 2.6

A frame F has been moved 10 units along the y-axis and 5 units along the z-axis of the reference frame. Find the new location of the frame.

$$F = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 5 \\ 0.369 & 0.819 & 0.439 & 3 \\ -0.766 & 0 & 0.643 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: Using Equation (2.15) or (2.16), we get:

$$F_{new} = Trans(d_x, d_y, d_z) \times F_{old} = Trans(0, 10, 5) \times F_{old}$$

and

$$F_{new} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 10 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.527 & -0.574 & 0.628 & 5 \\ 0.369 & 0.819 & 0.439 & 3 \\ -0.766 & 0 & 0.643 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.527 & -0.574 & 0.628 & 5 \\ 0.369 & 0.819 & 0.439 & 13 \\ -0.766 & 0 & 0.643 & 13 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ■$$

2.6.2 Representation of a Pure Rotation about an Axis

To simplify the derivation of rotations about an axis, let's first assume that the frame is at the origin of the reference frame and is parallel to it. We will later expand the results to other rotations as well as combinations of rotations.

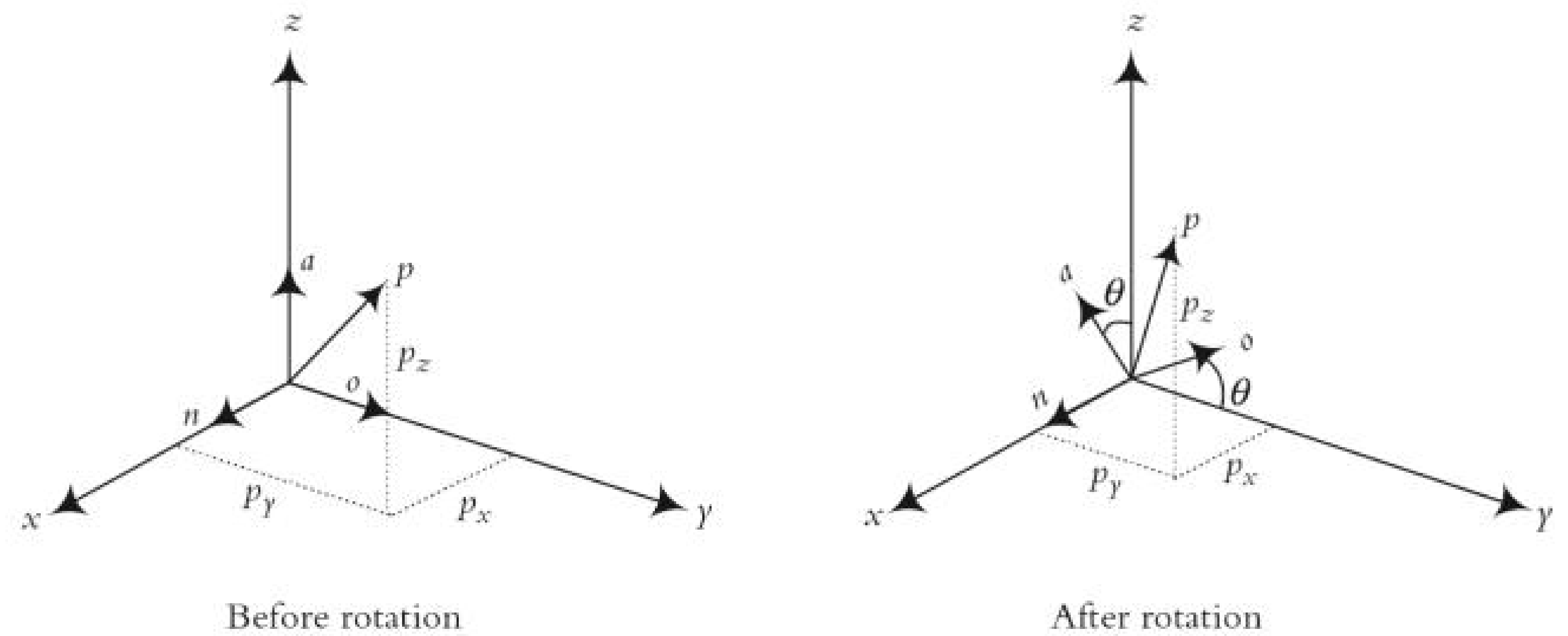


Figure 2.12 Coordinates of a point in a rotating frame before and after rotation.

Let's assume that a frame F_{noa}, located at the origin of the reference frame F_{xyz}, rotates an angle of θ about the x-axis of the reference frame. Let's also assume that attached to the rotating frame F_{noa}, is a point p, with coordinates p_x, p_y, and p_z relative to the reference frame and p_n, p_o, and p_a relative to the moving frame. As the frame rotates about the x-axis, point p attached to the frame will also rotate with it. Before rotation, the coordinates of the point in both frames are the same (remember that the two frames are at the same location and are parallel to each other). After rotation, the p_n, p_o, and p_a coordinates of the point remain the same in the rotating frame F_{noa}, but p_x, p_y, and p_z will be different in the F_{xyz} frame (Figure 2.12). We want to find the new coordinates of the point relative to the fixed reference frame after the moving frame has rotated.

Now let's look at the same coordinates in 2-D as if we were standing on the x-axis. The coordinates of point p are shown before and after rotation in Figure 2.13. The coordinates of point p relative to the reference frame are p_x, p_y, and p_z, while its coordinates relative to the rotating frame (to which the point is attached) remain as p_n, p_o, and p_a.

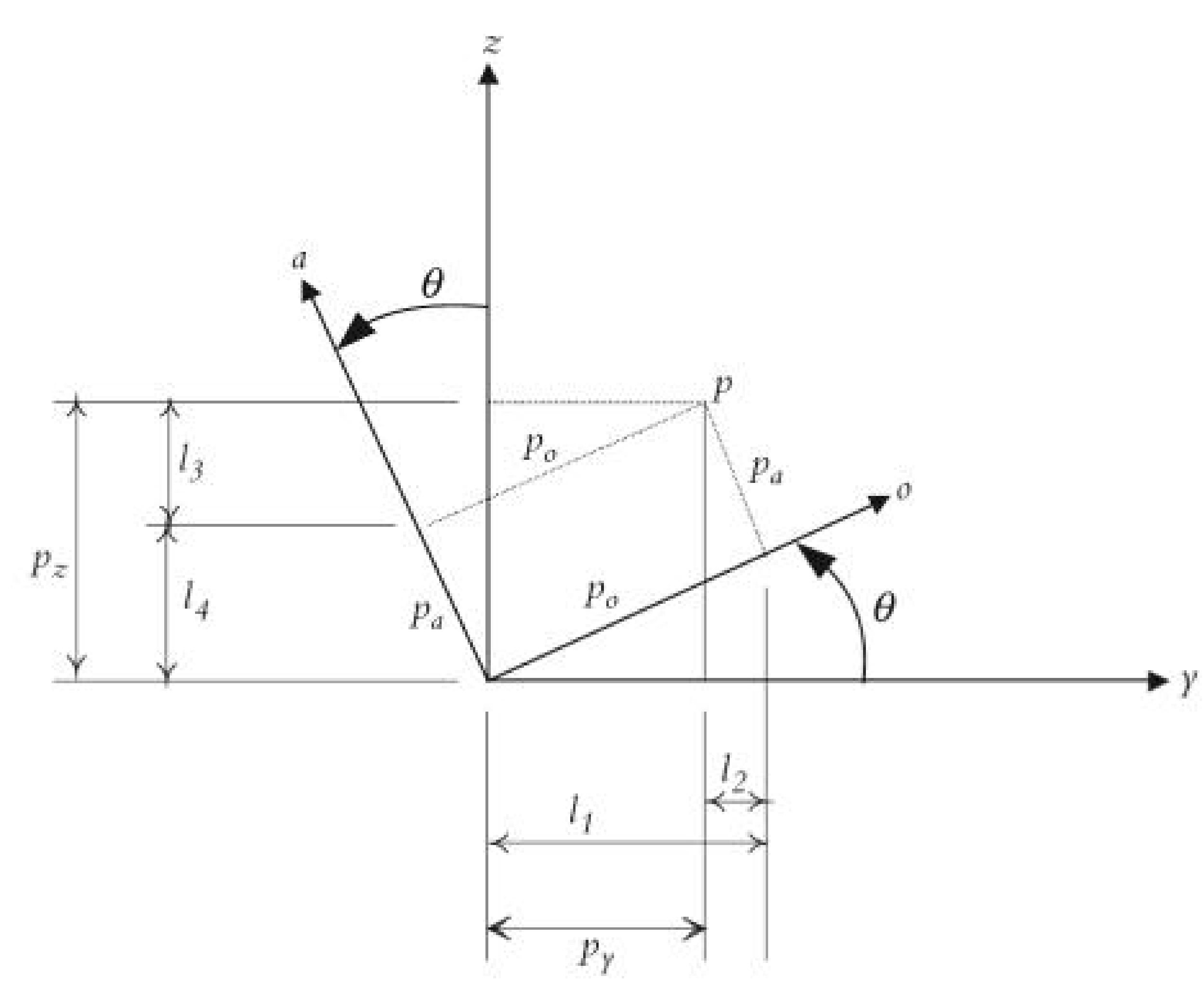


Figure 2.13 Coordinates of a point relative to the reference frame and rotating frame as viewed from the x-axis.

From Figure 2.13, you can see that the value of p_x does not change as the frame rotates about the x-axis, but the values of p_y and p_z do change. Please verify that:

$$\begin{aligned} p_x &= p_n \\ p_y &= l_1 - l_2 = p_o \cos\theta - p_a \sin\theta \\ p_z &= l_3 + l_4 = p_o \sin\theta + p_a \cos\theta \end{aligned} \tag{2.17}$$

and in matrix form:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p_n \\ p_o \\ p_a \end{bmatrix} \tag{2.18}$$

This means that the coordinates of the point p (or vector $\mathbf{p}$) in the rotated frame must be pre-multiplied by the rotation matrix, as shown, to get the coordinates in the reference frame. This rotation matrix is only for a pure rotation about the x-axis of the reference frame and is denoted as:

$$p_{xyz} = Rot(x, \theta) \times p_{noa} \tag{2.19}$$

Notice that the first column of the rotation matrix in Equation (2.18)—which expresses the location relative to the x-axis—has 1,0,0 values, indicating that the coordinate along the x-axis has not changed.

To simplify writing these matrices, it is customary to designate $C\theta$ to denote $\cos\theta$ and $S\theta$ to denote $\sin\theta$. Therefore, the rotation matrix may be also written as:

$$Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \tag{2.20}$$

You may want to do the same for the rotation of a frame about the y- and z-axes of the reference frame. Please verify that the results will be:

$$Rot(y, \theta) = \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \quad \text{and} \quad Rot(z, \theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.21}$$

Equation (2.19) can also be written in a conventional form that assists in easily following the relationship between different frames. Denoting the transformation as ${}^{U}T_R$ (and reading it as the transformation of frame R relative to frame U (for Universe)), denoting p_{noa} as ${}^{R}p$ (p relative to frame R), and denoting p_{xyz} as ${}^{U}p$ (p relative to frame U), Equation (2.19) simplifies to:

$${}^{U}p = {}^{U}T_R \times {}^{R}p \tag{2.22}$$

As you see, canceling the Rs will yield the coordinates of point p relative to U. The same notation will be used throughout this book to relate to multiple transformations.

Example 2.7

A point $p(2,3,4)^T$ is attached to a rotating frame. The frame rotates 90° about the x-axis of the reference frame. Find the coordinates of the point relative to the reference frame after the rotation, and verify the result graphically.

Solution: Of course, since the point is attached to the rotating frame, the coordinates of the point relative to the rotating frame remain the same after the rotation. The coordinates of the point relative to the reference frame will be:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \times \begin{bmatrix} p_n \\ p_o \\ p_a \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \\ 3 \end{bmatrix}$$

As shown in Figure 2.14, the coordinates of point p relative to the reference frame after rotation are 2, -4, 3, as obtained by the above transformation.

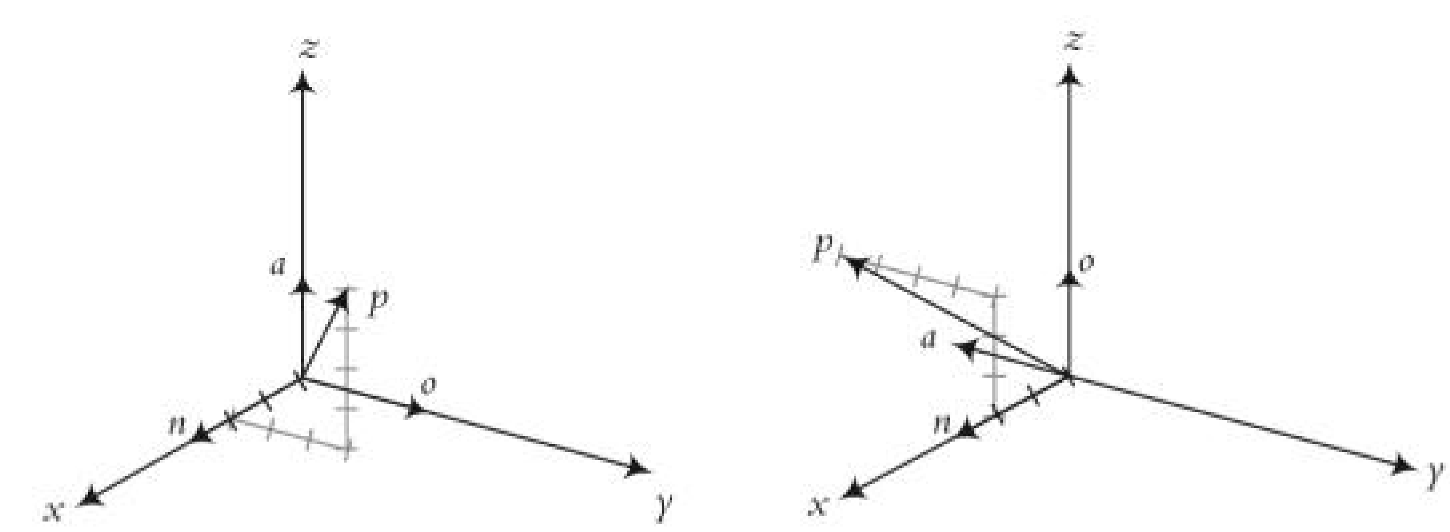


Figure 2.14 Rotation of a frame relative to the x-axis of the reference frame. ■

2.6.3 Representation of Combined Transformations

Combined transformations consist of a number of successive translations and rotations about the fixed reference frame axes or the moving current frame axes. Any transformation can be resolved into a set of translations and rotations in a particular order. For example, we may rotate a frame about the x-axis, then translate about the x-, y-, and z-axes, then rotate about the y-axis in order to accomplish the desired transformation. As we will see later, this order is very important, such that if the order of two successive transformations changes, the result may be completely different.

To see how combined transformations are handled, let's assume that a frame F_{noa} is subjected to the following three successive transformations relative to the reference frame F_{xyz}:

1. Rotation of α degrees about the x-axis,
2. Followed by a translation of $[l_1,l_2,l_3]$ (relative to the x-, y-, and z-axes respectively),
3. Followed by a rotation of β degrees about the y-axis.

Also, let's say that a point p_{noa} is attached to the rotating frame at the origin of the reference frame. As the frame F_{noa} rotates or translates relative to the reference frame, point p within the frame moves as well, and the coordinates of the point relative to the

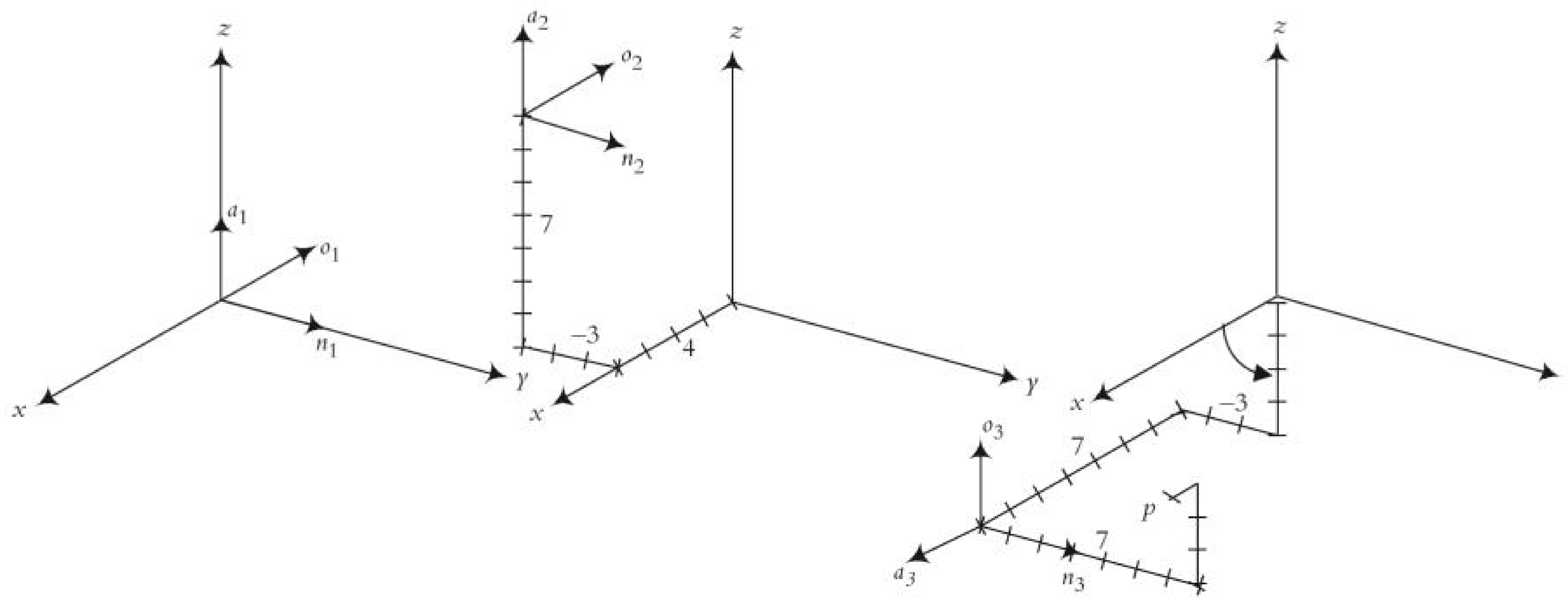


Figure 2.16 Changing the order of transformations will change the final result.

relative to the reference frame are $7 + 1 = 8$, $-3 + 7 = 4$, and $-4 + 3 = -1$, which is the same as the analytical result. ■

2.6.4 Transformations Relative to the Rotating Frame

All transformations we have discussed so far have been relative to the fixed reference frame. This means that all translations, rotations, and distances (except for the location of a point relative to the moving frame) have been measured relative to the reference frame axes. However, it is possible to make transformations relative to the axes of a moving or current frame. This means that, for example, a rotation of 90° may be made relative to the *n*-axis of the moving frame (also referred to as the current frame), and not the *x*-axis of the reference frame. To calculate the changes in the coordinates of a point attached to the current frame relative to the reference frame, the transformation matrix is post-multiplied instead. Note that since the position of a point or an object attached to a moving frame is always measured relative to that moving frame, the position matrix describing the point or object is also always post-multiplied.

Example 2.10

Assume that the same point as in Example 2.9 is now subjected to the same transformations, but all relative to the current moving frame, as listed below. Find the coordinates of the point relative to the reference frame after transformations are completed.

1. A rotation of 90° about the *a*-axis,
2. Then a translation of [4,−3,7] along *n*-, *o*-, *a*-axes
3. Followed by a rotation of 90° about the *o*-axis.

Solution: In this case, since the transformations are made relative to the current frame, each transformation matrix is post-multiplied. As a result, the equation

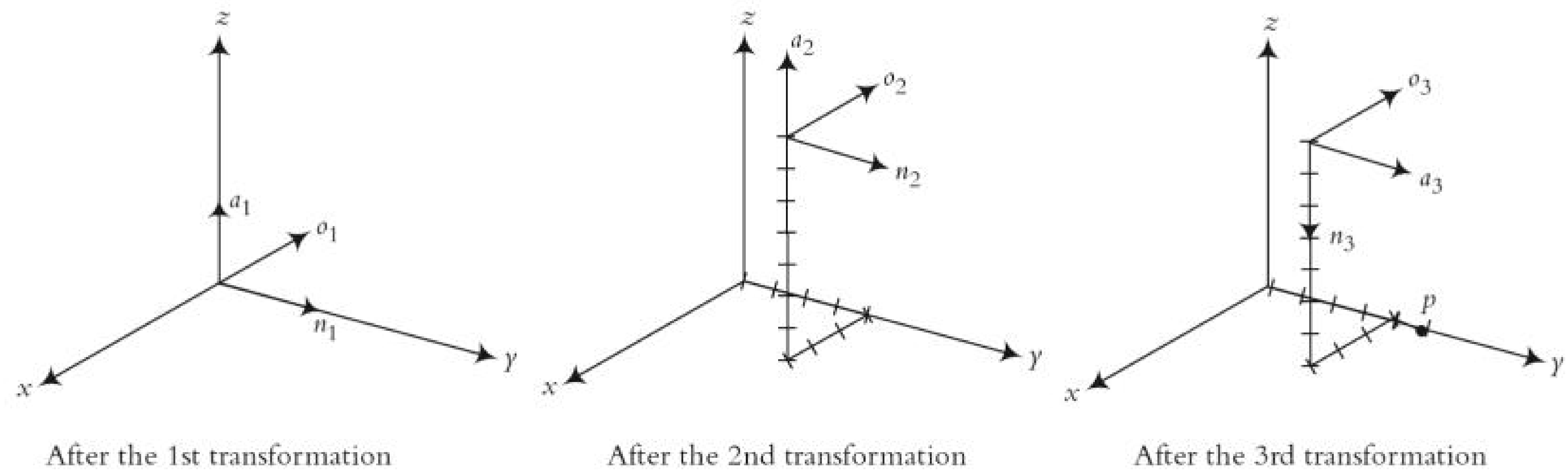


Figure 2.17 Transformations relative to the current frames.

representing the coordinates is:

$$p_{xyz} = Rot(a,90)\,Trans(4,-3,7)\,Rot(o,90)\,p_{noa}$$

$$= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 7 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 0 \\ 1 \end{bmatrix}$$

As expected, the result is completely different from the other cases, both because the transformations are made relative to the current frame, and because the order of the matrices is now different. Figure 2.17 shows the results graphically. Notice how the transformations are accomplished relative to the current frames.

Notice how the 7,3,1 coordinates of point p in the current frame will result in 0,5,0 coordinates relative to the reference frame. ■

Example 2.11

A frame B was rotated about the x-axis 90°, then it was translated about the current a-axis 3 inches before it was rotated about the z-axis 90°. Finally, it was translated about current o–axis 5 inches.

(a) Write an equation that describes the motions.
(b) Find the final location of a point $p(1,5,4)^T$ attached to the frame relative to the reference frame.

Solution: In this case, motions alternate relative to the reference frame and current frame.

(a) Pre- or post-multiplying each motion's matrix accordingly, we will get:

$$^{U}T_B = Rot(z,90)\,Rot(x,90)\,Trans(0,0,3)\,Trans(0,5,0)$$

(b) Substituting the matrices and multiplying them, we will get:

$$^{U}p = {}^{U}T_{B} \times {}^{B}p$$

$$= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 1 \\ 10 \\ 1 \end{bmatrix}$$

■

Example 2.12

A frame F was rotated about the y-axis 90°, followed by a rotation about the o-axis of 30°, followed by a translation of 5 units along the n-axis, and finally, a translation of 4 units along the x-axis. Find the total transformation matrix.

Solution: The following set of matrices, written in the proper order to represent transformations relative to the reference frame or the current frame describes the total transformation:

$$T = Trans(4,0,0)Rot(y,90)Rot(o,30)Trans(5,0,0)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.866 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0 \\ -0.5 & 0 & 0.866 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.5 & 0 & 0.866 & 1.5 \\ 0 & 1 & 0 & 0 \\ -0.866 & 0 & -0.5 & -4.33 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Please verify graphically that this is true. ■

2.7 Inverse of Transformation Matrices

As mentioned earlier, there are many situations where the inverse of a matrix will be needed in robotic analysis. One situation where transformation matrices may be involved can be seen in the following example. Suppose the robot in Figure 2.18 is to be moved toward part P in order to drill a hole in the part. The robot's base position relative to the reference frame U is described by a frame R, the robot's hand is described by frame H, and the end effector (let's say the end of the drill bit that will be used to drill the hole) is described by frame E. The part's position is also described by frame P. The location of the point where the hole will be drilled can be related to the reference frame U through two independent paths: one through the part, one through the robot. Therefore, the

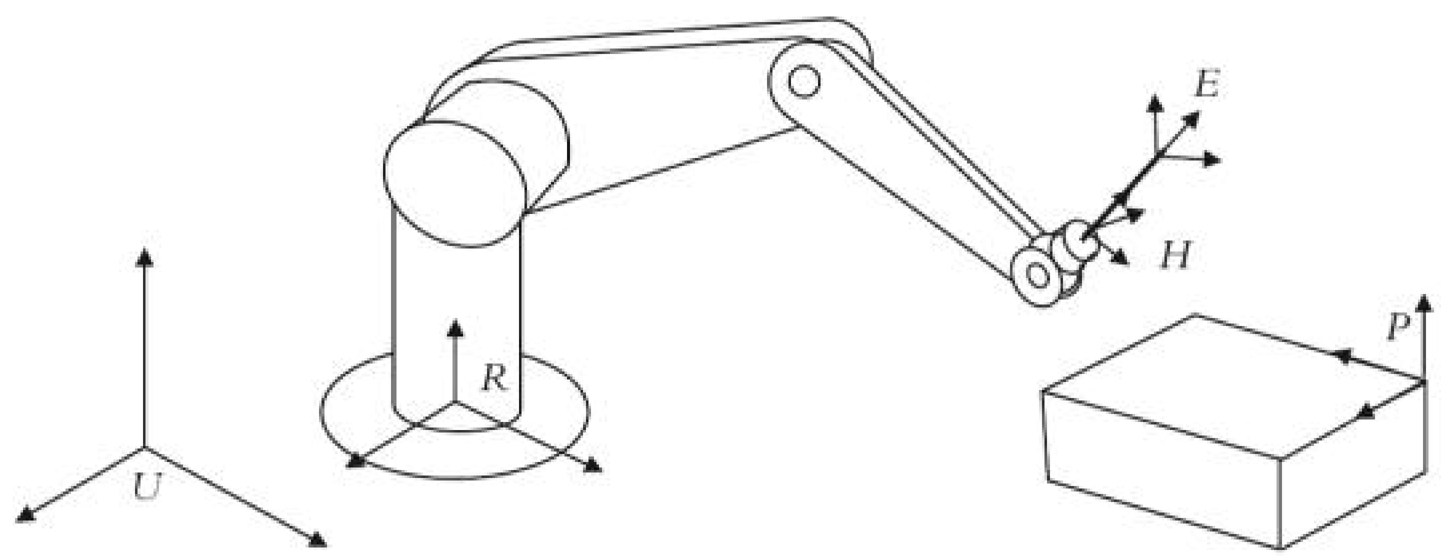


Figure 2.18 The Universe, robot, hand, part, and end effector frames.

following equation can be written:

$$^{U}T_{E} = {}^{U}T_{R}\,{}^{R}T_{H}\,{}^{H}T_{E} = {}^{U}T_{P}\,{}^{P}T_{E} \tag{2.25}$$

The location of point E on the part can be achieved by moving from U to P and from P to E, or it can alternately be achieved by a transformation from U to R, from R to H, and from H to E.

In reality, the transformation of frame R relative to the Universe frame $({}^{U}T_{R})$ is known since the location of the robot's base must be known in any set-up. For example, if a robot is installed in a work cell, the location of the robot's base will be known since it is bolted to a table. Even if the robot is mobile or attached to a conveyor belt, its location at any instant is known because a controller must be following the position of the robot's base at all times. The ${}^{H}T_{E}$, or the transformation of the end effector relative to the robot's hand, is also known since any tool used at the end effector is a known tool and its dimensions and configuration is known. ${}^{U}T_{P}$, or the transformation of the part relative to the universe, is also known since we must know where the part is located if we are to drill a hole in it. This location is known by putting the part in a jig, through the use of a camera and vision system, through the use of a conveyor belt and sensors, or other similar devices. ${}^{P}T_{E}$ is also known since we need to know where the hole is to be drilled on the part. Consequently, the only unknown transformation is ${}^{R}T_{H}$, or the transformation of the robot's hand relative to the robot's base. This means we need to find out what the robot's joint variables—the angle of the revolute joints and the length of the prismatic joints of the robot—must be in order to place the end effector at the hole for drilling. As you can see, it is necessary to calculate this transformation, which will tell us what needs to be accomplished. The transformation will later be used to actually solve for joint angles and link lengths.

To calculate this matrix, unlike in an algebraic equation, we cannot simply divide the right side by the left side of the equation. We need to pre- or post-multiply by inverses of appropriate matrices to eliminate them. As a result, we will have:

$$\left({}^{U}T_{R}\right)^{-1}\left({}^{U}T_{R}\,{}^{R}T_{H}\,{}^{H}T_{E}\right)\left({}^{H}T_{E}\right)^{-1} = \left({}^{U}T_{R}\right)^{-1}\left({}^{U}T_{P}\,{}^{P}T_{E}\right)\left({}^{H}T_{E}\right)^{-1} \tag{2.26}$$

or, since $\left({}^{U}T_{R}\right)^{-1}\left({}^{U}T_{R}\right) = I$ and $\left({}^{H}T_{E}\right)\left({}^{H}T_{E}\right)^{-1} = I$, the left side of Equation (2.26) simplifies to ${}^{R}T_{H}$ and we get:

$$^{R}T_{H} = {}^{U}T_{R}^{-1}\,{}^{U}T_{P}\,{}^{P}T_{E}\,{}^{H}T_{E}^{-1} \tag{2.27}$$

We can check the accuracy of this equation by realizing that $({}^{H}T_{E})^{-1}$ is the same as ${}^{E}T_{H}$. Therefore, the equation can be rewritten as:

$$^{R}T_{H} = {}^{U}T_{R}^{-1}\, {}^{U}T_{P}\, {}^{P}T_{E}\, {}^{H}T_{E}^{-1} = {}^{R}T_{U}\, {}^{U}T_{P}\, {}^{P}T_{E}\, {}^{E}T_{H} = {}^{R}T_{H} \tag{2.28}$$

It is now clear that we need to be able to calculate the inverse of transformation matrices for kinematic analysis as well. In order to see what transpires, let's calculate the inverse of a simple rotation matrix about the x-axis. Please review the process for calculation of square matrices in Appendix A. The rotation matrix about the x-axis is:

$$Rot(x,\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \tag{2.29}$$

Recall that the following steps must be taken to calculate the inverse of a matrix:

- Calculate the determinant of the matrix.
- Transpose the matrix.
- Replace each element of the transposed matrix by its own minor (adjoint matrix).
- Divide the converted matrix by the determinant.

Applying the process to the rotation matrix, we will get:

$$\det[Rot(x,\theta)] = 1(C^2\theta + S^2\theta) + 0 = 1$$
$$Rot(x,\theta)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & S\theta \\ 0 & -S\theta & C\theta \end{bmatrix}$$

Now calculate each minor. As an example, the minor for the 2,2 element will be $C\theta - 0 = C\theta$, the minor for 1,1 element will be $C^2\theta + S^2\theta = 1$, and so on. As you will notice, the minor for each element will be the same as the element itself. Therefore:

$$\text{Adj}[Rot(x,\theta)] = Rot(x,\theta)^T_{minor} = Rot(x,\theta)^T$$

Since the determinant of the original rotation matrix is 1, dividing the Adj[Rot (x, θ)] matrix by the determinant will yield the same result. Consequently, the inverse of a rotation matrix about the x-axis is the same as its transpose, or:

$$Rot(x,\theta)^{-1} = Rot(x,\theta)^T \tag{2.30}$$

Of course, you would get the same result with the second method mentioned in Appendix A. A matrix with this characteristic is called a unitary matrix. It turns out that all rotation matrices are unitary matrices. Therefore, all we need to do to calculate the inverse of a rotation matrix is to transpose it. Please verify that rotation matrices about the

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Substituting the matrices and the inverses in the above equation will result:

$$^E T_{obj} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 3 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

or

$$^E T_{obj} = \begin{bmatrix} -1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & -1 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

■

2.8 Forward and Inverse Kinematics of Robots

Suppose we have a robot whose configuration is known. This means that all the link lengths and joint angles of the robot are known. Calculating the position and orientation of the hand of the robot is called forward kinematic analysis. In other words, if all robot joint variables are known, using forward kinematic equations, we can calculate where the robot is at any instant. However, if we want to place the hand of the robot at a desired location and orientation, we need to know how much each link length or joint angle of the robot must be such that—at those values—the hand will be at the desired position and orientation. This is called inverse kinematic analysis. This means that instead of substituting the known robot variables in the forward kinematic equations of the robot, we need to find the inverse of these equations to enable us to find the necessary joint values to place the robot at the desired location and orientation. In reality, the inverse kinematic equations are more important since the robot controller will calculate the joint values using these equations and it will run the robot to the desired position and orientation. We will first develop the forward kinematic equations of robots; then, using these equations, we will calculate the inverse kinematic equations.

For forward kinematics, we will have to develop a set of equations that relate to the particular configuration of a robot (the way it is put together) such that by substituting the joint and link variables in these equations, we may calculate the position and orientation of the robot. These equations will then be used to derive the inverse kinematic equations.

You may recall from Chapter 1 that in order to position and orientate a rigid body in space, we attach a frame to the body and then describe the position of the origin of the frame and the orientation of its three axes. This requires a total of 6 DOF, or alternately, six pieces of information, to completely define the position and orientation of the body. Here too, if we want to define or find the position and orientation of the hand of the robot in space, we will attach a frame to it and define the position and orientation of the hand frame of the robot. The means by which the robot accomplishes this determines the forward kinematic equations. In other words, depending on the configuration of the links and joints of the robot, a particular set of equations will relate the hand frame of the robot to the reference frame. Figure 2.19 shows a hand frame, the reference frame, and their relative positions and orientations. The undefined connection between the two frames is related to the configuration of the robot. Of

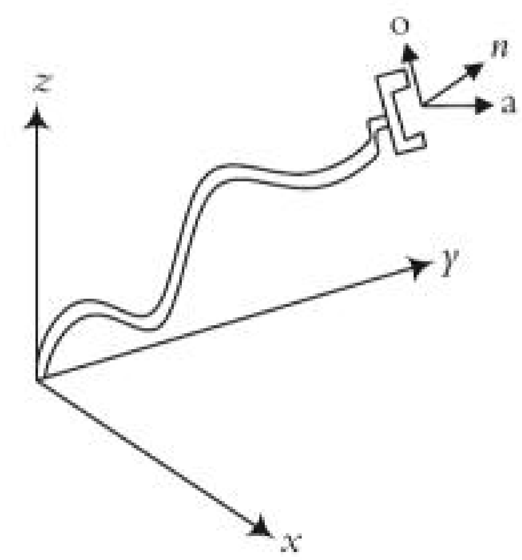


Figure 2.19 The hand frame of the robot relative to the reference frame.

course, there are many different possibilities for this configuration, and we will later see how we can develop the equations relating the two frames, depending on the robot configuration.

In order to simplify the process, we will analyze the position and orientation issues separately. First, we will develop the position equations, then we will do the same for orientation. Later, we will combine the two for a complete set of equations. Finally, we will see about the use of the Denavit-Hartenberg representation, which can model any robot configuration.

2.9 Forward and Inverse Kinematic Equations: Position

In this section, we will study the forward and inverse kinematic equations for position. As was mentioned earlier, the position of the origin of a frame attached to a rigid body has three degrees of freedom, and therefore, can be completely defined by three pieces of information. As a result, the position of the origin of the frame may be defined in any customary coordinates. As an example, we may position a point in space based on Cartesian coordinates, meaning there will be three linear movements relative to the x-, y-, and z-axes. Alternately, it may be accomplished through spherical coordinates, meaning there will be one linear motion and two rotary motions. The following possibilities will be discussed:

(a) Cartesian (gantry, rectangular) coordinates
(b) Cylindrical coordinates
(c) Spherical coordinates
(d) Articulated (anthropomorphic or all-revolute) coordinates

2.9.1 Cartesian (Gantry, Rectangular) Coordinates

In this case, there will be three linear movements along the x-, y-, and z-axes. In this type of robot, all actuators are linear (such as a hydraulic ram or a linear power screw), and the positioning of the hand of the robot is accomplished by moving the three linear joints along the three axes (Figure 2.20). A gantry robot is basically a Cartesian coordinate robot, except that the robot is usually attached to a rectangular frame upside down.

Of course, since there are no rotations, the transformation matrix representing this motion to point p is a simple translation matrix (shown next). Note that here we are only

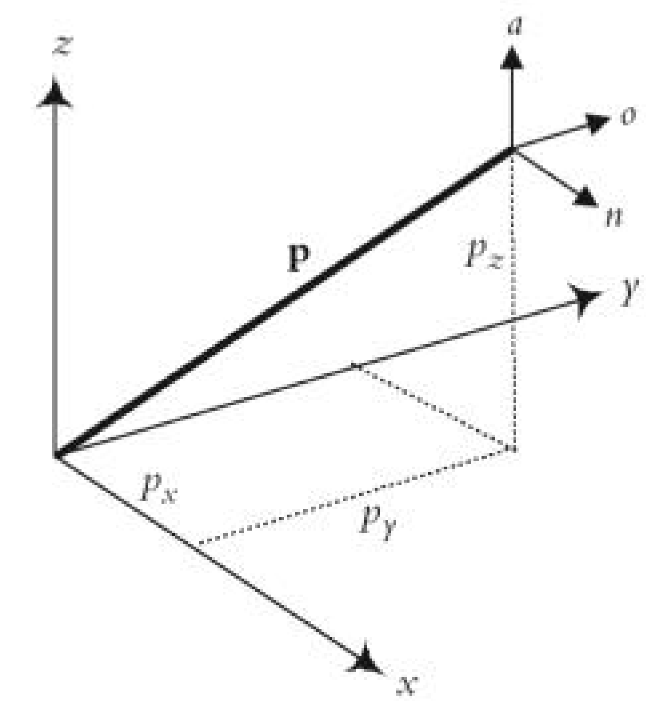


Figure 2.20 Cartesian coordinates.

concerned with the position of the origin of the frame—not its orientation. The transformation matrix representing the forward kinematic equation of the position of the hand of the robot in a Cartesian coordinate system will be:

$$^{R}T_{p} = T_{cart}(p_x, p_y, p_z) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.32}$$

where $^{R}T_{p}$ is the transformation between the reference frame and the origin of the hand p, and $T_{cart}(p_x, p_y, p_z)$ denotes Cartesian transformation matrix. For the inverse kinematic solution, simply set the desired position equal to p.

Example 2.16

It is desired to position the origin of the hand frame of a Cartesian robot at point $p = [3,4,7]^T$. Calculate the necessary Cartesian coordinate motions that need to be made.

Solution: Setting the forward kinematic equation, represented by the $^{R}T_{p}$ matrix of Equation (2.32), equal to the desired position will yield the following result:

$$^{R}T_{p} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad p_x = 3,\ p_y = 4,\ p_z = 7$$

■

2.9.2 Cylindrical Coordinates

A cylindrical coordinate system includes two linear translations and one rotation. The sequence is a translation of r along the x-axis, a rotation of α about the z-axis, and a

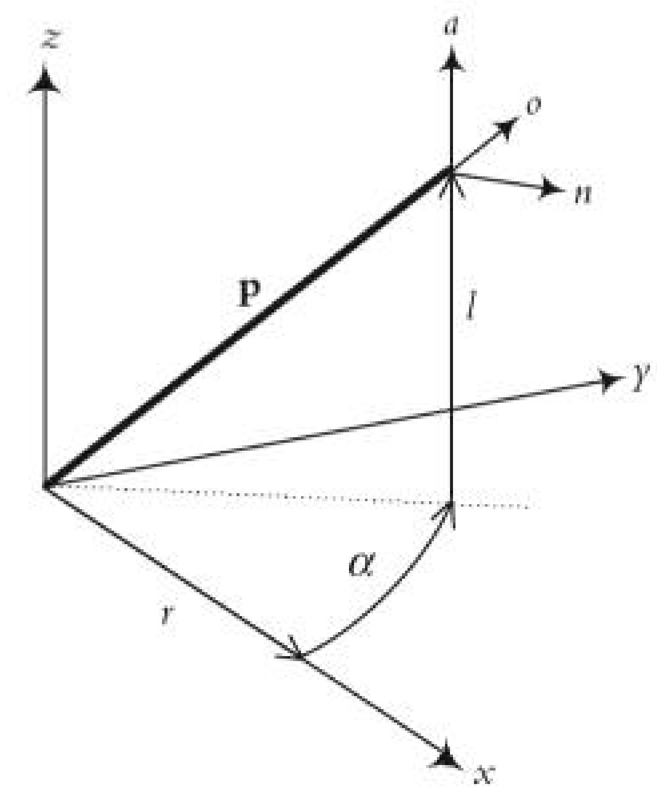


Figure 2.21 Cylindrical coordinates.

translation of l along the z-axis, as shown in Figure 2.21. Since these transformations are all relative to the Universe frame, the total transformation caused by these three transformations is found by pre-multiplying by each matrix, as follows:

$$^{R}T_{p} = T_{cyl}(r, \alpha, l) = Trans(0, 0, l)Rot(z, \alpha)Trans(r, 0, 0) \tag{2.33}$$

$$^{R}T_{p} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} C\alpha & -S\alpha & 0 & 0 \\ S\alpha & C\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{R}T_{p} = T_{cyl}(r, \alpha, l) = \begin{bmatrix} C\alpha & -S\alpha & 0 & rC\alpha \\ S\alpha & C\alpha & 0 & rS\alpha \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.34}$$

The first three columns represent the orientation of the frame after this series of transformations. However, at this point, we are only interested in the position of the origin of the frame, or the last column. Obviously, in cylindrical coordinate movements, due to the rotation of α about the z-axis, the orientation of the moving frame will change. This orientation change will be discussed later.

You may restore the original orientation of the frame by rotating the n,o,a frame about the a-axis an angle of $-\alpha$, which is equivalent of post-multiplying the cylindrical coordinate matrix by a rotation matrix of $Rot(a, -\alpha)$. As a result, the frame will be at the same location but will be parallel to the reference frame again, as follows:

$$T_{cyl} \times Rot(a, -\alpha) = \begin{bmatrix} C\alpha & -S\alpha & 0 & rC\alpha \\ S\alpha & C\alpha & 0 & rS\alpha \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} C(-\alpha) & -S(-\alpha) & 0 & 0 \\ S(-\alpha) & C(-\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & rC\alpha \\ 0 & 1 & 0 & rS\alpha \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As you can see, the location of the origin of the moving frame has not changed, but it was restored back to being parallel to the reference frame. Notice that the last rotation was performed about the local a-axis in order to not cause any change in the location of the frame, but only in its orientation.

Example 2.17

Suppose we desire to place the origin of the hand frame of a cylindrical robot at $[3,4,7]^T$. Calculate the joint variables of the robot.

Solution: Setting the components of the location of the origin of the frame from the T_{cyl} matrix of Equation (2.34) to the desired values, we get:

$l = 7$

$rC\alpha = 3$ and $rS\alpha = 4$ and therefore, $\tan\alpha = {}^4\!/_3$ and $\alpha = 53.1^\circ$

Substituting α into either equation will yield $r = 5$. The final answer is $r = 5$ units, $\alpha = 53.1^\circ$, and $l = 7$ units. Note: As discussed in Appendix A, it is necessary to ensure that the angles calculated in robot kinematics are in correct quadrants. In this example, $rC\alpha$ and $rS\alpha$ are both positive and the length r is always positive, therefore $S\alpha$ and $C\alpha$ are also both positive. Consequently, the angle α is in quadrant 1 and is correctly 53.1°. ■

Example 2.18

The position and restored orientation of a cylindrical robot are given. Find the matrix representing the original position and orientation of the robot before it was restored.

$$T = \begin{bmatrix} 1 & 0 & 0 & -2.394 \\ 0 & 1 & 0 & 6.578 \\ 0 & 0 & 1 & 9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: Since r is always positive, it is clear that $S\alpha$ and $C\alpha$ are positive and negative, respectively. Therefore, α is in the second quadrant. From T, we get:

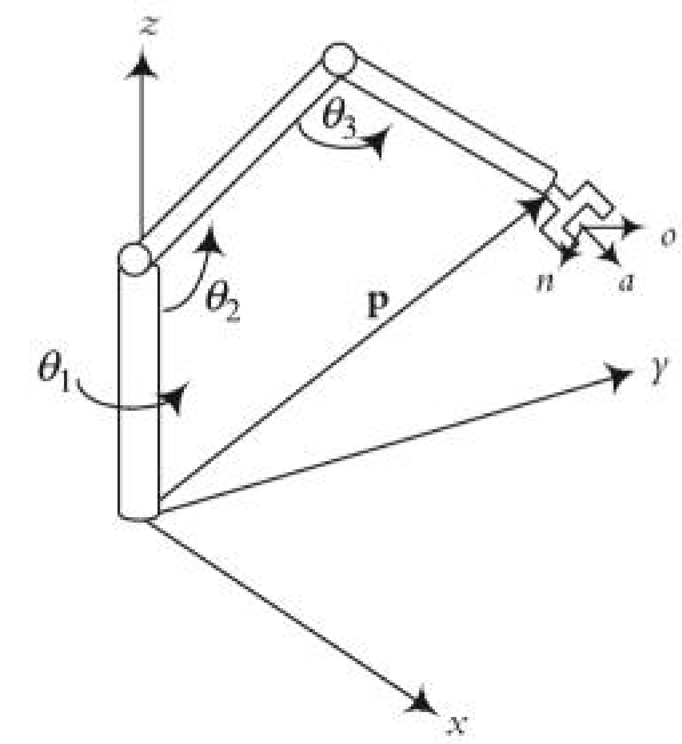


Figure 2.23 Articulated coordinates.

2.9.4 Articulated Coordinates

Articulated coordinates consist of three rotations, as shown in Figure 2.23. We will develop the matrix representation for this later, when we discuss the Denavit-Hartenberg representation.

2.10 Forward and Inverse Kinematic Equations: Orientation

Suppose the moving frame attached to the hand of the robot has already moved to a desired position—in Cartesian, cylindrical, spherical, or articulated coordinates—and is either parallel to the reference frame or is in an orientation other than what is desired. The next step will be to rotate the frame appropriately in order to achieve a desired orientation without changing its position. This can only be accomplished by rotating about the current frame axes; rotations about the reference frame axes will change the position. The appropriate sequence of rotations depends on the design of the wrist of the robot and the way the joints are assembled together. We will consider the following three common configurations:

(a) Roll, Pitch, Yaw (RPY) angles
(b) Euler angles
(c) Articulated joints

2.10.1 Roll, Pitch, Yaw (RPY) Angles

This is a sequence of three rotations about current *a*-, *o*-, and *n*-axes respectively, which will orientate the hand of the robot to a desired orientation. The assumption here is that the current frame is parallel to the reference frame; therefore, its orientation is the same as the reference frame before the application of RPY. If the current moving frame is not parallel to the reference frame, then the final orientation of the robot's hand will be a combination of the previous orientation, post-multiplied by the RPY.

It is very important to realize that since we do not want to cause any change in the position of the origin of the moving frame (we have already placed it at the desired

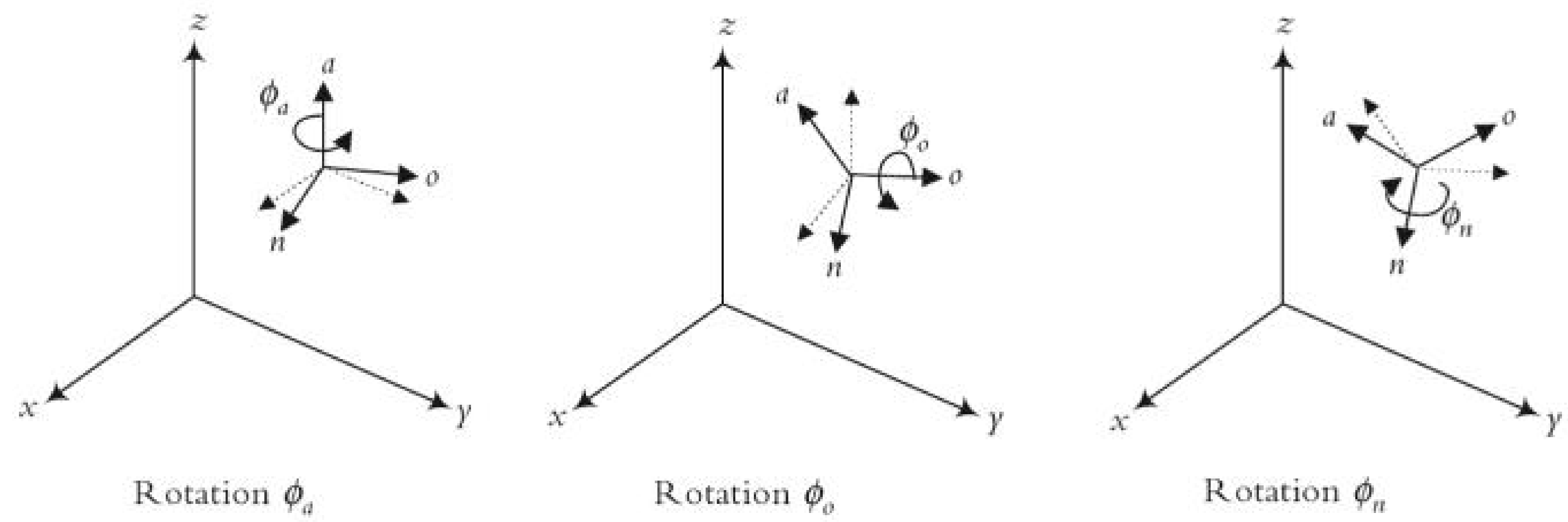


Figure 2.24 RPY rotations about the current axes.

location and only want to rotate it to the desired orientation), the movements relating to RPY rotations are relative to the current moving axes. Otherwise, as we saw before, the position of the frame will change. Therefore, all matrices related to the orientation change due to RPY (as well as other rotations) will be post-multiplied. Referring to Figure 2.24, the RPY sequence of rotations consists of:

Rotation of ϕ_a about the *a*-axis (*z*-axis of the moving frame) called Roll,
Rotation of ϕ_o about the *o*-axis (*y*-axis of the moving frame) called Pitch,
Rotation of ϕ_n about the *n*-axis (*x*-axis of the moving frame) called Yaw.

The matrix representing the RPY orientation change will be:

$$\mathrm{RPY}(\phi_a,\phi_o,\phi_n) = Rot(a,\phi_a)Rot(o,\phi_o)Rot(n,\phi_n)$$

$$= \begin{bmatrix} C\phi_a C\phi_o & C\phi_a S\phi_o S\phi_n - S\phi_a C\phi_n & C\phi_a S\phi_o C\phi_n + S\phi_a S\phi_n & 0 \\ S\phi_a C\phi_o & S\phi_a S\phi_o S\phi_n + C\phi_a C\phi_n & S\phi_a S\phi_o C\phi_n - C\phi_a S\phi_n & 0 \\ -S\phi_o & C\phi_o S\phi_n & C\phi_o C\phi_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.37}$$

This matrix represents the orientation change caused by the RPY alone. The location and the final orientation of the frame relative to the reference frame will be the product of the two matrices representing the position change and the RPY. For example, suppose that a robot is designed based on spherical coordinates and RPY. Then the robot may be represented by:

$$^{R}T_H = T_{sph}(r,\beta,\gamma) \times \mathrm{RPY}(\phi_a,\phi_o,\phi_n)$$

The inverse kinematic solution for the RPY is more complicated than the spherical coordinates because here there are three coupled angles, where we need to have information about the sines and the cosines of all three angles individually to solve for the angles. To solve for these sines and cosines, we will have to de-couple these angles. To do this, we will pre-multiply both sides of Equation (2.37) by the inverse of $Rot(a,\phi_a)$:

$$Rot(a,\phi_a)^{-1}\,\mathrm{RPY}(\phi_a,\phi_o,\phi_n) = Rot(o,\phi_o)Rot(n,\phi_n) \tag{2.38}$$

Assuming that the final desired orientation achieved by RPY is represented by the (n,o,a) matrix, we will have:

$$Rot(a,\phi_a)^{-1}\begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = Rot(o,\phi_o)Rot(n,\phi_n) \tag{2.39}$$

Multiplying the matrices, we will get:

$$\begin{bmatrix} n_xC\phi_a + n_yS\phi_a & o_xC\phi_a + o_yS\phi_a & a_xC\phi_a + a_yS\phi_a & 0 \\ n_yC\phi_a - n_xS\phi_a & o_yC\phi_a - o_xS\phi_a & a_yC\phi_a - a_xS\phi_a & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} C\phi_o & S\phi_oS\phi_n & S\phi_oC\phi_n & 0 \\ 0 & C\phi_n & -S\phi_n & 0 \\ -S\phi_o & C\phi_oS\phi_n & C\phi_oC\phi_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.40}$$

Remember that the n,o,a components in Equation (2.39) represent the final desired values normally given or known. The values of the RPY angles are the unknown variables. Equating the different elements of the right-hand and left-hand sides of Equation (2.40) will result in the following. Refer to Appendix A for an explanation of $ATAN2$ function.

From the 2,1 elements we get:

$$n_yC\phi_a - n_xS\phi_a = 0 \rightarrow \phi_a = ATAN2(n_y, n_x) \text{ and } \phi_a = ATAN2(-n_y, -n_x) \tag{2.41}$$

Note that since we do not know the signs of $\sin(\phi_a)$ or $\cos(\phi_a)$, two complementary solutions are possible. From the 3,1 and 1,1 elements we get:

$$\begin{aligned} S\phi_o &= -n_z \\ C\phi_o &= n_xC\phi_a + n_yS\phi_a \rightarrow \phi_o = ATAN2[-n_z, (n_xC\phi_a + n_yS\phi_a)] \end{aligned} \tag{2.42}$$

And finally, from the 2,2 and 2,3 elements we get:

$$\begin{aligned} C\phi_n &= o_yC\phi_a - o_xS\phi_a \\ S\phi_n &= -a_yC\phi_a + a_xS\phi_a \rightarrow \phi_n = ATAN2[(-a_yC\phi_a + a_xS\phi_a), (o_yC\phi_a - o_xS\phi_a)] \end{aligned} \tag{2.43}$$

Example 2.20

The desired final position and orientation of the hand of a Cartesian-RPY robot is given below. Find the necessary RPY angles and displacements.

$$^{R}T_{P} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.354 & -0.674 & 0.649 & 4.33 \\ 0.505 & 0.722 & 0.475 & 2.50 \\ -0.788 & 0.160 & 0.595 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: From the above equations, we find two sets of answers:

$$\begin{aligned} \phi_a &= ATAN2(n_y, n_x) = ATAN2(0.505, 0.354) = 55^\circ \text{ or } 235^\circ \\ \phi_o &= ATAN2(-n_z, (n_x C\phi_a + n_y S\phi_a)) = ATAN2(0.788, 0.616) = 52^\circ \text{ or } 128^\circ \\ \phi_n &= ATAN2((-a_y C\phi_a + a_x S\phi_a), (o_y C\phi_a - o_x S\phi_a)) \\ &= ATAN2(0.259, 0.966) = 15^\circ \text{ or } 195^\circ \\ p_x &= 4.33 \quad p_y = 2.5 \quad p_z = 8 \text{ units.} \end{aligned}$$

■

Example 2.21

For the same position and orientation as in Example 2.20, find all necessary joint variables if the robot is cylindrical-RPY.

Solution: In this case, we will use:

$$^{R}T_{P} = \begin{bmatrix} 0.354 & -0.674 & 0.649 & 4.33 \\ 0.505 & 0.722 & 0.475 & 2.50 \\ -0.788 & 0.160 & 0.595 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_{cyl}(r, \alpha, l) \times \mathrm{RPY}(\phi_a, \phi_o, \phi_n)$$

The right-hand side of this equation now involves four coupled angles; as before, these must be de-coupled. However, since the rotation of α about the z-axis for the cylindrical coordinates does not affect the a-axis, it remains parallel to the z-axis. As a result, the rotation of ϕ_a about the a-axis for RPY will simply be added to α. This means that the 55° angle we found for ϕ_a is the summation of $\phi_a + \alpha$ (see Figure 2.25). Using the position information given, the solution of Example 2.20, and referring to Equation (2.34), we get:

$$\begin{aligned} &rC\alpha = 4.33, \quad rS\alpha = 2.5 && \rightarrow \alpha = 30^\circ \\ &\phi_a + \alpha = 55^\circ && \rightarrow \phi_a = 25^\circ \\ &S\alpha = 0.5 && \rightarrow r = 5 \\ &p_z = 8 && \rightarrow l = 8 \end{aligned}$$

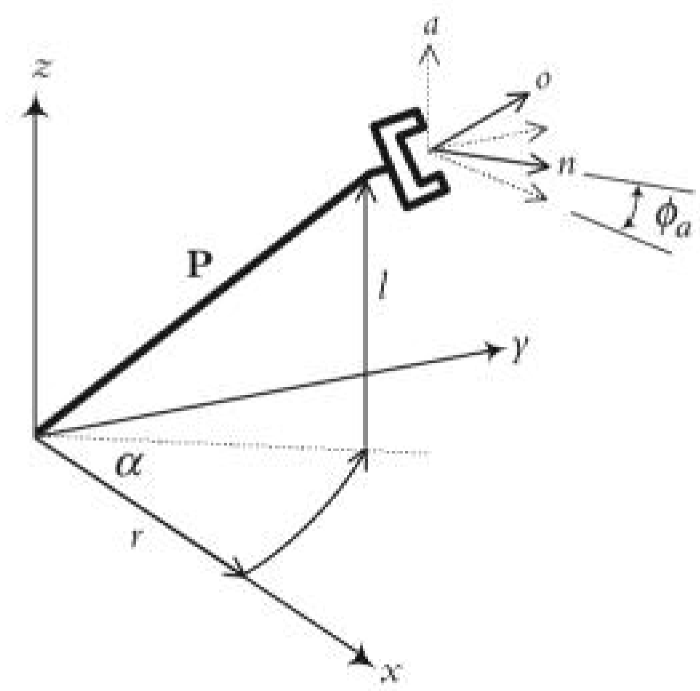


Figure 2.25 Cylindrical and RPY coordinates of Example 2.21.

As in Example 2.16:

$$\rightarrow \phi_o = 52°, \quad \phi_n = 15°$$

Of course, a similar solution may be found for the second set of answers. ■

2.10.2 Euler Angles

Euler angles are very similar to RPY, except that the last rotation is also about the current *a*-axis (Figure 2.26). We still need to make all rotations relative to the current axes to prevent any change in the position of the robot. Therefore, the rotations representing the Euler angles will be:

Rotation of ϕ about the *a*-axis (*z*-axis of the moving frame) followed by,
Rotation of θ about the *o*-axis (*y*-axis of the moving frame) followed by,
Rotation of ψ about the *a*-axis (*z*-axis of the moving frame).

The matrix representing the Euler angles orientation change will be:

$$\text{Euler}(\phi,\theta,\psi) = Rot(a,\phi)Rot(o,\theta), Rot(a,\psi)$$

$$= \begin{bmatrix} C\phi C\theta C\psi - S\phi S\psi & -C\phi C\theta S\psi - S\phi C\psi & C\phi S\theta & 0 \\ S\phi C\theta C\psi + C\phi S\psi & -S\phi C\theta S\psi + C\phi C\psi & S\phi S\theta & 0 \\ -S\theta C\psi & S\theta S\psi & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.44}$$

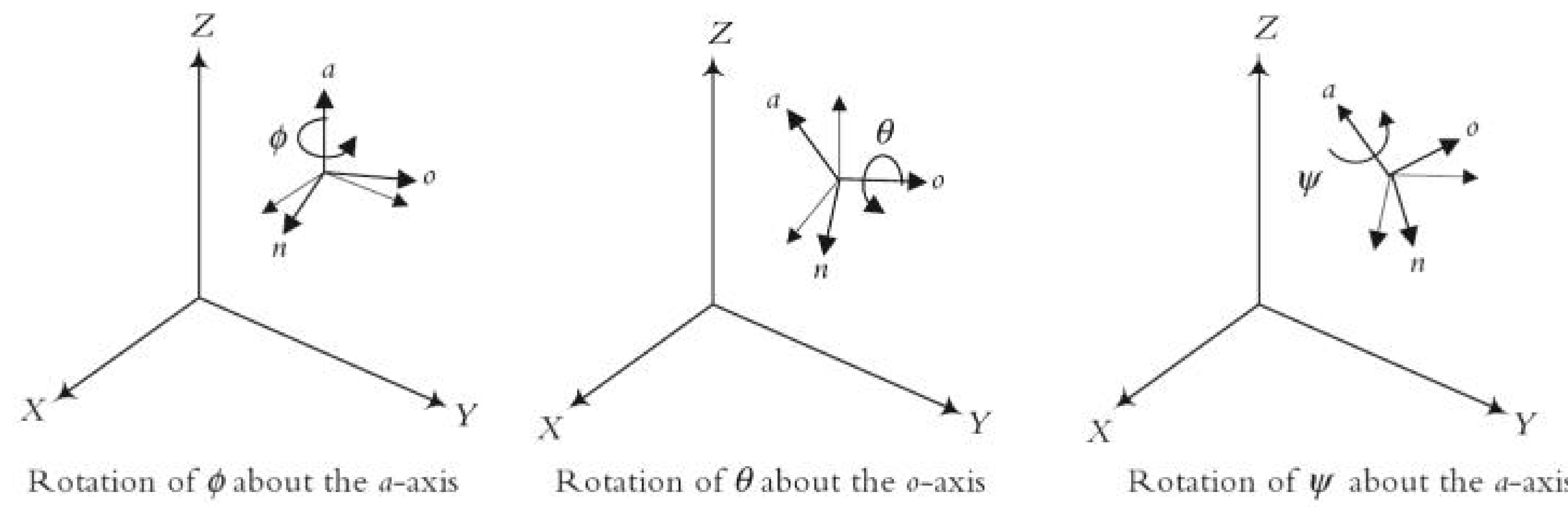


Figure 2.26 Euler rotations about the current axes.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

determined by the spherical coordinates, while the final orientation is affected by both the angles in the spherical coordinates as well as the Euler angles:

$$^{R}T_{H} = T_{sph}(r, \beta, \gamma) \times \text{Euler}(\phi, \theta, \psi) \tag{2.51}$$

The forward and inverse kinematic solutions for these cases are not developed here, since many different combinations are possible. Instead, in complicated designs, the Denavit-Hartenberg representation is recommended. We will discuss this next.

2.12 Denavit-Hartenberg Representation of Forward Kinematic Equations of Robots

In 1955, Denavit and Hartenberg[4] published a paper in the *ASME Journal of Applied Mechanics* that was later used to represent and model robots and to derive their equations of motion. This technique has become the standard way of representing robots and modeling their motions, and therefore, is essential to learn. The Denavit-Hartenberg (D-H) model of representation is a very simple way of modeling robot links and joints that can be used for any robot configuration, regardless of its sequence or complexity. It can also be used to represent transformations in any coordinates we have already discussed, such as Cartesian, cylindrical, spherical, Euler, and RPY. Additionally, it can be used for representation of all-revolute articulated robots, SCARA robots, or any possible combinations of joints and links. Although the direct modeling of robots with the previous techniques are faster and more straightforward, the D-H representation has an added benefit; as we will see later, analysis of differential motions and Jacobians, dynamic analysis, force analysis, and others are based on the results obtained from D-H representation.[5–9]

Robots may be made of a succession of joints and links in any order. The joints may be either prismatic (linear) or revolute (rotational), move in different planes, and have offsets. The links may also be of any length, including zero; may be twisted and bent; and may be in any plane. Therefore, any general set of joints and links may create a robot. We need to be able to model and analyze any robot, whether or not it follows any of the preceding coordinates.

To do this, we assign a reference frame to each joint, and later define a general procedure to transform from one joint to the next (one frame to the next). If we combine all the transformations from the base to the first joint, from the first joint to the second joint, and so on, until we get to the last joint, we will have the robot's total transformation matrix. In the following sections, we will define the general procedure, based on the D-H representation, to assign reference frames to each joint. Then we will define how a transformation between any two successive frames may be accomplished. Finally, we will write the total transformation matrix for the robot.

Imagine that a robot may be made of a number of links and joints in any form. Figure 2.27 represents three successive joints and two links. Although these joints and links are not necessarily similar to any real robot joint or link, they are very general and can easily represent any joints in real robots. These joints may be revolute or prismatic, or both. Although in real robots it is customary to only have 1-DOF joints, the joints in Figure 2.27 represent 1- or 2-DOF joints.

Figure 2.27(a) shows three joints. Each joint may both rotate and/or translate. Let's assign joint number n to the first joint, $n+1$ to the second joint, and $n+2$ to the third

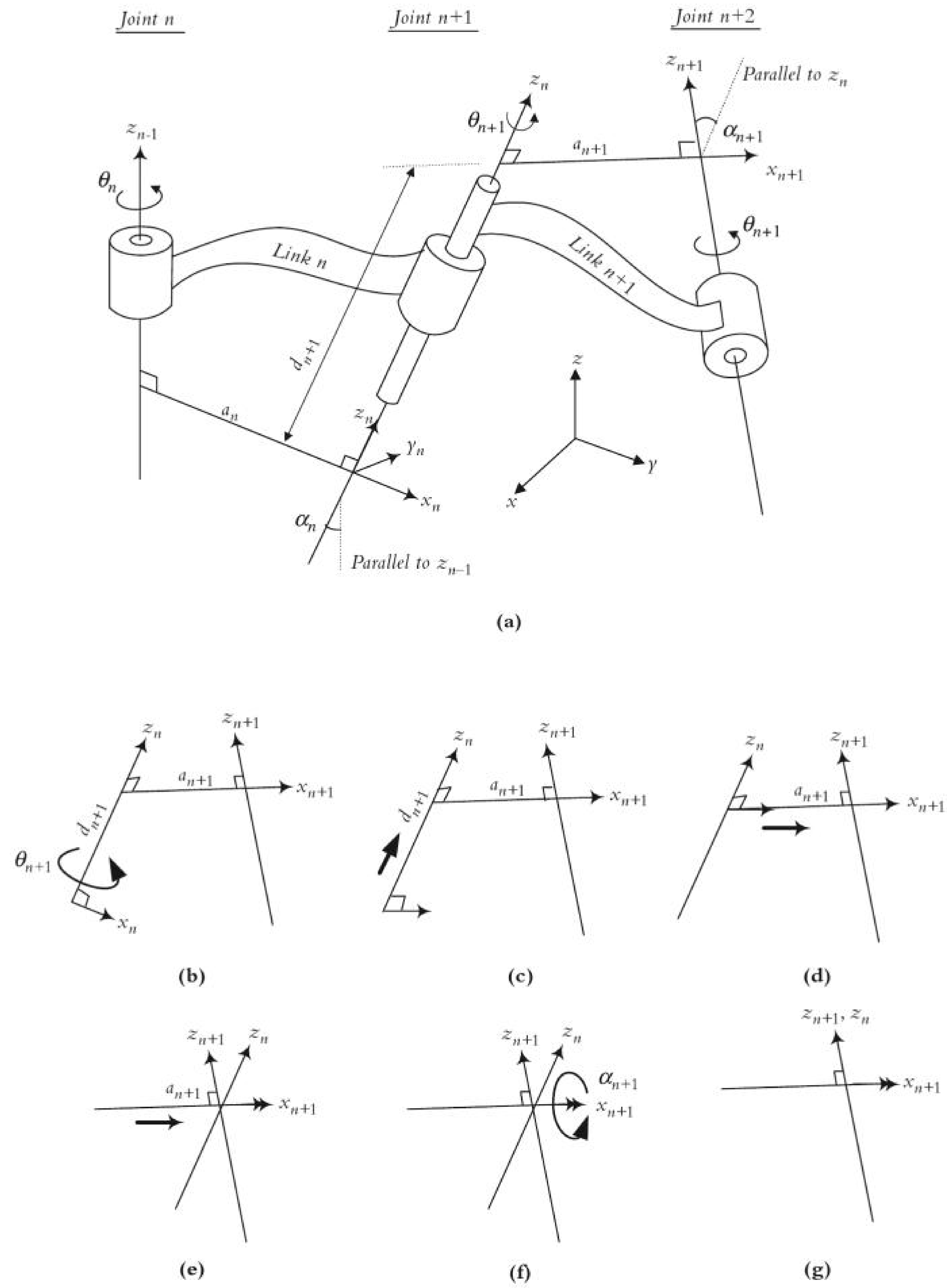


Figure 2.27 A Denavit-Hartenberg representation of a general purpose joint-link combination.

joint shown. There may be other joints before or after these. Each link is also assigned a link number as shown. Link n will be between joints n and $n+1$, and link $n+1$ is between joints $n+1$ and $n+2$.

To model the robot with the D-H representation, the first thing we need to do is assign a local reference frame for each and every joint. Therefore, for each joint, we will have to assign a z-axis and an x-axis. We normally do not need to assign a y-axis, since we always know that y-axes are mutually perpendicular to both x- and z-axes. In addition, the D-H representation does not use the y-axis at all. The following is the procedure for assigning a local reference frame to each joint.

- All joints, without exception, are represented by a z-axis. If the joint is revolute, the z-axis is in the direction of rotation as followed by the right-hand rule for rotations. If the joint is prismatic, the z-axis for the joint is along the direction of the linear movement. In each case, the index number for the z-axis of joint n (as well as the local reference frame for the joint) is $n-1$. For example, the z-axis representing motions about joint number $n+1$ is z_n. These simple rules will allow us to quickly assign z-axes to all joints. For revolute joints, the rotation about the z-axis (θ) will be the joint variable. For prismatic joints, the length of the link along the z-axis represented by d will be the joint variable.
- As you can see in Figure 2.27(a), in general, joints may not necessarily be parallel or intersecting. As a result, the z-axes may be skew lines. There is always one line mutually perpendicular to any two skew lines, called common normal, which is the shortest distance between them. We always assign the x-axis of the local reference frame in the direction of the common normal. Therefore, if a_n represents the common normal between z_{n-1} and z_n, the direction of x_n will be along a_n. Similarly, if the common normal between z_n and z_{n+1} is a_{n+1}, the direction of x_{n+1} will be along a_{n+1}. The common normal lines between successive joints are not necessarily intersecting or colinear. As a result, the origins of two successive frames may also not be at the same location. Based on the above, we can assign coordinate frames to all joints, with the following exceptions:
  - If two z-axes are parallel, there are an infinite number of common normals between them. We will pick the common normal that is colinear with the common normal of the previous joint. This will simplify the model.
  - If the z-axes of two successive joints are intersecting, there is no common normal between them (or it has a zero length). We will assign the x-axis along a line perpendicular to the plane formed by the two axes. This means that the common normal is a line perpendicular to the plane containing the two z-axes, which is the equivalent of picking the direction of the cross-product of the two z-axes. This also simplifies the model.

In Figure 2.27(a), θ represents a rotation about the z-axis, d represents the distance on the z-axis between two successive common normals (or joint offset), a represents the length of each common normal (the length of a link), and α represents the angle between two successive z-axes (also called joint twist angle). Commonly, only θ and d are joint variables.

The next step is to follow the necessary motions to transform from one reference frame to the next. Assuming we are at the local reference frame $x_n - z_n$, we will do the following four standard motions to get to the next local reference frame $x_{n+1} - z_{n+1}$:

1. Rotate about the z_n-axis an angle of θ_{n+1} (Figure 2.27(a) and (b)). This will make x_n and x_{n+1} parallel to each other. This is true because a_n and a_{n+1} are both perpendicular to z_n, and rotating z_n an angle of θ_{n+1} will make them parallel (and thus, coplanar).

Table 2.1 *D-H Parameters Table.*

#	θ	d	a	α
0–1				
1–2				
2–3				
3–4				
4–5				
5–6				

As an example, the transformation between joints 2 and 3 of a generic robot will simply be:

$$^{2}T_{3} = A_3 = \begin{bmatrix} C\theta_3 & -S\theta_3 C\alpha_3 & S\theta_3 S\alpha_3 & a_3 C\theta_3 \\ S\theta_3 & C\theta_3 C\alpha_3 & -C\theta_3 S\alpha_3 & a_3 S\theta_3 \\ 0 & S\alpha_3 & C\alpha_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.54}$$

At the base of the robot, we can start with the first joint and transform to the second joint, then to the third, and so on, until the hand of the robot and eventually the end effector. Calling each transformation an A_{n+1}, we will have a number of A matrices that represent the transformations. The total transformation between the base of the robot and the hand will be:

$$^{R}T_{H} = {}^{R}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3} \ldots {}^{n-1}T_{n} = A_1 A_2 A_3 \ldots A_n \tag{2.55}$$

where n is the joint number. For a 6-DOF robot, there will be six A matrices.

To facilitate the calculation of the A matrices, we will form a table of joint and link parameters, whereby the values representing each link and joint are determined from the schematic drawing of the robot and are substituted into each A matrix. Table 2.1 can be used for this purpose.

In the following examples, we will assign the necessary frames, fill out the parameters tables, and substitute the values into the A matrices. We will start with a simple robot, but will consider more difficult robots later.

Starting with a simple 2-axis robot and moving up to a robot with 6 axes, we will apply the D-H representation in the following examples to derive the forward kinematic equations for each one.

Example 2.23

For the simple 2-axis, planar robot of Figure 2.28, assign the necessary coordinate systems based on the D-H representation, fill out the parameters table, and derive the forward kinematic equations for the robot.

Solution: First, note that both joints rotate in the x-y plane and that a frame $x_H - z_H$ shows the end of the robot. We start by assigning the z-axes for the joints. z_0 will be assigned to joint 1, and z_1 will be assigned to joint 2. Figure 2.28 shows both

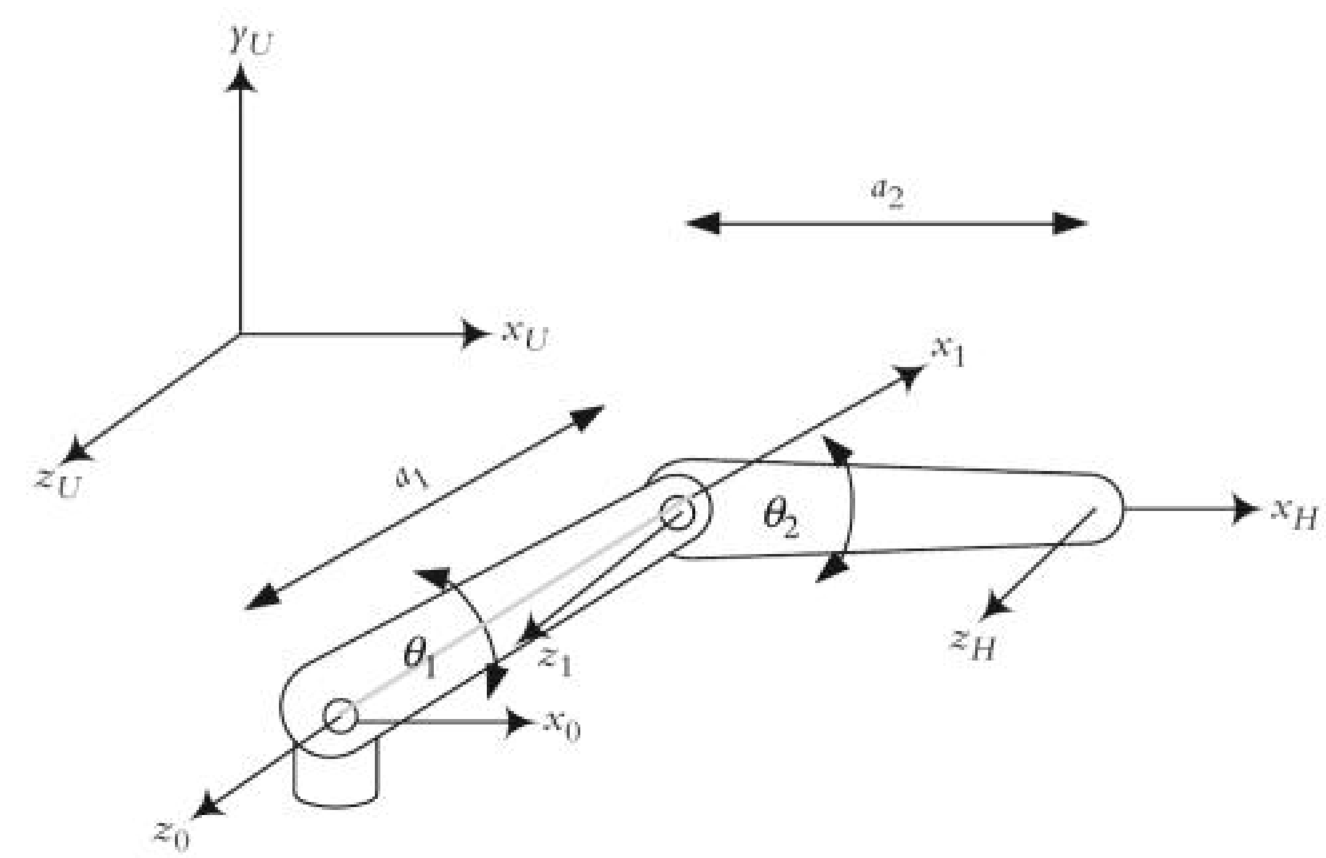


Figure 2.28 A simple 2-axis, articulated robot arm.

z-axes pointing out from the page (as are the z_U- and z_H-axes). Notice that the 0-frame is fixed and does not move. The robot moves relative to it.

Next, we need to assign the x-axes for each frame. Since the first frame (frame 0) is at the base of the robot, and therefore, there are no joints before it, the direction of x_0 is arbitrary. For convenience (only), we may choose to assign it in the same direction as the Universe x-axis. As we will see later, there is no problem if another direction is chosen; all it means is that if we were to specify ${}^UT_H$ instead of ${}^0T_H$, we would have to include an additional fixed rotation to indicate that x_U- and x_0-axes are not parallel.

Since z_0 and z_1 are parallel, the common normal between them is in the direction between the two, and therefore, the x_1-axis is as shown.

Table 2.2 shows the parameters table for the robot. To identify the values, follow the four necessary transformations required to go from one frame to the next, according to the D-H convention:

1. Rotate about the z_0-axis an angle of θ_1 to make x_0 parallel to x_1.
2. Since x_0 and x_1 are in the same plane, translation d along the z_0-axis is zero.
3. Translate along the (already rotated) x_0-axis a distance of a_1.
4. Since z_0 and z_1-axes are parallel, the necessary rotation α about the x_1-axis is zero.

The same can be repeated for transforming between frames 1 and H.

Note that since there are two revolute joints, the two unknowns are also joint angles θ_1 and θ_2. The forward kinematic equation of the robot can be found by

Table 2.2 *D-H Parameters Table for Example 2.23.*

#	θ	d	a	α
0–1	θ_1	0	a_1	0
1–H	θ_2	0	a_2	0

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Table 2.3 *D-H Parameters Table for Example 2.24.*

#	θ	d	a	α
0–1	θ_1	0	a_1	0
1–2	$90+\theta_2$	0	0	90
2–H	θ_3	d_3	0	0

applicable to this robot, but we need to add another frame for the new joint. Therefore, we will add a z_2-axis perpendicular to the joint, as shown. Since z_1 and z_2 axes intersect at joint 2, x_2-axis will be perpendicular to both at the same location, as shown.

Table 2.3 shows the parameters for the robot. Please follow the four required transformations between every two frames and make sure that you note the following:

- The direction of the H-frame is changed to represent the motions of the gripper.
- The physical length of link 2 is now a "d" and not an "a".
- Joint 3 is shown as a revolute joint. In this case, d_3 is fixed. However, the joint could have been a prismatic joint (in which case, d_3 would be a variable but θ_3 would be fixed), or both (in which case both θ_3 and d_3 would be variables).
- Remember that the rotations are measured with the right-hand rule. The curled fingers of your right hand, rotating in the direction of rotation, determine the direction of the axis of rotation along the thumb.
- Note that the rotation about z_1 is shown to be $90^\circ+\theta_2$ and not θ_2. This is because even when θ_2 is zero, there is a 90° angle between x_1 and x_2 (see Figure 2.30). This is an extremely important factor in real life, when the reset position of the robot must be defined.

Noting that $\sin(90+\theta)=\cos(\theta)$ and $\cos(90+\theta)=-\sin(\theta)$, the matrices representing each joint transformation and the total transformation of the robot are:

$$A_1=\begin{bmatrix} C_1 & -S_1 & 0 & a_1C_1 \\ S_1 & C_1 & 0 & a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\quad A_2=\begin{bmatrix} -S_2 & 0 & C_2 & 0 \\ C_2 & 0 & S_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\quad A_3=\begin{bmatrix} C_3 & -S_3 & 0 & 0 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^0T_H=A_1A_2A_3$$

$$=\begin{bmatrix} (-C_1S_2-S_1C_2)C_3 & -(-C_1S_2-S_1C_2)S_3 & C_1C_2-S_1S_2 & (C_1C_2-S_1S_2)d_3+a_1C_1 \\ (C_1C_2-S_1S_2)C_3 & -(C_1C_2-S_1S_2)S_3 & C_1S_2+S_1C_2 & (C_1S_2+S_1C_2)d_3+a_1S_1 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

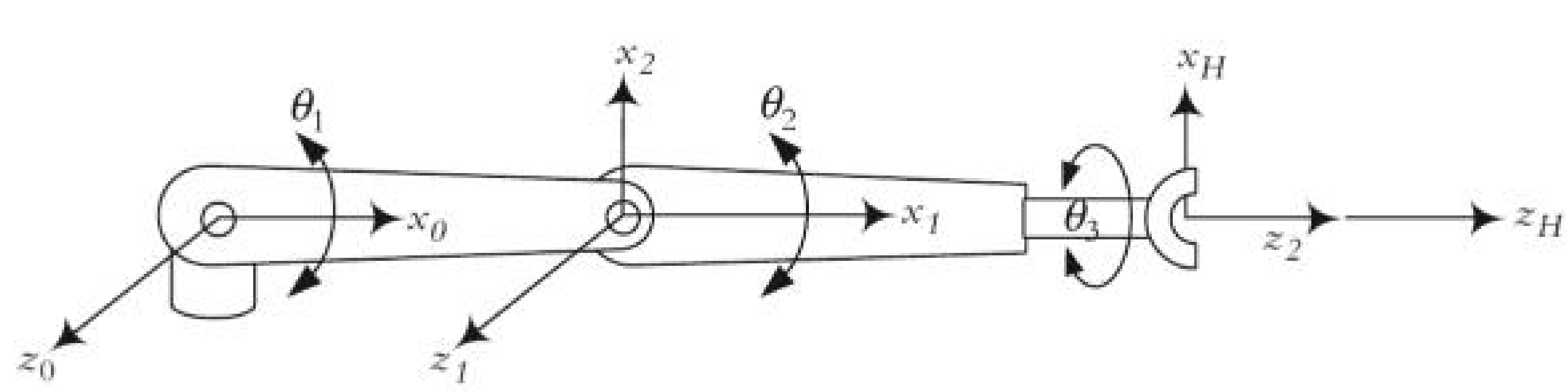


Figure 2.30 Robot of Example 2.24 in reset position.

Simplifying the matrix with $C_1C_2 - S_1S_2 = C_{12}$ and $S_1C_2 + C_1S_2 = S_{12}$, we get:

$$^0T_H = A_1A_2A_3 = \begin{bmatrix} -S_{12}C_3 & S_{12}S_3 & C_{12} & C_{12}d_3 + a_1C_1 \\ C_{12}C_3 & -C_{12}S_3 & S_{12} & S_{12}d_3 + a_1S_1 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{For} \quad \begin{cases} \theta_1 = 0 \\ \theta_2 = 0, \\ \theta_3 = 0 \end{cases} \qquad ^0T_H = \begin{bmatrix} 0 & 0 & 1 & d_3 + a_1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and}$$

$$\text{for} \quad \begin{cases} \theta_1 = 90 \\ \theta_2 = 0\ , \\ \theta_3 = 0 \end{cases} \qquad ^0T_H = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 + a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Please verify that these values represent the robot correctly. ■

Example 2.25

For the simple 6-DOF robot of Figure 2.31, assign the necessary coordinate frames based on the D-H representation, fill out the accompanying parameters table, and derive the forward kinematic equation of the robot.

Solution: As you will notice, when the number of joints increases, in this case to six, the analysis of the forward kinematics becomes more complicated. However, all principles apply the same as before. You will also notice that this 6-DOF robot is still simplified with no joint offsets or twist angles. In this example, for simplicity, we are assuming that joints 2, 3, and 4 are in the same plane, which will render their d_n values zero; otherwise, the presence of offsets will make the equations slightly more

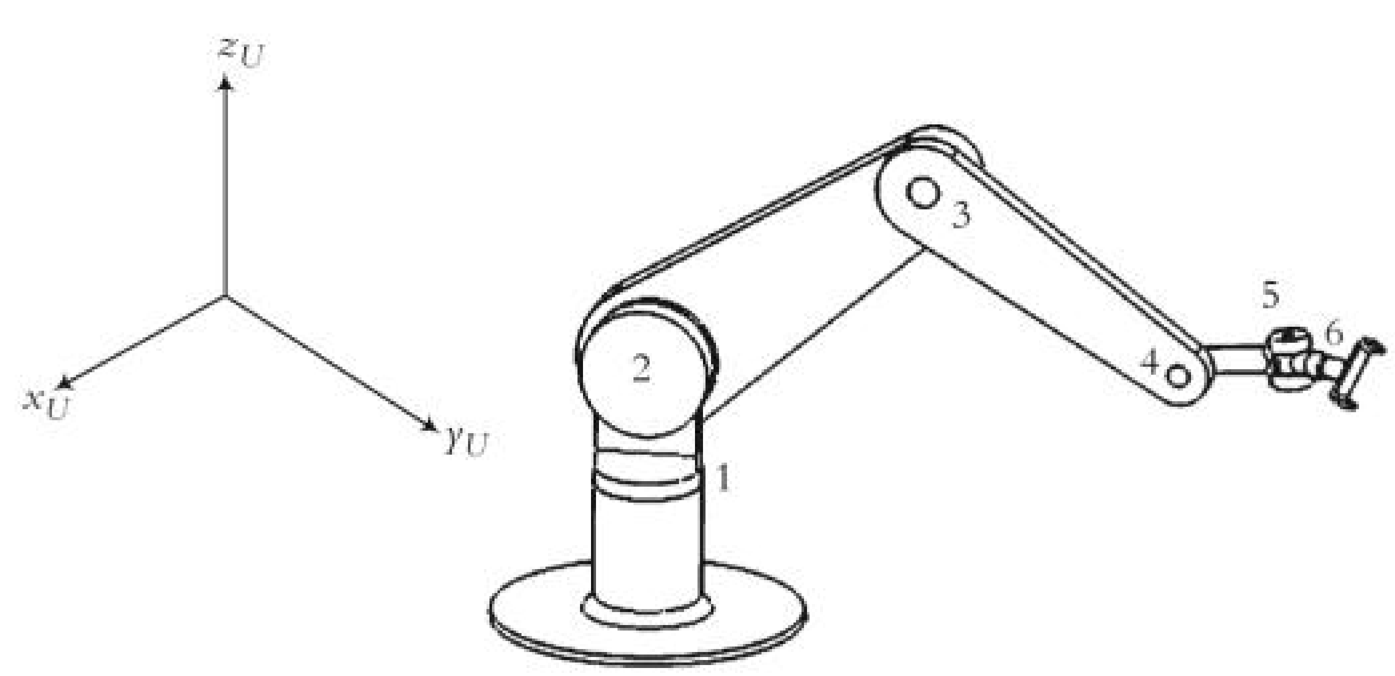


Figure 2.31 A simple 6-DOF articulate robot.

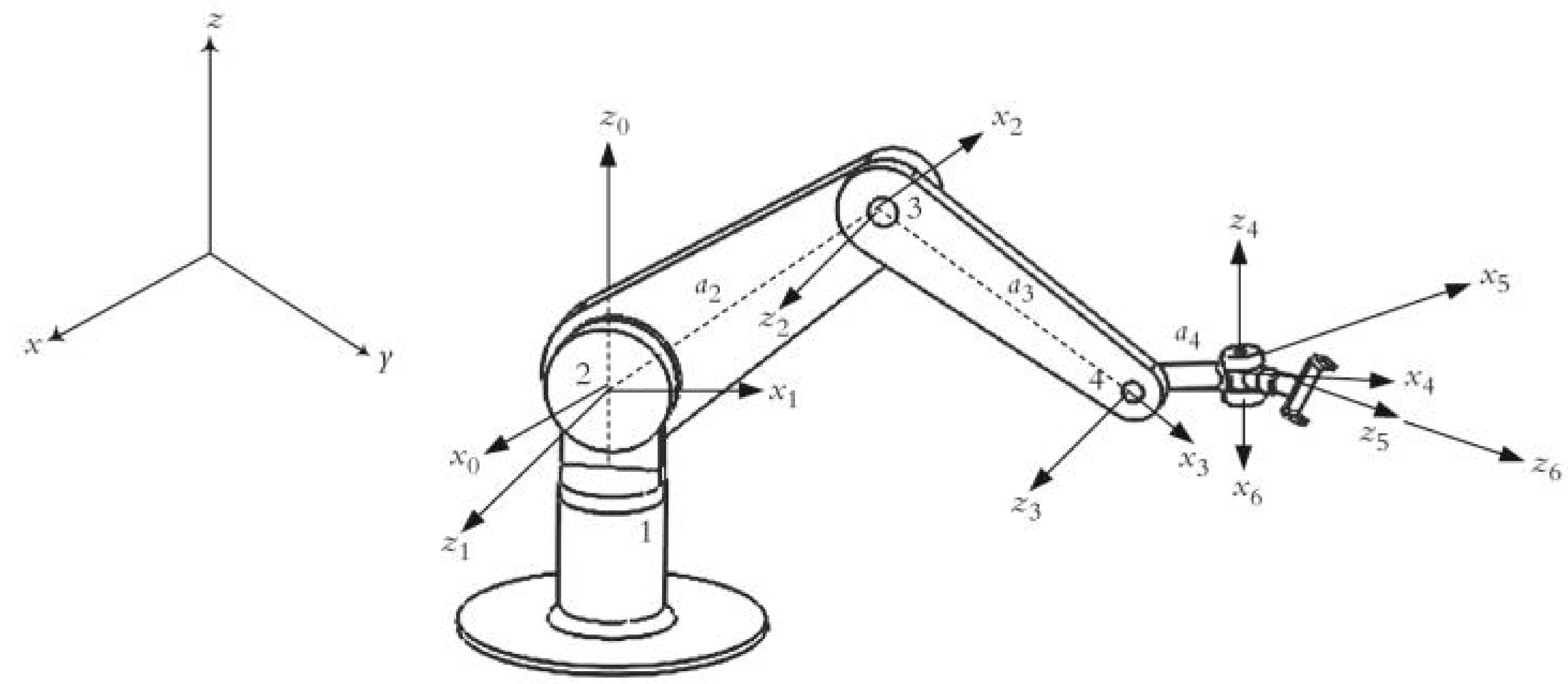


Figure 2.32 Reference frames for the simple 6-DOF articulate robot.

involved. Generally, offsets will change the position terms, but not orientation terms. To assign coordinate frames to the robot, we will first look for the joints (as shown). First, we will assign z-axes to each joint, followed by x-axes. Please follow the coordinates as shown in Figures 2.32 and 2.33. Figure 2.33 is a line drawing of the robot in Figure 2.31 for simplicity. Notice where the origin of each frame is, and why.

Start at joint 1. z_0 represents motions about the first joint. x_0 is chosen to be parallel to the reference frame x-axis. This is done only for convenience. x_0 is a fixed axis, representing the base of the robot, and does not move. The movement of the first joint *occurs around the* $z_0 - x_0$ axes. Next, z_1 is assigned at joint 2. x_1 will be normal to z_0 and z_1 because these two axes are intersecting. x_2 will be in the direction of the common normal between z_1 and z_2. x_3 is in the direction of the common normal between z_2 and z_3. Similarly, x_4 is in the direction of the common normal between z_3 and z_4. Finally, z_5 and z_6 are as shown, because they are parallel and colinear. z_5 represents the motions about joint 6, while z_6 represents the motions of the end effector. Although we normally do not include the end effector in the equations of motion, it is necessary to include the end effector frame because it will allow us to transform out of frame $z_5 - x_5$. Also important to notice is the location of the origins

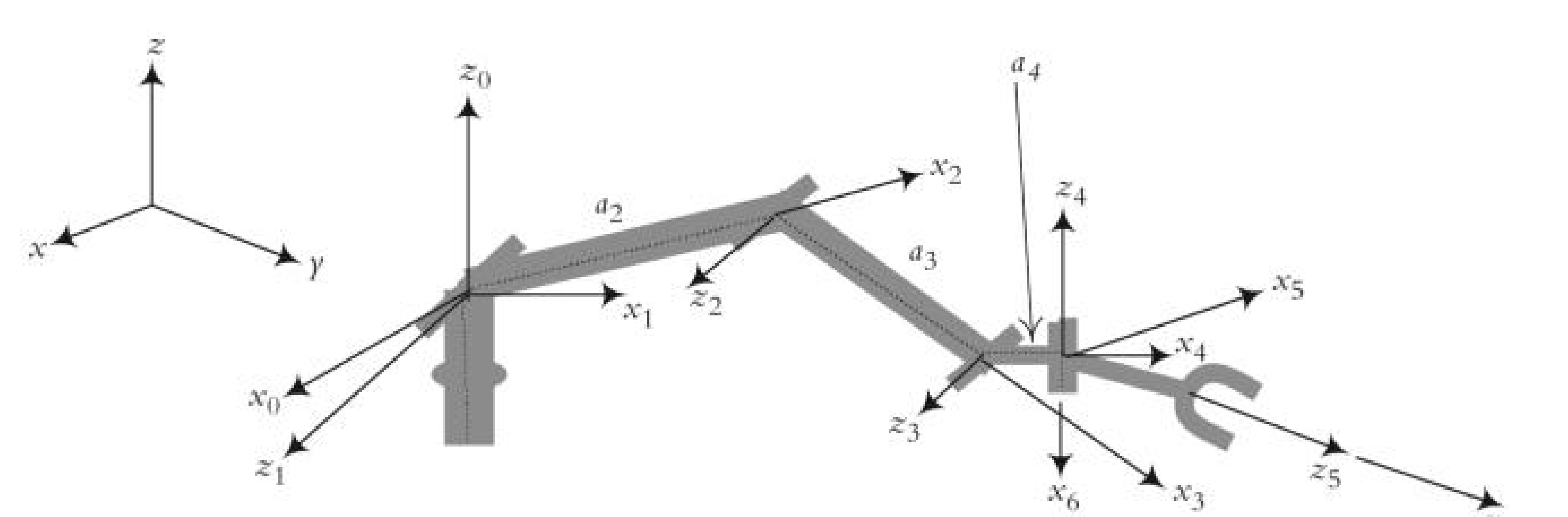


Figure 2.33 Line drawing of the reference frames for the simple 6-DOF articulate robot.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

will be necessary to make transformations along the y-axis. Therefore, the D-H methodology cannot represent these errors.

5. Note that frame $x_n - z_n$ represents link n before itself. It is attached to link n and moves with it relative to frame $n-1$. Motions about joint n are relative to frame $n-1$.
6. Obviously, you may use other representations to develop the kinematic equations of a robot. However, in order to be able to use subsequent derivations that will be used for differential motions, dynamic analysis, and so on—which are all based on the D-H representation—you may benefit from following this methodology.
7. So far, in all of our examples in this section, we derived the transformation between the base of the robot and the end effector (${}^0T_H$). It is also possible to desire the transformation between the Universe frame and the end effector (${}^UT_H$). In that case, we will need to pre-multiply ${}^0T_H$ by the transformation between the base and the Universe frames, or ${}^UT_H = {}^UT_0 \times {}^0T_H$. Since the location of the base of the robot is always known, this will not add to the number of unknowns (or complexity of the problem). The transformation ${}^UT_0$ usually involves simple translations and rotations about the Universe frame to get to the base frame. This process is not based on the D-H representation; it is a simple set of rotations and translations.
8. As you have probably noticed, the D-H representation can be used for any configuration of joints and links, whether or not they follow known coordinates such as rectangular, spherical, Euler, and so on. Additionally, you cannot use those representations if any twist angles or joint offsets are present. In reality, twist angles and joint offsets are very common. The derivation of kinematic equations based on rectangular, cylindrical, spherical, RPY, and Euler was presented only for teaching purposes. Therefore, you should normally use the D-H for analysis.

Example 2.26

The Stanford Arm: Assign coordinate frames to the Stanford Arm (Figure 2.34) and fill out the parameters table. The Stanford Arm is a spherical coordinate arm: the first two joints are revolute, the third is prismatic, and the last three wrist joints are revolute joints.

Solution: To allow you to work on this before you see the solution, the answer to this problem is included at the end of this chapter. It is recommended that before you look at the assignment of the frames and the solution of the Arm, you try to do this on your own.

The final forward kinematic solution of the Arm[5] is the product of the six matrices representing the transformation between successive joints, as follows:

$$ {}^RT_{H\,Stanford} = {}^0T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

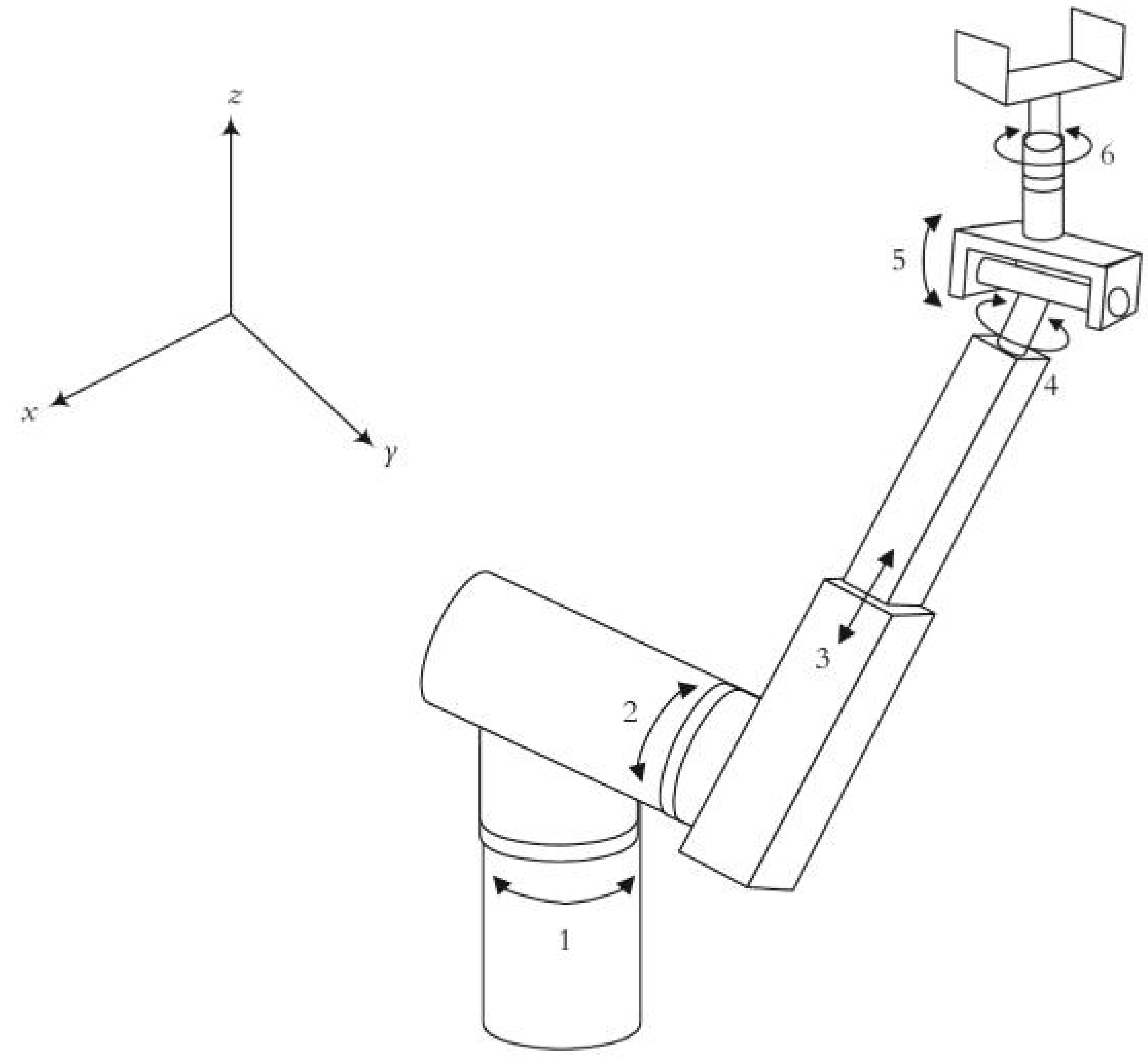


Figure 2.34 Schematic drawing of the Stanford Arm.

where

$$
\begin{aligned}
n_x &= C_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6)\\
n_y &= S_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + C_1(S_4C_5C_6 + C_4S_6)\\
n_z &= -S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6\\
o_x &= C_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] - S_1(-S_4C_5S_6 + C_4C_6)\\
o_y &= S_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] + C_1(-S_4C_5S_6 + C_4C_6)\\
o_z &= S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6\\
a_x &= C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5\\
a_y &= S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5\\
a_z &= -S_2C_4S_5 + C_2C_5\\
p_x &= C_1S_2d_3 - S_1d_2\\
p_y &= S_1S_2d_3 + C_1d_2\\
p_z &= C_2d_3
\end{aligned}
\tag{2.60}
$$

■

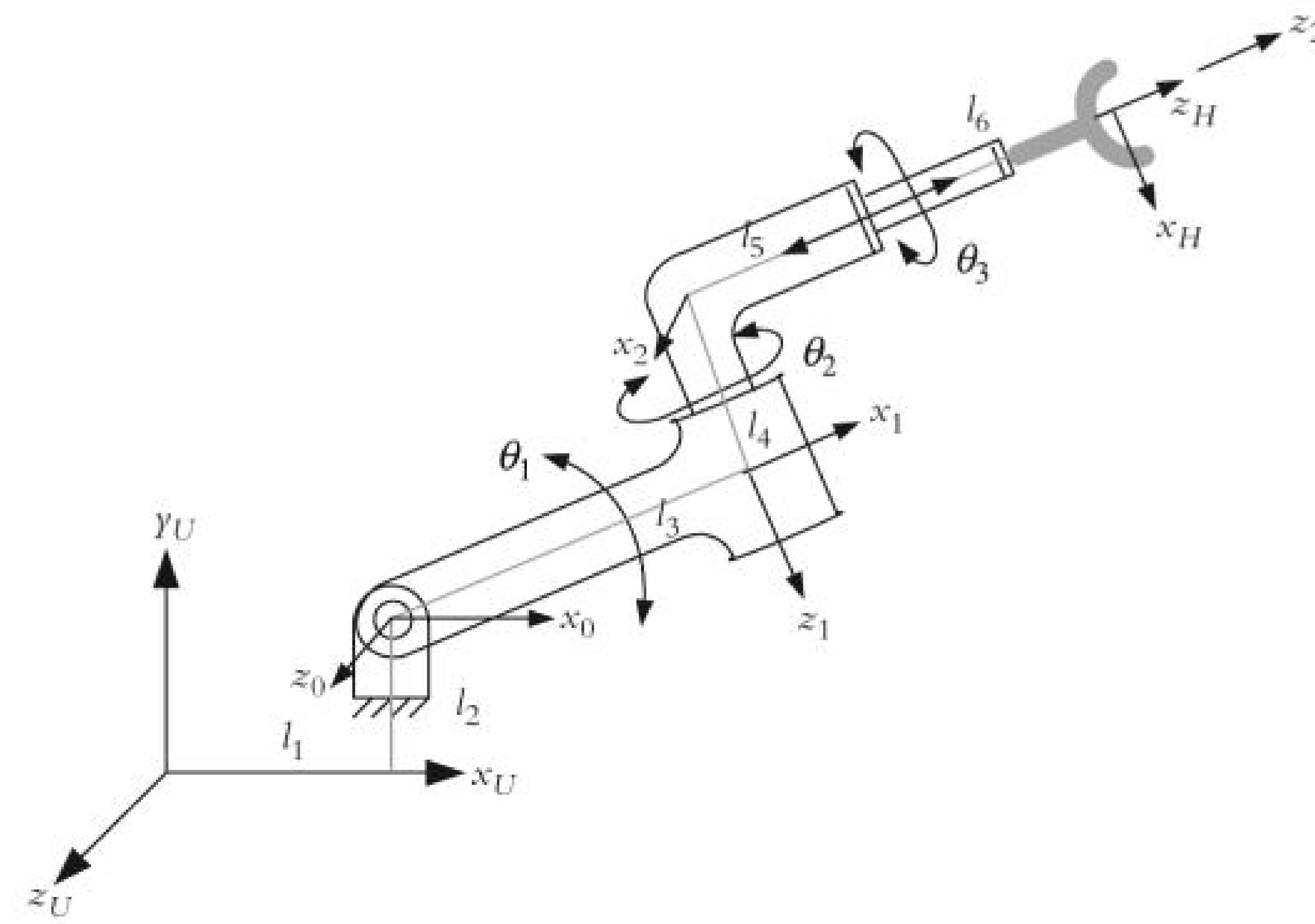


Figure 2.35 4-axis robot of Example 2.27.

Example 2.27

Assign required frames to the 4-axis robot of Figure 2.35 and write an equation describing ${}^{U}T_{H}$.

Solution: This example shows a robot with a twist angle, a joint offset, and a double-action joint represented by the same z-axis. Applying the standard procedure, we assign the frames. The parameters table is shown in Table 2.5.

The total transformation is:

$$
{}^{U}T_{H} = {}^{U}T_{0} \times {}^{0}T_{H} = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times A_1 A_2 A_H
$$

Table 2.5 *Parameters for the Robot of Example 2.27.*

#	θ	d	a	α
0–1	θ_1	0	l_3	90
1–2	θ_2	$-l_4$	0	90
2–H	θ_3	$l_5 + l_6$	0	0

■

2.13 The Inverse Kinematic Solution of Robots

As we mentioned earlier, we are actually interested in the inverse kinematic solutions. With inverse kinematic solutions, we will be able to determine the value of each joint in order to place the robot at a desired position and orientation. We have already seen the inverse kinematic solutions of specific coordinate systems. In this section, we will learn a general procedure for solving the kinematic equations.

As you have noticed by now, the forward kinematic equations have a multitude of coupled angles such as C_{234}. This makes it impossible to find enough elements in the matrix to solve for individual sines and cosines to calculate the angles. To de-couple some of the angles, we may multiply the ${}^{R}T_{H}$ matrix with individual A_{n}^{-1} matrices. This will yield one side of the equation free of an individual angle, allowing us to find elements that yield sines and cosines of the angle, and subsequently, the angle itself. We will demonstrate the procedure in the following section.

Example 2.28

Find a symbolic expression for the joint variables of the robot of Example 2.23.

Solution: The forward kinematic equation for the robot is shown as Equation (2.56), repeated here. Assume that we desire to place the robot at a position—and consequently, an orientation—given as **n**, **o**, **a**, **p** vectors:

$$
{}^{0}T_{H} = A_1 \times A_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_2C_{12} + a_1C_1 \\ S_{12} & C_{12} & 0 & a_2S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.56}
$$

Since this robot has only two degrees of freedom, its solution is relatively simple. We can solve for the angles either algebraically, or by de-coupling the unknowns. We will do both for comparison. Remember that whenever possible, we should look for values of both the sine and cosine of an angle in order to correctly identify the quadrant in which the angle falls.

I. Algebraic solution: Equating elements (2,1), (1,1), (1,4), and (2,4) of the two matrices, we get:

$$
\begin{aligned}
&S_{12} = n_y \text{ and } C_{12} = n_x \to \theta_{12} = ATAN2\,(n_y, n_x) \\
&a_2C_{12} + a_1C_1 = p_x \text{ or } a_2n_x + a_1C_1 = p_x \to C_1 = \frac{p_x - a_2n_x}{a_1} \\
&a_2S_{12} + a_1S_1 = p_y \text{ or } a_2n_y + a_1S_1 = p_y \to S_1 = \frac{p_y - a_2n_y}{a_1} \\
&\theta_1 = ATAN2(S_1, C_1) = ATAN2\left(\frac{p_y - a_2n_y}{a_1}, \frac{p_x - a_2n_x}{a_1}\right)
\end{aligned}
$$

Since θ_1 and θ_{12} are known, θ_2 can also be calculated.

From the 3,3 elements of the matrices in Equation (2.69):

$$-S_{234}(C_1a_x+S_1a_y)+C_{234}a_z=0\rightarrow\theta_{234}=\tan^{-1}\left(\frac{a_z}{C_1a_x+S_1a_y}\right)\text{ and }\theta_{234}=\theta_{234}+180^\circ \tag{2.70}$$

and we can calculate S_{234} and C_{234}, which are used to calculate θ_3, as previously discussed.

Now, referring again to Equation (2.65), repeated here, we can calculate the sine and cosine of θ_2 as follows:

$$\begin{cases}p_xC_1+p_yS_1=C_{234}a_4+C_{23}a_3+C_2a_2\\ p_z=S_{234}a_4+S_{23}a_3+S_2a_2\end{cases}$$

Since $C_{12}=C_1C_2-S_1S_2$ and $S_{12}=S_1C_2+C_1S_2$, we get:

$$\begin{cases}p_xC_1+p_yS_1-C_{234}a_4=(C_2C_3-S_2S_3)a_3+C_2a_2\\ p_z-S_{234}a_4=(S_2C_3+C_2S_3)a_3+S_2a_2\end{cases} \tag{2.71}$$

Treating this as a set of two equations and two unknowns and solving for C_2 and S_2, we get:

$$\begin{cases}S_2=\dfrac{(C_3a_3+a_2)(p_z-S_{234}a_4)-S_3a_3(p_xC_1+p_yS_1-C_{234}a_4)}{(C_3a_3+a_2)^2+S_3^2a_3^2}\\[2ex] C_2=\dfrac{(C_3a_3+a_2)(p_xC_1+p_yS_1-C_{234}a_4)+S_3a_3(p_z-S_{234}a_4)}{(C_3a_3+a_2)^2+S_3^2a_3^2}\end{cases} \tag{2.72}$$

Although this is a large equation, all its elements are known and it can be evaluated. Then:

$$\theta_2=\tan^{-1}\frac{(C_3a_3+a_2)(p_z-S_{234}a_4)-S_3a_3(p_xC_1+p_yS_1-C_{234}a_4)}{(C_3a_3+a_2)(p_xC_1+p_yS_1-C_{234}a_4)+S_3a_3(p_z-S_{234}a_4)} \tag{2.73}$$

Now that θ_2 and θ_3 are known:

$$\theta_4=\theta_{234}-\theta_2-\theta_3 \tag{2.74}$$

Remember that since there are two solutions for θ_{234} (Equation (2.70)), there will be two solutions for θ_4 as well. From 1,3 and 2,3 elements of Equation (2.69), we get:

$$\begin{cases}S_5=C_{234}(C_1a_x+S_1a_y)+S_{234}a_z\\ C_5=-C_1a_y+S_1a_x\end{cases} \tag{2.75}$$

$$\text{and}\quad\theta_5=\tan^{-1}\frac{C_{234}(C_1a_x+S_1a_y)+S_{234}a_z}{S_1a_x-C_1a_y} \tag{2.76}$$

As you have probably noticed, there is no de-coupled equation for θ_6. As a result, we have to pre-multiply Equation (2.69) by the inverse of A_5 to de-couple it. We get:

$$\begin{bmatrix} C_5\left[C_{234}\left(C_1 n_x + S_1 n_y\right) + S_{234} n_z\right] & C_5\left[C_{234}\left(C_1 o_x + S_1 o_y\right) + S_{234} o_z\right] & 0 & 0 \\ -S_5\left(S_1 n_x - C_1 n_y\right) & -S_5\left(S_1 o_x - C_1 o_y\right) & & \\ -S_{234}\left(C_1 n_x + S_1 n_y\right) + C_{234} n_z & -S_{234}\left(C_1 o_x + S_1 o_y\right) + C_{234} o_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.77}$$

From 2,1 and 2,2 elements of Equation (2.77) we get:

$$\theta_6 = \tan^{-1} \frac{-S_{234}\left(C_1 n_x + S_1 n_y\right) + C_{234} n_z}{-S_{234}\left(C_1 o_x + S_1 o_y\right) + C_{234} o_z} \tag{2.78}$$

Therefore, we have found six equations that collectively yield the values needed to place and orientate the robot at any desired location. Although this solution is only good for the given robot, a similar approach may be taken for any other robot.

It is important to notice that this solution is only possible because the last three joints of the robot are intersecting at a common point. Otherwise, it will not be possible to solve for this kind of solution, and as a result, we would have to solve the matrices directly or by calculating the inverse of the matrix and solving for the unknowns. Most industrial robots have intersecting wrist joints.

2.14 Inverse Kinematic Programming of Robots

The equations we found for solving the inverse kinematic problem of robots can directly be used to drive the robot to a desired position. In fact, no robot would actually use the forward kinematic equations in order to solve for these results. The only equations that are used are the set of six (or less, depending on the number of joints) equations that calculate the joint values. In other words, the robot designer must calculate the inverse solution and derive these equations and, in turn, use them to drive the robot to position. This is necessary for the practical reason that it takes a long time

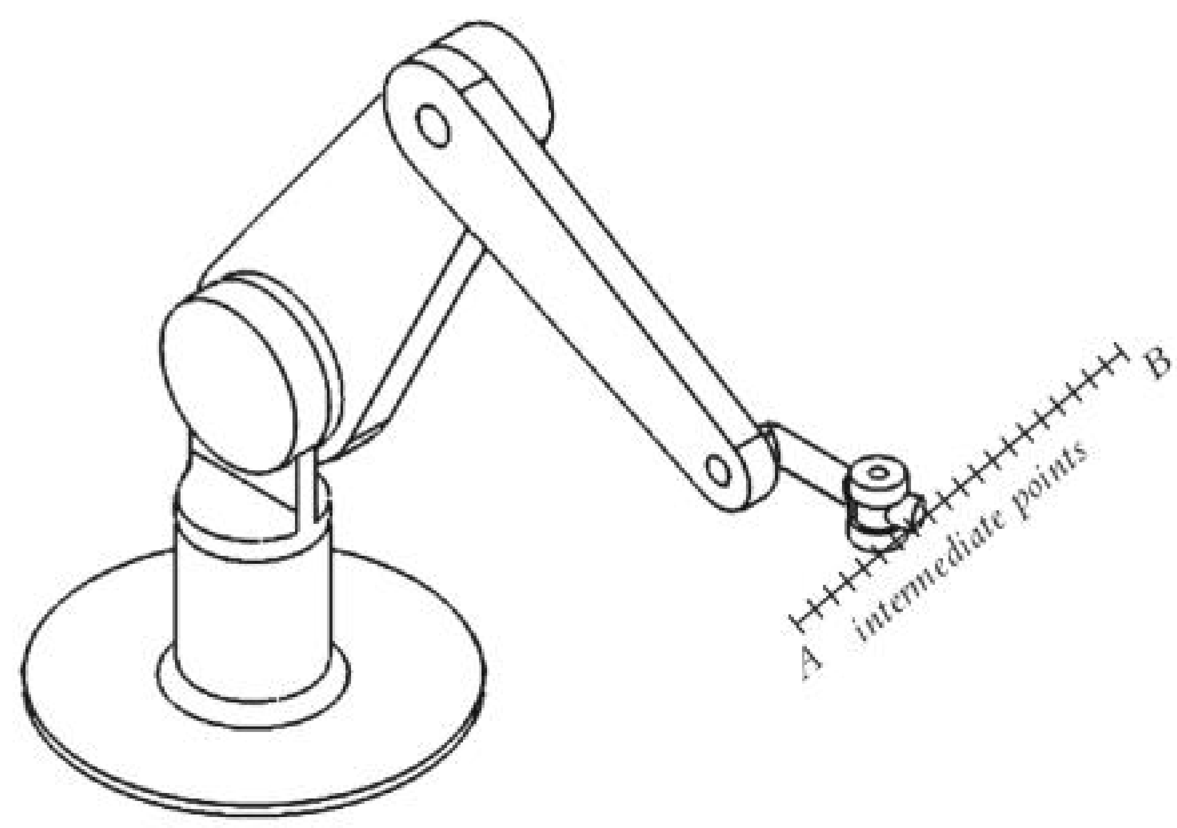


Figure 2.36 Small sections of movement for straight line motions.

for a computer to calculate the inverse of the forward kinematic equations or to substitute values into them and calculate the unknowns (joint variables) by methods such as Gaussian elimination.

For a robot to move in a predictable path, say a straight line, it is necessary to recalculate joint variables many times a second. Imagine that a robot needs to move in a straight line between a starting point *A* and a destination point *B*. If no other action is taken and the robot moves from point *A* to point *B*, the path is unpredictable. The robot moves all its joints until they are at the final value, which will place the robot at the destination point *B*. However, depending on the rate of change in each joint, the hand will follow an unknown path in between the two points. To make the robot follow a straight line, it is necessary to break the line into many small sections (Figure 2.36) and make the robot follow those very small sections sequentially between the two points. This means that a new solution must be calculated for each small section. Typically, the location may be recalculated between 50 to 200 times a second. This means that if calculating a solution takes more than 5 to 20 ms, the robot will lose accuracy or will not follow the specified path.[10] The shorter the time it takes to calculate a new solution, the more accurate the robot. As a result, it is vital to eliminate as many unnecessary computations as possible to allow the computer controller to calculate more solutions. This is why the designer must do all mathematical manipulations beforehand and only program the robot controller to calculate the final solutions. This will be discussed in more detail in Chapter 5.

For the 6-axis robot discussed earlier, given the final desired location and orientation as:

$$
{}^{R}T_{H_{Desired}} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Index

Index