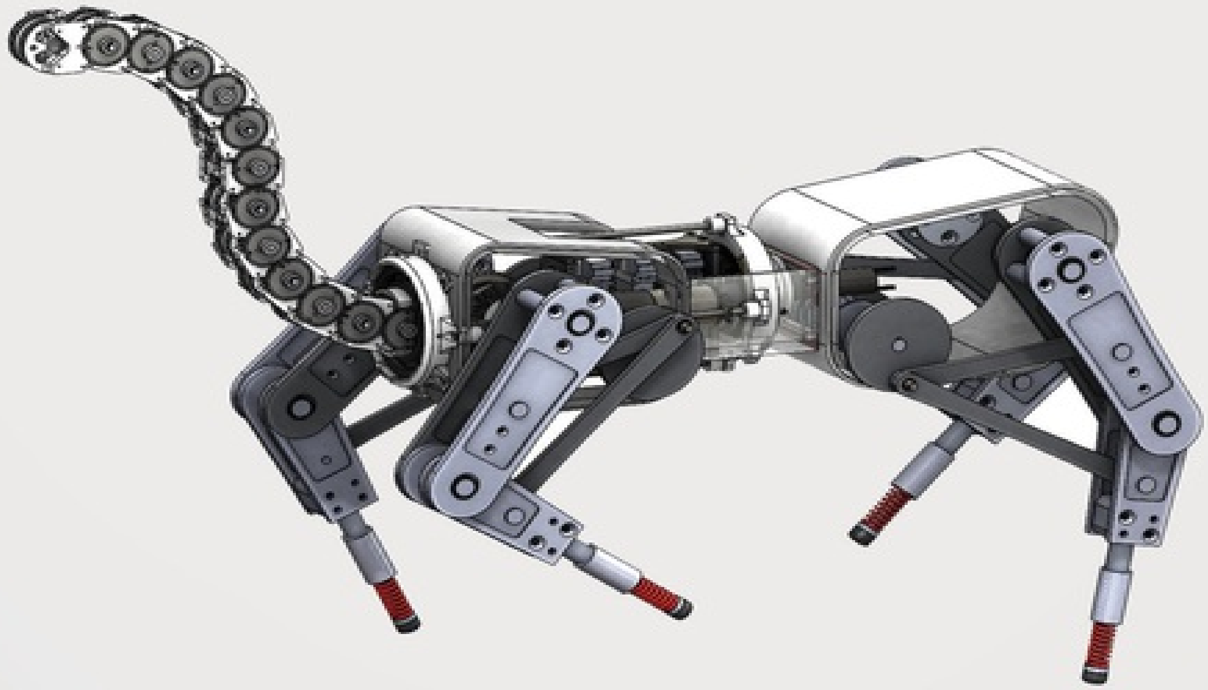

DYNAMICS AND CONTROL OF ROBOTIC SYSTEMS

ANDREW J. KURDILA | PINHAS BEN-TZVI

with website

WILEY

Table of Contents

List of Tables

List of Illustrations

Chapter 4

Appendix

Dynamics and Control of Robotic Systems

Andrew J. Kurdila and Pinhas Ben-Tzvi
Virginia Tech
Virginia
USA

WILEY

Copyright

This edition first published 2020

Registered Offices

John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Editorial Office

The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print - on - demand. Some content that appears in standard print versions of this book may not be available in other formats.

Library of Congress Cataloging‐in‐Publication Data

Names: Kurdila, Andrew J., author. | Ben-Tzvi, Pinhas, 1973‐ author.

Title: Dynamics and control of robotic systems / Andrew J. Kurdila, Virginia

Tech, Virginia, USA, Pinhas Ben‐Tzvi, Virginia Tech, Virginia, USA.

Description: Hoboken, NJ : Wiley, 2020. | Includes bibliographical references

and index. |

Identifiers: LCCN 2019007284 (print) | LCCN 2019009469 (ebook) | ISBN

9781119524908 (Adobe PDF) | ISBN 9781119524953 (ePub) | ISBN 9781119524830

(hardback)

Subjects: LCSH: Robots-Dynamics. | Robots–Control systems.

Classification: LCC TJ211.4 (ebook) | LCC TJ211.4 .K87 2020 (print) | DDC

629.8/92–dc23

LC record available at https://lccn.loc.gov/2019007284

Cover Design: Wiley

Cover Image: © Pinhas Ben‐Tzvi

Dedication

This book is dedicated to Patrick, Hannah and Justin, and to Timor, Jonathan, Daniel, and Sara

Preface

The goal of this book is to provide a modern, systematic, and thorough theoretical background for the study of the dynamics and control of robotic systems. The presentation of the material emphasizes the underlying principles of dynamics and control that can be employed in a host of contemporary applications. Consequently, at its core, the goal of this book is quite ambitious. Not only do we seek to give a detailed presentation of the precepts of robotics, but also we aim to provide methodologies that are applicable to realistic robotic systems. These robotic systems include the following well known examples: classical industrial manipulators, humanoid robots, autonomous ground vehicles, autonomous air vehicles, autonomous marine vehicles, robotic surgical assistants, space vehicles, and computer controlled milling machines. Modern robotic systems are inherently complex, and the representation of their dynamics and the synthesis of their control can be unavoidably complicated.

One of the principal reasons for creating this book has been to show how modern computational and analytical tools expand and enhance our ability to address problems in robotics. Even a few years ago, the complexity of modern robotic systems rendered intractable the solution by hand of all but the most simple examples. The formulation of dynamic models of common robotic systems was once too tedious for the classroom. The advent of symbolic, numeric, and general purpose computational engines over the past few decades is particularly relevant to the problems addressed in this book. With higher level computing environments such as MATLAB, Mathematica, Maple and similar programs, the envelope of problems that can be addressed by undergraduate and graduate students has expanded dramatically. These tools enable students to focus on principles and theory, and free them from tedious exercises in algebraic gymnastics that merely distract from the technical foundations. It is critical, in our opinion, that the student concentrates on the systematic application of the underlying principles.

This text evolved from class notes and problem assignments for courses in dynamics, control, and robotics taught by the authors over a period of several years. These courses have been taught at several top tier universities in the United States, and our approach in presenting the material has continuously evolved during this time. This material is suitable for a two semester sequence in dynamics, control, and robotics at the senior undergraduate or first year graduate student level. A course intended for the senior year of an undergraduate curriculum can focus on the fundamentals of kinematics and dynamics as applied to robotic systems. This first semester can be built primarily from topics extracted from Chapters 2, 3, and 4, and to a lesser degree from Chapter 5. A second semester can concentrate on the techniques of analytical mechanics in Chapter 5 and control theory in Chapter 6. Specific advanced topics such as the recursive order N formulation in Chapters 3 and 4, or the vision - based control methodologies in Chapter 7 can also be covered in the second semester.

The authors have worked hard to demonstrate that a wide array of design and analysis problems for robotic systems are made tractable through the use of modern computational and analytical tools. To this end, an extensive collection of examples and problems are included in the text. The solutions of many of the examples or problems have been carried out using either MATLAB, Simulink or Mathematica, or a combination of both. It is important that the students who use this book realize that the authors are not advocating the use of a particular computational tool, but rather espousing a common philosophy. For nearly every problem in this book, the computational tools are interchangeable: a student can use whatever software package with which he or she is most familiar. The theoretical foundations, however, are irreplaceable and constitute the common language for addressing any specific problem.

September 2019

Andrew J. Kurdila

Pinhas Ben - Tzvi

Acknowledgment

The authors would like to acknowledge William S. Rone for his review of the text and help with the figures in this textbook and to thank Jessica G. Gregory for her review of the text.

About the CompanionWebsite

The companion website for this book is at

www.wiley.com/go/kurdila/robotic-systems

The website includes:

- Matlab workbook

Scan this QR code to visit the companion website.

Chapter 1
Introduction

In this chapter the collection of*robotic systems* that are studied in this book are introduced. The field of robotics embraces topics requiring expertise in a number of technical disciplines including mechanical engineering, electrical engineering, computer science, applied mathematics, industrial engineering, cognitive science, psychology, biology, bio‐inspired design, and software engineering. Moreover, the family of robotic systems that can be designed and fabricated today is growing rapidly. Reasons for this trend are based in economics and the maturity of the technological infrastructure supporting robotics. A wide variety of sensing and actuation technologies that are portable, compact, and inexpensive are now readily available. These building blocks can be used to construct a plethora of robotic systems using commercial off‐the‐shelf technology. The broad scope of the robotics field precludes a comprehensive theoretical summary of the disciplines relevant to all of these diverse systems being given. Instead, this text specifically deals with the construction of models of the kinematics and dynamics of typical robotic systems, and the derivation of control strategies for these systems. Upon completion of this chapter, the student should be able to:

- Discuss a variety of definitions of a robotic system and explain their key attributes.
- Discuss the general structure and components of robotic systems.
- Describe a variety of methods for classifying robotic systems.
- Describe the classical robotic manipulators, including the Cartesian, cylindrical, spherical, SCARA, PUMA, and articulated robotic manipulators.
- Describe other common, contemporary robotic systems.

- Describe the fundamental problems of forward kinematics, inverse kinematics, forward dynamics, and control synthesis for robotic systems.

1.1 Motivation

Over the past few decades, the robotic systems that undergraduate and graduate students are expected to be able to design and analyze has expanded dramatically. It is now commonplace in varying engineering disciplines to ask relatively inexperienced engineers and researchers to design, analyze, and construct prototypical robotic systems. Students may encounter such challenges in either undergraduate or graduate design projects, or immediately upon taking a job in industry or at a national laboratory. Projects may be as varied as the development of a computer controlled, multi‐axis stage for positioning of laser Doppler vibration measurements, the development of a flapping wing autonomous flight vehicle, the modification of a commercial vehicle for autonomous operation, or the development of a humanoid robot. The diversity and complexity of this list continues to grow every year.

While the study of robotics has been popular for several decades, the recent rapid expansion of robotic systems in commercial markets can be attributed in part to the fact that sensors and actuators have become increasingly cost effective, modular, and portable. This trend has lead to the emergence of the field of*mechatronics*, which has played a key role in the spread of robotics technologies. Mechatronics is a multidisciplinary field of study that integrates aspects of mechanisms, electronics, computer hardware/software, systems theory, and information technologies into a unified practical design methodology. The fusion of these topical areas that define the study of mechatronics is depicted in Figure 1.1. A key feature of mechatronic systems is that they often feature built‐in intelligence that is applied to the task for which they are designed.

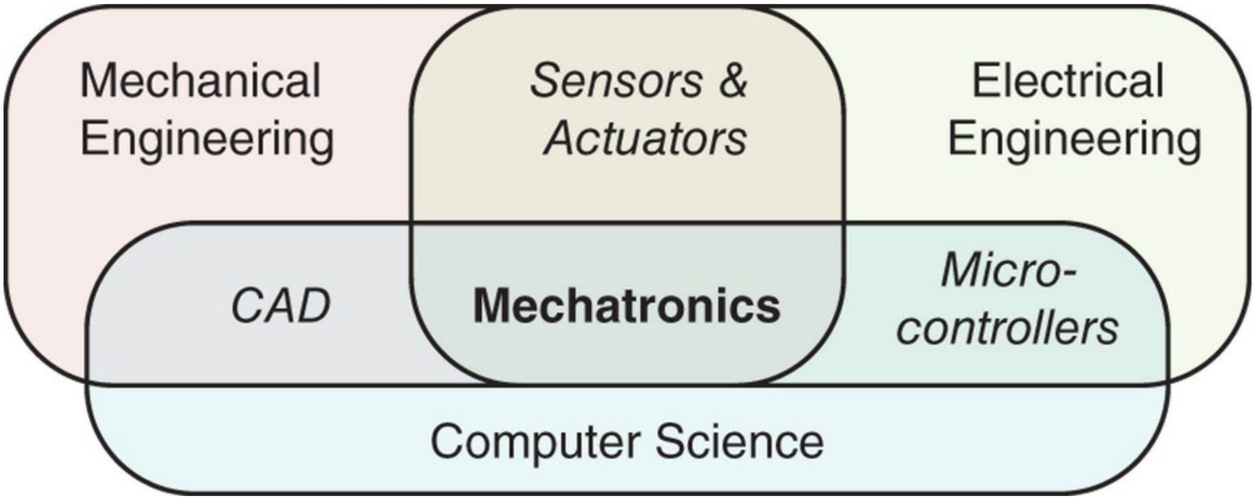

Figure 1.1 Fields of expertise associated with mechatronics.

Although the range of mechatronic systems is vast, there are features common to most, if not all, such systems. Figure 1.2 illustrates a schematic drawing of signal flow for a typical mechatronic system. Computer systems connect the mechatronic system to sources of intelligence, be it user inputs/outputs to include humans in the operation and/or algorithms to interpret sensor data and make decisions for the mechatronic systems. The electrical system conditions signals passing between the computer and mechanical systems, along with regulating the electrical power provided to the mechatronic system. The mechanical systems consist of the physical system(s) that interact with their environment. Commands from the digital computer systems to the analog electrical systems pass through a*digital - to - analog* converter, and these commands are implemented on actuators connecting the electrical and mechanical systems. Sensors integrated into the mechanical systems generate signals passed to the electrical systems, and these signals (after conditioning) are communicated to the computer systems through an*analog - to - digital* converter.

Mechatronics is elevated to a field distinct from its contributing fields by the need to balance consideration of mechanical, electrical and information technology factors when designing an overall system. Assessing the signal processing and algorithmic requirements for operating a physical system, and meeting these requirements intelligently and efficiently, distinguishes mechatronics as a unique discipline and not simply an exercise in hardware connectivity. While some systems may require complex multi - core processors to operate in real time, others may simply require a simple embedded controller. Interested readers can refer to the following textbooks for a more in - depth study of mechatronics as an integrating approach to engineering design [1,8,11].

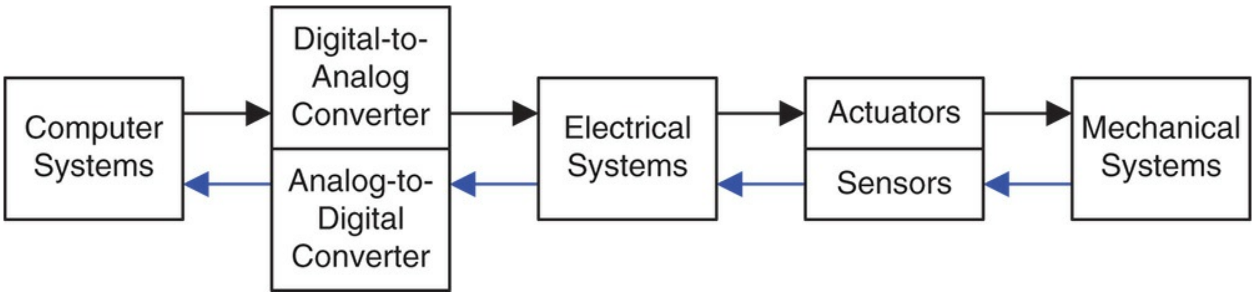

Figure 1.2 Structure of a typical mechatronic system.

As the robotics infrastructure has matured, expectations of students in the field of robotics has correspondingly increased. A decade ago a beginning student might have been asked to create a simple two - dimensional model of a robotic system. Older textbooks are filled with such introductory problems that serve to familiarize students with the fundamentals. However, technical tools and analytical skills are now required that facilitate modeling of robot kinematics and dynamics in three spatial dimensions.

Fortunately, the tools that are applicable throughout the design and analysis process have also evolved and matured. A few years ago, the computational tools available for the systematic design, analysis, and study of complex robotic systems were limited in number. At that time a student faced with the creation of a detailed model of a realistic robotic system was confronted with a daunting task. The determination of the kinematics and dynamics of robotic systems via hand calculation was a lengthy and tedious job for all but the simplest cases. Once the heroic effort of deriving a formulation was complete, the student was faced with coding the governing equations in a low level programming language such as C or Fortran. It is no exaggeration that the time involved in this task could be measured in months, or worse, years, of effort.

Now, two separate and complementary collections of commercial software packages make this problem much more manageable. First, there is an ever expanding list of specialized three dimensional modeling programs such as

- Autodesk Inventor
- SolidWorks
- Pro Engineer
- MSC Adams
- LabView

that are available for building highly detailed and general models of the kinematics, dynamics, and control of robotic systems. These packages vary in the generality of their simulation capabilities, but all allow numerical approximation of the solutions of the forward kinematics and dynamics problems. Some also incorporate programming interfaces for the introduction of user - defined controls. These software packages can be expensive to purchase. However, most universities have software contracts with the vendors of these packages. Most large engineering firms or government laboratories also have licenses for a portfolio of these analysis programs. Many of the more complex examples in this book have been modeled by students under an academic license for Autodesk Inventor.

As useful as the programs above can be, sometimes greater flexibility is needed in formulating the governing equations of dynamics or in deriving a control architecture for a robotic system. As an example, when a model is created for the purpose of constructing a controller for a specific robot system, a symbolic set of equations for hardware implementation is often required. Some programs have the option of explicitly generating symbolic code that is suitable for hardware implementation. It should be noted that the packages listed above vary dramatically in the ways that they handle code generation. There is currently a highly competitive market of software tools to download controller equations to specific hardware platforms. Still, it is often the case that a standard commercially available software simulation tool, such as those listed above, does not allow the flexibility that a practicing control engineer requires. It can also be the case that an analyst wants to implement a controller in terms of a highly efficient algorithm, like the recursive formulations discussed in Chapters or . These algorithms may not be supported by a specific commercial software package. It should come as no surprise that no matter how well a commercial package is designed, a user will often desire some functionality that is not available.

In such cases, the software packages that support symbolic computation can be used to great advantage. These are general purpose, object - oriented, high level programs that define their own computing languages. Examples include:

- MATLAB
- Mathematica
- Mathcad
- Maple

Each of these software programs has developed its own object oriented, high level language that performs calculations on a large number of different types of mathematical objects. For example, they usually have a large library of operators based on linear algebra, signal processing, and calculus. The mathematical objects may be matrices and vectors, or they can be discrete dynamical systems, or they might take the form of systems of ordinary differential equations. A few lines of code in the language of these packages can replace thousands of lines of code in a low level programming language like C, C++ or Fortran. Perhaps most importantly for this text, each of these programs has a syntax that enables symbolic computation. This is a computing engine that incorporates most well known operations defined in differential or integral calculus. For the most part, tedious operations can be performed using these symbolic variables with minimal input from the analyst. Both public domain and commercial packages designed expressly for the study of robotic systems have been written in several of these computing languages. This text makes extensive use of some of these packages in solving the examples in the text and the problems at the end of each chapter. In many cases the solutions of the problems are carried out by writing general purpose programs that address fundamental robotics problems; a family of high level functions that solve core robotics problems are provided with the solutions for this text.

1.2 Origins of Robotic Systems

Robotic systems have been traced historically to efforts by early artists, artisans, craftsmen, engineers, and scientists to create machines that mimic humans in action or reasoning. The modern notion of a robotic system emerged as society sought to create surrogates that can replace human labor in jobs that are menial, tiresome or even dangerous. Even before industrial robots became commonplace, the potentially transformative role of automatons in the workplace was imagined. The role of robots as factory workers has been noted repeatedly over the years. The word “robot” was coined by the Czech writer Karel Capek in the play*Rossum's Universal Robots* published in 1920. Capek wanted to describe the repetitive and boring nature of robotic tasks. The word “robot” originates from the Czech word “robota” which means “work” or “forced labor”. The play studied moral questions arising in the creation and use of digitally programmed slaves. This has been a recurring theme in novels, plays, and movies. For example, the novelist Kurt Vonnegut explores the angst and disillusionment of a society with the displacement of human workers by automation in the more recent novel*Player Piano*.

Despite these cautionary tales, robots have proliferated as a means of replacing human labor in adverse environments. The first reprogrammable digitally controlled robot was created in 1954 by George Devol. This robot,*Unimate*, was an industrial manipulator having a spherical workspace and was used to lift and move heavy production parts in a factory setting. It was purchased by General Motors in 1960 and was the forerunner of the large collection of industrial robots that are now commonplace along modern factory assembly lines. Demands on performance have been a driving force in the use of robotics in industry. The load capacity, repeatability, precision, and speed afforded by modern robotic systems far exceed the capabilities of man.

Current definitions of what constitutes a robotic system vary dramatically, but all definitions convey the idea that robots perform menial or repetitive tasks. Merriam - Webster's Dictionary defines a robot as

- a machine that looks like a human being and performs various complex tasks, or
- a device that automatically performs complicated often repetitive tasks.

The first definition above requires that robots appear to be humanoid, and while some robotic systems do indeed have a humanoid appearance, this definition would exclude many of the robotic systems in this book. A critical attribute of robotic systems that this definition omits, one that is important to engineers and scientists who actually build robotic systems, is that robots are controlled by computers. This fact is made explicit in the Cambridge Dictionary which defines a robot as

- a machine used to perform jobs automatically, which is controlled by a computer.

Some definitions of robots have arisen in view of the historical concentration of robots in factories and along assembly lines. The Robotics Institute of America defines a robot as

- a reprogrammable, multi - functional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks.

This definition focuses on the robot as a*multifunctional manipulator* or*robotic arm*, but neglects a wide range of mobile robots designed to explore and map environments without the need for a manipulator to interact with these environments.

All of the definitions of robots above are accurate in some contexts, but do not describe the breadth of systems that will be considered in this book. The definition of a robotic system that will be used in this book is given below.

Definition 1.1 (Robotic System)

A robotic system is a reprogrammable, computer - controlled mechanical system that may sense and react to attributes of its surroundings as it performs assigned tasks with some degree of autonomy.

This definition expands those previously introduced and is broad enough to encompass the examples encountered in this book. A robot need not have humanoid form, and it does not necessarily have the form of a multi - functional manipulator. The above definition emphasizes that robotic systems exhibit some level of autonomy. They operate, to varying degrees, independent of human intervention. They have sensors such as cameras, laser ranging sensors, acoustic proximity sensors, or force transducers that allow them to sense their environment via measurements. This data is subsequently used by the robot to react to its environment. For example, an autonomous ground, air, marine, or space vehicle may change course to avoid obstructions or debris; a dexterous manipulator may change the pressure with which a tool is gripped based on force transducer measurements; a robotic manipulator may use camera measurements to position a tool in the workspace. Finally, the definition makes explicit that a robot is a mechanical system, one that is built from the interconnection of components.

In summary, a robotic system is made possible through the synthesis of theory and techniques from many fields, perhaps most notably mechanical engineering, electrical engineering, and information technology. The field of mechatronics facilitates and enables the development of complex robotic systems from standard sub - systems and has accelerated the maturation of the robotics field in recent years. This relationship among some of the primary fields contributing to robotics is depicted in Figure 1.3.

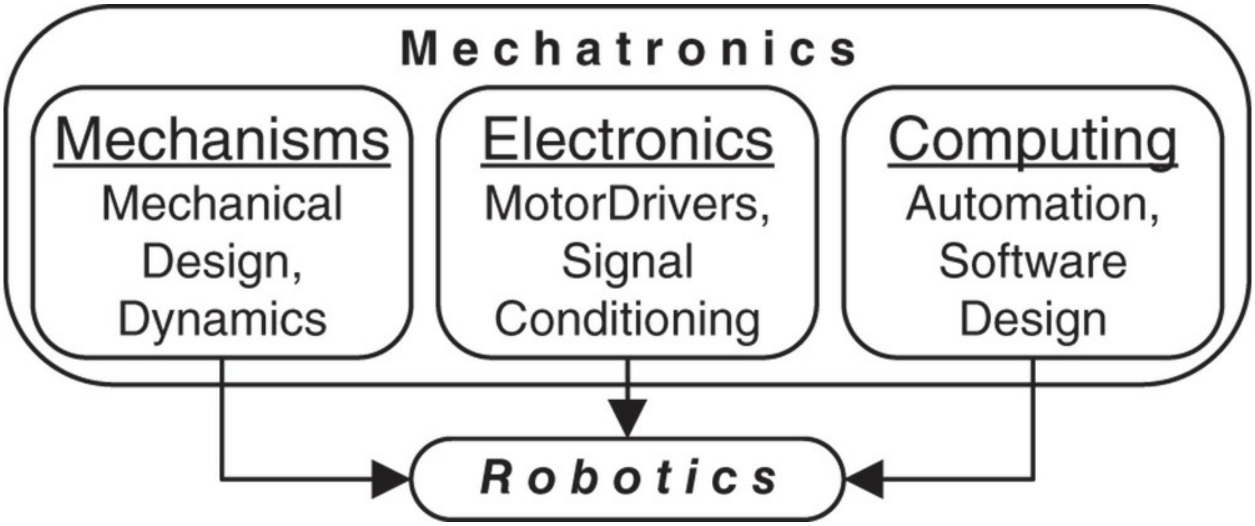

Figure 1.3 Fields contributing to robotics.

1.3 General Structure of Robotic Systems

As will be discussed in the following sections, there are a diverse population of robots that have been developed over the years, ranging from robotic manipulators to mobile robots that traverse the air, land, or sea. These robots may emulate humans or animals, or have novel topologies to accomplish desired tasks. However, despite these differences, there are some common features that many robots share that are discussed in this section.

Figure 1.4 depicts several components of a typical robotic system. Nearly all robotic systems feature*actuators*. The actuators serve as the muscles of the system and produce motion. Their power is usually supplied electrically, pneumatically, or by hydraulics. Since many robots are either controlled remotely or make provision for interruptions to their autonomous operation from outside agents, many robots include a*communicator* of some sort. The communicator is a unit that transmits information to a host and/or receives instructions from a remote operator. As noted earlier, an essential feature of any robot is that it exhibit some level of autonomy or intelligence. A*control unit* is a vital component of nearly all robotic systems. It may consist of a single processor, or may be a central computer that integrates the activities of several microprocessors. Many robotic manipulator systems, underwater autonomous vehicles, or space robots must directly mechanically manipulate their environment. An*end effector* that consists of a*gripping device* at the end of a manipulator arm can therefore be essential to the operation of the robot. The end effector can be used to make intentional contact with an object or to produce the robot's final effect on its surroundings. In some cases there may be several manipulator arms or gripping mechanisms. Since a robot must interact with its environment, and usually lacks much information about its surroundings, many robots also include*sensor suites* that include a variety of sensing modalities. Each*sensor* is usually a transducer of some kind whose inputs are physical phenomena and whose outputs consist of electronic signals. Finally, since mobility, sensing and actuation require energy expenditure, a robot must have a*power supply* of some type. Most frequently this is an energy storage device such as a battery. In some instances the robot may be tethered to a fixed power supply. For example, a military or industrial exoskeleton may require so much power that it is only feasible to connect to a remote local power supply while the suit is worn in a warehouse to move heavy payloads.

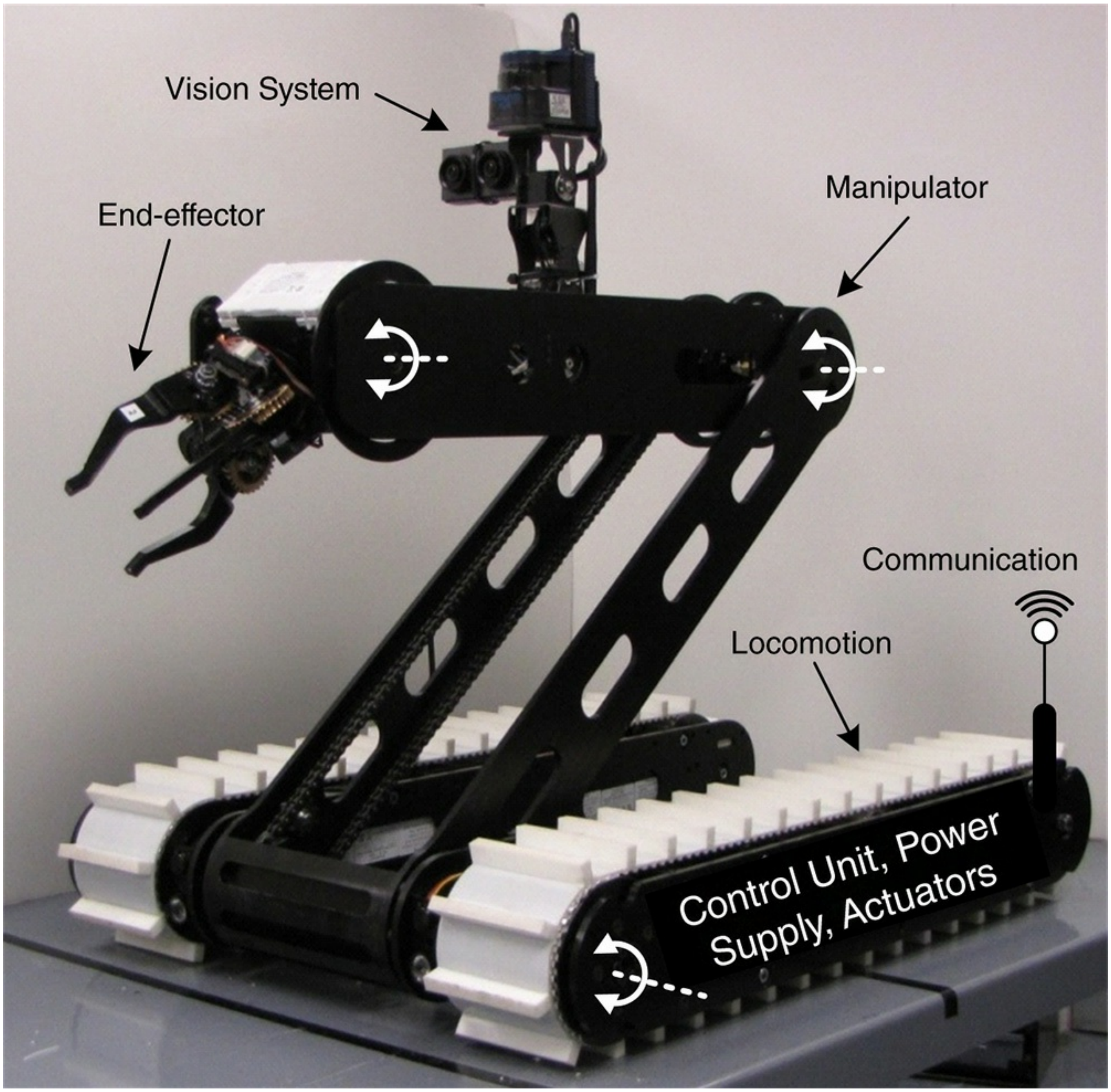

Figure 1.4 Typical mobile robotic system components [4–6].

Any particular robotic system may include many of these components, or simply a few in each category. An autonomous military ground vehicle will usually host a wide variety of vision sensors, motion sensors including a ground positioning system (GPS) and compass, thermal sensors, and chemical sensors. A simple table top robotic manipulator in a laboratory might only have joint encoders to sense motion. Figure 1.5 illustrates a typical robotic manipulator that might be suitable for a laboratory benchtop. The figure emphasizes the data flow within the robotic system. In this system, the robot is usually equipped with rotary encoders that return measurements of angular motion at the joints to the controller and computer. In this particular system, a vision or laser tracking sensor is also configured to provide measurements of the end effector position and velocity. This measurement is returned to the computer to assist controlling the position tracking of the end effector along a desired trajectory. It should also be noted that this figure, while giving a general picture of the topology and connectivity of a robotic system, lacks many details that are necessary for a real robotic system. For example, the motor controllers are not shown in the figure, nor are the amplifiers or signal conditioners that may be required between the primary components.

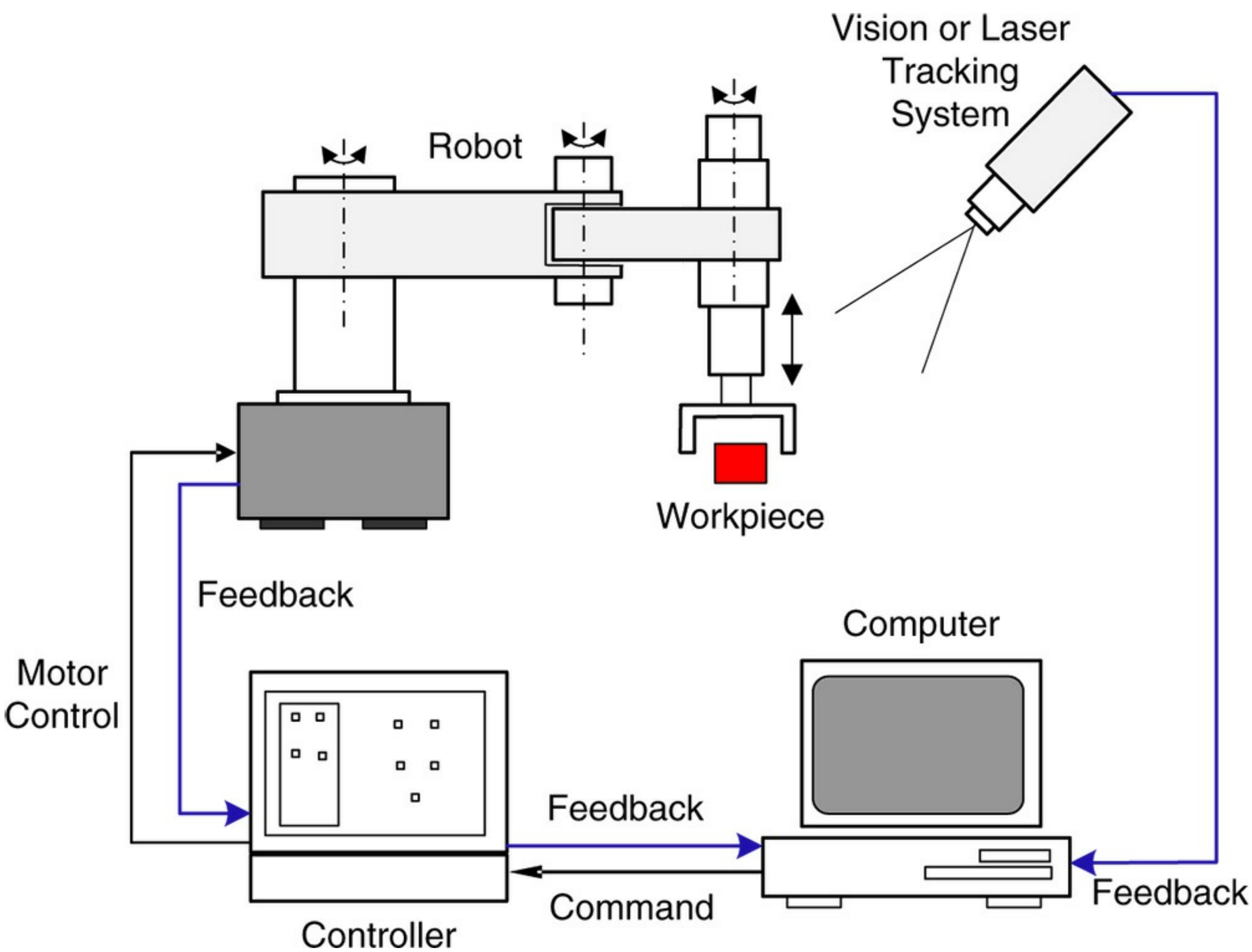

Figure 1.5 Typical robotic manipulator system components.

1.4 Robotic Manipulators

An important type of robotic system that is studied often in this text is the*robotic manipulator* or*robotic arm*. Robotic systems of this kind were some of the first to achieve widespread use in industry. As noted in the previous section, robotic manipulators have become a standard feature of modern assembly lines. They perform a host of tasks including welding, spraying, pick and place operations, drilling, cutting, and lifting. Many of the analytical techniques, modeling methodologies, and control strategies introduced in this text are demonstrated on examples that treat robotic manipulators. The reasons for this choice are numerous. Robotic manipulators are some of the simplest examples of practical robotic systems. Their study helps clarify the underlying principles and problems encountered when studying more complex systems. Although an autonomous marine vehicle may not resemble a robot on an assembly line, the general form of the mathematical problem that must be solved to control these two types of systems can be surprisingly similar. The same is true for modeling and control of autonomous ground or air vehicles. General methodologies applicable to one system can often be a starting point for the development of models and controllers for others. Moreover, it is often the case that a sub - system of an autonomous robotic system can be modeled or controlled using techniques developed for robotic manipulators. For example, the arms or legs of a humanoid robot or an imaging payload that actively controls the line of sight of a camera on an autonomous air vehicle may be modeled using techniques from robotic manipulators.

1.4.1 Typical Structure of Robotic Manipulators

Many robots consist of a number of individual bodies or*links* that are connected by*joints*. The individual bodies that make up the robot are often treated as rigid bodies, and that is the assumption throughout this text. However, for high speed or highly loaded mechanisms, elastic effects of the material body become significant and should be taken into consideration. The joints that connect the links in the robot can be quite complex and may themselves exhibit highly non - trivial mechanics including flexibility, hysteresis, backlash, or friction. An ideal joint is an interconnection between rigid bodies of a robotic system that allows only specific, predefined relative motions such as translation or rotation. Mathematically, an ideal joint imposes a kinematic constraint on the motion between rigid bodies that is based on the joint geometry. Common types of ideal joints include *revolute*, *prismatic*, *universal*, *spherical* or *screw joints*. Figure 1.6 depicts a few of these ideal joints and summarizes some of their properties.

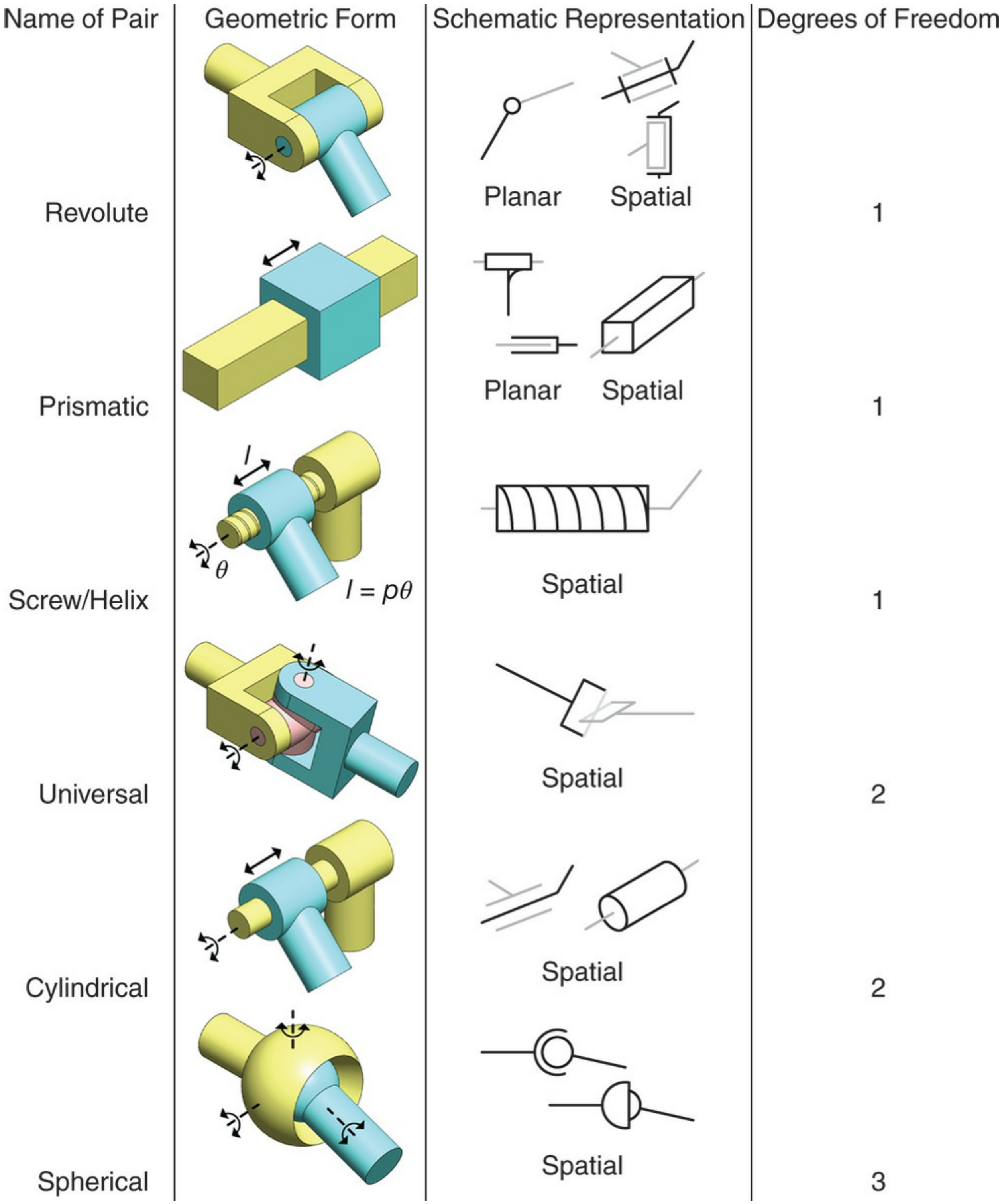

Figure 1.6 Ideal joints and their properties.

The two simplest types are the prismatic joint or revolute joint. Nearly all of the robotic systems studied in this text consist of these two types.

Many of the other types of ideal joints can be modeled by combining these two. For example, a universal joint consists of a pair of revolute joints with their joint axes orthogonal to one another. A prismatic joint allows only relative translation between two links along a prescribed axis, while the revolute joint permits only relative rotation about a prescribed axis.

An independent variable that is used to describe the motion of a robot, or the relative motion allowed by an ideal joint, is often called a*degree of freedom*. The number of degrees of freedom of an ideal joint is the number of independent variables required to model the relative motion that the joint permits. A robot has N *degrees of freedom* if it requires N independent variables to describe all of its possible configurations. The revolute and prismatic joints are consequently*single degree of freedom joints*. If the joint constraints are independent of one another, the number of degrees of freedom N for a general mechanism can be calculated as

$$N = \lambda(n-1) - \sum_{i=1}^{k}(\lambda - f_i) \tag{1.1}$$

where n is then number of links, k is the number of joints, and f_i is the number of degrees of freedom for joint i . For planar mechanisms $\lambda = 3$ and for spatial mechanisms $\lambda = 6$.

Example 1.1

Consider the*pantograph* mechanism depicted in Figure 1.7.

Figure 1.7 Pantograph mechanism.

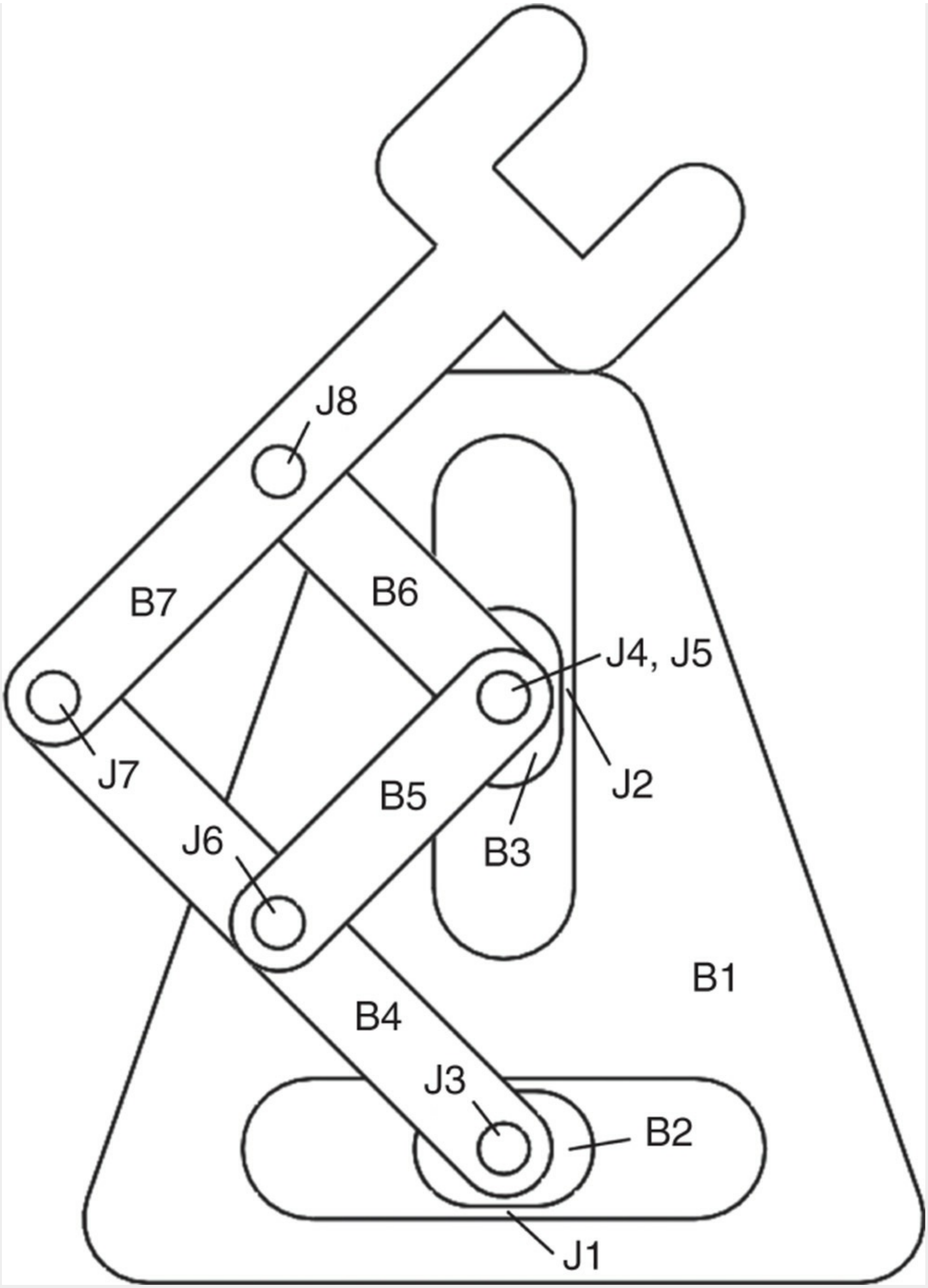

Figure 1.7 Pantograph mechanism.

When Equation ([1.1]) is applied to the pantograph in Figure 1.7, the values $\lambda = 3$, $n = 7$ (including the ground link), $k = 8$, and $f_i = 1$ for $i = 1, \ldots, 8$ are determined. It follows that the number of degrees of freedom $N = 2$.

More details on the properties of the ideal joints are presented in Chapters and . Precise mathematical definitions of the degrees of freedom for mechanical systems, and robots in particular, are discussed in Chapter .

1.4.2 Classification of Robotic Manipulators

Now that the basic definitions of links, joints, and degrees of freedom for typical robotic manipulators have been defined, a summary of different ways in which robots are classified is provided. Again, although this discussion focuses on robotic manipulators, some of the classifications are pertinent to other classes of robots. For example, classification of robots by driver technology and drive power applies equally well to all types of mobile robots whether they operate in the air, on land, under water, or on the water's surface.

1.4.2.1 Classification by Motion Characteristics

One of the most common means of differentiating among different robot architectures considers motion characteristics. A*planar manipulator* is one in which all the moving links in the mechanism perform planar motions that are parallel to one another. In contrast, a*spatial manipulator* is one in which at least one of the moving links demonstrates a general spatial motion. In other words $\lambda = 6$ in Equation (1.1). In some cases the manipulator is constructed so that only very specific kinds of motion are possible. A*spherical manipulator* is constructed so that the moving links perform spherical motions about a common stationary point. A*cylindrical manipulator* is constructed so that the end effector travels on the surface of a cylinder. More details of these two types of manipulators are discussed in Sections 1.4.3.2 and 1.4.3.4.

1.4.2.2 Classification by Degrees of Freedom

Another means of classifying robots is based on the number and type of degrees of freedom. A*general purpose robot* possesses $\lambda = 3$ degrees of freedom if it is a planar robot or $\lambda = 6$ degrees of freedom if it is spatial robot. A robot is*redundant* if it posses more than λ degrees of freedom. A redundant robot can be used to move around obstacles and operate in tightly confined spaces. A robot is*deficient* if it has less than λ degrees of freedom.

1.4.2.3 Classification by Driver Technology and Drive Power

Robots are often characterized by the nature and type of their drive technology. An*electric robot* employs DC servo motors or stepper motors. These robots have the advantage that they are clean and relatively easy to control. A*hydraulic robot* is preferred for tasks that require a large load carrying capacity. Care and maintenance is required to handle leaks and fluid compressibility problems. For high speed applications, a*pneumatic robot* is often preferred. These robots are generally clean, but can be hard to control due to challenges associated with air compressibility.

A*direct drive manipulator* is one in which each joint is driven directly by an actuator without any torque transmission mechanism. These drives can be bulky and heavy but do not exhibit backlash or drive flexibility, which can render robotic control more difficult. Finally, a*conventional manipulator* generates a driver torque that is magnified by a transmission mechanism. Usually this is achieved via gear reduction or by a harmonic drive unit. This design allows the use of smaller actuators. However, the gear mechanisms suffer from backlash, and the harmonic drives inherently exhibit flexibility effects.

1.4.2.4 Classification by Kinematic Structure

Kinematic structure is a topic of great importance to robotics and is yet another means that can be used to classify different types of robots. The kinematic structure of a robot results from its*system connectivity*. This topic has been studied extensively in multibody dynamics and has had a profound impact on robotics. The study of*multibody dynamics* is closely related to robotics, and strong references for the basic theory can be found in [14,46,24]. Many of the results discussed in this book can be considered as special cases within the general study of multibody dynamics. Generally speaking, the field of robotics is usually more concerned with problems of forward kinematics, inverse kinematics, or control synthesis, and the field of multibody dynamics tends to focus more on the study of numerical methods for approximations of the solution of the forward dynamics problem. It has been known for some time in the field of multibody dynamics that the*connectivity topology* of a system can have a dramatic influence on the complexity of simulating or deriving a control strategy for a system.

A robotic system is said to have the connectivity of a*kinematic chain* if there is one and only one connected path that traverses a system from the first to the last link. Such a robot is also often referred to as a*serial manipulator* or as a*open loop manipulator* in the robotics literature. A single arm or leg of a humanoid robot is a good example of a kinematic chain. Multibody systems that form a kinematic chain have the simplest connectivity topology. It is this class of robotic systems for which the richest collection of formulations and control strategies have been derived. The kinematics of chains is studied carefully in Chapter , their dynamics is studied in Chapters and , and their control is the topic of Chapters and .

A multibody system is said to have*tree topology connectivity* when it is built from an assembly of kinematic chains and no closed loops are formed by their interconnection. A full body humanoid robot or a space station in orbit are two familiar examples of systems having a tree topology connectivity. It is relatively straightforward to extend the techniques for modeling and control of kinematic chains to treat systems that have tree topology connectivity, although such methods must often be extended to account for the rigid body motion of the robotic system as a whole.

Finally, a robotic system is said to have*closed loop connectivity* whenever it is possible to construct a continuous path that starts at one link, traverses several other links, and finally connects to the original link. The multibody model of an autonomous ground vehicle is an example of a system that has closed loop topology if its suspension system has closed loops. Two robotic manipulators that cooperate in the task of lifting a large payload also form a system that has closed loop topology. The Stewart platform depicted in Figure 1.8 is a common robotic platform that has closed loop connectivity.

Robotic manipulator systems with closed loop connectivity are commonly referred to as*parallel manipulators* in the field of robotics. General robotic systems that have closed loop topology are not addressed in this introductory book.

Of course, some systems are constructed from sub - systems that constitute both open and closed loop chains. In some industrial manipulators, such as the Fanuc S - 900W, a four - bar push - rod linkage is used to drive the intermediate joints, which in turn are mounted on the robot base or waist. This design reduces the inertia of the manipulator. Such a system, which contains both open and closed loop chains as sub - systems, is known as a*hybrid manipulator*.

Figure 1.8 Industrial Stewart platforms.

In summary, robotic systems that have the form of a kinematic chain are the most basic; other more complicated robotic systems can be assembled from them. Methods for analyzing, simulating, or synthesizing a controller for kinematic chains can be applied to sub‐systems having more complex connectivity. Robotic manipulators serve as prototypical examples of robots that form kinematic chains.

1.4.2.5 Classification by Workspace Geometry

The last method for classifying the robots discussed looks at their*workspace geometry*. The*manipulator workspace* is the volume of space that the end effector can reach. The set of points where every point can be reached by the end effector in at least one orientation or pose is the*reachable workspace*. The set of points where every point can be reached by the end effector in all possible orientations or poses is called the*dextrous workspace*. By definition, it follows that the dexterous workspace is a subset of the reachable workspace. It should be noted that most industrial serial manipulators are designed with their first three moving links longer than the remaining links. These inner links are used primarily for controlling the end effector position. The remaining outboard links are used typically for controlling the end effector pose or orientation. Often, the sub‐assembly associated with the first three links is denoted the arm, and the remaining outboard links constitute the wrist. Figure 1.9 shows four common types of workspaces.