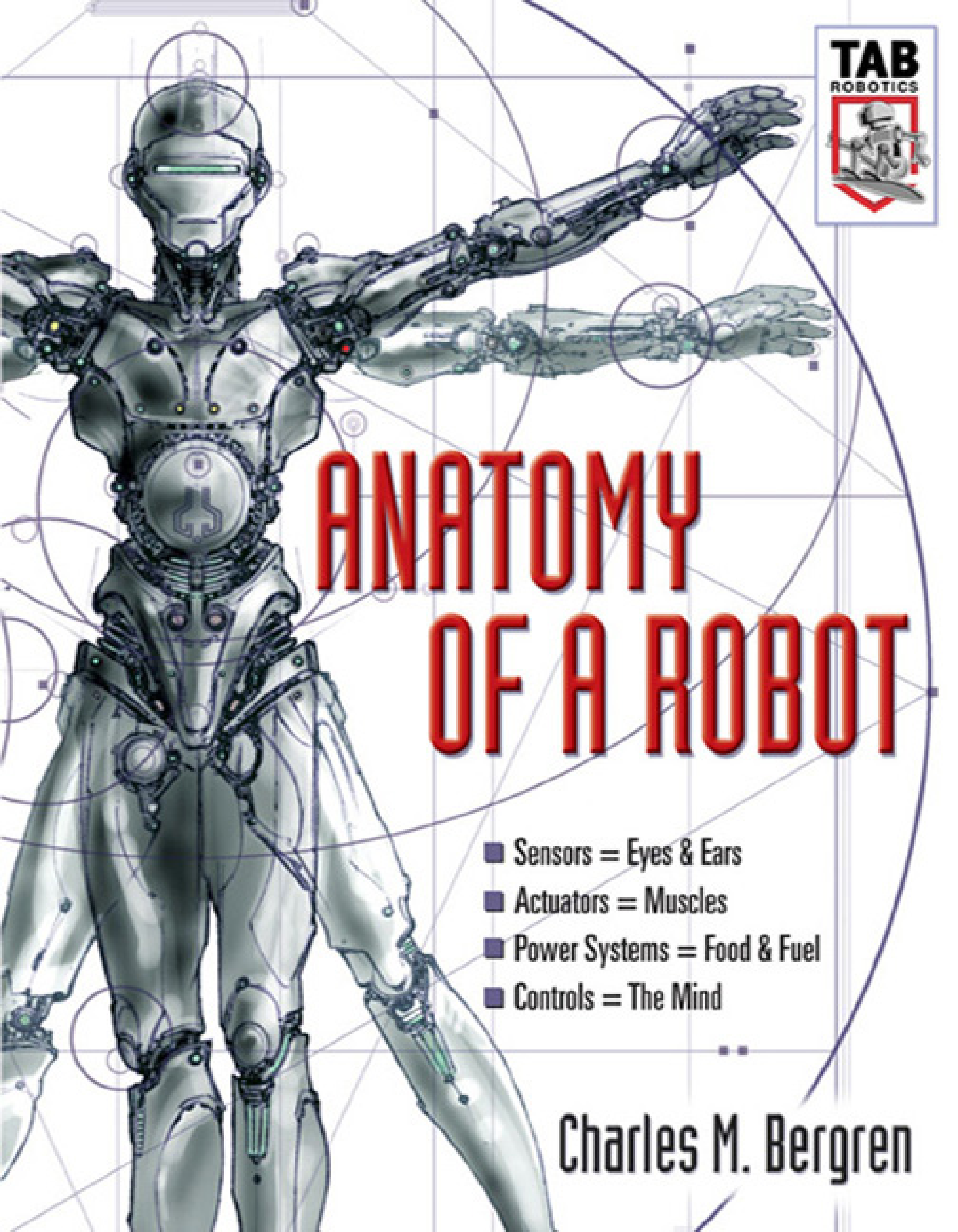
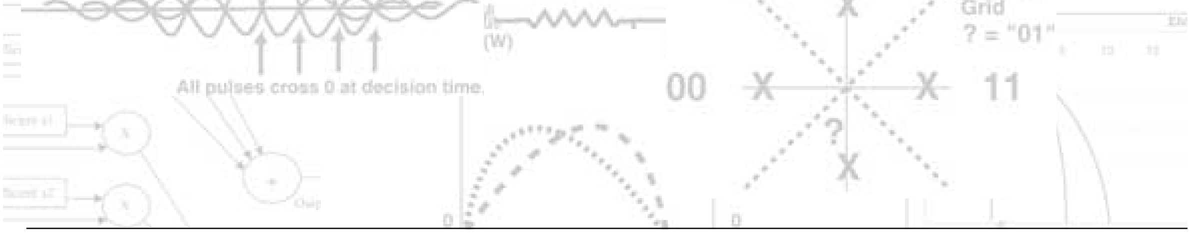
TAB ROBOTICS



ANATOMY OF A ROBOT

- Sensors = Eyes & Ears
- Actuators = Muscles
- Power Systems = Food & Fuel
- Controls = The Mind

Charles M. Bergren



ANATOMY OF A ROBOT

CHARLES M. BERGREN

McGraw-Hill

New York Chicago San Francisco Lisbon London Madrid
Mexico City Milan New Delhi San Juan Seoul
Singapore Sydney Toronto

The McGraw-Hill Companies

0-07-142930-1

The material in this eBook also appears in the print version of this title: 0-07-141657-9

TERMS OF USE

DOI: 10.1036/0071429301

To my son and my wonderful family

This page intentionally left blank.

CONTENTS

This page intentionally left blank.

PREFACE

Two years ago, I took my six-year-old son to a “robot race” up in the Rockies near Boulder. It was held in the community center of a small mountain town. Nevertheless, it was packed with about 100 enthusiastic people and many interesting exhibits. The central event was to be a timed race along a prescribed course. Several school-aged kids had entered plastic robots clearly built from parts from the same toy manufacturer. The racecourse was a plastic mat approximately 15 feet on each side. The robots had to follow a one-inch-wide, serpentine black line on the mat from beginning to end. The winner would be the robot finishing with the fastest time.

I watched the kids tuning up their robots on the racecourse before the race. Each robot had a sensor on each side that could detect the black line. If the robot moved forward and started to cross the line, the electronics would correct the steering and move the robot back on course.

It was clear the kids were all having trouble. None of the robots could follow the course from beginning to end. They would invariably lurch too far over the black racecourse line and get lost, spinning in useless circles. Legions of adult advisors huddled with the kids, making all sorts of changes, yet nobody was making progress. To me, the answer was obvious and I wanted to help.

Off in the corner, a bit cowed and unsure of himself, was the youngest competitor. Let's call him Sam. He may have been 13 and was there with his mom. They, too, were making changes without good results. I approached Sam's mom, discretely asked permission to help, and joined their team. Without going into the theory, I explained that the robots were all too fast and powerful for their own control systems. I recommended slowing down Sam's robot by adding more weight at the back end. We finally decided to build a sled for the robot to drag and set about finding the materials. With the race deadline approaching, Sam himself came up with the solution. With a quick glance to ask permission, he grabbed his mom's handheld camera and slipped the wrist strap over a post on the rear of the robot. We confirmed the robot could still move slowly down the racecourse line towing the camera. Sam took the batteries out of the camera until it was near the right weight. All too soon, race time came and we had to halt our experiment.

One after another, the older competitors' robots raced down the course only to stray off the black line and be disqualified. A couple of the robots did finish after wandering around lost and wasting a good deal of time. Eventually, the time came for Sam to race his robot. He placed his robot on the starting line, plopped his mom's camera down behind it, confidently put the wrist strap on the rear bumper, and pushed the start button with a bit of ceremony. As Sam's robot left the starting line, it lurched forward, tugging the camera behind it. The crowd started to buzz and I watched the highly amused advisors talking among themselves. It was clear some of them understood what was going on.

To make a long story short, Sam's robot reliably chugged around the racecourse and he won. The look on his face alone was worth the effort. Sam's nominal reward was a kit for a bigger robot, but I think he walked away with much more than that.

After the race, Sam was eager to know how I knew the solution. I took Sam aside and gave him a glimpse of the college-level mathematics and graphs that were behind his victory. My intention was to stimulate his curiosity and point him in the direction that would lead him to further accomplishments. I went home feeling wonderful, proud for myself, and happy for Sam.

After all, everyone seeks direction in life. We experience a feeling of comfort when we discover that our problems are definitive, comprehensible, and tractable. To build a successful robot, it takes a disciplined approach. Many pitfalls are possible, but they are not inevitable. The subjects you will have to master are many and difficult, but not incomprehensible.

To be clear, it is not the intention of my book to teach you how to build a robot. Others can find the nuts and bolts better than I, but if you want to come away enriched with the seminal knowledge of the academic and professional disciplines necessary to be successful in the field, then this book is for you. Each major discipline is the subject of a separate chapter. Each chapter will cover the basics but will also lead you to theory and reasoning that can capture the imagination. For each discipline, legions of engineers and professors spend their entire careers sweating the details.

Sam, if you're out there, I hope one of them is you.

INTRODUCTION

The boundless energy of youth often must give way to the laws of physics. All too often I've seen bright ideas flounder for a lack of fundamental knowledge. If this book can foster the development of the art, if it can encourage and educate the robotic community, if it can provide the missing ingredients—the secret sauce—then I did my job right. If you have a sense that a robot is more than wires and wheels, then this book is for you.

Math rules physics, and physics rules robots. This book sheds light on the math and physics behind a robot design, and does so in an accessible way. The text was written for all ages, from high school through college and beyond. The math used in the book includes algebra, calculus, and differential equations. For readers unacquainted with these subjects, I made sure the text "returns to Earth" often. Nobody should be left behind. The laws of physics and math are evident in everyday life, and several examples are given in this book. Throughout the history of science and technology, the path to great discoveries has almost always started with the observation of simple events. Newton's apple, Einstein's empty room in space, and Shannon's word games are clear examples. Proceeding from an intuitive, personal understanding of the basic laws of physics and math, you can take your understanding further. Using this knowledge, you can predict the behavior of your robot in advance. As problems crop up, you'll have the basic knowledge to move effectively toward solutions.

Throughout the book, I've also thrown in experience gained from 32 years of engineering design. I can't be there when you build your robot, but I can put tools in your belt and pass on such wisdom as we both can sit still for.

Originally, I started this project for the fame, fortune, and groupies. As the chapters rolled out, I got my true rewards. I relearned the basic technologies to better explain them. I dug into the larger questions lurking behind the equations and technology. And as the book developed, I found an outlet for other thoughts I've had for quite some time. I hope my philosophical asides prove entertaining.

The book is divided into chapters that deal with monolithic subjects like computer hardware, computer software, *digital signal processing* (DSP), communications, power, and control systems. It is my hope that readers will find these individual subjects compelling enough to pursue them further. In each chapter, I've included URLs for web sites that explain the technologies in more depth. The Web can be a great place to obtain a continuing education.

Chapter 1 covers project management. More robots bite the dust for a lack of management discipline than any other reason. Building robots is much like going into battle. You can do great damage coming straight out of the gate and swinging swords, but it takes planning to make sure only the enemy gets cut. The chapter outlines how to approach a robot project from the outset. It includes development process flowcharts, checklists, and lots of tips. Robots are not built; they are born. With forethought and preparation, the process can be much less painful. And lest we forget, the project depends on people. Motivation and management, of self and others, are required for success.

Chapter 2 covers control systems. This is a complex field with a language of its own and many disciplines. If someone were to gather data about why robot projects fail, I'm guessing mechanics and power problems would come first. Control system problems would be right up there, too. The chapter discusses control system architecture; distributed and centralized control systems are compared and contrasted. Most robots have centralized systems and use open-loop and closed-loop control methods. The text outlines the basic behavior of a second order-control system, a good model for the behavior of many robotic systems. The text explains the math needed to understand and control system behavior. Specific examples of ways to design and correct such a control system are also given. Last of all, I've thrown in all the tricks of the trade that I know.

Chapter 3 covers computer hardware. I've outlined many of the reasons for using a computer in a robot and ways to accelerate the design process. Several computer architectures are listed, including analog, general-purpose digital, DSP, neural networks, and parallel processors. I've outlined the basic architecture of general-purpose digital microprocessors and commented on the applicability of various computer options. Just as the lack of planning can ruin a robot project, so too can the wrong choice of microprocessor. The last part of the chapter has a large checklist that can help you through the process of selecting a computer.

Chapter 4 covers reliability, safety, and compliance. The first section defines reliability and provides methods for predicting and measuring it. The chapter also includes a list of components to be wary of and some advice about using them. In the safety section is a list of dangers that can sneak up on even the most experienced designers, and it also offers advice about managing risks. The compliance and testing section covers environmental considerations, emissions, and many tips for forestalling problems.

Chapter 5 covers the early stage of the design process, the *high-level design* (HLD). The text covers where to start, what to consider first, and how to make the design gel early. Although every robotic project will be different, I wanted the chapter to document how I would go about designing a robot. I closed my eyes, gave myself a phantom team of engineers, and wrote down what I'd do. Let me know if you'd do it differently.

Chapter 6 covers power and energy. First, I discuss how to determine the robot's energy requirements. It outlines a series of considerations that should be taken into account in the selection and use of an energy source, with a specific concentration on batteries.

Chapter 7 covers energy and software control systems, with an emphasis on energy management. It includes a list of specific actions to take in the design of an energy-efficient robot. I mentioned many considerations that should be kept in mind during the selection and design of robotic software. The chapter outlines a coordinated approach to the selection of a processor, a battery, a power supply, operating software, and application software. Included are many software techniques that have proven successful, including a discussion of braking methods.

Chapter 8 covers DSP and the chapter starts with an example of DSP processing that is familiar to all of us. This leads to the two basic theorems of DSP. Specific examples illustrate the need for both learning and using the theorems. The chapter includes different methods of constructing a classic DSP control system. I've included rules of thumb for picking components, methods for programming them, and ways to test them.

Chapter 9 covers communication, which is vital to the effectiveness and power of people, and robots are not far behind in this need. The chapter starts with the definition of communication, the concept of noise, and Shannon's theorem for the capacity of a noisy communications link. I discuss baseband transmission, the basic techniques for sending pulses down a wire, and the common baseband communication links, including the Ethernet. The chapter outlines the reasons for modulated communication and some of the methods for doing so. The emphasis is on the transmission of digital data and the control of errors in a noisy communication channel. I've explained several methods of encoding the data that make modern wireless communication possible. The chapter lists and explains many of the standard tools used by communication engineers, including coding, multiuser access, security, and compression. Lastly, I've described a few of the most popular communication protocols that can be used in a robot project.

Chapter 10 covers motors. Engineers classify motors by the type of power they consume. AC and DC motors (including stepping motors) are discussed along with the different internal structures that make them work. The advantages and disadvantages of each type are presented as well.

Chapter 11 covers mechanics and covers the selection and the relevant properties of materials. Many robots have mechanical problems, so several design tips are included. In addition, short sections are dedicated to static and dynamic calculations.

1

PROJECT MANAGEMENT

Act 1 Scene 1:

The graying professor stands in his graying tweed suit in an overly heated classroom with high windows and ceramic tiles on the wall. The asbestos-covered steam pipes clank and bang as he stares out from behind his ridiculously high podium over a classroom of eager, young robot builders seated in hard, creaking wooden chairs. There is a long silence until his lowers his glasses, leans forward, and slowly intones the following in his best Stentorian English.

"So you want to build a robot, do you? Well, I am reminded of a wonderful scene in the movie *Young Frankenstein* by Mel Brooks. The son of the famous Dr. Frankenstein is addressed in a conversation by his proper last name pronounced 'Frankenstein.' What follows is an embarrassing, slow, pregnant pause in the conversation. The young doctor leans forward and slowly corrects his friend, 'That's pronounced "frahnkensteen."

Just what is the fascination with robots anyway? If you remember nothing else in this book, remember this frahnkensteen phrase. Like no other, this technical field engenders passion.

It's important that you let that phrase sink in a couple of days before picking up a screwdriver. For from passion springs forces that we cannot understand. Love, joy, creativity, heartbreak, grief, and ruin all lurk to snare us as we move forward in this endeavor. And passion makes it all possible! Personally, I feel it's just as important to understand why I'm doing these things as it is to actually do them. I am old enough to realize that I will never fully understand my motives, nor should I. If I really found out exactly why I liked this field, the fantasy would probably be gone and I'd have to move on to something else.

Something is deliciously evil about trying to construct robots to carry out our bidding when we do not even know our own wishes and desires. Think about that. Have you ever seen the movie *Forbidden Planet*? It's a great, old science fiction movie partially based on Shakespeare's play *The Tempest*. I won't give away the movie's plot, but suffice it to say that a bright human gains control of a robot built by an advanced civilization. What ensues, as the robot carries out his new master's "will," is mayhem.

My point is this. Let me persuade you to stop and think first. Spend the time to analyze your motives and desires. Take the time to plan. This is not just a spiritual or psychological exercise, but it has a practical application and tangible rewards covering the spectrum from personal growth to the success of the project.

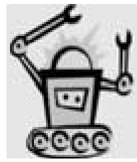
Taking this a step further, let me teach you something about the "nontechnical" art of project management first. It's a little known fact, but practicing a bit of project management makes it far less likely that your robot will run amuck and blow up the planet or that your family members will have to change their names to show their faces in public.

Project Management

Classically, a project is an endeavor to carry out some specific purpose. One English dictionary defines it as "a planned undertaking." We should note, for the record, that the Ape-English dictionary at www.ac.wwu.edu/~stephan/Tarzan/tarzan.dict.html has no entries for the words project, plan, or management. So if we are to maintain our species' lead over the apes, let's elevate our project management practices.

Why does a project require management? *Webster's* dictionary says a project requires planning. Webster did, after all, successfully finish his dictionary. Then again, we know of few people who have heeded Webster's advice in life. So let's look deeper than Webster's definition. Generally, a project has three elements: a deadline, a required outcome, and a budget. Maybe the project has no deadline, and maybe we don't know what the outcome is to be yet, but the project probably has a budget; any project always has some kind of financial limit, beyond which it will be cancelled. I'd like to make a case for having all three elements in the project.

The following discussion is based on project management processes used within a large company. The robot hobbyist, despite that fact that he or she wears all the hats in the project, should still perform the basic tasks of a *project manager* (PM). This is due to two reasons. First, the project will suffer if steps are skipped. Second, learning the art of being a PM is well worth it and will further any career.



The classic reason for managing a project is that some of the requirements will not otherwise be met. The truth is, even the most professional PMs have difficulty meeting all their goals at the same time. Half the time, a project will be late, be over budget, or fail to deliver the required results. If these goals were easy to attain, PMs would not be required in the first place. By implication, if no projects had PMs, the results would be much worse.

Many projects do not have formal PMs. Often, an engineer on the project handles some of the PM duties as a side task. Sometimes the PM duties are distributed among a few people, often with poor results. One person should be the PM and should be in complete charge of the project. That person should have all the powers and responsibilities of a PM. If you are the PM in your spare time, that's fine, as long as you can perform the tasks in the time you have to devote to the job.

First and foremost, a PM in a robot project is responsible for getting the robot done within all the restraints and requirements imposed at the outset. Certainly, a project can be executed and managed in almost any manner. To bring order to the situation, and to give all participants a clear picture of what's expected, it makes sense to use established methods and rules. The following discussion lays out the basics of project management processes but omits some of the details and reasoning to make it more readable.

Projects come in all shapes and sizes, and they are executed in all shapes and forms. This document provides a standard way to manage projects that is known to all responsible parties. It provides management tools that PMs can use to alter the course of a project and make corrections. This makes information easier to find, decreases the amount of negotiations involved, provides reliable channels of communication, and brings a level of comfort to all involved.

Project Process Flowchart

Figure 1-1 is a graphical representation of the various processes and procedures that will occur during the overall development cycle of the robot. The overall process is flexible, and deviations are acceptable as befits the situation. However, in general, deviations from the set process come at a sacrifice (see Figure 1-1).

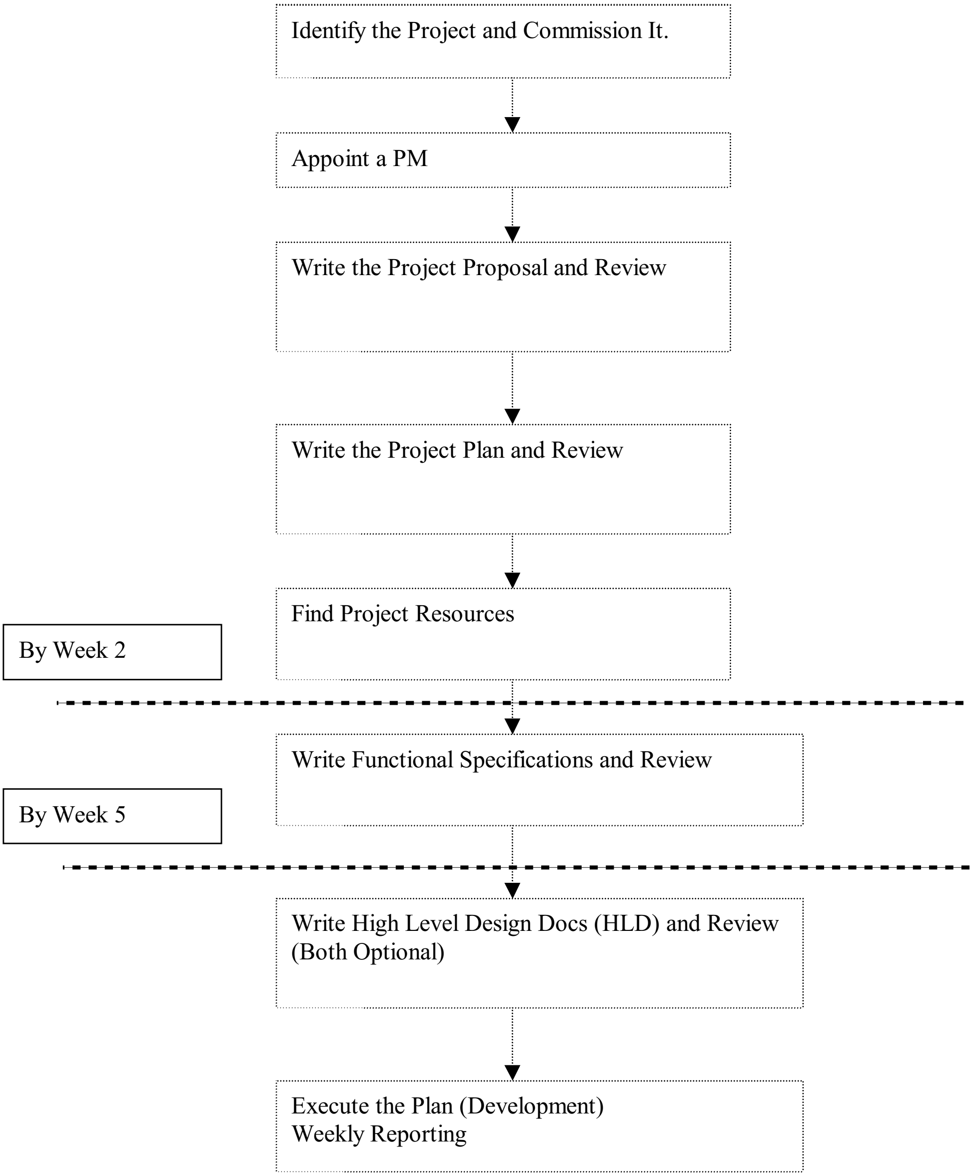
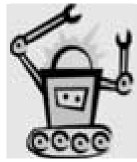


FIGURE 1-1 **Steps in managing a project**

How This Works When It's Implemented Right



In no particular order, these are some of the results and understandings that should come out of the proper application of this process.

The User's Manual for the "Boss"

The following words of advice pertain to the management of your robot development effort. If you are a lone robot hobbyist or operator, you are the management as well and should heed these general rules. This also applies to employees of a company involved in such a project.

PROJECTS ARE SHORT

Projects should be kept under six months. Ideally, most projects should be three to four months long at most. This means that any very long term robot projects should be broken up into a series of smaller projects. Divide the project up into functional blocks like power, chassis, control systems, and so on. This automatically engenders a complete review of all aspects of a long project at periodic intervals. By default, this includes the choice of PM, all project plans, all project resources, and so on. The following is a list of benefits that will accrue if short projects are the norm.

- PMs don't delay the project work while they get a long plan worked out. They can afford to make some mistakes over a shorter time period. These mistakes will be corrected in the next leg of the project.
- Long-term goals can be accomplished using a series of short-term goals and making corrections along the way.

PMS RUN THE PROJECTS

The PM is responsible for all aspects of the project after kickoff. "Management" might spawn the project and set the major goals, but it is the PM that runs with that information, makes a project proposal, makes a project plan (including the schedule, budget, resources, and so on.), finds the resources, executes the plan, builds the robot, and reports on a regular basis.

APPOINTING PMS

A PM must be well matched to a project. Don't overlook the fact that you might not be the right choice to be the PM! Some engineer make good PMs; others don't. The skill sets required for the two disciplines are much different.

KEEP THE PROJECT STABLE

Here are a few rules to observe:

- Don't change the tasks. Keep the specification (hereafter referred to as *spec*) stable after the project starts. The PM should give all parties the chance to change the spec up to the point when it is reviewed and development begins. If the spec must change, rewrite the project plan to accommodate the changes. Changing the spec is the second fastest way to scuttle a project's schedule and budget.
- Don't mess with the resources. Yes, this is the fastest way to scuttle a project's schedule and budget. Do not shift out resources once they have been allocated to a project. Don't borrow people, don't borrow equipment, and don't borrow space.

CORRECTIVE ACTIONS

When things are going well, about a third of all projects will still run into schedule or budget problems. Often, these projects can be identified early and corrective action can be taken. What can be done?

- Schedule a project review.
- Ask the PM for changes in the project plan, the project resources, and the project task as necessary.
- Change the PM. This is often a drastic solution, but it should not be avoided. Nor should the loss of a project be considered a significant black mark. Many new opportunities will arise for a PM to prove his or her mettle.
- Add more management. Sometimes a PM needs sub-PMs. This is often useful in large projects and can even be set up before the project starts.

The User's Manual for PMs

A checklist is provided at the end of this section that can serve as your guide throughout your robot project. The following paragraphs explain this checklist.