Linear Controller Design: Limits of Performance

Stephen Boyd **Craig Barratt**

Originally published 1991 by Prentice-Hall.

Contents

Preface

This book is motivated by the following technological developments: high quality integrated sensors and actuators, powerful control processors that can implement complex control algorithms, and powerful computer hardware and software that can be used to design and analyze control systems. We believe that these technological developments have the following ramifications for linear controller design:

- When many high quality sensors and actuators are incorporated into the design of a system, sophisticated control algorithms can outperform the simple control algorithms that have sufficed in the past.
- Current methods of computer-aided control system design underutilize available computing power and need to be rethought.

This book is one small step in the directions suggested by these ramifications.

We have several goals in writing this text:

- To give a clear description of how we might formulate the linear controller design problem, without regard for how we may actually solve it, modeling *fundamental specifications* as opposed to specifications that are artifacts of a particular method used to solve the design problem.
- To show that a wide (but incomplete) class of linear controller design problems can be cast as convex optimization problems.
- To argue that solving the controller design problems in this restricted class is in some sense fundamentally tractable: although it involves more computing than the standard methods that have "analytical" solutions, it involves much less computing than a global parameter search. This provides a partial answer to the question of how to use available computing power to design controllers.

- To emphasize an aspect of linear controller design that has not been emphasized in the past: the determination of *limits of performance*, *i.e.*, specifications that cannot be achieved with a given system and control configuration.

It is *not* our goal to survey recently developed techniques of linear controller design, or to (directly) teach the reader how to design linear controllers; several existing texts do a good job of that. On the other hand, a clear formulation of the linear controller design problem, and an understanding that many of the performance limits of a linear control system can be computed, are useful to the practicing control engineer.

Our intended audience includes the sophisticated industrial control engineer, and researchers and research students in control engineering.

We assume the reader has a basic knowledge of linear systems (Kailath [KAI80], Chen [CHE84], Zadeh and Desoer [ZD63]). Although it is not a prerequisite, the reader will benefit from a prior exposure to linear control systems, from both the "classical" and "modern" or state-space points of view. By *classical control* we refer to topics such as root locus, Bode plots, PI and lead-lag controllers (Ogata [OGA90], Franklin, Powell, Emami [FPE86]). By *state-space control* we mean the theory and use of the linear quadratic regulator (LQR), Kalman filter, and linear quadratic Gaussian (LQG) controller (Anderson and Moore [AM90], Kwakernaak and Sivan [KS72], Bryson and Ho [BH75]).

We have tried to maintain an informal, rather than completely rigorous, approach to the mathematics in this book. For example, in chapter 13 we consider linear functionals on infinite-dimensional spaces, but we do not use the term *dual space*, and we avoid any discussion of their continuity properties. We have given proofs and derivations only when they are simple and instructive. The references we cite contain precise statements, careful derivations, more general formulations, and proofs.

We have adopted this approach because we believe that many of the basic ideas are accessible to those without a strong mathematics background, and those with the background can supply the necessary qualifications, guess various generalizations, or recognize terms that we have not used.

A Notes and References section appears at the end of each chapter. We have not attempted to give a complete bibliography; rather, we have cited a few key references for each topic. We apologize to the many researchers and authors whose relevant work (especially, work in languages other than English) we have not cited. The reader who wishes to compile a more complete set of references can start by computing the transitive closure of ours, *i.e.*, our references along with the references in our references, and so on.

Our first acknowledgment is to Professor C. Desoer, who introduced the idea of the Q-parametrization (along with the advice, "this is good for CAD") to Stephen Boyd in EECS 290B at Berkeley in 1981. We thank the reviewers, Professor C. Desoer, Professor P. Kokotovic, Professor L. Ljung, Dr. M. Workman of IBM, and Dr. R. Kosut of Integrated Systems, Inc. for valuable suggestions. We are very grateful to Professor T. Higgins for extensive comments on the history and literature of control engineering, and a thorough reading of our manuscript. We thank S. Norman, a coauthor of the paper [BBN90], from which this book was developed, and V. Balakrishnan, N. Boyd, X. Li, C. Oakley, D. Pettibone, A. Ranieri, and Q. Yang for numerous comments and suggestions.

During the research for and writing of this book, the authors have been supported by the National Science Foundation (ECS-85-52465), the Office of Naval Research (N00014-86-K-0112), and the Air Force Office of Scientific Research (89-0228).

Stephen Boyd
Craig Barratt

Stanford, California
May, 1990

This book was typeset by the authors using LaTeX. The simulations, numerical computations, and figures were developed in MATLAB and C, under the UNIX operating system. We encourage readers to attempt to reproduce our plots and figures, and would appreciate hearing about any errors.

Chapter 1

Control Engineering and Controller Design

Controller design, the topic of this book, is only a part of the broader task of control engineering. In this chapter we first give a brief overview of control engineering, with the goal of describing the context of controller design. We then give a general discussion of the goals of controller design, and finally an outline of this book.

1.1 Overview of Control Engineering

The goal of control engineering is to improve, or in some cases enable, the performance of a system by the addition of *sensors*, *control processors*, and *actuators*. The sensors measure or sense various signals in the system and operator commands; the control processors process the sensed signals and drive the actuators, which affect the behavior of the system. A schematic diagram of a general *control system* is shown in figure 1.1.

This general diagram can represent a wide variety of control systems. The system to be controlled might be an aircraft, a large electric power generation and distribution system, an industrial process, a head positioner for a computer disk drive, a data network, or an economic system. The signals might be transmitted via analog or digitally encoded electrical signals, mechanical linkages, or pneumatic or hydraulic lines. Similarly the control processor or processors could be mechanical, pneumatic, hydraulic, analog electrical, general-purpose or custom digital computers.

Because the sensor signals can affect the system to be controlled (via the control processor and the actuators), the control system shown in figure 1.1 is called

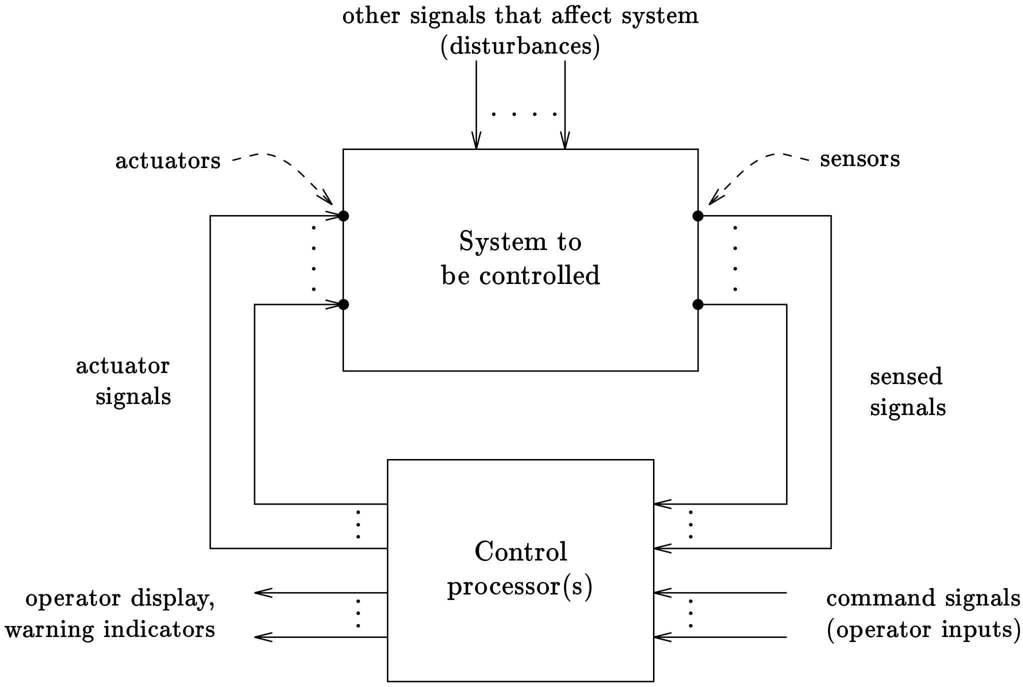

Figure 1.1 A schematic diagram of a general control system.

a *feedback* or *closed-loop* control system, which refers to the signal "loop" that circulates clockwise in this figure. In contrast, a control system that has no sensors, and therefore generates the actuator signals from the command signals alone, is sometimes called an *open-loop* control system. Similarly, a control system that has no actuators, and produces only operator display signals by processing the sensor signals, is sometimes called a *monitoring system.*

In industrial settings, it is often the case that the sensor, actuator, and processor signals are *boolean*, *i.e.* assume only two values. Boolean sensors include mechanical and thermal limit switches, proximity switches, thermostats, and pushbutton switches for operator commands. Actuators that are often configured as boolean devices include heaters, motors, pumps, valves, solenoids, alarms, and indicator lamps. Boolean control processors, referred to as *logic controllers*, include industrial relay systems, general-purpose microprocessors, and commercial *programmable logic controllers.*

In this book, we consider control systems in which the sensor, actuator, and processor signals assume real values, or at least digital representations of real values. Many control systems include both types of signals: the real-valued signals that we will consider, and boolean signals, such as fault or limit alarms and manual override switches, that we will not consider.

In control systems that use digital computers as control processors, the signals are sampled at regular intervals, which may differ for different signals. In some cases these intervals are short enough that the sampled signals are good approximations of the continuous signals, but in many cases the effects of this sampling must be considered in the design of the control system. In this book, we consider control systems in which all signals are continuous functions of time.

In the next few subsections we briefly describe some of the important tasks that make up control engineering.

1.1.1 System Design and Control Configuration

Control configuration is the selection and placement of the actuators and sensors on the system to be controlled, and is an aspect of system design that is very important to the control engineer. Ideally, a control engineer should be involved in the design of the system itself, even before the control configuration. Usually, however, this is not the case: the control engineer is provided with an already designed system and starts with the control configuration. Many aircraft, for example, are designed to operate without a control system; the control system is intended to improve the performance (indeed, such control systems are sometimes called *stability augmentation* systems, emphasizing the secondary role of the control system).

Actuator Selection and Placement

The control engineer must decide the type and placement of the actuators. In an industrial process system, for example, the engineer must decide where to put actuators such as pumps, heaters, and valves. The specific actuator hardware (or at least, its relevant characteristics) must also be chosen. Relevant characteristics include cost, power limit or authority, speed of response, and accuracy of response. One such choice might be between a crude, powerful pump that is slow to respond, and a more accurate but less powerful pump that is faster to respond.

Sensor Selection and Placement

The control engineer must also decide which signals in the system will be measured or sensed, and with what sensor hardware. In an industrial process, for example, the control engineer might decide which temperatures, flow rates, pressures, and concentrations to sense. For a mechanical system, it may be possible to choose *where* a sensor should be placed, *e.g.*, where an accelerometer is to be positioned on an aircraft, or where a strain gauge is placed along a beam. The control engineer may decide the particular type or relevant characteristics of the sensors to be used, including the type of transducer, and the signal conditioning and data acquisition hardware. For example, to measure the angle of a shaft, sensor choices include a potentiometer, a rotary variable differential transformer, or an 8-bit or 12-bit absolute or differential shaft encoder. In many cases, sensors are smaller than actuators, so a change of sensor hardware is a less dramatic revision of the system design than a change of actuator hardware.

There is not yet a well-developed theory of actuator and sensor selection and placement, possibly because it is difficult to precisely formulate the problems, and possibly because the problems are so dependent on available technology. Engineers use experience, simulation, and trial and error to guide actuator and sensor selection and placement.

1.1.2 Modeling

The engineer develops mathematical models of

- the system to be controlled,
- noises or disturbances that may act on the system,
- the commands the operator may issue,
- desirable or required qualities of the final system.

These models might be deterministic (*e.g.*, ordinary differential equations (ODE's), partial differential equations (PDE's), or transfer functions), or stochastic or probabilistic (*e.g.*, power spectral densities).

Models are developed in several ways. *Physical modeling* consists of applying various laws of physics (*e.g.*, Newton's equations, energy conservation, or flow balance) to derive ODE or PDE models. *Empirical modeling* or *identification* consists of developing models from observed or collected data. The a priori assumptions used in empirical modeling can vary from weak to strong: in a "black box" approach, only a few basic assumptions are made, for example, linearity and time-invariance of the system, whereas in a physical model identification approach, a physical model structure is assumed, and the observed or collected data is used to determine good values for these parameters. Mathematical models of a system are often built up from models of subsystems, which may have been developed using different types of modeling.

Often, several models are developed, varying in complexity and fidelity. A simple model might capture some of the basic features and characteristics of the system, noises, or commands; a simple model can simplify the design, simulation, or analysis of the control system, at the risk of inaccuracy. A complex model could be very detailed and describe the system accurately, but a complex model can greatly complicate the design, simulation, or analysis of the system.

1.1.3 Controller Design

Controller design is the topic of this book. The *controller* or *control law* describes the algorithm or signal processing used by the control processor to generate the actuator signals from the sensor and command signals it receives.

Controllers vary widely in complexity and effectiveness. Simple controllers include the *proportional* (P), the *proportional plus derivative* (PD), the *proportional plus integral* (PI), and the *proportional plus integral plus derivative* (PID) controllers, which are widely and effectively used in many industries. More sophisticated controllers include the *linear quadratic regulator* (LQR), the estimated-state-feedback controller, and the *linear quadratic Gaussian* (LQG) controller. These sophisticated controllers were first used in state-of-the-art aerospace systems, but are only recently being introduced in significant numbers.

Controllers are designed by many methods. Simple P or PI controllers have only a few parameters to specify, and these parameters might be adjusted empirically, while the control system is operating, using "tuning rules". A controller design method developed in the 1930's through the 1950's, often called *classical controller design*, is based on the 1930's work on the design of vacuum tube feedback amplifiers. With these heuristic (but very often successful) techniques, the designer attempts to synthesize a compensation network or controller with which the closed-loop system performs well (the terms "synthesize", "compensation", and "network" were borrowed from amplifier circuit design).

In the 1960's through the present time, state-space or "modern" controller design methods have been developed. These methods are based on the fact that the solutions to some optimal control problems can be expressed in the form of a feedback law or controller, and the development of efficient computer methods to solve these optimal control problems.

Over the same time period, researchers and control engineers have developed methods of controller design that are based on extensive computing, for example, numerical optimization. This book is about one such method.

1.1.4 Controller Implementation

The signal processing algorithm specified by the controller is implemented on the control processor. Commercially available control processors are generally restricted to logic control and specific types of control laws such as PID. Custom control processors built from general-purpose microprocessors or analog circuitry can implement a very wide variety of control laws. General-purpose *digital signal processing* (DSP) chips are often used in control processors that implement complex control laws. Special-purpose chips designed specifically for control processors are also now available.

1.1.5 Control System Testing, Validation, and Tuning

Control system testing may involve:

- extensive computer simulations with a complex, detailed mathematical model,
- real-time simulation of the system with the actual control processor operating ("hardware in the loop"),
- real-time simulation of the control processor, connected to the actual system to be controlled,
- field tests of the control system.

Often the controller is modified after installation to optimize the actual performance, a process known as tuning.

1.2 Goals of Controller Design

A well designed control system will have desirable performance. Moreover, a well designed control system will be tolerant of imperfections in the model or changes that occur in the system. This important quality of a control system is called *robustness.*

1.2.1 Performance Specifications

Performance specifications describe how the closed-loop system should perform. Examples of performance specifications are:

- *Good regulation against disturbances.* The disturbances or noises that act on the system should have little effect on some critical variables in the system. For example, an aircraft may be required to maintain a constant bearing despite wind gusts, or the variations in the demand on a power generation and distribution system must not cause excessive variation in the line frequency. The ability of a control system to attenuate the effects of disturbances on some system variables is called *regulation.*
- *Desirable responses to commands.* Some variables in the system should respond in particular ways to command inputs. For example, a change in the commanded bearing in an aircraft control system should result in a change in the aircraft bearing that is sufficiently fast and smooth, yet does not excessively overshoot or oscillate.
- *Critical signals are not too big.* Critical signals always include the actuator signals, and may include other signals in the system. In an industrial process

control system, for example, an actuator signal that goes to a pump must remain within the limits of the pump, and a critical pressure in the system must remain below a safe limit.

Many of these specifications involve the notion that a signal (or its effect) is small; this is the subject of chapters 4 and 5.

1.2.2 Robustness Specifications

Robustness specifications limit the change in performance of the closed-loop system that can be caused by changes in the system to be controlled or differences between the system to be controlled and its model. Such *perturbations* of the system to be controlled include:

- The characteristics of the system to be controlled may change, perhaps due to component drift, aging, or temperature coefficients. For example, the efficiency of a pump used in an industrial process control system may decrease, over its life time, to 70% of its original value.
- The system to be controlled may have been inaccurately modeled or identified, possibly intentionally. For example, certain structural modes or nonlinearities may be ignored in an aircraft dynamics model.
- Gross failures, such as a sensor or actuator failure, may occur.

Robustness specifications can take several forms, for example:

- *Low differential sensitivities.* The derivative of some closed-loop quantity, with respect to some system parameter, is small. For example, the response time of an aircraft bearing to a change in commanded bearing should not be very sensitive to aerodynamic pressure.
- *Guaranteed margins.* The control system must have the ability to meet some performance specification despite some specific set of perturbations. For example, we may require that the industrial process control system mentioned above continue to have good regulation of product flow rate despite any decrease in pump effectiveness down to 70%.

1.2.3 Control Law Specifications

In addition to the goals and specifications described above, there may be constraints on the control law itself. These *control law specifications* are often related to the implementation of the controller. Examples include:

- The controller has a specific form, *e.g.*, PID.

- The controller is linear and time-invariant (LTI).
- In a control system with many sensors and actuators, we may require that each actuator signal depend on only one sensor signal. Such a controller is called *decentralized*, and can be implemented using many noncommunicating control processors.
- The controller must be implemented using a particular control processor. This specification limits the complexity of the controller.

1.2.4 The Controller Design Problem

Once the system to be controlled has been designed and modeled, and the designer has identified a set of design goals (consisting of performance goals, robustness requirements, and control law constraints), we can pose the controller design problem:

> **The controller design problem:** Given a model of the system to be controlled (including its sensors and actuators) and a set of design goals, find a suitable controller, or determine that none exists.

Controller design, like all engineering design, involves tradeoffs; by *suitable*, we mean a satisfactory compromise among the design goals. Some of the tradeoffs in controller design are intuitively obvious: *e.g.*, in mechanical systems, it takes larger actuator signals (forces, torques) to have faster responses to command signals. Many other tradeoffs are not so obvious.

In our description of the controller design problem, we have emphasized the determination of whether or not there is any controller that provides a suitable tradeoff among the goals. This aspect of the controller design problem can be as important in control engineering as finding or synthesizing an appropriate controller when one exists. If it can be determined that no controller can achieve a suitable tradeoff, the designer must:

- relax the design goals, or
- redesign the system to be controlled, for example by adding or relocating sensors or actuators.

In practice, existing controller design methods are often successful at finding a suitable controller, when one exists. These methods depend upon talent, experience, and a bit of luck on the part of the control engineer. If the control engineer is successful and finds a suitable controller, then of course the controller design problem has been solved. However, if the control engineer *fails* to design a suitable controller, then he or she *cannot* be sure that there is no suitable controller, although the control engineer might suspect this. Another design approach or method (or indeed, control engineer) could find a suitable controller.