1
Introduction

1.1 What Is This Book About?

Control theory is the engineering science of manipulating physical systems to achieve the desired functionality. Its history is centuries old and spans rich areas from the mathematical sciences. A brief list includes differential equations, operator theory and linear algebra, functional analysis, harmonic analysis, and more. However, it mostly does not concern modern algorithmic theory or computational complexity from computer science.

The advance of the deep learning revolution has brought to light the power and robustness of algorithms based on gradient methods. The following quote is attributed to deep learning pioneer Yann Lecun during a Barbados workshop on deep learning theory circa February 2020.

"Control = reinforcement learning with gradients."

This book concerns this exact observation: *How can we exploit the differentiable structure in natural physical environments to create efficient and robust gradient-based algorithms for control?*

The answer we give is **not** to apply gradient descent to any existing algorithm. Rather, we rethink the paradigm of control from the start. What can be achieved in a given dynamical system, whether known or unknown, fully observed or not, linear or nonlinear, and without assumptions on the noise sequence?

This is a deep and difficult question that we fall short of answering. However, we do propose a different way of answering this question: We define the control problem from the point of view of an online player in a repeated game. This gives rise to a simple but powerful framework called nonstochastic control theory. The framework is a natural fit to applying techniques from online convex optimization, resulting in new, efficient, gradient-based control methods.

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press

1.2 The Origins of Control

The industrial revolution was in many ways shaped by the steam engine that came to prominence in the eighteenth century. One of its most critical components was introduced by the Scottish inventor James Watt: the *centrifugal governor*, which is amongst the earliest examples of a control mechanism. A major source of efficiency losses of the earlier designs was the difficulty in regulating the engine. Too much or too little steam could easily cause engine failure. The centrifugal governor was created to solve this problem (Figure 1.1).

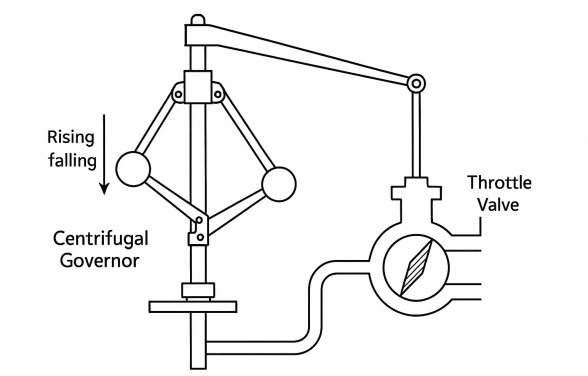


Figure 1.1 A centrifugal governor.

The centrifugal governor consists of a shaft connected to two metal balls that are allowed to rotate. The increased shaft speed causes the balls to spin faster, and the resulting centrifugal force causes the balls to lift. The balls are in turn attached to a pin that controls the amount of steam entering the engine, so as the balls spin faster, the pin lifts and limits the amount of fuel that can enter the system. However, if the balls spin slower, the pin is lowered, allowing more steam to enter the system. This provides natural, direct, and proportional control to the system.

The impact of the centrifugal governor on the steam engine, and in turn on the industrial revolution, illustrates the importance of the field of control in engineering and the sciences. Since the eighteenth century, our world has changed quite a bit, and the importance of control has only grown.

The challenges of controlling systems have moved from the steam engine to the design of aircraft, robotics, and autonomous vehicles. The importance of designing efficient and robust methods remains as crucial as in the industrial revolution.

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press

1.3 Formalization and Examples of a Control Problem

With the physical example of the centrifugal governor in mind, we give a mathematical definition of a control problem. The first step is to formally define a dynamical system in the following way.

Definition 1.1 (Dynamical system) A dynamical system evolving from an initial state $\mathbf{x}_1$ is given by the following equations:

$$\mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t), \quad \mathbf{y}_t = g_t(\mathbf{x}_t).$$

Here $\mathbf{x}_t \in \mathbb{R}^{d_x}$ is the state of the system, $\mathbf{y}_t \in \mathbb{R}^{d_y}$ is the observation, $\mathbf{u}_t \in \mathbb{R}^{d_u}$ is the control input, and $\mathbf{w}_t \in \mathbb{R}^{d_w}$ is the perturbation.

The function g_t is a mapping from the state to the observed state, and the function f_t is the transition function, given the current state $\mathbf{x}_t$, the input control $\mathbf{u}_t$, and the perturbation $\mathbf{w}_t$. The subscript indicates the relevant quantities at time step t.

Given a dynamical system, there are several possible objectives that we can formulate. We give a generic formulation below that captures many common control formulations, notably the optimal control problem and the robust control problem. These will be addressed in much more detail later in this book. In this problem formulation, the goal is to minimize the long-term horizon cost given by a sequence of cost functions.

Definition 1.2 (A generic control problem) Given a dynamical system with starting state $(\mathbf{x}_1, \mathbf{y}_1)$, iteratively construct controls u_t to minimize a sequence of loss functions $c_t(\mathbf{y}_t, \mathbf{u}_t)$, $c_t : \mathbb{R}^{d_y+d_u} \to \mathbb{R}$. This is given by the following mathematical program:

$$\min_{\mathbf{u}_{1:T}} \sum_t c_t(\mathbf{y}_t, \mathbf{u}_t)$$
$$s.t. \ \mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t), \quad \mathbf{y}_t = g_t(\mathbf{x}_t).$$

At this point, many aspects of the generic problem have not been specified. These include:

- Are the transition functions and observation functions known or unknown to the controller?
- How accurately is the transition function known?
- What is the mechanism that generates the perturbations $\mathbf{w}_t$?
- Are the cost functions known or unknown ahead of time?

Different theories of control differ by the answers they give to the above questions, and these in turn give rise to different methods.

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press

Instead of postulating a hypothesis about how our world should behave, we proceed to give an intuitive example of a control problem. We can then reason about the different questions and motivate the nonstochastic control setting.

1.3.1 Example: Control of a Medical Ventilator

A recent relevant example of control is in the case of medical ventilators. The ventilator connects the trachea in the lungs to a central pressure chamber, which helps maintain positive-end expiratory pressure (PEEP) to prevent the lungs from collapsing and simulate healthy breathing in patients with respiratory problems.

The ventilator must take into account the structure of the lung to determine the optimal pressure to induce. Such structural factors include *compliance*, or the change in lung volume per unit pressure, and *resistance*, or the change in pressure per unit flow.

A simplistic formalization of the dynamical system of the ventilator and lung can be derived from the physics of a connected two-balloon system. The dynamics equations can be written as

$$p_t = C_0 + C_1 v_t^{-1/3} + C_2 v_t^{5/3},$$
$$v_{t+1} = v_t + \Delta_t u_t,$$

where p_t is the observed pressure, v_t is the volume of the lung, u_t is the input flow (control), and Δ_t is the unit of time resolution of the device. C_0, C_1, and C_2 are constants that depend on the particular system/lung–ventilator pair.

The goal of ventilator control is to regulate the pressure according to some predetermined pattern. As a dynamical system, we can write $v_{t+1} = f(v_t, u_t, w_t)$, $p_t = g(v_t)$, and we can define the cost function to be $c_t(p_t, u_t) = |p_t - p_t^*|^2$, where p_t^* is the desired pressure at time t. The goal, then, is to choose a sequence of control signals $\{u_t\}_{t=1}^T$ to minimize the total cost over T iterations.

In Figure 1.2 we see an illustration of a simple controller, which we will soon describe, called the PID controller, as it performs on a ventilator and lung simulator.

1.4 Simple Control Algorithms

The previous section detailed the importance of control problems, specifically through the example of a critical care medical device. In this section, we detail

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press

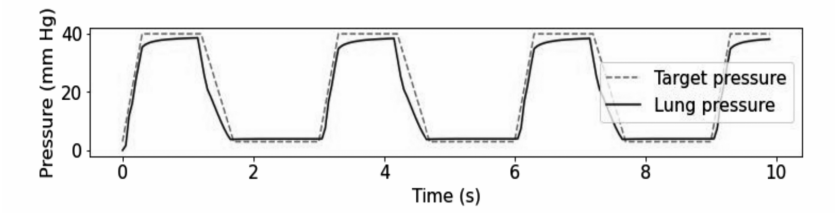


Figure 1.2 Performance of the PID controller on a mechanical ventilator, from Suo et al. (2021).

methods for controlling a medical ventilator, or indeed any physical device that can be modeled as a dynamical system.

The following methods are simplistic but still widely used. For example, the PID algorithm is still used in most medical ventilators today. In later sections, we describe a modern optimization-based framework that yields more sophisticated algorithms that are capable of better performance.

1.4.1 The Bang-Bang Controller

Bang-bang control is one of the simplest control strategies. Consider a scalar dynamical system, where it is desirable to keep the state x_t within a prespecified dynamic range $[x_{\min}, x_{\max}]$ (which can vary over time) and where the scalar control input u_t must at all times be within the interval $[u_{\min}, u_{\max}]$, which is assumed to contain zero for simplicity. A bang-bang controller chooses the control input at time t as

$$u_t = \mathbf{1}_{\{x_t < x_{\min}\}} \times u_{\max} + \mathbf{1}_{\{x_t > x_{\max}\}} \times u_{\min}.$$

A bang-bang controller saturates its inputs to extreme values in an attempt to confine the state of the system to a desirable range. For example, in the context of mechanical ventilation, one could choose $[p_t^* - \Delta p, p_t^* + \Delta p]$ as a reasonable region around the prescribed pressure p_t^* as a desirable range for p_t. In such a case, a bang-bang controller would allow oxygen to enter at the highest possible rate whenever the pressure drops below $p_t^* - \Delta p$.

This discussion highlights two unsatisfactory aspects of the bang-bang control scheme: one, it ignores the cost associated with acting with the extreme values of the control input; two, it produces oscillatory and rapidly changing behavior in the control state.

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press

1.4.2 The PID Controller

A more sophisticated control algorithm is the PID control method. Consider a control method that chooses the control to be a linear function of the observed states, that is:

$$\mathbf{u}_t = \sum_{i=0}^{k} A_i \mathbf{x}_{t-i},$$

for some coefficients A_i, where we consider only the $k+1$ most recent states. One of the most useful special cases of this linear family is the PID controller.

A special subfamily of linear controllers is that of linear controllers that optimize on a predefined basis of historical observations. More precisely,

(i) Proportional control (e.g., centrifugal governor): $\mathbf{u}_t = \alpha_0 \mathbf{x}_t$.
(ii) Integral control: $\mathbf{u}_t = \beta \sum_{i<t} \mathbf{x}_{t-i}$.
(iii) Derivative control: $\mathbf{u}_t = \gamma(\mathbf{x}_t - \mathbf{x}_{t-1})$.

The PID controller, named after the nature of the three components that make up it, specifies three coefficients α, β, γ to generate a control signal. In that sense, it is extremely sparse, which from a learning-theoretic perspective is a good indication of generalization.

In fact, the PID controller is the method of choice for numerous engineering applications, including control of medical ventilators.

1.5 Classical Theory: Optimal and Robust Control

Classical theories of control differ within the generic control problem formulation 1.2 primarily in one significant aspect: the perturbation model. A rough divide between the two main theories – *optimal control* and *robust control* – is as follows.

Optimal control postulates a probabilistic model for the perturbations, which are typically independent and identically distributed (i.i.d.) draws from a Gaussian distribution. However, other types of distribution can also be considered. As such, the control problem can be rephrased in a more precise stochastic mathematical program,

$$\min_{\mathbf{u}_{1:T}} \mathop{\mathbf{E}}_{\mathbf{w}_{1:T}} \left[\sum_t c_t(\mathbf{y}_t, \mathbf{u}_t) \right]$$
$$s.t. \;\; \mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t), \quad \mathbf{y}_t = g(\mathbf{x}_t).$$

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press

In contrast, robust control theory plans for the worst-case noise from a given set of constraints, denoted $\mathcal{K}$. Thus, the robust control problem can be rephrased as the following mathematical program:

$$\min_{\mathbf{u}_{1:T}} \max_{\mathbf{w}_{1:T} \in \mathcal{K}} \left[\sum_t c_t(\mathbf{y}_t, \mathbf{u}_t) \right]$$
$$s.t. \ \ \mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t), \quad \mathbf{y}_t = g(\mathbf{x}_t).$$

The two formulations have been debated for decades, with pros and cons for each side. Optimal control focuses on the average case and often yields simpler formulations for optimization, resulting in more efficient methods. Robust control, in contrast, allows for more difficult noise models and some amount of model misspecification. However, it is too pessimistic for many natural problems.

Nonstochastic control theory takes the best of both worlds, simultaneously granting robustness to adversarial noise and allowing the use of optimistic behavior when appropriate while providing computationally efficient methods.

1.6 The Need for a New Theory

Consider the problem of flying a drone from the source to the destination, subject to unknown weather conditions. The aerodynamics of flight can be modeled sufficiently well by time-varying linear dynamical systems, and existing techniques are perfectly capable of doing a great job for indoor flight. However, the wind conditions, rain, and other uncertainties are a different story. Certainly, the wind is not an i.i.d. Gaussian random variable! Optimal control theory, which assumes this zero-mean noise, is therefore overly optimistic and inaccurate for our problem.

The designer might resort to robust control theory to account for all possible wind conditions, but this is overly pessimistic. What if we encounter a bright sunny day after all? Planning for the worst case would mean slow and fuel-inefficient flight.

We would like an adaptive control theory that allows us to attain the best of both worlds: an efficient and fast flight when the weather permits, and careful drone maneuvering when this is required by the conditions. Can we design a control theory that will allow us to take into account the specific instance perturbations and misspecifications, and give us finite-time provable guarantees? This is the subject of online nonstochastic control!

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press

1.6.1 Online Nonstochastic Control Theory

This book is concerned with nonstochastic noise. When dealing with nonstochastic, that is, arbitrary or even adversarial, perturbations, the optimal policy, that is, a decision-making rule mapping observations to control inputs, is not clear a priori. Rather, an optimal policy for the observed perturbations can be determined *only in hindsight*. This is a significant difference from optimal and robust control theory, where the mapping from states to actions can be completely predetermined based upon the noise model.

Since optimality is defined only in hindsight, the goal is not predetermined. We thus shift to a different performance metric borrowed from the world of learning in repeated games, namely regret. Specifically, we consider regret with respect to a reference class of policies. In online control, the controller iteratively chooses a control $\mathbf{u}_t$. The controller then observes the next state of the system $\mathbf{x}_{t+1}$ and suffers a loss of $c_t(\mathbf{x}_t, \mathbf{u}_t)$ according to an adversarially chosen loss function. For simplicity, we consider the case of full state observation where $\mathbf{y}_t = \mathbf{x}_t$. Let $\Pi = \{\pi : \mathbf{x} \to \mathbf{u}\}$ be a class of policies. The regret of the controller with respect to Π is defined as follows.

Definition 1.3 Given a dynamical system, the regret of an online control algorithm $\mathcal{A}$ over T iterations with respect to a class of policies Π is given by

$$\text{regret}_T(\mathcal{A}, \Pi) = \max_{\mathbf{w}_{1:T}:\|\mathbf{w}_t\|\le 1} \left(\sum_{t=1}^{T} c_t(\mathbf{x}_t, \mathbf{u}_t) - \min_{\pi\in\Pi} \sum_{t=1}^{T} c_t(\mathbf{x}_t^\pi, \mathbf{u}_t^\pi) \right),$$

where $\mathbf{u}_t = \mathcal{A}(\mathbf{x}_t)$ are the controls generated by $\mathcal{A}$, and $\mathbf{x}_t^\pi, \mathbf{u}_t^\pi$ are the counterfactual state sequence and controls under the policy π, that is:

$$\mathbf{u}_t^\pi = \pi(\mathbf{x}_t^\pi),$$
$$\mathbf{x}_{t+1}^\pi = f_t(\mathbf{x}_t^\pi, \mathbf{u}_t^\pi, \mathbf{w}_t).$$

Henceforth, if T, Π, and $\mathcal{A}$ are clear from the context, we drop these symbols when defining regret.

At this point, the reader may wonder why we should compare to a reference class, as opposed to all possible policies. There are two main answers to this question:

(i) First, the best policy in hindsight out of all possible policies may be very complicated to describe and to reason about. This argument was made by the mathematician Tyrell Rockafellar; see the bibliographic section for more details.

(ii) Second, it can be shown that it is impossible, in general, to obtain sublinear regret versus the best policy in hindsight. Rather, a different performance

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press

metric called *competitive ratio* can be analyzed, as we discuss later in the book.

The nonstochastic control problem can now be stated as finding an efficient algorithm that minimizes the worst-case regret vs. meaningful policy classes that we examine henceforth. It is important to mention that *the algorithm does not have to belong to the comparator set of policies!* Indeed, in the most powerful results, we will see that the algorithm will learn and operate over a policy class that is strictly larger than the comparator class.

1.6.2 A New Family of Algorithms

The policy regret minimization viewpoint in control leads to a new approach to algorithm design. Instead of a priori computation of the optimal controls, online convex optimization suggests modifying the controller according to the costs and dynamics, in the flavor of adaptive control, in order to achieve provable bounds on the policy regret.

As a representative example, consider the *gradient perturbation controller* described in Algorithm 1.1. The algorithm maintains matrices that are used to create the control as a linear function of past perturbations, according to line 3. These parameters change over time according to a gradient-based update rule, where the gradient is calculated as the derivative of a counterfactual cost function with respect to the parametrization, according to line 5.

Algorithm 1.1 Gradient Perturbation Controller (GPC) – Simplified Version

1: Input: h, η, initial parameters $M^1_{1:h}$, dynamics f.
2: **for** $t = 1 \dots T$ **do**
3: Use control $\mathbf{u}_t = \sum_{i=1}^{h} M_i^t \mathbf{w}_{t-i}$
4: Observe state $\mathbf{x}_{t+1}$, compute perturbation $\mathbf{w}_t = \mathbf{x}_{t+1} - f(\mathbf{x}_t, \mathbf{u}_t)$
5: Construct loss $\ell_t(M_{1:h}) = c_t(\mathbf{x}_t(M_{1:h}), \mathbf{u}_t(M_{1:h}))$
Update $M^{t+1}_{1:h} \leftarrow M^t_{1:h} - \eta \nabla \ell_t(M^t_{1:h})$
6: **end for**

The version given here assumes that the time-invariant dynamics f are known. The GPC computes a control that is a linear function of past perturbations. It modifies the linear parameters according to the loss function via an iterative gradient method. In Chapter 7, we prove that this algorithm attains sublinear regret bounds for a large class of time-varying and time-invariant dynamical systems. In later chapters, we explore extensions to nonlinear dynamical systems and unknown systems, which lie at the heart of nonstochastic control theory.

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press

1.7 Bibliographic Remarks

Control theory is a mature and active discipline with both deep mathematical foundations and a rich history of applications. In terms of application, self-regulating feedback mechanisms that limit the flow of water, much like the modern float valve, were reportedly used in ancient Greece over two thousand years ago. The first mathematical analysis of such systems appeared in a paper by James Clerk Maxwell on the topic of governors (Maxwell, 1868), shortly after Maxwell had published his treatise on the unification of electricity and magnetism. For an enlightening historical account, see Fernández Cara and Zuazua Iriondo (2003).

For a historical account on the development of the PID controller, see Bennett (1993). The theoretical and practical properties of this controller are described in detail in Åström and Hägglund (1995).

There are many excellent textbooks on control theory, which we point to in later chapters. The book of Bertsekas (2007) offers extensive coverage of topics from the point of view of dynamic programming. The texts Stengel (1994) and Zhou et al. (1996a), in contrast, delve deeply into results on linear control, the most well-understood branch of control theory. Rockafellar (1987) proposed the use of convex cost functions to model the cost in a linear dynamical system to incorporate constraints on state and control and suggested computational difficulties with this approach. The text Slotine and Li (1991) gives a thorough introduction to adaptive control theory.

Online nonstochastic control theory and the gradient perturbation control algorithm were proposed in Agarwal et al. (2019c). A flurry of novel methods and regret bounds for online nonstochastic control have appeared in recent years, many of which are surveyed in Hazan and Singh (2021). Notable among them are logarithmic regret algorithms (Agarwal et al., 2019b; Foster and Simchowitz, 2020; Simchowitz, 2020), nonstochastic control for unknown systems (Hazan et al., 2020), black-box online nonstochastic control (Chen and Hazan, 2021), and nonstochastic control with partial observability (Simchowitz et al., 2020).

The theory is heavily based on recent developments in decision making and online learning, notably the framework of online convex optimization (Hazan, 2016) and especially its extensions to learning with memory (Anava et al., 2015).

https://doi.org/10.1017/9781009499705.003 Published online by Cambridge University Press