Operational Technology for Smart and Sustainable Systems

A Multidisciplinary Introduction

Ahsan Rahman

Operational Technology for Smart and Sustainable Systems

A Multidisciplinary Introduction

Ahsan Rahman

All examples in this book are abstracted and generalised for pedagogical purposes. The reflections and views expressed are those of the author.

Author: Ahsan Rahman

To contribute to and return to the world that has granted me the time and opportunity to learn.

Preface

Most engineering students arrive at a graduate program already fluent in one technical language. Some think in code and data structures. Some think in voltages, currents, and signals. Some think in forces, flows, and thermodynamic cycles. Each of these languages is powerful, and each is, by itself, incomplete for the subject of this book.

Operational Technology — OT — is the engineering of systems that *do* things in the physical world: systems that move water through a city, keep a factory running through the night, hold a turbine at constant speed while the wind gusts, and shut a pipeline valve in the half-second before a pressure spike becomes an explosion. OT is not a single discipline. It is the place where several disciplines are forced to cooperate, because no real industrial plant can be built by software engineers alone, or by power engineers alone, or by mechanical engineers alone.

This book was written for a classroom in which all of those engineers sit side by side. It does not assume you have taken a course in control theory. It does not assume you can read a ladder-logic diagram, size a pump, or configure a network switch. It assumes only that you have completed an undergraduate engineering degree of some kind, that you are comfortable with quantitative reasoning, and that you are willing to learn ideas that originated outside your home discipline.

A word on what this book is *not*. It is not a vendor manual, and it will not teach you to operate a specific manufacturer's controller. It is not a standards compendium, and it will not reproduce the clauses of IEC 62443 or IEC 61511 — though it will explain what those standards are for and why they exist. Product names and model numbers age badly; principles do not. This book is built from first principles, and its goal is that when you later sit in front of a real industrial system — any real industrial system — you will recognize what you are looking at and know which questions to ask.

The book is organized in five parts spanning thirteen chapters. **Part 0** is unusual: it exists specifically to bridge the diverse backgrounds in the room, and it is the only part that speaks to disciplines by name. From **Part I** onward, the book speaks one language — the language of OT itself — and every concept is developed from the ground up so that no reader is left behind.

Each chapter ends with review questions arranged in four tiers: *recall*, *application*, *analysis*, and *design*. The design questions have no single correct answer; they are the ones worth arguing about in a seminar. The book closes with a capstone project in which mixed teams design a complete OT-enabled sustainable system. By that point, the disciplinary boundaries you walked in with should feel a good deal less important than they once did.

How to Use This Book

If you are a student. Read Part 0 carefully even if it feels elementary — it establishes the shared vocabulary that the rest of the book relies on. From Part I onward, do the practical exercises; OT is a physical subject and reading about a control loop is no substitute for watching one oscillate. The four-tier review questions are a self-diagnostic: if you can answer the recall and application questions but stall on analysis, you have memorized the chapter without understanding it.

If you are an instructor. The 13 chapters map onto a 14- to 16-week semester (a suggested schedule appears in Appendix G). Parts I and II are the technical core and should not be rushed. Part III can be weighted toward whichever sustainability domain — energy, water, or manufacturing — best matches your program's emphasis. The capstone in Chapter 13 is most effective when teams are deliberately assembled to mix backgrounds.

A note on figures. Figures in this book are deliberately schematic. A diagram that shows you the *structure* of an idea is worth more, at this stage, than a photograph of a particular machine. Where a figure is drawn in text, read it slowly; the layout carries meaning.

Table of Contents

PART 0 — WHERE DO YOU COME FROM?

Part 0 is the only part of this book that speaks to academic disciplines by name. Its job is to get every reader — whatever their undergraduate background — to the same starting line. Once Part I begins, the book speaks a single language: the language of Operational Technology.

Chapter 1 — Finding Your Entry Point

1.1 What This Book Is About

Consider an ordinary morning in any city. Before breakfast, a person has already depended on perhaps a dozen industrial systems without thinking about a single one of them. The water that came out of the tap was lifted from a source, filtered, disinfected to a controlled chemical concentration, pumped up to a holding reservoir, and delivered at a regulated pressure. The electricity that boiled the kettle was generated somewhere far away, stepped up to a high transmission voltage, carried across the country, stepped back down, and balanced — generation against demand — continuously, second by second, so that the frequency of the supply never drifted far from its nominal value. The bread was baked in an oven whose temperature was held within a narrow band by a controller that no baker ever sees.

None of this happens by accident, and none of it happens by human hands alone. Behind each of these everyday miracles is a layer of technology whose entire purpose is to *sense the physical world, decide what to do, and act on that decision* — reliably, continuously, and often without anyone watching. That layer is the subject of this book. We call it **Operational Technology**.

Operational Technology is the hardware and software that monitors and controls physical processes, physical equipment, and physical infrastructure. The word *operational* is the important one. OT does not exist to produce reports, store records, or send messages between people. It exists to keep a physical operation running correctly. When OT works well, nobody notices it. When it fails, the consequences are immediate and physical: a flooded street, a dark neighborhood, a production line stopped, in the worst cases an injury or an environmental release.

It is worth stating clearly, at the very beginning, what makes OT distinct from the technology most engineering students already know well. The laptop you are reading this on is a marvel of **Information Technology** — IT. Its job is to handle information: to compute, to store, to display, to communicate. If your laptop freezes for two seconds, you are mildly annoyed. If it crashes, you lose unsaved work and reboot. IT systems are built on the assumption that occasional failure is tolerable and recoverable.

OT cannot make that assumption. If the controller regulating the pressure in a municipal water main "freezes for two seconds," the pressure does not politely wait. Water is a physical substance with mass and momentum; the process keeps evolving whether the controller is paying attention or not. If the controller holding a chemical reactor at a safe temperature crashes and reboots, the reaction does not pause for the reboot. **The defining feature of OT is that it is coupled, in real time, to a physical process that will not wait.** Everything else that is distinctive about

OT — its obsession with reliability, its conservative attitude toward change, its particular security problems — flows from that single fact.

This book is about how such systems are designed, how they work, how they are connected together, how they are secured, and — the theme that runs through the entire second half — how they are being used to make our use of energy, water, and material resources more sustainable.

Box 1.1 — Three sentences that define OT

> *Operational Technology is technology that is coupled to a physical process in real time.*
> *Because the process will not wait, OT prizes reliability and predictability above almost everything else.*
> *Because the process is physical, the consequences of OT failure are physical.*

If you remember nothing else from this chapter, remember Box 1.1. Every later chapter is, in one way or another, an elaboration of those three sentences.

1.2 The Many Faces of Engineering — And Why They All Lead Here

OT: a working industrial system is genuinely multidisciplinary, and no single undergraduate degree prepares a person to design one alone.

To see why, imagine the design of something deliberately modest — an automated pumping station that lifts treated water from a low reservoir into an elevated tank, so that the elevated tank can then supply a town by gravity. This is about as simple as a real OT system gets, and it still requires every kind of engineer in the room.

Someone must choose the pump itself and the pipework: the pump must deliver enough flow at enough pressure to overcome the height difference and the friction of the pipe, and it must do so efficiently across the range of conditions the station will actually see. This is a question of fluids, of rotating machinery, of materials and mechanical wear.

Someone must supply the pump with electrical power and protect it: a motor of this size needs a starter that does not collapse the local voltage every time it switches on, protective devices that disconnect the motor if it draws a dangerous current, and a means of varying the motor's speed if the design calls for it. This is a question of power, of machines, of protection.

Someone must give the station its logic: when the elevated tank falls below a chosen level, start the pump; when the tank is full, stop it; if the pump runs but the flow sensor reports no flow, conclude that the pump is running dry and shut it down before it destroys itself. This logic must be written, structured, and made dependable. This is a question of programming, of computation, of software discipline.

Someone must connect the station to the wider system: the operator in a distant control room needs to see the tank level and the pump status, needs to be alerted when something is wrong, and may need to start or stop the pump remotely. This is a question of networks, of communication protocols, of data.

And in a modern station, someone must make it *smart and sustainable*: analyze the energy the station consumes, shift its pumping to hours when electricity is cheaper or cleaner, predict when the pump's bearings are beginning to fail before they fail outright, and report the station's water and energy efficiency. This draws on data analysis, on modeling, on optimization.

Five distinct kinds of engineering thinking, and we have not yet left the pump house. Scale this up to a desalination plant, a power grid, or a smart factory, and the multidisciplinary nature of OT becomes overwhelming. The lesson is not that you must master all five kinds of engineering. The lesson is that **you must be able to think alongside the other four** — to understand enough of their concerns to design something that works as a whole rather than as five disconnected pieces. Teaching you to do exactly that is the central purpose of this book.

1.3 Your Background and How It Connects to OT

Because readers of this book arrive from different places, this section offers each of you a short, honest map: what your undergraduate training has already given you that OT will draw on directly, and where OT will ask you to stretch into less familiar ground. Read your own paragraph closely; skim the others, because by the end of the program your teammates will be the people described in them.

If your background is in computing or software. You already think naturally about logic, state, and the discipline of writing programs that behave predictably. This transfers directly: an industrial controller is, at heart, a computer executing a program in a loop, and the habits of clear, testable, well-structured code are exactly what OT control logic needs. What will be less familiar is that your program is now wired to physical reality. There is no "undo." A variable in your code corresponds to a real valve that is really open. Timing is not a performance nicety but a correctness requirement: a control decision that is correct but late can be worse than useless. You will also need to build physical intuition — to develop a feel for how fast a tank fills, how much momentum a spinning machine carries, why a pipe cannot change its flow instantaneously. The physical world is your new runtime environment, and it is unforgiving.

If your background is in electrical or power engineering. You are already comfortable with signals, with the difference between analog and digital, with how electrical power is generated, transmitted, and protected, and very likely with feedback in the specific form of regulator circuits. Much of OT's lowest layer — sensors producing electrical signals, actuators consuming electrical power, the protection of motors and feeders — is close to home. Where you will stretch is upward, into the layers built on top of the electrical substrate: the structure of control programs, the design of communication networks, the architecture that ties hundreds of devices into one coherent system, and the software-centered way of thinking about security. You know the physics of the wire; OT will ask you to also think about the meaning of the message travelling along it.

If your background is in mechanical, mechatronic, or manufacturing engineering. You already have the physical intuition that the computing-trained reader must work to build. You understand processes — flows, pressures, temperatures, thermodynamic cycles, the behavior of rotating equipment. You have very likely met feedback already, in the form of a governor or a thermostat or a servo. The actuators and final control elements of Chapter 7 are, to you, simply machine elements. Where you will stretch is into the informational and computational layers: how control logic is actually programmed and structured, how devices communicate, how data is gathered and turned into insight, and how the whole system is secured against deliberate attack. You know the process; OT will ask you to also think rigorously about the information that governs it.

If your background is in another engineering discipline — civil, environmental, chemical, industrial, or otherwise — you bring domain knowledge that OT exists to serve. A water network, a treatment process, a building's energy system, a production line: each is a physical operation that OT controls, and your understanding of *what the process is for* and *what "correct" means for that process* is exactly what a control system must be designed around. You may need to build up both the computational and the electrical sides from a more basic starting point, and the early chapters of Part I are written so that you can. Your advantage is that you are least likely to mistake the control system for the point. The control system is never the point. The process is the point.

The honest summary is this: **every reader of this book has a genuine head start in some part of OT and a genuine gap in another part.** Part 0 exists to make the gaps smaller and the common ground larger. Nobody in the room is starting from zero, and nobody in the room is starting complete.

1.4 What You Will Gain From This Book

It is worth being concrete about what you should be able to do after working through this book — not vague aspirations, but specific capabilities.

You will be able to **look at an unfamiliar industrial system and decompose it correctly**: to identify what is being sensed, where decisions are being made, what is being actuated, and how the parts communicate. This skill of structured "reading" of a system is the foundation of everything else, and it is the explicit goal of Part I.

You will be able to **reason about control without being intimidated by it**. You will not emerge a control-theory specialist — that is a course in its own right — but you will understand, at the level of genuine intuition rather than slogan, the difference between acting blindly and acting on feedback, why feedback systems can become unstable, and what a controller is actually trying to do. Chapter 5 builds this understanding from first principles and deliberately keeps the mathematics light.

You will be able to **understand how modern OT connects to the wider digital world** — to IIoT platforms, to data analytics, to artificial intelligence, to the cloud — and, just as importantly, to reason about what that connectivity costs as well as what it buys. Part II is devoted to this.

You will be able to **apply OT thinking to sustainability problems**: to see how the same sensing-deciding-acting loop that runs a factory can be turned toward reducing energy waste, integrating renewable generation, conserving water, and lowering the carbon and material footprint of industrial activity. Part III is the heart of the book's purpose, and it is the reason the program is called *Smart and Sustainable Systems* rather than simply *Industrial Automation*.

You will be able to **think clearly about OT security and safety** — about why industrial systems are attractive targets, why they are unusually hard to defend, and what genuine defense looks like — without needing a prior background in cybersecurity. Part IV develops this.

And you will be able to **work inside a multidisciplinary team** on a real design problem, which is the explicit aim of the capstone in Chapter 21.

Notice what is *not* on this list. You will not become an expert operator of any particular vendor's equipment. You will not memorize the clauses of any standard. Those are valuable things, but they are the kind of knowledge that is best acquired on the specific system in front of you, and they age quickly. This book aims at the knowledge that does not age: the principles, the structure, the way of thinking.

1.5 How All Engineering Disciplines Converge in OT

We end this opening chapter with a single unifying picture, because it is the picture the rest of the book hangs on.

Strip away every detail and every piece of jargon, and *every* OT system — without exception, from the smallest pump controller to the national power grid — is doing the same three things, over and over, in a loop:

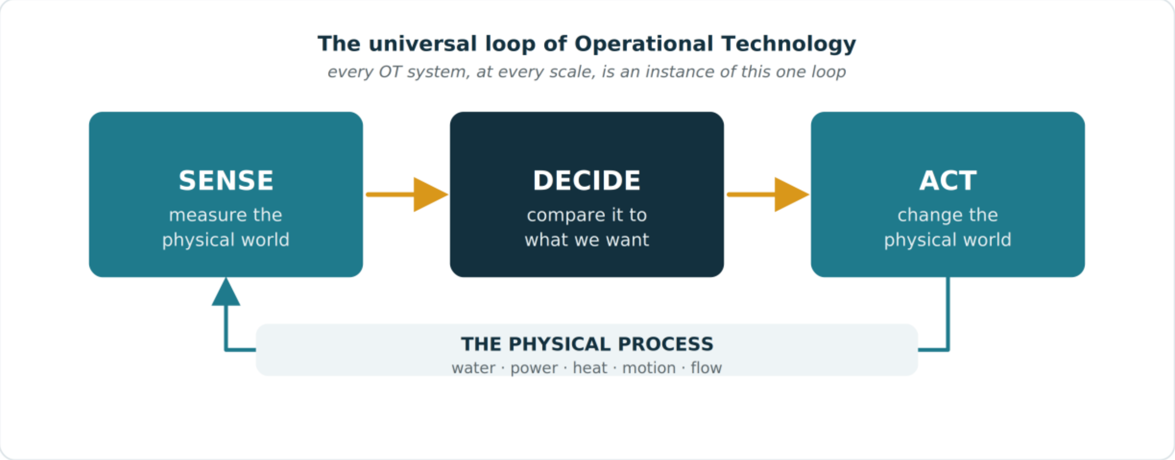

Figure 1.1 The universal loop of Operational Technology. *Every OT system, at every scale — from a single thermostat to a national power grid — is an instance of this same three-stage loop: sense the physical world, decide what to do, and act on the process. The loop never stops while the system is running.*

Sense. The system measures something physical — a level, a pressure, a temperature, a speed, an electrical quantity. (Chapter 4.)

Decide. The system compares what it measured against what it wants, and works out what to do about the difference. (Chapters 5 and 6.)

Act. The system changes something physical — it opens a valve, starts a motor, adjusts a heater. (Chapter 7.)

And then, crucially, the loop closes: the action changes the physical process, the changed process is sensed again, and the cycle repeats. The loop never stops while the system is running.

Now observe how the disciplines converge onto this single loop. The mechanically and process-trained engineer owns the physical process at the bottom of the figure and much of the *Act* block. The electrically trained engineer owns the signals flowing into *Sense* and the power flowing out of *Act*. The computing-trained engineer owns the *Decide* block and the structure of the program that implements it. The data-and-analytics-minded engineer works with the stream of measurements the loop produces and feeds insight back into how *Decide* behaves. And the domain engineer defines what the process is for, and therefore what the word *want* in the *Decide* block actually means.

Five kinds of engineer, one loop. They are not five separate jobs that happen to share a building. They are five views of a single system, and the system only works when the views are consistent. That is the convergence the chapter title promises, and it is the reason this program puts all of you in the same room.

Everything from here forward is an unfolding of Figure 1.1. Chapter 2 establishes the shared vocabulary you need to discuss it precisely. Part I takes the loop apart, block by block. Parts II and

III make the loop smart and turn it toward sustainability. Part IV asks how the loop can be made safe and secure. Keep the figure in mind; we will return to it constantly.

Review Questions — Chapter 1

Recall

1.1 In one sentence, state the defining feature that distinguishes Operational Technology from Information Technology.

1.2 Name the three actions that, repeated in a loop, constitute every OT system.

1.3 List the five kinds of engineering contribution involved in designing the automated pumping station of Section 1.2.

Application

1.4 A home thermostat senses room temperature, compares it to a setpoint, and switches a heater on or off. Identify the *Sense*, *Decide*, and *Act* blocks of Figure 1.1 for this system, and describe how the loop closes.

1.5 Explain, using the idea of a process "that will not wait," why a two-second freeze is a minor annoyance in an IT system but potentially serious in an OT system.

Analysis

1.6 The chapter claims that "the control system is never the point; the process is the point." Choose any industrial system you are familiar with and argue, with specific reference to that system, why this claim is correct — or where you think it is too strong.

1.7 A colleague argues that because modern controllers are really just computers, OT is simply "IT applied to factories" and needs no separate study. Using the distinctions drawn in this chapter, construct a reasoned rebuttal.

Design

1.8 Sketch — in the style of Figure 1.1 — the sense/decide/act loop for an automatic irrigation system intended to keep the soil moisture of a date palm plantation within a healthy band. Identify what is sensed, what is decided, and what is acted upon. Then list three ways the physical process might "refuse to wait" and what each implies for how the controller must be designed.

Chapter 2 — Common Ground for Every Engineer

Chapter 1 ended with a promise: that from Part I onward this book would speak a single language. This chapter is where that language is assembled. Its purpose is to take the handful of ideas that *every* engineer needs in order to discuss OT precisely — systems, processes, feedback, signals, energy, information — and to define them carefully enough that a reader from any background means the same thing by them. These ideas are not difficult. But they are often used loosely, and looseness here will cost us later. By the end of the chapter the vocabulary will be shared, and Part 0 will have done its job.

2.1 Systems Thinking — The Universal Engineering Language

The word *system* is used so freely that it has nearly worn smooth. Engineers speak of a control system, a power system, an operating system, a system of equations. If the word is to do real work for us, we must sharpen it.

A **system**, in the sense this book will always use, is a collection of interacting parts that together produce a behavior that none of the parts produces alone. Three ideas are doing work in that sentence, and each deserves a moment.

First, *parts*. A system has components, and they are identifiable. In the pumping station of Chapter 1 the parts included a pump, a motor, a level sensor, a controller, and the pipework. We can point to each.

Second, *interacting*. The parts are not merely collected in one place; they affect one another. The controller's output affects the motor; the motor's rotation affects the pump; the pump's action affects the water level; the water level affects the sensor; the sensor affects the controller. Remove the interactions and you do not have a system, only an inventory.

Third, and most importantly, *behavior that none of the parts produces alone*. This is the idea that makes systems thinking worth the name. A pump cannot keep a tank at a constant level. A sensor cannot keep a tank at a constant level. A controller cannot keep a tank at a constant level. But a pump, a sensor, and a controller, correctly interconnected, *can* — and "keeping a tank at a constant level" is a behavior that belongs to the assembled whole and to none of its parts. Engineers call such a behavior **emergent**. The central claim of systems thinking is that the interesting behavior of an OT installation is almost always emergent, and therefore cannot be understood by examining components one at a time.

This has a hard practical consequence. It is entirely possible to assemble a system from components that are each, individually, perfect — every part meeting its specification exactly — and to obtain a system that does not work. The fault, when this happens, lies not in any part but in the *interactions*: in the way the parts were connected, in mismatched assumptions about timing, in a signal that one part produces in a form another part cannot use. A great deal of OT engineering is the engineering of interactions rather than of parts, and a great deal of OT failure is interaction failure. Keep this in mind; we will see it again in Chapter 9, where the "parts" are whole subsystems and the interactions are network messages.

Box 2.1 — The boundary of a system

> *Every system has a* ***boundary****: a line, drawn by the engineer, that separates what is inside the system from what is outside. The boundary is a choice, not a physical fact. For the pumping station we might draw the boundary around the pump house alone, treating the incoming water and the electrical supply as things that arrive from "outside." Or we might draw it around the whole water network, in which case the pumping station becomes one part inside a larger system. Neither choice is wrong. But the choice must be made and then held, because everything we say afterward — what counts as an input, what counts as a disturbance, what we are responsible for — depends on where the boundary was drawn. A surprising number of engineering disputes dissolve once the parties notice they had drawn the boundary in different places.*

Systems thinking, then, is the discipline of asking, before anything else: *What are the parts? How do they interact? Where is the boundary? What behavior do we want to emerge?* It is the one habit of mind common to every engineer in the room, and it is the frame onto which every later chapter is hung.

2.2 Inputs, Processes, and Outputs — A Framework for All

Once a boundary is drawn, a system can be described in a way that is completely independent of what the system is made of. Whatever crosses the boundary inward is an **input**. Whatever crosses the boundary outward is an **output**. Whatever happens between them — the transformation of inputs into outputs — is the **process**.

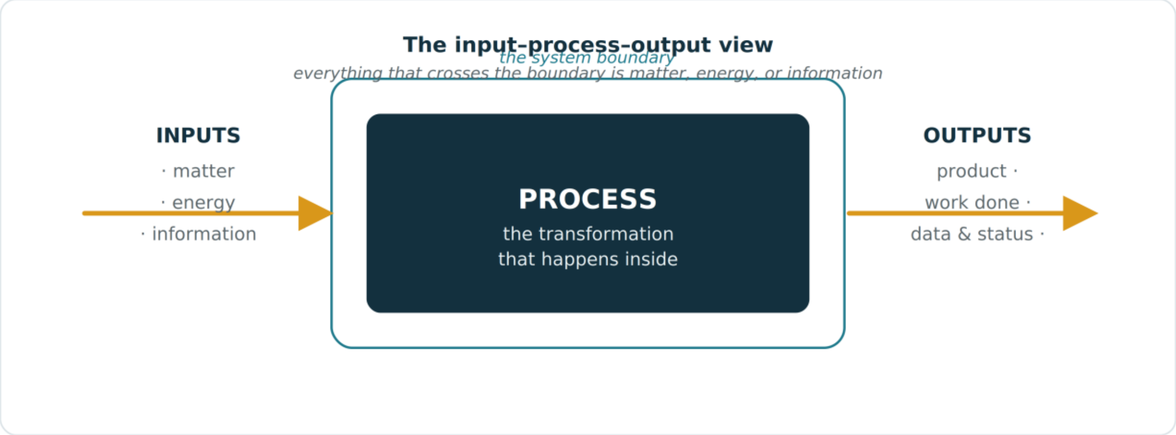

Figure 2.1 The input–process–output (IPO) view of a system. *Everything inside the boundary is the process; everything crossing the boundary is an input or an output. Drawing the boundary first — and deciding deliberately what lies inside it — is the engineer's first act of analysis.*

This **input–process–output** framework, abbreviated IPO, is deliberately abstract, and its abstraction is its power. It applies without modification to a chemical reactor (inputs: reactants, heat; process: reaction; outputs: product), to an electrical substation (input: high-voltage power; process: transformation; output: lower-voltage power), to a single controller (inputs: sensor readings; process: computation; output: a command), and to an entire factory. Because the framework does not care what the parts are made of, it is a description that every engineer in the room can read, and that is exactly why we adopt it.

Two refinements make the framework genuinely useful rather than merely tidy.

The first is the distinction between inputs we *chose* and inputs we did not. Some inputs are deliberate: the electrical power we supply to a pump is an input we arrange on purpose. Other inputs arrive uninvited: the rise in ambient temperature on a hot afternoon is also an input to a process, but no engineer asked for it. Uninvited inputs have a special name — **disturbances** — and they are not a minor footnote. The entire reason control exists, as the next section will argue, is that disturbances exist. In a world without disturbances we could set every process running once and walk away. We cannot, so we cannot.

The second refinement is that outputs divide into the *wanted* and the *unwanted*. A power station's wanted output is electricity; its unwanted outputs include waste heat and combustion gases. A pump's wanted output is water delivered; its unwanted outputs include noise, vibration, and heat in the motor windings. This distinction will matter intensely in Part III, because sustainability engineering is very largely the engineering of *unwanted outputs* — reducing them, capturing them, or turning them back into inputs somewhere else. For now simply register that "output" is not a synonym for "the thing we wanted."

Worked Example 2.1 — Drawing the IPO of a domestic water heater

Consider an ordinary electric storage water heater: an insulated tank of water with an electric heating element inside it, supplying hot water to a household.

We first draw the boundary: around the tank and its element, excluding the household plumbing beyond it.

Inputs (deliberate): cold water entering the tank; electrical energy entering the element.

Inputs (disturbances): heat leaking *in* or *out* through the insulation as room temperature varies; the unpredictable timing and volume of hot-water draw-off when someone opens a tap.

Process: electrical energy is converted to heat in the element and transferred to the water, raising its temperature; simultaneously, stored hot water mixes and loses heat to the surroundings.

Outputs (wanted): hot water delivered at the tap.

Outputs (unwanted): heat lost through the tank wall; a small amount of scale deposited over time.

Notice that the draw-off — someone opening a tap — is classed here as a *disturbance*, an uninvited input, because from the heater's point of view it arrives unpredictably and the heater must cope with it. That single modeling decision already tells us the heater will need *control*: if draw-off were predictable we could schedule heating in advance, but because it is a disturbance we must instead sense the water temperature and respond. We have, without yet using the word, just discovered why the heater contains a thermostat. The next section makes that discovery general.

2.3 Physical Processes and Why They Need Control

We can now state, plainly and from first principles, the single most important idea in this book. It is the answer to the question a newcomer most reasonably asks: *why is control necessary at all? Why can we not simply build the process correctly and leave it running?*

The answer has three parts, and all three are unavoidable features of the physical world.

First: disturbances exist. As Section 2.2 established, every real process receives uninvited inputs. The ambient temperature changes. The supply voltage sags. The composition of a raw material varies from one delivery to the next. Demand rises and falls. A process that was perfectly balanced for one set of conditions is, the moment conditions change, no longer balanced. Because disturbances never stop arriving, a process left to itself never stops drifting.

Second: processes have their own dynamics. A physical process is not a passive conduit that instantly turns inputs into outputs. It has *internal* behavior of its own, governed by the storage of mass, energy, and momentum. A tank integrates flow: the level depends not on the flow right now but on the entire history of flow in and out. A heated mass has thermal inertia: its temperature lags changes in heating, sometimes by a long way. A spinning machine stores rotational momentum and resists changes in speed. These internal dynamics mean a process *responds slowly, lags, overshoots, and sometimes oscillates* — and a fixed, unattended input cannot accommodate any of that.

Third: we usually do not get to choose the output directly. This is the subtle one. We almost never have a knob marked "tank level" or "reactor temperature." What we have are knobs marked "pump speed" or "heater power" — and the quantity we actually care about emerges from the process in response to those knobs, filtered through the dynamics just described and corrupted by the disturbances mentioned before. There is always a gap between the variable we can *touch* and the variable we *want*.

Put these three facts together and the necessity of control is no longer mysterious; it is forced. We want some quantity — the **controlled variable** — to stay at, or follow, some desired value, which we will call the **setpoint**. We cannot set the controlled variable directly; we can only adjust some other quantity, the **manipulated variable**, and hope. Disturbances are continually pushing the controlled variable away from the setpoint. The process's own dynamics mean our adjustments take effect slowly and imperfectly. Therefore the only thing that can possibly work is a *continuing activity*: measure the controlled variable, compare it to the setpoint, adjust the manipulated variable to reduce the difference, and repeat — forever, or for as long as the process runs. That continuing activity is **control**, and the loop it forms is the *sense–decide–act* loop of Figure 1.1, now equipped with proper names.

Box 2.2 — The vocabulary of control, fixed once and for all

> ***Controlled variable*** *— the physical quantity we want to keep at, or move to, a desired value (the tank level, the reactor temperature, the machine speed).*
> ***Setpoint*** *— the desired value of the controlled variable (the level we want, the temperature we want).*
> ***Manipulated variable*** *— the quantity the controller can actually adjust (the pump speed, the heater power, the valve position).*
> ***Disturbance*** *— any uninvited input that pushes the controlled variable away from the setpoint.*
> ***Error*** *— the difference between setpoint and controlled variable: error = setpoint − measurement. Control is, in one phrase, the business of driving error toward zero.*
> *These five words appear in every remaining chapter. Learn them now and the rest of the book reads smoothly.*

It is worth pausing on how complete this answer is. We have not appealed to any particular technology, any vendor, any era. We have appealed only to three facts — disturbances exist, processes have dynamics, we cannot touch the output directly — that were as true of a steam engine in 1820 as they are of a desalination plant today. Control is not a feature that engineers add because it is fashionable. It is the unavoidable response to the structure of physical reality. That is why control sits in the title of Chapter 4 as *the heart of OT*.

2.4 Data, Signals, and Decision Making

Section 2.3 described the control loop in terms of physical quantities. But notice that the *decide* step does not act on water or heat or rotation directly — it acts on *information about* water and heat and rotation. Between the physical process and the decision sits a chain of representation, and an engineer of OT must understand that chain, because much of what goes right and wrong in real systems happens inside it.

Begin with the cleanest distinction. A **physical quantity** is a feature of the world: the actual temperature of actual water. A **signal** is a physical carrier of *information about* that quantity — most often, in OT, an electrical voltage or current whose size stands for the size of the temperature. A **datum** (plural **data**) is a signal once it has been turned into a number a computer can store and manipulate. The journey of one temperature reading is therefore: *the water has a temperature* → *a sensor produces a voltage representing that temperature* (a signal) → *a converter turns that voltage into the number 64* (a datum) → *the controller uses the number 64 to decide*.

Two properties of signals are worth fixing now because they will return constantly.

The first is the distinction between **analog** and **digital**. An analog signal varies *continuously*: at every instant it has a value, and between any two values it passes through all the values in between, just as the water temperature itself does. A digital signal is *discrete*: it exists as distinct, separated values — at its simplest, just two, which we may call 0 and 1, off and on, open and closed. The physical world is overwhelmingly analog; the computers that make decisions are digital; therefore somewhere in every OT system there must be a translation between the two, and that translation is never perfectly faithful. Chapter 4 examines it in detail. For now, hold onto the asymmetry: *reality is analog, decision is digital, and a translation stands between them*.

The second property is that **every signal carries error**. The voltage a sensor produces is never an exact image of the physical quantity. It is corrupted by **noise** — small, random, unwanted fluctuations added by the electrical environment — and it is distorted by the imperfections of the sensor itself. The number the controller finally acts on is therefore an *estimate*, not a truth. A mature OT engineer never forgets this. Decisions are made on data; data are imperfect images of signals; signals are imperfect images of reality. The chain from world to decision is a chain of approximations, and designing OT well means designing so that the approximations are good enough — not pretending they are exact.

Box 2.3 — "Good enough" is an engineering quantity

> *A beginner asks whether a measurement is correct. An engineer asks whether it is good enough — and treats "good enough" as something to be defined numerically before the system is built. Good enough for what decision? With what consequence if wrong? A temperature reading that may be off by two degrees is entirely good enough to decide whether to run a building's air conditioning, and nowhere near good enough to control a chemical reaction near a hazardous threshold. The same measurement, the same error, two different verdicts. "Good enough" is not an excuse; it is a specification, and writing it down honestly is part of the engineering.*

Decision making, finally, is what the *decide* block does with the data. At its simplest a decision is a comparison — *is the measurement above or below the setpoint?* — producing an on/off command. At its most elaborate it is the work of an artificial-intelligence model weighing hundreds of inputs, as Chapter 8 will describe. But however simple or elaborate, every decision in OT shares one structure: it takes in *data representing the present state of the process*, combines it with *a notion of what is wanted*, and produces *a command that will change the process*. Sense, decide, act. We have now seen the loop three times — as a slogan in Chapter 1, as physical necessity in Section 2.3, and as a flow of information here. It is the same loop. The repetition is deliberate.

2.5 Energy, Information, and Matter in Industrial Systems

There is one more lens to install before Part 0 closes, and it is the lens that connects this book's two title words — *smart* and *sustainable* — to a single physical picture.

Every industrial system, viewed at the most fundamental level, traffics in exactly three things: **matter**, **energy**, and **information**. Everything that flows across the boundary of Figure 2.1 is one of the three. The water moving through a treatment plant is matter. The electricity driving its pumps is energy. The sensor readings, commands, and alarms coordinating it are information. There is no fourth category. An industrial system is, at bottom, an arrangement for moving and transforming matter, energy, and information — and the three behave according to very different rules.

Matter is conserved and it is local. It does not appear or vanish; if a tonne of feedstock enters a process, a tonne of products plus wastes must leave. And matter moves by being physically transported, which takes time, energy, and equipment. The conservation of matter is the bedrock of every *mass balance* an engineer draws, and mass balances are how Part III will account for material waste and material recovery.

Energy is conserved too, but it degrades. The total quantity of energy is never lost — but its *quality*, its usefulness, falls every time it is converted. Electrical energy driving a motor becomes mechanical energy, and a portion becomes low-grade heat that warms the room and can never be fully recovered. This one-way degradation, from concentrated useful energy toward diffuse useless heat, is the deepest reason that *efficiency* is never 100% and that sustainability is hard. Part III's energy chapters are, in the end, an extended argument about managing energy *quality*, not just energy quantity.

Information obeys neither rule, and that is what makes it special. Information is not conserved: it can be copied endlessly, created, and destroyed. It does not degrade with use: a temperature reading is no less useful for having been read a thousand times. And it can be moved at the speed of light for almost no energy. Information is the one ingredient of an industrial system that is *cheap to move and cheap to multiply* — and that single asymmetry is the whole engine of modern OT. The word *smart*, in "smart and sustainable systems," means precisely this: *using information — which is cheap, fast, and freely copied — to reduce the consumption of matter and energy, which are none of those things*. A smart grid uses information to waste less electricity. A smart water network uses information to lose less water. Predictive maintenance uses information to consume fewer spare parts and less downtime.

Box 2.4 — The central bargain of smart and sustainable systems

> *Matter and energy are conserved, costly to move, and — in the case of energy — degrade with every use. Information is none of these things. The entire project of smart, sustainable OT is the deliberate trading of cheap, abundant information for reductions in expensive, limited matter and energy. When you read "smart" in this book, read "spending information to save matter and energy."*

This is the thought that unifies the book. Part I builds the machinery that gathers and acts on information. Part II makes that information abundant and intelligent. Part III spends it, deliberately, to save matter and energy. Part IV protects it. The whole arc is one idea, and Box 2.4 states it.

2.6 Bridging Your World to the Industrial World

We can now gather the chapter's threads and speak directly to the moment of transition that Part 0 was written to ease.

Every reader of this book has spent years inside one *world* of engineering, and each world has its own instincts. The computing world's instinct is that systems are made of logic and state, that anything can be versioned and rolled back, that the right response to a bug is a patch deployed quickly. The electrical world's instinct is that systems are made of signals and power, that behavior is continuous, that the laws are physical and fixed. The mechanical and process world's instinct is that systems are made of flows and forces and stored energy, that change is gradual and physical, that nothing happens instantly. The domain engineer's instinct is that the system exists to serve a purpose — clean water, safe buildings, delivered product — and that the purpose, not the mechanism, is what matters.

The industrial world does not ask you to discard any of these instincts. It asks you to *hold all of them at once*, because an OT system genuinely is all of those things simultaneously. It is logic and state — the controller runs a program. It is signals and power — the sensors and actuators are electrical. It is flows and forces and stored energy — the process is physical and gradual. And it exists entirely to serve a purpose. The single hardest mental adjustment in moving from your world to the industrial world is not learning new facts; it is learning to keep four kinds of instinct active at the same time and to notice when they conflict.

They *will* conflict, and the conflicts are instructive. The computing instinct says "deploy the fix now"; the process instinct says "the reactor is at temperature, you cannot just restart the controller." The computing instinct is not wrong and the process instinct is not wrong — they are two true views of one system, and the engineering lies in honoring both. Chapter 1 promised that the disciplinary boundaries you walked in with would come to feel less important. This is how: not by any boundary being erased, but by each of you learning to think, a little, in all of the others' terms. The IPO framework, the control vocabulary, the matter–energy–information lens — these are the shared instruments that make such cross-thinking possible. They belong to no single discipline, and from here on they belong to all of us equally.

2.7 What OT Looks Like Across Different Engineering Lenses

This is the final section of Part 0, and therefore the last place in this book where engineering disciplines are named. We use it deliberately: to look at one single OT system through each lens in turn, so that every reader sees both their own view and their teammates' views laid side by side. After this section the lenses merge, and the book speaks only of OT.

The system we will look at is a **single industrial heat-treatment furnace**: an insulated chamber in which metal parts are held at a high, precisely controlled temperature for a fixed time to alter their material properties. It is small enough to grasp whole and rich enough that every lens sees something real.

Through the process and mechanical lens, the furnace *is* a thermal process. What matters is the heat balance: energy supplied by the burners or elements against energy lost through the walls and carried away by the parts and the flue. What matters is thermal inertia — the chamber and its contents store heat and so respond slowly, with a lag of minutes. What matters is uniformity: every part must see the same temperature history, so the physical arrangement of heating, airflow, and load placement is itself an engineering problem. To this lens the furnace is a problem of energy storage and transfer in a physical body.

Through the electrical lens, the furnace is a load and a set of signals. The heating elements draw substantial electrical power and must be switched and protected; the switching itself, done many times an hour, is an engineering concern. The temperature is sensed by devices that turn heat into small electrical signals, signals weak enough that how they are wired and shielded determines whether the reading is trustworthy. To this lens the furnace is a problem of delivering power cleanly and recovering a faithful signal from a noisy electrical environment.

Through the computing lens, the furnace is a program. A controller runs a loop: read the temperature, compare to the setpoint profile, decide how much heat to call for, command the elements, repeat — and within that loop sit the logic of the heating recipe, the sequencing of its stages, the handling of the case where a sensor reading is missing or absurd. To this lens the furnace is a problem of correct, well-structured, dependable logic executing in step with a physical clock.

Through the data and analytics lens, the furnace is a source of a stream of measurements, and that stream has value beyond the immediate control loop. Recorded over many cycles it reveals whether the process is drifting, whether one part of the chamber runs cooler than another, whether the elements are slowly degrading and will soon need replacement. To this lens the furnace is a problem of turning a history of measurements into foresight.

Through the domain lens, the furnace exists for one reason: to give the treated metal the properties the customer needs. Every other view is subordinate to this. A furnace that holds temperature beautifully but to the wrong profile is a failure; a furnace that is elegant in every technical respect but produces parts that do not meet specification is simply broken. To this lens the furnace is a problem of reliably delivering a metallurgical outcome.

Five lenses, one furnace. Notice three things about the list. First, no lens is wrong — each sees something genuinely present in the system. Second, no lens is complete — each, alone, would miss something a working furnace requires. Third, and most important, the lenses are not looking at five different furnaces; they are looking at one, and the furnace only works when all five views are *simultaneously* satisfied. The heat balance must close, the signal must be clean, the logic must be correct, the data must be used, and the metallurgical outcome must be delivered — all at once, in the same machine.

That is the truth Part 0 was written to deliver, and we can now state it in its final form. **An OT system is not a collection of disciplinary problems sitting next to each other; it is a**

single problem that happens to be visible from several disciplinary directions. From the next chapter onward this book stops naming the directions. It will simply describe the furnace, the pump, the grid, and the factory as what they are — OT systems — and trust that each reader, having seen all five lenses here, can supply whichever view a passage most needs. Part 0 is complete. The common ground is laid. Part I begins on it.

Review Questions — Chapter 2

Recall

2.1 Define *system* in the precise sense used in this book, and explain what is meant by an *emergent* behavior.

2.2 State the input–process–output framework and define *disturbance* within it.

2.3 Write down the five terms of the control vocabulary fixed in Box 2.2, with a one-line definition of each.

2.4 Name the three fundamental things that, according to Section 2.5, every industrial system traffics in.

Application

2.5 For the domestic water heater of Worked Example 2.1, identify the controlled variable, the setpoint, the manipulated variable, and one disturbance. Then write the error in the form given in Box 2.2.

2.6 Classify each of the following as analog or digital, and justify your choice: (a) the position of a valve that may sit anywhere from fully shut to fully open; (b) a signal reporting only whether a door is open or closed; (c) the speed of a rotating shaft.

2.7 A pressure measurement in a particular system may be in error by up to 3%. Describe one decision for which this measurement is "good enough" and one for which it is not, and explain what makes the difference.

Analysis

2.8 Section 2.3 argues that control is *forced* by three unavoidable facts about the physical world. Construct the argument in your own words, and then attempt to describe a hypothetical process that would genuinely need no control. What does your attempt reveal about the three facts?

2.9 The chapter claims that "it is possible to assemble a system from perfect components and obtain a system that does not work." Give a concrete example of such an interaction failure, drawn from any system you know, and identify which interaction was at fault.

2.10 Explain, using the matter–energy–information distinction, why the word *smart* in "smart and sustainable systems" is not merely a marketing adjective but names a specific engineering strategy.

Design

2.11 Choose an everyday physical system not discussed in this chapter. (a) Draw its boundary explicitly and justify where you placed it. (b) Produce its full IPO description, separating deliberate inputs from disturbances and wanted from unwanted outputs. (c) Identify a controlled variable, a plausible setpoint, and a manipulated variable. (d) Finally, describe the same system through any two of the five lenses of Section 2.7, and state one thing each lens sees that the other misses.