

THE RESEARCH LEDGER

Volume II

Engineering Agentic Systems for Smart Sustainable Agriculture

An Undergraduate Introduction to Building,
Constraining, and Governing Autonomous Farm Agents

Ngo Trung Kien

Contents

Preface	v
How to use this book	vii
Notation and abbreviations	xi
List of figures	xx
List of tables	xxii
1 From Precision Agriculture to Agentic Agriculture	1
1.1 Four paradigms and one discontinuity	1
1.2 Why the farm is a hard environment for a learning agent	3
1.2.1 The feedback horizon exceeds the correction horizon	3
1.2.2 Irreversibility is not uniform, and it is not a property of the action	3
1.2.3 The plant is a low-pass filter with a memory	3
1.3 The autonomy ladder	4
1.3.1 Choosing a level: risk scoring	6
1.4 The agent's operating loop	8
1.5 Evidence, and how this book labels it	8
1.6 The running example	10
Chapter summary	10
Review questions	10
Hands-on exercises	11
Further reading	12
2 The Farm as a Cyber-Physical System	13
2.1 Scales, and the loops that live at each	13
2.2 What can be measured	15
2.3 Data quality as a first-class concern	15
2.3.1 Gaps, and the honest way to fill them	18
2.4 Observable, derived, and latent state	18
2.5 Actuation beyond valves	20
2.6 The reference site	21
Chapter summary	22
Review questions	22
Hands-on exercises	23
Further reading	23
3 Communication, Computation, and Energy	25
3.1 Why no single link will do	25

3.2	The radio link through a growing crop	26
3.2.1	Log-distance path loss	26
3.2.2	Visible light inside the greenhouse	29
3.3	Handover between two unequal links	30
3.4	Placing the computation	32
3.4.1	Degradation as a designed behaviour	34
3.5	Energy	34
	Chapter summary	37
	Review questions	38
	Hands-on exercises	38
	Further reading	39
4	System Model, Requirements, and Reference Architecture	41
4.1	Six layers, six characteristic failures	41
4.1.1	Cross-layer paths that must exist	41
4.2	Trust boundaries	42
4.3	Requirements that can be tested	43
4.4	The ten agent roles	45
4.5	The reference architecture	46
4.5.1	Interfaces	46
4.6	Topologies and the autonomy they support	46
4.7	Failure and recovery	49
	Chapter summary	51
	Review questions	51
	Hands-on exercises	52
	Further reading	52
5	The Agent's Decision Problem	53
5.1	The formalism, and where it breaks	53
5.1.1	Partial observability	53
5.2	The state vector	55
5.3	The action space and its constraints	55
5.4	Reward design, and the exploits against it	57
5.4.1	The delayed reward, and shaping	58
5.5	Multiple objectives, three formulations	59
5.6	The sensitivity of a single constant	61
	Chapter summary	62
	Review questions	63
	Hands-on exercises	63
	Further reading	64
6	Learning to Act Without a Second Chance	65
6.1	Why the obvious approach is unavailable	65
6.2	Offline reinforcement learning	66
6.2.1	The extrapolation-error problem	66
6.2.2	Conservatism as the remedy	66
6.3	Knowing whether the new policy is better	67
6.3.1	The promotion rule	68
6.4	Constraints during learning, and what they are worth	68
6.5	An empirical comparison, done fairly	70

6.6	The learning pipeline that follows from all this	73
	Chapter summary	76
	Review questions	76
	Hands-on exercises	77
	Further reading	78
7	Digital Twins, Calibration, and Drift	79
7.1	What a twin is for	79
7.2	The model	79
7.2.1	Soil water balance	80
7.2.2	Parameters that must be calibrated	81
7.3	Auditing a donated environment	82
7.4	Calibration and validation	84
7.5	Drift, and the autonomy ratchet	85
7.6	The sim-to-real gap compounds	86
	Chapter summary	89
	Review questions	89
	Hands-on exercises	90
	Further reading	90
8	Coordination Among Many Agents	93
8.1	Why coordination is a safety problem	93
8.2	Allocating a scarce resource	94
8.2.1	Fairness and value, measured	95
8.3	Architecture: how many agents, arranged how	96
8.4	Learning to coordinate	98
8.5	Emergent failures	99
8.5.1	The system-wide stop	100
	Chapter summary	101
	Review questions	101
	Hands-on exercises	102
	Further reading	102
9	Safety Engineering for Agricultural Agents	103
9.1	Hazard analysis before architecture	103
9.2	Constraints as formal predicates	103
9.3	Four enforcement layers	106
9.4	Fail-safe states	106
9.5	Human override, and when it cannot work	107
9.6	Verifying safety-critical software	108
9.6.1	The adversarial test suite	110
9.7	Safety metrics and the test schedule	111
	Chapter summary	113
	Review questions	113
	Hands-on exercises	113
	Further reading	114
10	Security of an Agentic Farm	115
10.1	What is different about attacking a farm	115
10.2	The threat model	115

10.2.1 Assets	116
10.2.2 Adversaries	117
10.2.3 Threats and attack paths	117
10.2.4 Residual risk	119
10.3 Defence in depth	120
10.4 Identity, authorisation, and tool control	121
10.5 Language-model agents: a new hazard class	123
Chapter summary	126
Review questions	126
Hands-on exercises	127
Further reading	127
11 Governance, Oversight, and Farmer Data Rights	129
11.1 Oversight does not scale	129
11.2 The alert budget	130
11.3 Trust calibration	131
11.4 Explanation as a governance requirement	132
11.5 The runtime governance engine	133
11.6 Audit and accountability	136
11.7 Incidents	137
11.8 Data rights	137
Chapter summary	140
Review questions	141
Hands-on exercises	141
Further reading	142
12 Sustainability, Economics, and Deployment	143
12.1 The boundary problem	143
12.2 Metrics	144
12.2.1 Definitions	144
12.2.2 A ten-hectare scenario, accounted honestly	145
12.3 The rebound effect	147
12.4 Economics	149
12.5 Deployment maturity	151
12.6 Benchmarking	152
12.7 What remains unsolved	153
12.8 The capstone	155
Chapter summary	156
Review questions	157
Hands-on exercises	157
Further reading	158
A Glossary	159
B Canonical Parameter Register	161
C Safety Constraint Catalogue	165
D Threat Catalogue	167
E The AgriFarm-AI Reference Site	169

F Capstone Project	171
G Laboratory Exercises	173
H Software and Reproduction	175
Bibliography	177
Index	187

CHAPTER 1

From Precision Agriculture to Agentic Agriculture

Learning objectives

By the end of this chapter you should be able to:

1. distinguish an *agent* from the measurement and analytics systems that precede it, using the criterion of closed-loop authority over an actuator rather than the sophistication of the model;
2. place an agricultural decision on the six-level autonomy ladder AL-0–AL-5 and justify the placement from the reversibility and severity of the decision, not from the capability of the software;
3. compute a risk score for a proposed autonomous action and read off the autonomy ceiling it licenses;
4. explain why the temporal structure of agricultural feedback makes reinforcement learning harder here than in the domains where it was developed;
5. state the epistemic status of a claim using the five-level evidence ladder, and identify the level at which most published agricultural-AI performance claims actually sit.

1.1 Four paradigms and one discontinuity

Agriculture has automated itself three times, and each time the automation left the decision with the farmer. Mechanisation replaced human and animal power with engines; it changed what could be done to a field, and how fast, but the question of what *should* be done to which part of the field remained a human judgement applied uniformly across the whole. Precision agriculture, from the early 1990s, broke the uniformity: positioning systems, yield monitors, and variable-rate applicators made it possible to treat a hectare as a mosaic of sub-hectare management zones with different requirements (Liakos et al., 2018). The farmer now decided differently for different parts of the field, but still decided. Smart agriculture, roughly the 2010s, instrumented the mosaic: networked sensors, cloud analytics and dashboards converted the field into a data stream and the farmer into a consumer of that stream (Walter et al., 2017; Klerkx et al., 2019). The decision remained human; what changed was the volume of evidence placed in front of the human, and, as Chapter 11 will argue at length, the volume of evidence placed in front of a human is not an unmixed good.

Figure 1.1 sets out the sequence. The discontinuity is at the fourth step. An **agentic system** is one in which software closes the loop: it perceives state, forms an intention, and commands an actuator, without a human in the path of each individual action. The change is not that the model became better. A gradient-boosted yield predictor embedded in a dashboard may be a far more sophisticated

statistical object than the threshold rule that opens a valve. The change is that the threshold rule has *authority*, and the yield predictor does not.

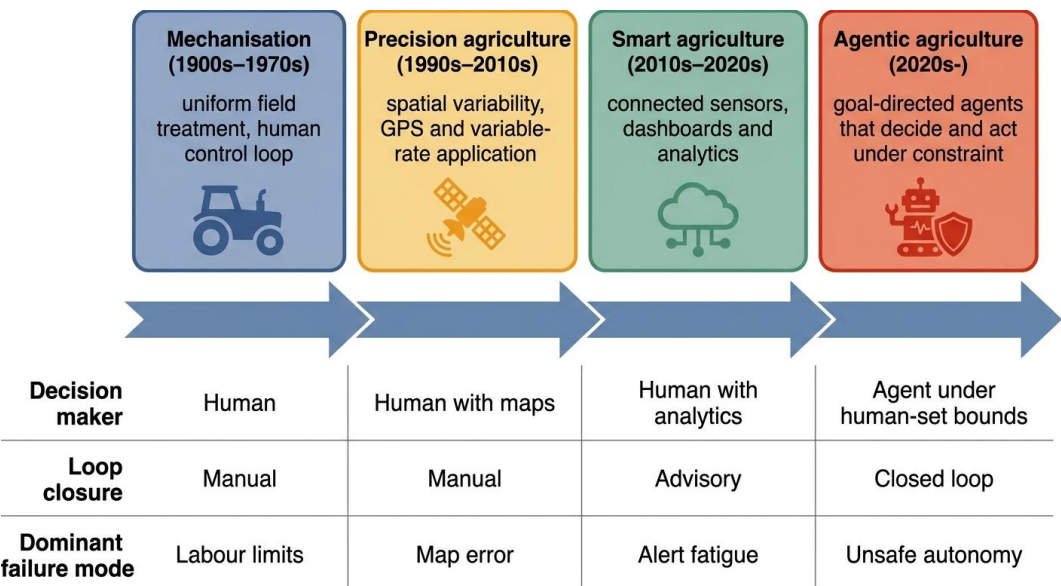


Figure 1.1: Four paradigms of agricultural automation. The first three change what the farmer can do and what the farmer can see; only the fourth changes who decides. Note the accompanying shift in dominant failure mode: precision agriculture fails through map error, smart agriculture fails through alert fatigue, and agentic agriculture fails through unsafe autonomy—a failure whose consequences land in the soil rather than on a screen.

Definition 1.1 (Agricultural agent). An *agricultural agent* is a software entity that (i) maintains an estimate of the state of a biophysical production system from sensor observations, (ii) selects actions from an action space that includes commands to physical actuators, (iii) does so in pursuit of an explicitly represented objective, and (iv) does so repeatedly over time without a human authorising each individual action. A system that satisfies (i)–(iii) but not (iv) is a *decision support system*; the distinction is legal and practical as much as technical, and it is the one that matters for engineering.

Definition 1.1 is deliberately silent about the implementation. It admits a fifteen-line threshold controller on an ESP32 and it admits a large language model orchestrating tool calls across a farm’s entire API surface. That is the correct level of abstraction for a safety argument, because the actuator does not know which one sent the command, and neither does the crop. It also has an uncomfortable corollary that is worth confronting immediately: a great deal of equipment that has been installed on farms for decades already satisfies the definition. The float-valve controller that maintains a stock tank is an agricultural agent under Definition 1.1. What makes contemporary agentic systems a different engineering problem is not that they act autonomously but that the space of actions they can select from is large, the mapping from state to action is learned rather than specified, and the consequences of an unanticipated action are correspondingly unbounded.

1.2 Why the farm is a hard environment for a learning agent

1.2.1 The feedback horizon exceeds the correction horizon

The defining structural feature of agricultural control is that the interval between an action and the observable consequence of that action is longer than the interval within which the action could have been corrected. Consider the canonical case. An irrigation agent under-waters a maize crop during the flowering window. The soil moisture deficit is observable within a day. The reduction in kernel set is not observable until grain-filling begins, some three weeks later, and the yield consequence is not quantified until harvest, some ten weeks later. By the time the reward signal arrives, the state that produced it is gone and cannot be revisited within that season. The agent gets one trajectory per year.

This is a quantitative claim about sample complexity, and it is worth making the arithmetic explicit. A deep reinforcement learning algorithm of the kind routinely applied to this problem—[Mnih et al. \(2015\)](#) for value-based methods, [Schulman et al. \(2017\)](#) for policy-gradient methods—typically requires 10^5 to 10^7 environment transitions to converge on a control task of modest dimensionality. An irrigation agent making one decision per day over a 120-day season generates 120 transitions per zone-season. A farm with ten zones generates 1200 transitions per year. At 10^5 transitions the agent requires roughly eighty-three farm-years of experience, which is longer than the working life of the farmer, the sensors, and in most cases the company that supplied the software.

There are exactly three ways out of this arithmetic, and this book uses all three. The first is to shorten the timestep, which trades sample count against relevance: a greenhouse climate controller acting every five minutes generates 288 transitions per day, but climate control is a different and easier problem than seasonal water allocation. The second is to learn in a simulator and transfer, which relocates the difficulty into the fidelity of the simulator and is the subject of Chapter 7. The third is to learn from logged data generated by whatever policy the farm was already running—offline reinforcement learning ([Levine et al., 2020](#))—which relocates the difficulty into the coverage of the logged data and is the subject of Chapter 6. None of the three removes the problem. Each converts it into a different problem with better-understood failure modes, and understanding those conversions is most of the intellectual content of agricultural agent design.

1.2.2 Irreversibility is not uniform, and it is not a property of the action

Engineering intuition from other control domains suggests classifying actions by magnitude. Agriculture requires classifying them by reversibility, and reversibility turns out to depend on *when* the action occurs rather than on what the action is. Table 1.1 makes the point with a set of actions that a farm agent might plausibly command.

The third row is the one that overturns the usual intuition. Inaction is an action, and in agriculture it is frequently the irreversible one. A safety architecture that constrains only what the agent may *do*, and imposes no obligation on what it must not *fail* to do, has left the most consequential failure mode unguarded. This is why Chapter 9 specifies safety constraints as two-sided invariants on the state—soil moisture must remain above a floor as well as below a ceiling—rather than as one-sided limits on the action.

1.2.3 The plant is a low-pass filter with a memory

A third structural feature deserves mention because it shapes the reward design of Chapter 5. Crop response to water and nutrient supply is not instantaneous, not linear, and not memoryless. The standard empirical relation, due to [Doorenbos and Kassam \(1979\)](#), expresses the fractional yield loss as proportional to the fractional deficit in seasonal transpiration,

$$1 - \frac{Y_a}{Y_m} = k_y \left(1 - \frac{ET_a}{ET_c} \right), \quad (1.1)$$

Table 1.1: Reversibility and severity of a representative set of agent actions. The same physical action—applying water—occupies three different rows depending on crop stage and antecedent condition. Reversibility is measured as the time within which the system state can be restored to what it would have been; severity is the consequence if it cannot. Classifications are engineering judgements, offered as a starting template to be revised per crop and site.

Action in context	Reversibility	Recovery time	Severity
Irrigation, vegetative stage, soil below field capacity	partial	hours–days	low
Irrigation, flowering stage, heat-stressed canopy	partial	days	medium
Withholding irrigation, flowering stage	none	—	high
Greenhouse ventilation	full	minutes	low
Greenhouse heating during frost event	full	minutes	medium
Nitrogen application above crop demand	none (leaching)	weeks	medium–high
Pesticide application	none (ecological)	days–seasons	high
Field drainage	full	hours	low
Harvest initiation	none	—	high
Actuation of machinery with people in the work envelope	none	—	critical

where Y_a and Y_m are actual and maximum attainable yield, ET_a and ET_c are actual and potential crop evapotranspiration accumulated over the period of interest, and k_y is a stage-specific yield-response factor. The engineering content of Equation (1.1) lies almost entirely in the stage dependence of k_y . For maize, k_y is of order 0.4 during establishment and rises above 1.4 at flowering; the same water deficit imposed at two moments six weeks apart differs in consequence by a factor of three or more. An agent that optimises a season-aggregate water objective without stage weighting will, entirely rationally, spend its water uniformly, and will be wrong.

1.3 The autonomy ladder

If the consequences of autonomous action vary this much across decisions, then autonomy cannot be a property of the system. It must be a property of the decision. This is the single most important structural commitment in this book, and everything in Chapters 9 through 11 is machinery for enforcing it.

Table 1.2 defines the six levels, adapted from the levels-of-automation tradition in human factors (Parasuraman et al., 2000) and from the driving-automation taxonomy whose numbering it deliberately echoes. Figure 1.2 renders the same ladder alongside the human role and the engineering controls that each level presupposes.

Five properties of this ladder are worth stating explicitly because each is violated routinely in practice.

Table 1.2: The autonomy ladder AL-0–AL-5. The final column is the operative one: it states what must be *built* before the level may be claimed, and it is the basis of the deployment gates in Chapter 12.

Level	Name	Agent authority and human role	Controls that must exist
AL-0	No autonomy	Agent displays data. Human decides and acts.	Nothing beyond data integrity.
AL-1	Advisory	Agent recommends. Human approves <i>and</i> executes.	Decision logging; explanation on demand.
AL-2	Supervised execution	Agent executes actions the human has approved, one approval per action cycle.	Action gating; per-action audit record.
AL-3	Bounded autonomy	Agent acts without per-action approval, inside an envelope the human approved in advance.	Safety shield; resource budget enforcement; anomaly monitor.
AL-4	Delegated autonomy	Agent handles routine operation and exceptions. Human sets objectives and reviews periodically.	All of AL-3 plus drift detection, automatic rollback, and a tested downgrade path.
AL-5	Audited autonomy	Agent operates independently inside an approved envelope. Human audits and retains emergency override.	All of AL-4 plus a formally specified envelope, independent hardware interlocks, and an incident-liability arrangement.

Design principle 1.1 (Autonomy is per-decision). A single agent operates at different autonomy levels for different decisions simultaneously. The same irrigation agent may hold AL-4 for greenhouse ventilation, AL-3 for vegetative-stage irrigation, AL-2 for irrigation during flowering, and AL-1 for any fertiliser application. A system that exposes one global “autonomy setting” has committed a category error and will be operated at the lowest level any of its decisions requires, which wastes the autonomy it was built for.

Design principle 1.2 (Escalation is automatic, de-escalation is not). Reduction of autonomy in response to a trigger—sensor loss, model drift, out-of-distribution conditions, a near-miss—occurs immediately and without waiting for human confirmation. Restoration of autonomy requires an explicit human decision after a defined observation period. The asymmetry is deliberate: safety must not depend on a human being reachable, and trust must be re-earned rather than restored by timeout.

The remaining three properties are that autonomy must be revocable at any moment including during execution; that a claim to a level requires *evidence* obtained at that level in a comparable context, which is the content of the maturity gates in Chapter 12; and that emergency override

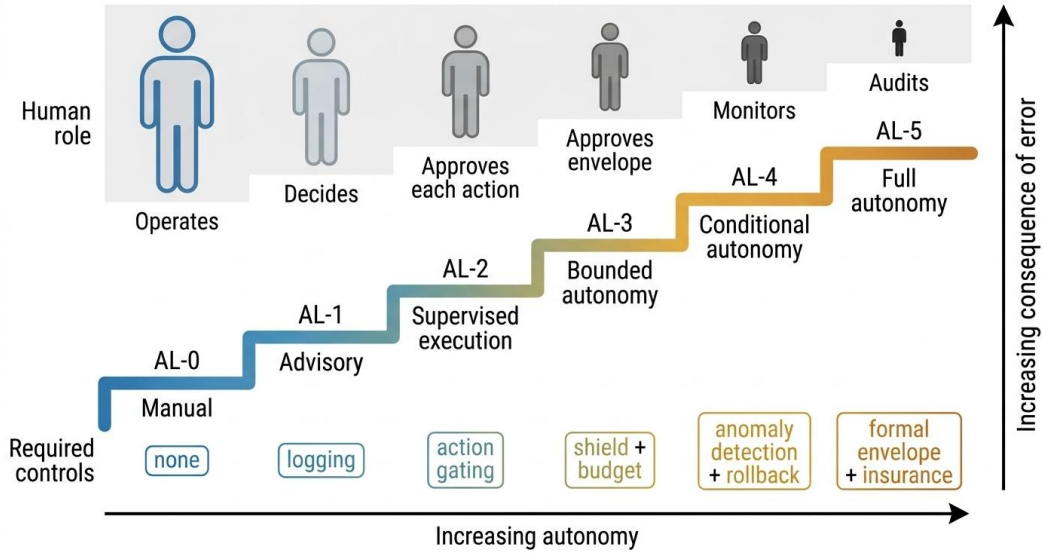


Figure 1.2: The autonomy ladder. Ascending the staircase does not consist of granting the agent more permission; it consists of building the controls in the bottom row. A system that has been configured to AL-4 without an automatic downgrade path is not at AL-4; it is at AL-2 with the supervision removed.

remains absolute at every level including AL-5, implemented in hardware, and tested on a schedule (Chapter 9).

1.3.1 Choosing a level: risk scoring

The mechanism that connects Table 1.1 to Table 1.2 is a risk score. We use the conventional ordinal product

$$R = L \times C, \quad (1.2)$$

with likelihood $L \in \{1, \dots, 5\}$ from *rare* to *almost certain* and consequence $C \in \{1, \dots, 6\}$ from *negligible* to *catastrophic*, and we map the resulting score onto an autonomy ceiling. Figure 1.3 shows the mapping used throughout this book.

Two warnings about Equation (1.2) belong here rather than in a footnote, because misuse of ordinal risk products is endemic. First, L and C are ordinal scales, so R is ordinal too: a score of 20 is not twice as bad as a score of 10, and averaging risk scores across a portfolio of hazards is meaningless. The score is useful only for ranking and for triggering a threshold. Second, the product form systematically under-weights low-probability catastrophic outcomes relative to how a farmer, an insurer, or a regulator will weight them after the fact. Where consequence reaches the top of the scale—injury to a person, irreversible loss of a season’s crop—the correct treatment is not a score but a categorical prohibition, which is why AL-0 and AL-1 appear in the top-right of the matrix regardless of likelihood.

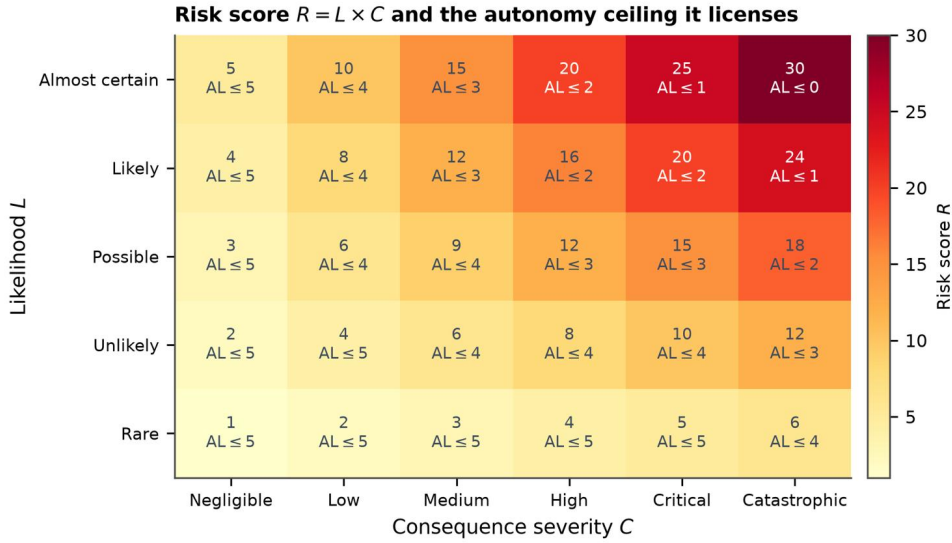


Figure 1.3: The ordinal risk score $R = L \times C$ and the autonomy ceiling it licenses. Read the matrix as a veto, not as a recommendation: it states the highest level at which a decision may be automated, never that automation at that level is desirable. The banding is an engineering convention adopted for this book, not a standardised scale.

Worked example 1.1 Assigning an autonomy ceiling to three decisions

Consider three decisions on a maize and greenhouse-tomato operation.

(a) Greenhouse ventilation on a warm afternoon. Failure mode: vents open when they should be closed, or fail to open in heat. Likelihood of a wrong command in any given week, given a validated controller and a redundant temperature sensor: *possible*, $L = 3$. Consequence: transient departure from the optimal climate band, fully reversible within minutes by the next control cycle, $C = 2$ (low). Hence $R = 6$, which the matrix maps to an AL-4 ceiling. Ventilation is a good candidate for delegated autonomy, and this is the standard finding.

(b) Withholding irrigation on the third day of the flowering window. Failure mode: the agent defers irrigation on a forecast of rain that does not arrive. Likelihood: forecasts of convective rainfall at 24-hour lead miss frequently enough that we take *likely*, $L = 4$. Consequence: with $k_y > 1.4$ at flowering (Equation 1.1), a multi-day deficit at this stage is irreversible within the season and costs a substantial fraction of yield on the affected zones, $C = 5$ (critical). Hence $R = 20$, mapping to an AL-2 ceiling: the agent may propose the deferral, but a human approves each such action. This is exactly the decision that Chapter 6 will show a naively trained policy getting wrong.

(c) Powering a gantry that traverses a greenhouse aisle. Failure mode: motion commanded while a worker is in the aisle. Likelihood, with an interlocked light curtain: *rare*, $L = 1$. Consequence: injury, $C = 6$ (catastrophic). The product $R = 6$ would suggest AL-4. *This is where the product form fails.* Human injury is subject to categorical treatment: the ceiling is AL-1, the interlock is hardware, and no software argument may raise the level. Record the reasoning in the risk register, not the arithmetic.

1.4 The agent's operating loop

Figure 1.4 shows the loop that every chapter of this book elaborates. It is drawn deliberately to separate three things that implementations habitually merge.

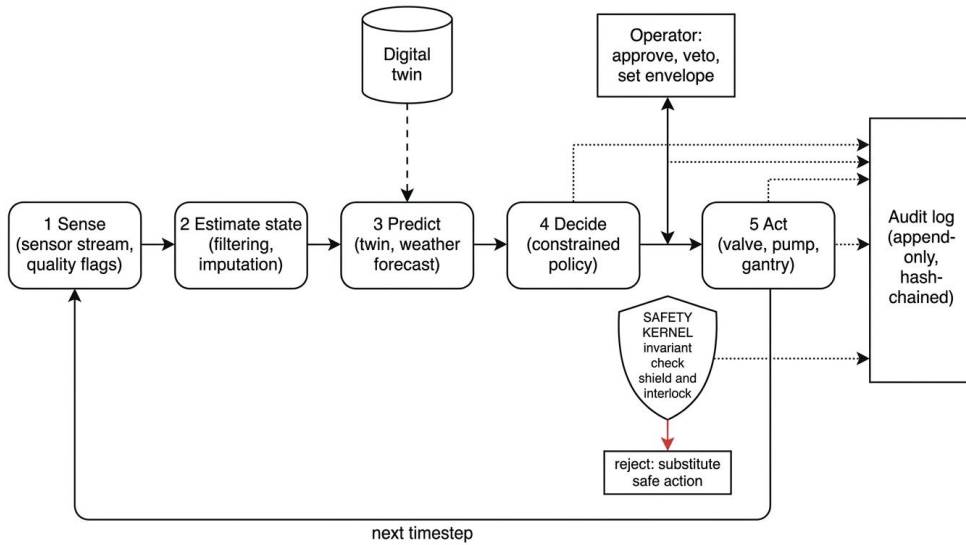


Figure 1.4: The agent's operating loop. The *safety kernel* sits between decision and actuation, not inside the policy; the *audit log* is written by three separate stages so that a missing record is itself detectable; and the operator interacts with the kernel's envelope rather than with the policy. Chapters 5 to 8 build the upper half of this diagram and Chapters 9 to 11 build the lower half.

The first separation is between the policy and the constraint. A learned policy $\pi(a | s)$ that has been trained with a penalty term for constraint violation is not a constrained system; it is an unconstrained system with a preference. The safety kernel of Chapter 9 is a distinct component with distinct code, distinct failure modes, and, at its lowest layer, distinct hardware. The second separation is between the decision and the record: the agent writes an audit entry before it acts, not after, so that “no log, therefore no action” is enforceable. The third separation is between the operator's authority over the *envelope* and the operator's authority over the *action*. At AL-3 and above the operator does not approve actions; the operator approves the set of actions the agent may take, and the distinction is what makes oversight scale beyond a handful of zones—a point Chapter 11 quantifies.

1.5 Evidence, and how this book labels it

Agricultural AI produces more claims than evidence. This is not a moral failing peculiar to the field; it follows from the structural facts of Section 1.2. Field validation of a seasonal control policy requires a season, replicated plots, an independent measurement chain, and a control treatment, and it produces a single noisy observation per year. Simulation produces a thousand seasons in an afternoon. The incentive gradient points in an obvious direction, and the literature shows it: a substantial fraction of published performance improvements for agricultural reinforcement learning are simulation results reported in the vocabulary of field results, and the more careful assessments find that learned policies

frequently fail to beat well-tuned conventional scheduling once the comparison is made fairly (Kelly et al., 2024).

This book therefore labels every substantive claim with one of five levels.

The evidence ladder

ESTABLISHED

A mathematical result, a settled physical relation, or the text of a standard or regulation. Not a measurement claim, and therefore not something an uncertainty interval applies to. This rung exists so that the next one is not stretched to cover theorems.

VERIFIED

Measured in the field, with a control or baseline, by a party other than the system's designer, and reported with an uncertainty interval.

DOCUMENTED

Specified in detail sufficient for an independent team to reimplement, but not measured. Most reference architectures live here.

INFERRED

Derived from a validated model or from established physics, with the inference chain stated. Link budgets live here.

CLAIMED

Asserted by a designer or vendor without an independent measurement. Most vendor performance figures live here.

UNKNOWN

Not established at any level. Saying so is a finding, not an omission.

A seventh label, **ILLUSTRATIVE**, is attached to individual numbers rather than to claims. It marks a value chosen so that an example computes—a plausible field capacity, an assumed water price—and it means: the arithmetic that follows is exact, the input is not evidence. Worked examples say so explicitly in their closing line.

Evidence note

The claims of this chapter sit as follows. The four-paradigm periodisation is **DOCUMENTED** and is a framing device, not a finding. The sample-complexity arithmetic of §1.2.1 is **INFERRED**: the transition counts follow from stated assumptions, and the algorithmic sample requirements are typical figures from Mnih et al. (2015) and Schulman et al. (2017) rather than measurements on an agricultural task. Equation (1.1) and the stage dependence of k_y are **VERIFIED** in the agronomic literature (Doorenbos and Kassam, 1979) though the specific coefficient values vary by cultivar and site. The autonomy ladder is **DOCUMENTED**: it is an engineering proposal adapted from Parasuraman et al. (2000), and no study establishes that systems built to it are safer than systems built without it. The risk-to-autonomy mapping in Figure 1.3 is a **CLAIMED** convention of this book.

Pitfall: the percentage without a denominator

“Our system reduces water use by 25%” is not a claim until three quantities are supplied: the baseline it is measured against, the boundary over which water is counted, and the effect on yield. Reduction against a deliberately naive fixed-calendar baseline is easy and uninformative—Chapter 6 obtains 51 % against exactly such a baseline in simulation, and

(continued)

the number means almost nothing. Reduction measured at the plot while the irrigated area expands is not a reduction at all (Chapter 12). Reduction achieved by accepting a yield loss the farmer did not agree to is a different product. Demand all three.

1.6 The running example

A textbook needs a system concrete enough to be attacked. Throughout this book we use AGRIFARM-AI, a reference deployment specified in full in Appendix E and introduced incrementally as the machinery to describe it becomes available.

Reference site | AgriFarm-AI

Introduction. AGRIFARM-AI is a 7.6 ha mixed operation comprising two open-field maize zones (Zone A, 4.0 ha; Zone C, 3.0 ha) and one protected tomato zone (Zone B, a 0.6 ha greenhouse). The three zones share a single pump rated at 40 m³/h and a seasonal water allocation of 19 000 m³ issued by a basin authority. Field telemetry runs over a LoRaWAN star topology to a single gateway; the greenhouse additionally carries a visible-light link from its luminaires. Compute is a farm-edge server with an intermittent cellular uplink.

Status. AGRIFARM-AI is a DOCUMENTED design. It is fully specified, its components are commercially available, and the design has been costed. It has not been built, and no performance figure in this book that is attributed to it is a measurement. Where the text reports numbers for AGRIFARM-AI they come from the simulation described in Chapter 6 or from the analytic models of Chapter 3, and are labelled accordingly. The reason for constructing the example this way, rather than borrowing a real deployment, is given in Appendix E: the deployments whose performance is publicly claimed do not publish enough detail to be attacked, and a case study that cannot be attacked cannot teach security engineering.

Chapter summary

Agricultural automation crossed a discontinuity when software acquired authority over actuators, and the discontinuity is about authority rather than about model sophistication. Three structural features of the farm make that authority difficult to grant safely: the reward horizon exceeds the correction horizon by weeks, so an agent cannot learn from consequences fast enough to avoid them; irreversibility is a property of the action-in-context rather than of the action, so inaction at the wrong moment is frequently the irreversible choice; and crop response is stage-weighted, so uniform optimisation of a seasonal objective is systematically wrong. The engineering response is to make autonomy a per-decision property, assigned from a risk score and bounded by machinery that is independent of the policy. Because the field's evidence base is thinner than its claim base, every substantive statement in this book carries a label from a five-level ladder, and UNKNOWN is a legitimate and frequently correct label.

Review questions

- R1.1 State the criterion in Definition 1.1 that separates an agent from a decision support system, and explain why the criterion refers to authority rather than to model complexity.

- R1.2 A float valve maintaining a stock tank satisfies Definition 1.1. Why does this not trivialise the definition? Identify the property of modern agentic systems that changes the engineering problem.
- R1.3 An irrigation agent makes one decision per day per zone over a 130-day season on a farm with six zones. How many transitions does it observe per year? If a learning algorithm needs 3×10^5 transitions, how many farm-years does it require?
- R1.4 Explain why withholding irrigation can be the irreversible action, and state the consequence for how safety constraints must be specified.
- R1.5 Using Equation (1.1), compute the fractional yield loss for a crop that transpires at 70 % of potential through a stage with $k_y = 1.5$. Repeat for $k_y = 0.4$. What does the ratio imply for reward design?
- R1.6 Distinguish Principle 1.1 from the common design in which a system exposes a single global autonomy setting. What operational cost does the global setting impose?
- R1.7 Why is de-escalation of autonomy human-approved while escalation is automatic? Construct a scenario in which the reverse convention causes harm.
- R1.8 Two hazards score $R = 12$ and $R = 6$. Is the first “twice as risky”? Justify your answer from the measurement scale of L and C .
- R1.9 In Worked Example 1.1(c) the product $R = L \times C$ gives 6, yet the assigned ceiling is AL-1. Explain the reasoning and give a second decision on a farm that should receive the same categorical treatment.
- R1.10 A vendor reports “30 % water saving.” List the three quantities you must obtain before the claim can be evaluated, and state what each one could conceal.

Hands-on exercises

- E1.1 **Autonomy audit.** Choose a real agricultural operation you can find public documentation for. Enumerate at least twelve distinct decisions it makes in a season. For each, assign L , C , and an autonomy ceiling using Figure 1.3, and write one sentence justifying the consequence rating. Produce a table sorted by ceiling. Identify the decisions where your rating is most sensitive to an assumption you cannot check.
- E1.2 **Sample-complexity study.** Write a short script that, for timesteps of 1 day, 1 hour, and 5 minutes, computes the number of transitions a single-zone agent observes per season and the number of seasons needed to reach 10^5 , 10^6 and 10^7 transitions. Plot the result on log axes. Then write two paragraphs on what is lost, agronomically, when the timestep is shortened to obtain samples.
- E1.3 **Reversibility classification.** Extend Table 1.1 with ten actions drawn from a cropping system other than the ones used here—orchard, paddy rice, or protected horticulture. For each, state the recovery mechanism explicitly (what physical process restores the prior state?) and the time it requires. Flag any action whose reversibility changes sign across the season.
- E1.4 **Stage-weighted objective.** Using k_y values from a published source for a crop of your choice, construct a stage-weighted seasonal water objective and compare it numerically against an unweighted one for a hypothetical deficit of 20 % imposed (a) uniformly, (b) entirely during the most sensitive stage, (c) entirely during the least sensitive stage. Report the three yield outcomes and the implication for an agent that is allowed to choose when to impose the deficit.

E1.5 Evidence relabelling. Take any three performance claims from the marketing material of a commercial agricultural AI product. Assign each a level from the evidence ladder in §1.5 and justify the assignment in two sentences. Then write the sentence the vendor would have to publish for the claim to qualify as **VERIFIED**.

Further reading

Walter et al. (2017) and Klerkx et al. (2019) give complementary framings of the transition to digital agriculture, the first technological and the second social-scientific; read them together, because the failure modes in Chapter 11 are mostly social. Liakos et al. (2018) surveys machine learning applications across the agricultural task space and is useful for locating where agentic control sits relative to the much larger body of work on prediction. Parasuraman et al. (2000) remains the clearest statement of why automation must be allocated by function rather than granted wholesale, and its argument transfers to the ladder of §1.3 with almost no modification. For the sample-complexity argument, Levine et al. (2020) is the standard reference on what can be recovered when online interaction is unavailable.