Magnetic Fluidic Computing

Admissible Circulation, Thermal Transport, and Contracts at Module Boundaries

Flyxion
Independent Researcher

November 2026

Abstract

Magnetic fluidic computing is an architecture in which compute pods circulate through a closed loop of docking, computation, ejection, cooling, and return under magnetic control, so that heat removal, routing, and scheduling are one process and no pod occupies a fixed position. The monograph argues that the architecture replaces a metaphysics of storage with a metaphysics of trajectory, and turns that argument into a mathematical program. The primitive object is an admissibility structure, a directed graph of configurations whose edges depend monotonically on pod temperature. From it the monograph derives, with proofs, a Lawvere metric of accessibility, an accessibility entropy that cooling provably restores, a notion of trajectory identity that determines that entropy, and a trajectory memory whose convergence and loss are stated precisely. The pod dynamics are treated exactly. An overdamped-limit theorem justifies the gradient flow, a piecewise-exact thermal model gives the period and duty cycle in closed form and shows that, for prescribed transit times and threshold crossings, the reduced thermal cycle is reached in finite time and recurs, a claim for the full machine requiring separate return conditions on mechanical state, shared coolant, queues, and checkpoint state, and a closed-network analysis gives the pool size at which the hub saturates, the heat-rejection ceiling on utilisation, and the entropy generated per cycle. Viability theory supplies the safe ejection threshold. A second architecture, in which a conductive carrier slides through the cores of a segmented accelerator, is analysed as the continuum limit of a stream of circulating pods; its temperature profile makes thermal admissibility a global property of the power assignment, and placing the heaviest loads upstream minimizes the peak. Contracts at module boundaries are then developed for circulating modules: gluing and repair obstructions, a necessary and sufficient interval condition for exchange windows that fall in the times when both pods are docked, a rejoin condition that couples cooling time to training staleness, and a typed commit chain that gives trajectory memory a checkable form. No device has been built, every numerical value is an assumption, and the final chapter states the evidentiary status of each result together with the measurements that would test it.

Keywords: magnetic fluidic computing, admissibility, Lawvere metric, accessibility entropy, viability theory, pod circulation, sliding thermal carrier, exchange contracts, typed commit.

Contents

I Admissibility **1**

1 Introduction **2**
1.1 The Problem of Continuation 2
1.2 What the Program Establishes 3
1.3 Conventions 3

2 The Pattern of Return **4**

3 Admissibility Structures **6**
3.1 Configurations and Thermal Classes 6
3.2 Accessibility Distance as a Lawvere Metric 6
3.3 Accessibility Entropy and Its Restoration 7
3.4 The Scope of Restoration 8
3.5 Observable Continuations and the Refinement Obligation 9
3.6 Trajectory Identity 9
3.7 Trajectory Memory 10
3.8 Order, Frequency, and Receipts 11

4 Geometry of Accessibility **12**
4.1 The Accessibility Metric and Its Curvature 12
4.2 The Stationary Density 12

II Circulation **14**

5 Pod Dynamics **15**
5.1 The Pod as a Hybrid System 15
5.2 The Force on a Pod 15
5.3 The Overdamped Limit 16
5.4 Why Circulation Requires Switching 16
5.5 The Thermal Cycle 17
5.6 Timed and Threshold Returns 18

6 The Circulation Network **20**
6.1 Throughput and Pool Size 20
6.2 The Heat-Rejection Ceiling 20
6.3 Two Entropies 21
6.4 Capacity Is Necessary, Not Sufficient 21
6.5 Plant Boundary and Absorption Cooling 22

7 Viability and Tangency **23**
7.1 Invariance of a Constraint Set . . . 23
7.2 The Safe Ejection Threshold . . . 23
7.3 Learned Controllers on the Constraint Set . . . 24

III Transport by Moving Solids **26**

8 Parcel Streams and Sliding Carriers **27**
8.1 A Stream of Parcels . . . 27
8.2 Heat Paths in Present Accelerators . . . 27
8.3 A Segmented Accelerator with Sliding Carriers . . . 28
8.4 Steady-State Heat Transport . . . 29
8.4.1 A worked example . . . 31
8.4.2 Carrier speed . . . 32
8.5 Friction, Wear, and the Cost of Motion . . . 33
8.6 Degraded Operation and the Thermal Contract . . . 34
8.7 Relation to Other Heat Paths . . . 36
8.8 What the Two Architectures Share . . . 36

IV Contracts at Module Boundaries **37**

9 Local Transitions and Global Admissibility **38**
9.1 Local Checks and a Shared Limit . . . 38
9.2 Constraints No Pairwise Check Can Reproduce . . . 38
9.3 Overlaps and the State Sheaf . . . 39
9.4 Repair and Its Obstruction . . . 39

10 Exchange Windows under Circulation **41**
10.1 The Interval Condition . . . 41
10.2 Docking Windows . . . 42
10.3 A Worked Example . . . 42

11 Commit, Staleness, and Recovery **44**
11.1 The Typed Commit . . . 44
11.2 Drift and the Rejoin Condition . . . 45
11.3 Receipts, Replacement, and Recovery . . . 45

V Synthesis **47**

12 Determination and Invariance **48**
12.1 Invariants of an Admissibility Structure . . . 48
12.2 The Coarsest Faithful Abstraction . . . 48
12.3 Liveness and Viability . . . 49
12.4 Apparent Goal Direction . . . 50

13 Protocols and Limits **51**
13.1 Pod and Contract Runs 51
13.2 Carrier Runs 52
13.3 The Status of the Results 53
13.4 Limits and Open Problems 53

A Selected Exercise Solutions **55**

Bibliography **55**

Part I

Admissibility

Chapter 1

Introduction

A conventional computer fixes its parts and moves its data. Processors sit in sockets, memory sits at addresses, and heat is a nuisance that a separate subsystem carries away. Magnetic fluidic computing reverses the assignment. The compute units, which are called pods, are encapsulated modules that circulate continuously through a closed loop. A pod docks at a hub, computes until its temperature reaches an ejection threshold, is carried by magnetic forces to a cooling bath, gives up its heat there, and returns to the hub once it has cooled to a return threshold. The loop Dock, Compute, Heat, Eject, Cool, Return, and back to Dock is the unit of the architecture. Routing is cooling, and cooling is scheduling, because the same circulation performs all three.

Magnetic fluidic computing treats this circulation as the organizing fact of the machine. The architecture embodies a transition from storage to trajectory: memory, entropy, identity, and agency are all functions of a single pair, a routing space and a set of admissible paths. This monograph develops that proposal as a mathematical program. Each notion is given a definition that can be checked and each result a proof that can be read line by line, and every claim carries a stated status.

1.1 The Problem of Continuation

A compute pod leaves its dock with a partially completed job, a rising temperature, and a claim on a route to cooling. What must be preserved so that a different pod can finish the job? What can be gained by moving the hardware rather than moving only its heat or its data? This book treats these as related problems in the design of continuations. A continuation has a physical path, a computational meaning, a resource cost, and conditions under which its result can be committed, and the four are studied together because each constrains the others.

The thermodynamic and mechanical parameters used throughout are hypothetical. The engineering program compares the circulating machine with fixed compute modules cooled by stationary cold plates and, separately, with stationary modules serviced by a moving thermal carrier. The mathematics starts from a single ideal pod and introduces, in turn, timing, contested resources, partial observation, replacement, and learned control. Exercises appear throughout and include calculations, counterexamples, and design tasks. The last part gives an experimental program in which every claim has a stated way of failing.

1.2 What the Program Establishes

The program has four layers, and each is developed in a part of the book. The first layer is the structure of admissibility itself. Part I defines an admissibility structure as a finite directed graph of pod configurations whose edges are available or unavailable according to the pod's thermal class, with the single axiom that a cooler pod has every move that a hotter pod has. Memory, entropy, identity, and distance are then derived from this structure with proofs, in a form that can be computed for a given design.

The second layer is the physics of circulation. Part II treats the pod as a magnetizable body in a viscous fluid whose motion is, in a precisely stated limit, a gradient flow. It treats the pod's temperature with exact solutions of the heating and cooling phases, and it treats the population of pods as a closed network whose throughput, saturation, and heat-rejection limits follow from operational laws. Viability theory is used to say what it means for a thermal constraint to be maintained and to compute the threshold at which a pod must leave the hub.

The third layer is a second architecture that turns out to be the same one seen from another side. In a segmented accelerator a conductive carrier slides through a bore in each core and returns cooled from a liquid-cooled node. Part III shows that a carrier is the continuum limit of a stream of circulating pods and derives the temperature profile, the dependence of each core on every core upstream of it, the ordering that minimizes the peak temperature, the role of cooling stations, the ceiling on the temperature at which heat can be delivered to a coolant, the diminishing return of speed, and the consequences of a stall.

The fourth layer is the service that surrounds the hardware. When circulating modules are sold as a cloud service, what is promised is a commitment about what happens at module boundaries. Part IV develops gluing and repair, a feasibility theorem for exchanges that must fit in the times when both endpoints are docked, a rejoin condition that couples the cooling time to the staleness that a training job can tolerate, and a typed commit chain that gives the notion of trajectory memory a checkable form. Part V proves that the derived quantities are invariants of the admissibility structure and states, result by result, what has been proved, what depends on a physical model, and what is interpretation.

1.3 Conventions

Results are numbered by chapter. Definitions fix terms, axioms state assumptions about the physical systems that are used and not derived, and theorems are proved from the axioms and definitions in force. Where a result depends on a physical model, the model's assumptions are listed before the statement, and the proof is conditional on them. Interpretive remarks are marked as such and are not used in any proof. Numerical examples use assumed parameters, collected in the chapters where they are introduced. No example reports a measurement. The bibliography is at the end of the book, and the companion works cited by name are the author's own, listed there.

The philosophical background of the program, namely process metaphysics [1, 2, 3], the physics of computation and information [4, 8, 9], attractor networks [6], trail-based collective routing [7], and the questions about mechanism, mind, and life raised by Penrose and Rosen [10, 11], is background and is not used in any proof.

Chapter 2

The Pattern of Return

This chapter has a different voice from the rest. It contains no equations and no results, and nothing in later chapters depends on it. It states the picture that the mathematics is meant to make exact.

When most computing systems remember something, they retrieve a stored representation. Information is placed somewhere, a search later finds it, and the memory exists because a record persists. A circulating architecture begins from a different assumption, that a memory need not be a thing. A memory may be a circulation.

In conventional machines computation and maintenance are separate activities. The processor works, the cooling system removes the waste, and storage keeps the results. Each subsystem is distinguished by its place, and the paths between places exist to serve them. Inside a circulating machine the distinctions begin to dissolve. A computational element works only for a time. As its temperature rises it migrates into the cooling region, and once it has recovered it returns to work. What looks like a cooling cycle is also a routing cycle, and what looks like a routing cycle is also a memory cycle, because a pattern survives by coming back. Nothing needs to stay put for the system to remember. The same configuration needs to keep returning.

The inversion changes the character of several inherited ideas. Memory becomes a special case of circulation. A frequent computation is a pod that makes the same circuit often, and the path is worn smooth by use. A forgotten memory has not necessarily been deleted. It has lost the pathways that allowed it to recur. To remember is less like searching an archive and more like re-entering a stable current, so that one does not find a memory but re-enacts a trajectory.

Entropy changes meaning with it. In a conventional computer heat is waste to be expelled. In a circulating architecture temperature is information, because it decides which regions of the routing space a pod may enter. A hot pod must cool before it docks, and a cool pod may dock wherever there is room. Cooling restores possibilities. The refrigeration cycle is then an engine of accessibility, a device whose purpose is not to eliminate disorder in the thermodynamic sense but to widen the set of futures available to a pod that has exhausted them.

Routing is the navigation of that set. Every decision about where a pod may go next is a decision about which futures to keep open. Agency, if the word is allowed, is what the shape of the constraints looks like from outside. A pod moves toward the hub when it is cool and retreats to the bath when it is hot, and it does so without representing the hub or planning a route. The appearance of purpose comes from the field.

A river remains a river while its water is replaced, an organism remains itself while its cells are replaced, and a flame persists while it consumes different fuel from one moment to

the next. In a circulating machine, as in these, what endures is neither matter nor position. What endures is the pattern of return, and more exactly the structure of constraints that makes some trajectories admissible and others not. The chapters that follow say precisely what that means and which parts of it can be proved.

Chapter 3

Admissibility Structures

3.1 Configurations and Thermal Classes

A pod is at every moment in some configuration. The configuration records which cell of the routing space it occupies, which may be a docking position, a segment of channel, or a slot in the bath, and it records the pod's thermal class, a coarse grading of its temperature into finitely many ordered levels. A thermal class does not replace the temperature. It records which side of each threshold the temperature lies on, which is what the routing layer can act on.

Definition 3.1 (Routing system)**.** A *routing system* consists of a finite set S of cells, thermal classes $0, 1, \dots, J$ ordered from coolest to hottest, for each class j a set of directed edges $E_j \subseteq S \times S$ that are available to a pod of that class, and a cost $c(e) > 0$ for each edge $e \in E_0$, which is the time a pod needs to traverse it. The graph $G_j = (S, E_j)$ is the *admissibility graph at class* j.

The decisive property of a thermal constraint is that it only removes options. A pod that is too hot to dock has a subset of the options of the same pod when cool. This is stated as an axiom because it is a property of the control policy and of the physics together, and it is the only assumption on which the main theorem of the chapter depends.

Axiom 3.2 (Thermal monotonicity)**.** For classes $j \le j'$, every edge available at class j' is available at class j, that is, $E_{j'} \subseteq E_j$.

A walk in G_j is a finite sequence of cells in which each consecutive pair is an edge of E_j. The set of walks of a given length is finite. The routing space is the set of cells and the admissible path set is the set of walks, so the pair of the program becomes the pair $(S, \{E_j\})$.

Remark 3.3 (Frozen classes). The walks counted below keep the pod's thermal class fixed at j. This is the right object for the question what is admissible now, which is how an admission controller uses temperature. A walk during which the pod's class changes is a walk in a product graph whose states are pairs of cell and class, and Chapter 5 supplies the dynamics that govern the class. Nothing in this chapter depends on the class remaining frozen along the walk.

3.2 Accessibility Distance as a Lawvere Metric

The distance that matters to a pod is the time needed to get somewhere, and the time needed to return need not equal it. The right notion is the one introduced by Lawvere,

in which a metric space is a category enriched over the nonnegative reals with the reverse order and addition [12]. The axioms are weaker than those of a metric. Symmetry is not required, distinct points may be at distance zero, and distances may be infinite.

Definition 3.4 (Accessibility distance)**.** For cells s, s' and class j, let $d_j(s, s')$ be the minimum total cost of a walk from s to s' in G_j, and $d_j(s, s') = \infty$ if there is none.

Proposition 3.5. *For each j, (S, d_j) is a Lawvere metric space: $d_j(s, s) = 0$ and $d_j(s, s'') \leq d_j(s, s') + d_j(s', s'')$. Moreover $d_j(s, s') \leq d_{j'}(s, s')$ whenever $j \leq j'$. The distance is not symmetric in general.*

Proof. The empty walk has cost zero, which gives the first property. Concatenating a walk from s to s' with a walk from s' to s'' gives a walk from s to s'' whose cost is the sum, and the minimum over walks is at most the cost of this one, which gives the triangle inequality. If $j \leq j'$ then $E_{j'} \subseteq E_j$ by Axiom 3.2, so every walk of $G_{j'}$ is a walk of G_j with the same cost, and the minimum over the larger set is not larger. For asymmetry take cells a, b, c with edges $a \to b \to c \to a$ of cost one each. Then $d(a, b) = 1$ and $d(b, a) = 2$. □

Remark 3.6 (Distances on full configurations). The proposition concerns costs that are zero for the empty execution and additive under concatenation. On full configurations, where free state changes put distinct configurations at distance zero, the distance is in general an extended pseudometric. Projection to cells can destroy the triangle inequality: a cheap path into a dock may end with a hot pod, while the cheap path out of it may require a cool pod at its start. The definition freezes the thermal class so that the distance is taken in a single graph, where concatenation is legitimate.

The loop of the architecture is exactly such a cycle, and it is irreversible by design, so the asymmetry is the typical case. A symmetric accessibility metric is recovered in the continuum in Chapter 4, where the flow is replaced by a gradient and the field has no preferred sense of traversal.

3.3 Accessibility Entropy and Its Restoration

Definition 3.7 (Accessibility entropy)**.** Let A_j be the adjacency matrix of G_j and $N_L(s, j) = (A_j^L \mathbf{1})_s$ the number of walks of length L in G_j that start at s. The *finite-horizon accessibility entropy* is $S_L(s, j) = \log N_L(s, j)$, with $\log 0 = -\infty$, and the *accessibility entropy* is $S_{\mathrm{acc}}(s, j) = \limsup_{L \to \infty} \frac{1}{L} \log N_L(s, j)$.

Counting walks avoids the need for a measure on a space of continuous paths. The limit exists and equals the logarithm of the largest spectral radius among the strongly connected components of G_j that are reachable from s whenever that spectral radius is positive, by the Perron–Frobenius theory of nonnegative matrices [13]. A cell from which no cycle is reachable has $N_L = 0$ for large L and $S_{\mathrm{acc}} = -\infty$, which records that a pod at such a cell is stranded.

Theorem 3.8 (Accessibility restoration)**.** *Under Axiom 3.2, for classes $j \leq j'$, every cell s, and every L, $N_L(s, j) \geq N_L(s, j')$. Consequently $S_L(s, j) \geq S_L(s, j')$ and $S_{\mathrm{acc}}(s, j) \geq S_{\mathrm{acc}}(s, j')$. If some walk of length L in G_j starting at s uses an edge of $E_j \setminus E_{j'}$, the inequality at horizon L is strict.*

Proof. Since $E_{j'} \subseteq E_j$, every walk of length L in $G_{j'}$ is a walk of length L in G_j, so the set counted by $N_L(s, j')$ is a subset of the set counted by $N_L(s, j)$. Taking logarithms and then limits preserves the order. A walk of G_j that uses an edge outside $E_{j'}$ is in the larger set and not in the smaller, so the subset is proper and the counts differ. □

The theorem is the precise content of the statement that cooling restores accessibility. Cooling lowers the class of a pod, the admissible edge sets grow, and the number of admissible futures cannot decrease. The theorem needs no differentiability and no measure on continuous paths, and it gives a criterion for when the restoration is strict, namely that the cooler class admits an edge that is actually on some walk from the pod. Chapter 6 contrasts this semantic entropy with the thermodynamic entropy that cooling also changes.

3.4 The Scope of Restoration

Theorem 3.8 compares graphs. A physical statement compares pods, and the passage from one to the other needs a hypothesis that can fail. Fix a horizon H, an observation alphabet Σ, an explicit cap L on the number of events, and let $\mathcal{A}_H(s) \subseteq \Sigma^{\leq L}$ be the set of event words realized from the hybrid state s within H. The cap, or a minimum separation between events, prevents unbounded word length at a finite horizon. Put $S_H(s) = \log |\mathcal{A}_H(s)|$.

Proposition 3.9 (Conditional restoration)**.** *Let s_c, s_h differ only in initial pod temperature, $T_c \leq T_h$. Suppose every admissible controlled execution from s_h has a counterpart from s_c with the same observed word within H. Then $\mathcal{A}_H(s_h) \subseteq \mathcal{A}_H(s_c)$ and $S_H(s_h) \leq S_H(s_c)$, and the inequality is strict if the inclusion is proper and $\mathcal{A}_H(s_c)$ is nonempty.*

Proof. Choose a realization of any word of $\mathcal{A}_H(s_h)$ and use its colder counterpart. This is an inclusion of finite sets, and cardinalities and their logarithms are monotone under inclusion, strictly so for a proper inclusion. □

The hypothesis can be discharged by replaying the same controls at the same jump times. This requires order-preserving thermal evolutions and resets, mechanical evolution that does not depend on temperature under replay, guards that are closed under lowering temperature, and permission for the colder pod to take the same jumps at the same times. The last requirement is where the hypothesis fails for a machine with compulsory actions. A dock that must eject exactly at T_e ejects a colder pod later. With the parameters of Table 5.1, a pod docked at $T_r = 45$ °C reaches T_e after 78.3 s, and a pod docked at 55 °C reaches it after $\tau_h \ln \frac{T_\infty - 55}{T_\infty - T_e} = 59.4$ s. For the horizon $H = 70$ s the hot pod realizes a word containing an ejection and the cold pod does not, so $\mathcal{A}_H(s_h) \not\subseteq \mathcal{A}_H(s_c)$. Restoration of options is therefore a statement about availability. It holds for the graphs of Axiom 3.2, in which thresholds only gate edges, and it does not hold for executions in which a threshold forces an action. A claim about cooling on an actual execution must also account for elapsed time, changed position, and what other pods did during the cooling interval.

Cardinality is one choice among several. Let w be a nonnegative weight on words and $S_H^w(s) = \log \sum_{\omega \in \mathcal{A}_H(s)} w(\omega)$. Under inclusion the weighted entropy is monotone as well, and two states with equal cardinality can have different weighted entropy, which is the form in which, for instance, deadline-relevant continuations are counted above idle ones. Any comparison must use the same vocabulary, horizon, and weights. A continuum of physical paths may induce a single word, so cardinality measures the continuations that are distinguishable at the chosen resolution. The thermodynamic balance of Theorem 6.4 is a separate account in different units. An increase in S_H takes no part in it, and a natural research question is how much physical energy is spent to restore a particular set of accepted continuations.

Exercise 3.10. Give a three-temperature finite model in which S_H has a plateau as the temperature falls. Then assign weights to the words so that two states of equal cardinality have different weighted accessibility.

3.5 Observable Continuations and the Refinement Obligation

Definition 3.11 (Observable continuation)**.** For a given contract and horizon, the *continuation language* of a state is the set of observable event words obtainable by admissible execution from that state. The alphabet may distinguish successful docking, a verified commit, a rejected proposal, a timeout, and a suspension.

A physical route can implement several event words, and several physical routes can implement one word. A computation enters through an encoding, a rule that decides which executions count as successful, and a decoding of their outputs. For input b and accepted output y the *refinement obligation* is

$$\mathrm{Enc}(b, s_0) \ \wedge \ s_0 \xRightarrow{\gamma} s_1 \ \wedge \ \mathrm{Accept}(\gamma, s_1, y) \implies R(b, y), \tag{3.1}$$

where R specifies the service. This is a partial correctness statement. A separate progress claim must give conditions under which an accepted execution exists, or occurs by a deadline. A controller that suspends every job satisfies many partial correctness implications vacuously, which is why the two obligations are kept apart in every contract of Part IV.

Exercise 3.12. Construct two pods that traverse the same geometric path and produce different results. Then construct two paths that produce the same result while differing in their recovery behaviour. Specify an event alphabet that distinguishes the second pair.

3.6 Trajectory Identity

Two pods might be called architecturally equivalent if they follow the same path, but a usable statement needs a definition of equivalence that is stated independently of paths.

Definition 3.13 (Labelled routing system and equivalence)**.** A routing system is *labelled* if each edge carries a label from a finite set Λ of operations, such as dock, eject, cool, and return. The *traces* of a cell s at class j are the finite sequences of labels along walks from s in G_j. Two cells are *trace equivalent* if they have the same traces. The system is *deterministic* if for each cell, class, and label there is at most one edge with that label.

A bisimulation is a relation $\sim$ on cells such that whenever $s \sim s'$ and there is an edge $s \xrightarrow{a} t$ there is an edge $s' \xrightarrow{a} t'$ with $t \sim t'$, and symmetrically. Cells that are related by some bisimulation are indistinguishable by any procedure that interacts with them through the operations.

Theorem 3.14 (Trajectory identity)**.** *In a deterministic labelled routing system, trace equivalence is a bisimulation, so two cells are trace equivalent if and only if they are bisimilar.*

Proof. Suppose s and s' have the same traces and $s \xrightarrow{a} t$. Then a is a trace of s, hence of s', so there is an edge $s' \xrightarrow{a} t'$, which is unique by determinism. A sequence w is a trace of t if and only if aw is a trace of s. By determinism the same holds for t' and s', and s and s' have the same traces, so t and t' have the same traces. The case of an edge from s' is symmetric. Hence trace equivalence is a bisimulation. A bisimulation preserves traces by induction on length, so bisimilar cells are trace equivalent. □

The theorem says that for the architecture as it is controlled, with at most one admitted operation of each kind at a given place, the identity of a pod from the scheduler’s point

of view is the set of operation sequences it can perform, and two pods are interchangeable exactly when those sets coincide. The equivalence classes are the machine's notion of identity, and the next proposition shows that they determine the entropy.

A partition of S is *equitable* for G_j if the number of edges of E_j from a cell s into a block B' depends only on the block B containing s and on B'. This is the counting refinement of bisimulation, and it is the condition under which cells of a block can be merged without changing how many walks there are.

Proposition 3.15. *If s and s' lie in the same block of an equitable partition of G_j, then $N_L(s, j) = N_L(s', j)$ for every L, and hence $S_{\mathrm{acc}}(s, j) = S_{\mathrm{acc}}(s', j)$.*

Proof. For $L = 0$ both counts equal one. Suppose the claim holds for $L-1$, so that N_{L-1} is constant on blocks, with value $n_{L-1}(B')$ on block B'. Then $N_L(s, j) = \sum_{s \to t} N_{L-1}(t, j) = \sum_{B'} m(B, B')\, n_{L-1}(B')$, where $m(B, B')$ is the number of edges from s into B', which depends only on the block of s. The same expression holds for s'. □

Accessibility entropy is therefore a function of the identity class when identity is taken in the counting sense, and the quotient graph of an equitable partition has the same entropy as the graph. This is the first of the determination results collected in Chapter 12.

3.7 Trajectory Memory

A circuit is a closed walk that begins and ends at a docking cell. In a design with bounded transit times the set $\mathcal{C}$ of circuits of bounded length is finite. The memory of a pod is the set of circuits traversed, weighted by frequency. The weighting is what carries the information, so the definition below keeps it.

Definition 3.16 (Trajectory memory)**.** Let $\gamma_1, \gamma_2, \dots \in \mathcal{C}$ be the circuits traversed by a pod in order. The *trajectory memory* after n circuits is the empirical measure $\mu_n = \frac{1}{n} \sum_{i \le n} \delta_{\gamma_i}$ on $\mathcal{C}$, and its support $M_n = \operatorname{supp} \mu_n$ is the set of circuits that have been traversed.

Proposition 3.17 (Convergence and loss)**.** *Suppose the sequence of circuits is an irreducible Markov chain on the finite set $\mathcal{C}$ with stationary distribution π. Then $\mu_n \to \pi$ almost surely, and almost surely there is a finite random n_0 such that $M_n = \mathcal{C}$ for all $n \ge n_0$. If at some time the available edge set changes from E to E', a circuit $\gamma \in \mathcal{C}$ can recur if and only if all of its edges lie in E'.*

Proof. The first statement is the ergodic theorem for irreducible Markov chains on a finite state space. Since $\pi(\gamma) > 0$ for every state of an irreducible chain, every circuit is visited infinitely often almost surely, and since $\mathcal{C}$ is finite all of its elements have been visited by a finite random time. A circuit is traversable exactly when each of its edges is admissible, which gives the last statement. □

The proposition formalizes the fragility and resilience of trajectory memory. Replacing a pod does not change π, which depends on the routing policy and the edge set, so the memory survives the replacement. Changing the edge set so that a circuit is no longer admissible removes it from what can recur, and recall of that circuit requires a restoration of its edges. This is a failure mode that depends on the routing topology and not on any stored value.

3.8 Order, Frequency, and Receipts

The empirical measure of Proposition 3.17 preserves frequency and discards order. The circuit sequences $ABAB$ and $AABB$ have the same empirical measure and different two-step statistics. A memory that must preserve order needs a different object. Let $h_{k+1} = H(h_k, r_{k+1})$, where H is a digest function and r_{k+1} is a canonical record containing the job and attempt identifiers, the input and software identifiers, the accepted output, and references to the evidence used for admission. The sequence (h_k) is a *receipt chain*.

Proposition 3.18 (Binding)**.** *If H is collision resistant, then no efficient procedure finds two distinct record sequences $(r_1, \dots, r_n) \neq (r'_1, \dots, r'_n)$ with equal heads $h_n = h'_n$ computed from the same h_0.*

Proof. Suppose $h_n = h'_n$ for distinct sequences and let j be the largest index at which $(h_{j-1}, r_j) \neq (h'_{j-1}, r'_j)$. Such an index exists, since if the pairs agreed for all j the sequences would be equal. By maximality $h_j = h'_j$, so the pairs are distinct inputs with equal digest, which is a collision. □

Adjacent hashes bind records under the stated integrity assumption. Retrieval also requires the referenced bytes and permission to read them, so a receipt chain is evidence that a history occurred and not a copy of it. The statistics of circuits are still governed by the chain they come from.

Proposition 3.19 (Return statistics)**.** *Let s_k be a finite irreducible Markov chain with transition matrix P and stationary law π, and for a circuit mark $W_k = f(s_k, s_{k+1})$ let $\widehat{\mu}_n(w) = n^{-1} \sum_{k \le n} \mathbf{1}\{W_k = w\}$. Then $\widehat{\mu}_n(w) \to \sum_{i,j:\, f(i,j)=w} \pi_i P_{ij}$ almost surely.*

Proof. The pairs (s_k, s_{k+1}) with $P_{s_k s_{k+1}} > 0$ form a finite chain on the set of positive-probability edges. It is irreducible, because from edge (i, j) one can reach any edge (k, l) by a positive-probability path from j to k followed by the edge (k, l). Its stationary law at (i, j) is $\pi_i P_{ij}$, as one checks by summing over predecessor edges. The time-average theorem for irreducible finite chains applied to the indicator of the mark gives the claim. □

Stationarity of a policy alone supplies neither irreducibility nor a common limit. An irreducible periodic chain can have one-time marginals that oscillate while its time averages converge, and the periodic control of a circulating loop is of this kind. It is represented by augmenting the return state with its phase, after which irreducibility must be checked on the augmented chain. A path observer may also erase exactly the information needed to choose the next route, so that the update from one circuit word to the next is in general a kernel conditional on the marked state and not an operator on words.

Exercise 3.20. Give an irreducible chain with alternating marginals and convergent edge frequencies. Give a second chain with two closed classes whose time averages depend on the initial state. Say what each failure would look like in a fleet log.

Chapter 4

Geometry of Accessibility

The continuum version of the theory replaces the cell graph by a planar region and the accessibility entropy by a smooth function. It supplies a symmetric metric, a curvature, and a stationary density, and its statements concern a model in which pods move in a plane under a drift and noise. Chapter 3 does not depend on it.

4.1 The Accessibility Metric and Its Curvature

Let $\Omega \subset \mathbb{R}^2$ be a connected open region and $S : \Omega \to \mathbb{R}$ a smooth function, a smoothed accessibility entropy. Define the Riemannian metric $g_S = e^{-2S} g_0$, where g_0 is the Euclidean metric. The length of a piecewise smooth curve γ is $\int e^{-S(\gamma(t))}\, |\dot\gamma(t)|\, dt$. The integral is over arclength, so it does not depend on the parametrization.

Theorem 4.1 (Accessibility metric)**.** *The infimum $d_S(x_1, x_2)$ of the length of curves in Ω from x_1 to x_2 is a metric on Ω, and the Gaussian curvature of g_S is $K_S = e^{2S}\, \Delta S$.*

Proof. A Riemannian distance on a connected manifold is a metric. For a conformal metric $g = e^{2u} g_0$ on a planar domain the Gaussian curvature is $K = -e^{-2u} \Delta u$ [14]. Here $u = -S$, which gives $K_S = e^{2S} \Delta S$. □

Corollary 4.2 (Sign of curvature at attractors)**.** *Let $\dot x = \nabla S(x)$ be the accessibility flow. At a local maximum of S, which is an attractor of the flow, $\Delta S \le 0$, so $K_S \le 0$. At a local minimum, which is a repeller, $K_S \ge 0$.*

In the metric g_S the neighbourhood of an attractor is a region where lengths are short, because e^{-S} is smallest where S is largest, and negative curvature is the geometry of a basin that draws curves in.

4.2 The Stationary Density

Let a pod move by the stochastic equation $dX_t = \nabla S(X_t)\, dt + \sqrt{2D}\, dW_t$ on a bounded region with reflecting boundary, where $D > 0$ is the strength of the fluctuations in pod temperature and routing field.

Proposition 4.3. *The stationary density is $\rho(x) \propto e^{S(x)/D}$. In particular $\rho \propto e^{S}$ only when $D = 1$.*

Proof. The stationary Fokker–Planck equation is $0 = -\nabla \cdot J$ with flux $J = \rho \nabla S - D \nabla \rho$, and the reflecting boundary imposes $J \cdot n = 0$. For $\rho \propto e^{S/D}$ one has $D \nabla \rho = \rho \nabla S$, so $J = 0$

everywhere and ρ is stationary. A nondegenerate diffusion with reflection on a bounded region has a unique stationary density [15], so this is it. □

Remark 4.4 (Circulation carries current). The density of Proposition 4.3 has zero stationary flux, because the drift is a gradient. A circulating population sustains a nonzero stationary current, and predicting its occupancy requires the actual transport law including the queues. High occupancy of a region may indicate congestion and not a large number of options, so occupancy alone cannot identify accessibility. The three-cycle $a \to b \to c \to a$ with unit costs has directed distances $d(a, b) = 1$, $d(b, a) = 2$, and adding a long waiting edge at one cell changes its occupancy without changing which cells are reachable.

Remark 4.5 (Hidden curvature, interpretive). The program called Hidden Curvature holds that the apparent geometry of space is derived from a more basic structure of admissibility. The architecture gives a finite instance in which the two geometries visibly differ. Two docking positions may be a few centimetres apart in Euclidean terms and unreachable from one another in the admissibility graph at every thermal class, so that the Lawvere distance between them is infinite, while two positions on opposite sides of the loop are connected by a single channel. In this instance the admissibility distance, not the Euclidean one, is what determines how a computation can move. The claim that physical spacetime is of this kind is not tested here.

Part II

Circulation

Chapter 5

Pod Dynamics

5.1 The Pod as a Hybrid System

A pod is described at every moment by

$$s = (q, x, p, T, e, z, h, a),$$

where q is the discrete mode, x, p are position and momentum, T is the lumped temperature, e is the available onboard energy, z is the working computational state, h is a reference to a committed record, and a records authorization. The global state also includes dock occupancy, cooling capacity, outstanding reservations, and any controller clock. These variables are a modelling choice, and removing one is justified when what remains still suffices for the predictions or decisions being made. The cells and thermal classes of Chapter 3 are the discrete skeleton of this state, namely the mode with the docking position, and a coarse grading of T.

Definition 5.1 (Hybrid trajectory)**.** For each mode q let S_q be a continuous state domain, $\dot{s} = f_q(s, u)$ the evolution with control $u \in U_q$, and for modes q, q' let $G_{qq'} \subset S_q$ be a guard and $J_{qq'} \subset G_{qq'} \times S_{q'}$ a reset relation. A *trajectory* is a finite succession of flow segments and permitted jumps, marked with elapsed time and event records, in which the end state of each segment equals the start state of the next.

A docking guard can require a sufficiently cool pod and a valid reservation, and a successful commit reset changes the record reference h. The guard and the reset make separate assertions: a cool pod with a valid reservation can still fail to establish a usable data link. If the routing fields are externally clocked, time or clock phase belongs in the state, so that concatenation retains the fact that a route open at dusk may be closed at dawn.

5.2 The Force on a Pod

A pod carries a magnetically permeable shell, and the routing layer is an array of electromagnets in the channel walls. The pod is modelled as a magnetizable body of volume V and small effective volume susceptibility χ in a nonmagnetic fluid of pressure field $p(x)$. To leading order in χ the magnetic energy of the body in a field $\mathbf{B}(x)$ is $-\frac{\chi V}{2\mu_0}\|\mathbf{B}(x)\|^2$, and the pressure contributes the work $Vp(x)$ [5]. Define

$$\Phi(x) = -a\,\|\mathbf{B}(x)\|^2 + b\,p(x), \qquad a = \frac{\chi V}{2\mu_0} > 0,\ b = V > 0. \tag{5.1}$$

The force on the pod is $-\nabla\Phi$, which points toward stronger fields and lower pressure, so a pod is attracted to regions of strong field. A thermal term $\gamma(T - T_{\mathrm{opt}})^2$ would not depend on position and would exert no force, so it is omitted from the mechanics. Temperature enters the dynamics through the admission thresholds of Chapter 3 and through the phases of the thermal model below.

With a linear Stokes drag of coefficient λ and an effective mass m that includes the mass of the displaced fluid, the equation of motion is

$$m\ddot{x} = -\nabla\Phi(x) - \lambda\dot{x}. \tag{5.2}$$

The Lagrangian $L = \tfrac{1}{2}m\|\dot{x}\|^2 - \Phi$ and the Hamiltonian $H = \|\pi\|^2/2m + \Phi$ describe the conservative part of this equation, and the drag removes energy at the rate $\lambda\|\dot{x}\|^2$.

5.3 The Overdamped Limit

The gradient flow $\dot{x} = -\nabla\Phi$ is obtained when drag dominates inertia, and the statement can be proved with an explicit error bound. Let $g = \nabla\Phi/\lambda$ and $\varepsilon = m/\lambda$, the inertial time scale. Equation (5.2) becomes $\varepsilon\ddot{x} = -g(x) - \dot{x}$.

Theorem 5.2 (Overdamped limit)**.** *Let g be C^1 on a convex region containing the trajectories considered, with $\|g\| \le G$ and $\|Dg\| \le L$ there. Let x_ε solve $\varepsilon\ddot{x} = -g(x) - \dot{x}$ with $x_\varepsilon(0) = x_0$ and $\dot{x}_\varepsilon(0) = v_0$, and let x solve $\dot{x} = -g(x)$ with $x(0) = x_0$. Put $w_0 = v_0 + g(x_0)$. If $\varepsilon L < 1$, then for $0 \le t \le T$,*

$$\|x_\varepsilon(t) - x(t)\| \le \varepsilon\, e^{LT}\Big[\|w_0\| + TL\,(G + W)\Big], \qquad W = \frac{\|w_0\| + \varepsilon LG}{1 - \varepsilon L}.$$

Proof. Put $v = \dot{x}_\varepsilon$ and $w = v + g(x_\varepsilon)$, so that $\dot{x}_\varepsilon = -g(x_\varepsilon) + w$. Then $\varepsilon\dot{v} = -g - v = -w$. Since $\dot{w} = \dot{v} + Dg\,v$ with $v = w - g$, we have $\dot{w} = -w/\varepsilon + Dg(x_\varepsilon)(-g(x_\varepsilon) + w)$. The last term has norm at most $L(G + \|w\|)$. Variation of constants gives $\|w(t)\| \le \|w_0\|e^{-t/\varepsilon} + \varepsilon L(G + W_T)$ with $W_T = \sup_{[0,T]}\|w\|$. Taking the supremum gives $W_T \le \|w_0\| + \varepsilon L(G + W_T)$, hence $W_T \le W$. Integrating the bound, $\int_0^t \|w\| \le \varepsilon\|w_0\| + t\,\varepsilon L(G + W)$. The difference $e = x_\varepsilon - x$ satisfies $\dot{e} = -(g(x_\varepsilon) - g(x)) + w$, so $\frac{d}{dt}\|e\| \le L\|e\| + \|w\|$, and Gronwall's inequality gives $\|e(t)\| \le e^{Lt}\int_0^t \|w\|$. □

The error is of order ε uniformly on bounded time intervals, apart from the initial layer of duration ε in which the velocity relaxes to the slow value. For a pod of mass 1 kg in a fluid with a drag coefficient of 50 kg/s, $\varepsilon = 0.02$ s, which is small against transit times of seconds. Under the overdamped flow the speed along a channel is $\|\nabla\Phi\|/\lambda$, so a one-way transit along a path ℓ takes

$$t_x = \lambda \int_\ell \frac{ds}{\|\nabla\Phi\|}, \tag{5.3}$$

which supplies the transit time of the thermal model below from the magnetic design.

5.4 Why Circulation Requires Switching

The potential of Equation (5.1) is a static function of position, and the gradient flow of a static potential cannot circulate. Both the inertial and the overdamped forms are covered.

Proposition 5.3 (No loop in a static potential)**.** *Let U be C^1, D symmetric positive definite, M constant positive definite. A nonconstant solution of $\dot{x} = -D^{-1}\nabla U(x)$ is not periodic, and neither is a nonconstant solution of $M\ddot{x} = -\nabla U(x) - D\dot{x}$.*

Proof. Along the first flow $\frac{d}{dt}U(x(t)) = -\nabla U^\top D^{-1}\nabla U$, which is strictly negative wherever $\nabla U \neq 0$. If x is periodic with period T and $\nabla U(x(0)) \neq 0$, then $U(x(T)) < U(x(0))$, a contradiction, and if $\nabla U(x(0)) = 0$ the solution is constant. For the second equation the energy $E = \frac{1}{2}\dot{x}^\top M\dot{x} + U$ satisfies $\dot{E} = -\dot{x}^\top D\dot{x}$. Periodicity forces $\int_0^T \dot{x}^\top D\dot{x}\,dt = 0$, so $\dot{x} \equiv 0$. □

A closed circuit therefore cannot be a limit cycle of a flow in a fixed potential. Circulation requires driving: the commanded field pattern must change with the mode of the pod, through the guards of the hybrid model, so that the potential at the moment of departure differs from the potential at the moment of arrival. The ledger of such a switched system is exact.

Proposition 5.4 (Switching ledger)**.** *Let U_σ be a family of potentials selected by a piecewise constant signal $\sigma(t)$ with switching times $t_1 < t_2 < \dots$, and let x solve $M\ddot{x} = -\nabla U_{\sigma(t)}(x) - D\dot{x}$ with $x, \dot{x}$ continuous. Put $E_\sigma = \frac{1}{2}\dot{x}^\top M\dot{x} + U_\sigma(x)$. Then E_σ decreases at the rate $\dot{x}^\top D\dot{x}$ between switches and jumps by $\Delta_k = U_{\sigma(t_k^+)}(x(t_k)) - U_{\sigma(t_k^-)}(x(t_k))$ at t_k. For a periodic orbit of period T with switches $t_1, \dots, t_m$ in one period,*

$$\sum_{k=1}^{m} \Delta_k = \int_0^T \dot{x}^\top D\dot{x}\,dt.$$

Proof. Between switches the computation of the previous proof applies. At t_k velocity and position are continuous, so only the potential term changes, by Δ_k. Periodicity of E_σ over one period equates the total of the jumps with the total dissipation. □

The ledger is reduced. The electrical power drawn by the magnets, and the losses in the drive, are not in it and must be metered separately, so it cannot estimate any part of a facility electricity bill without further measurements.

Exercise 5.5. Write the energy ledger for a magnetic gate that switches between two potentials at a fixed position. Identify the energy jump at the switch and say which part of a facility electricity bill the reduced ledger cannot estimate.

5.5 The Thermal Cycle

The pod is treated as a single thermal mass of heat capacity C. The loop has four phases. In the *docked* phase the pod dissipates power P and loses heat to its surroundings at the hub temperature T_a through a conductance h, so that $C\dot{T} = P - h(T - T_a)$. In the *outbound* transit it neither dissipates nor exchanges heat and its temperature is constant. In the *bath* phase it exchanges heat with the bath at temperature T_b through a conductance k, so that $C\dot{T} = -k(T - T_b)$. In the *return* transit it is again inert. The transit durations are t_x each. The control rule is that a docked pod is ejected when its temperature reaches T_e and a pod in the bath is returned when its temperature reaches T_r. The assumptions are that $T_b < T_r < T_e < T_\infty$, where $T_\infty = T_a + P/h$ is the temperature the docked pod would approach if it were never ejected. Write $\tau_h = C/h$ and $\tau_c = C/k$.

Proposition 5.6 (Phase durations)**.** *A pod docked at temperature T_r reaches T_e after*

$$\tau_D = \tau_h \ln \frac{T_\infty - T_r}{T_\infty - T_e},$$

and a pod entering the bath at T_e reaches T_r after $\tau_B = \tau_c \ln \dfrac{T_e - T_b}{T_r - T_b}$. As $h \to 0$, $\tau_D \to C(T_e - T_r)/P$, the adiabatic dwell time.

Quantity	Value
Heat capacity, dissipated power	$C = 900$ J/K, $P = 500$ W
Hub loss, hub temperature	$h = 1$ W/K, $T_a = 25$ °C, so $T_\infty = 525$ °C
Bath coupling, bath temperature	$k = 30$ W/K, $T_b = 10$ °C
Thresholds	$T_e = 85$ °C, $T_r = 45$ °C
Transit time (each leg)	$t_x = 3$ s
Derived dwell, adiabatic dwell	$\tau_D = 78.3$ s, $C(T_e - T_r)/P = 72.0$ s
Derived cooling time	$\tau_B = 22.9$ s
Derived period, duty cycle	$\tau = 107.2$ s, $\eta = 0.731$

Table 5.1: Assumed parameters of one pod and the quantities that follow from Proposition 5.6 and Theorem 5.7. All input values are assumptions.

Proof. The docked equation has the solution $T(t) = T_\infty - (T_\infty - T_r)e^{-t/\tau_h}$, and solving $T(\tau_D) = T_e$ gives τ_D. The bath equation has the solution $T(t) = T_b + (T_e - T_b)e^{-t/\tau_c}$, and solving $T(\tau_B) = T_r$ gives τ_B. For the limit write $\tau_D = \tau_h \ln(1 + \frac{T_e - T_r}{T_\infty - T_e})$. As $h \to 0$, $T_\infty - T_e \sim P/h$ and $\tau_h = C/h$, so $\tau_D \sim \frac{C}{h} \cdot \frac{(T_e - T_r)h}{P} = C(T_e - T_r)/P$. □

Theorem 5.7 (Exact limit cycle)**.** *Under the control rule above, from any initial state the pod is, after at most one period, on a periodic orbit of period $\tau = \tau_D + \tau_B + 2t_x$, and the orbit is unique. The circulation is free of chattering if and only if $T_e > T_r$. The duty cycle is $\eta = \tau_D/\tau$.*

Proof. Whatever the initial phase and temperature, the pod finishes that phase and passes through the remaining phases in order. During the transits the temperature is constant, so the pod arrives at the bath at T_e, since ejection occurs at T_e, and arrives at the hub at T_r, since return occurs at T_r. From the first docking onward the temperature at every docking is T_r, and the evolution over a circuit is the deterministic evolution of Proposition 5.6. Hence the trajectory from the first docking is exactly periodic with period τ. Uniqueness follows because the periodic orbit is determined by the docking temperature T_r. If $T_e > T_r$ both τ_D and τ_B are positive, so phase durations are bounded below. If $T_e \le T_r$ the thresholds do not separate and the pod ejects as soon as it docks, which is chattering. □

The orbit is a limit cycle reached in finite time and not only asymptotically, because every threshold crossing resets the temperature to a fixed value, so no appeal to a theorem on planar flows is needed. The energy lost to the surroundings during a dwell is $P\tau_D - C(T_e - T_r) \ge 0$, and it is the amount by which the dwell exceeds the adiabatic estimate multiplied by the dissipated power.

A numerical example

Table 5.1 gives assumed parameters for one pod and the quantities that follow from them. With $h = 1$ W/K the dwell is 78.3 s against an adiabatic estimate of 72.0 s, so the correction for loss to the hub is under ten percent and is not negligible for a design that wants the duty cycle to three figures.

5.6 Timed and Threshold Returns

The exact cycle of Theorem 5.7 is threshold-triggered. A machine can instead run each stage for a fixed time, and the return map is then a contraction, which gives a quantitative

statement about perturbations. Let the dock and bath stages last d and c and let transit be insulated. The map F from one dock inlet temperature to the next is

$$F(T) = T_b + e^{-c/\tau_c}\big[T_\infty + e^{-d/\tau_h}(T - T_\infty) - T_b\big], \qquad F'(T) = e^{-d/\tau_h - c/\tau_c} =: r \in (0,1).$$

Theorem 5.8 (Return under perturbation)**.** *Let I be a closed interval with $F(I) \subseteq I$, so that F has a unique fixed point $T_* \in I$. If perturbed returns stay in I and satisfy $|T_{j+1} - F(T_j)| \le \delta$, then*

$$|T_n - T_*| \le r^n |T_0 - T_*| + \delta\,\frac{1 - r^n}{1 - r}.$$

Proof. By the triangle inequality $|T_{j+1} - T_*| \le |T_{j+1} - F(T_j)| + |F(T_j) - F(T_*)| \le \delta + r|T_j - T_*|$. Iterating this scalar recurrence gives the bound. □

The ultimate error is at most $\delta/(1-r)$, so a margin m around T_* is held in the long run when $\delta < (1-r)m$. A finite-time guarantee also keeps the decaying term, and an initial point outside the margin cannot be certified by the ultimate bound alone. For a threshold-triggered cycle the map F does not describe the return, because threshold crossings determine travel times and the return temperature is reset at each crossing, a map of derivative zero. An exactly enforced thermal return guard can fix the temperature at a section while leaving position, coolant state, charge, and queue evolving, so a periodic orbit of the full machine needs a return condition on those variables as well. Verification of $F(I) \subseteq I$ and verification that the intermediate temperatures obey the dock and cooling guards are distinct checks, and a fixed point can describe a repeatable but inadmissible cycle.

A second example

Take $C = 100$ J/K, $P = 200$ W, $h = 2$ W/K, $k = 10$ W/K, $T_a = 30$ °C, $T_b = 25$ °C, $T_r = 45$ °C, $T_e = 65$ °C, so $T_\infty = 130$ °C and $\tau_h = 50$ s. Then $\tau_D = 50\ln(85/65) = 13.4$ s and $\tau_B = 10\ln 2 = 6.9$ s. With two seconds of transit in total the useful fraction is $13.4/(13.4 + 6.9 + 2) = 0.60$, a calculation from assumed parameters to which docking, verification, and queueing add further overhead. Run as a timed cycle with $d = 13.4$ s and $c = 6.9$ s, the map has $r = 0.382$ and fixed point $T_* = 45.0$ °C, so a per-cycle error of 1 K produces an ultimate error of at most 1.62 K.

Exercise 5.9. Calculate the dock temperature immediately before ejection for the timed map at its fixed point. Explain why its return temperature and its peak must both be checked. Compute the largest per-cycle error δ permitted if the return temperature must remain within a prescribed margin of T_*.

Chapter 6

The Circulation Network

A pod does not circulate alone. A hub has N docking positions, a pool of M pods shares them, and the bath has K cooling slots. The population is a closed network in which each pod cycles through four stations, and its throughput, utilisation, and heat rejection obey laws that hold for any such network without assuming a distribution of service times [16]. Throughout, a *circuit* is one pass of a pod through the loop and X is the long-run average number of circuits completed per unit time by the whole pool.

6.1 Throughput and Pool Size

Theorem 6.1 (Throughput bounds)**.** *In any long-run operation of M pods that always occupy one of the phases, with minimal circuit time $\tau = \tau_D + \tau_B + 2t_x$,*

$$X \le \min\Big\{ \frac{M}{\tau},\ \frac{N}{\tau_D},\ \frac{K}{\tau_B} \Big\}.$$

Proof. By Little's law, the mean number of pods in the loop equals the throughput times the mean circuit time [16]. Since every pod is in the loop, $M = X\bar{\tau}$ with $\bar{\tau} \ge \tau$, which gives $X \le M/\tau$. The mean number of pods docked is $X\tau_D$, and it cannot exceed the number N of docking positions, which gives the second bound. The mean number of pods in the bath is $X\tau_B$, and it cannot exceed K. □

Corollary 6.2 (Pool size)**.** *If $K \ge M$, the hub can be saturated, meaning $X = N/\tau_D$, only if $M \ge N\tau/\tau_D = N/\eta$. For $M < N/\eta$ the utilisation of the hub is at most $M\eta/N$.*

Proof. Saturation requires $N/\tau_D \le M/\tau$ by Theorem 6.1, which is $M \ge N/\eta$. For smaller M the bound $X \le M/\tau$ gives utilisation $X\tau_D/N \le M\tau_D/(N\tau) = M\eta/N$. □

This is the pool-size statement $M \ge N/\eta$, as an inequality that holds for any scheduling policy. The bound is attained when the circuit times can be staggered so that hub arrivals are evenly spaced, and it need not be attained otherwise. In the example of Table 5.1 with $N = 8$, $N/\eta = 10.95$, so a pool of 11 pods suffices for saturation in principle, and a pool of 9 gives a utilisation of at most 0.82.

6.2 The Heat-Rejection Ceiling

Cooling is the part of the loop with an external cost. Under the lumped model each circuit removes exactly $C(T_e - T_r)$ from the pod in the bath, so the bath must reject that heat.

Theorem 6.3 (Utilisation ceiling)**.** *Let the cooling plant reject at most $\dot{Q}_{\max}$ at the bath. Then*

$$X \le \frac{\dot{Q}_{\max}}{C(T_e - T_r)}, \qquad u \le \frac{\dot{Q}_{\max}\,\tau_D}{N\,C(T_e - T_r)},$$

where $u = X\tau_D/N$ is the hub utilisation.

Proof. The mean heat rejected at the bath is the product of the throughput and the heat removed per circuit, $X\,C(T_e - T_r) \le \dot{Q}_{\max}$. Dividing by N/τ_D gives the bound on u. □

With the pod of Table 5.1 and a cooling plant of 3.5 kW, the ceiling is $X \le 0.0972$ circuits per second and $u \le 0.952$ for $N = 8$. The hub could not run at saturation even with a pool large enough to supply it, because the plant is the binding constraint. Combined with Theorem 6.1, the utilisation of the hub is bounded by the smaller of the pool-size bound and the ceiling, and the plant sets the second.

6.3 Two Entropies

Cooling changes two entropies, and they move in opposite directions, which can be stated with numbers. Temperatures in this section are absolute.

Theorem 6.4 (Entropy generated in the bath)**.** *Let a pod of heat capacity C cool from T_e to T_r in contact with a bath held at $T_b < T_r$. The entropy of the pod changes by $-C\ln(T_e/T_r)$, the entropy of the bath changes by $C(T_e - T_r)/T_b$, and the total entropy generated is*

$$S_{\text{gen}} = C\Big[\frac{T_e - T_r}{T_b} - \ln\frac{T_e}{T_r}\Big] > 0.$$

Proof. For a body of constant heat capacity, $\Delta S = C\ln(T_r/T_e)$. The bath absorbs the heat $Q = C(T_e - T_r)$ at the constant temperature T_b, so its entropy rises by Q/T_b. Using $\ln x \le x - 1$ with $x = T_e/T_r$ gives $\ln(T_e/T_r) \le (T_e - T_r)/T_r < (T_e - T_r)/T_b$, so the sum is positive. □

For the pod of Table 5.1, $S_{\text{gen}} = 20.6$ J/K per circuit. The pod's own thermodynamic entropy falls by 106.6 J/K, and the bath's rises by 127.1 J/K. By Theorem 3.8 the same operation does not decrease the pod's accessibility entropy, and strictly increases it whenever a thermally gated edge lies on a walk from the pod. The cooling circuit therefore lowers the thermodynamic entropy of the pod, raises its accessibility entropy, and raises the entropy of the world by a positive amount. The accessibility entropy is a count of futures and carries no units, and the statement is not that it equals a thermodynamic quantity but that the two change in the stated directions under the same operation.

6.4 Capacity Is Necessary, Not Sufficient

For M identical pods with cycle duration τ and productive dock dwell τ_D, the aggregate ideal productive dock-time rate is $M\tau_D/\tau$, and sustaining N fully productive docks requires $M\tau_D/\tau \ge N$. This is the necessary capacity bound of Corollary 6.2. Occupancy constraints, phase alignment, cooling contention, and missed docking windows can leave idle time even when it holds. Average capacity does not guarantee that each pod meets its cooling deadline either. For a time interval I, demand that must be served entirely inside I has to fit the capacity available there, which is the interval test developed in Chapter 10. Pods occupying

exclusive berths, or travelling through blocked channels, need discrete reservations or a corresponding scheduling model.

The same bottleneck can be represented by a transition graph with resource tokens. A pod enters a dock by acquiring a dock token and returns it on release, and a route segment is represented similarly. Token conservation does not prevent deadlock, because a pod holding a dock may wait for a transit slot while another pod holds that slot and waits for the dock. A progress proof must rule out the circular wait or provide a maneuver that breaks it. In a unidirectional ring the matter is simple.

Proposition 6.5 (Ring progress)**.** *In a unidirectional ring of S slots, each holding at most one pod and each pod able to advance into the next slot when it is empty, if the number of pods is $M < S$ then in every configuration some pod can advance.*

Proof. Since $M < S$ some slot is empty. The slot behind it in the direction of motion is either empty or occupied, and following the ring backwards from an empty slot one reaches an occupied slot that has an empty slot ahead of it, because $M \geq 1$. That pod can advance. □

At $M = S$ with every slot full, no pod can advance without synchronous rotation. A pool that must meet the saturation bound $M \geq N/\eta$ therefore needs the loop to hold at least one more slot than pods, or a declared buffer that a pod may always enter.

6.5 Plant Boundary and Absorption Cooling

The cooling plant is an optional engineering choice, and so is any plant that uses waste heat to drive it. For an absorption machine, if Q_c is the heat removed from the cold side, Q_g the external driving heat, and W the auxiliary work, the first law at steady state for a boundary around the machine gives $Q_{\text{reject}} = Q_c + Q_g + W$ when these are the only energy streams. A claim that the device runs its own cooling from its waste heat requires a separate balance of temperature grade and energy with every reservoir accounted for, because the temperature of the waste heat limits the plant that can use it, by the ceiling that Proposition 8.11 states for the carrier. The cooler should first be characterized against a controlled external bath, and the plant that supplies the bath varied only afterwards. The driving heat is metered separately from electrical work.

Chapter 7

Viability and Tangency

7.1 Invariance of a Constraint Set

Chapter 3 treated admissibility as a graph. In the continuum, the question whether a constraint such as a temperature limit can be maintained along a trajectory is a question about the invariance of a set under a flow, and it has a classical answer. A set $K \subseteq \mathbb{R}^n$ is *forward invariant* for a vector field F if trajectories that start in K stay in K. The Bouligand tangent cone $T_K(x)$ at $x \in K$ is the set of directions v with $\operatorname{dist}(x+hv, K) = o(h)$ as $h \downarrow 0$.

Theorem 7.1 (Nagumo)**.** *Let K be closed and F locally Lipschitz. Then K is forward invariant for $\dot{x} = F(x)$ if and only if $F(x) \in T_K(x)$ for every $x \in K$.*

This is the theorem of Nagumo and the starting point of viability theory [17, 18]. For a set $K = \{g \leq 0\}$ with smooth boundary the condition reads $\nabla g \cdot F \leq 0$ on $\{g = 0\}$, which says that the field does not point out of the set. For a smooth manifold $M = \{h = 0\}$ the tangent cone is the tangent space, and the condition reads $\Pi_{N_x M} F(x) = 0$, where $\Pi_{N_x M}$ is the orthogonal projection onto the normal space at x. This is the tangency condition that the author's companion work on prediction targets places on admissible update fields [29], and the channel walls of a pod loop are the physical instance. A magnetic force that points into a wall at its surface leaves the routing region invariant only if it is inward or tangent, which is the Nagumo condition for the region.

If the update direction is isotropic in an n-dimensional ambient space and the tangent space has dimension d, the expected squared fraction of the direction that lies in the normal space is $(n-d)/n$. The calculation is elementary. For u uniform on the unit sphere, $\mathbb{E}\|\Pi_N u\|^2 = \operatorname{tr}(\Pi_N)/n = (n-d)/n$. It is a baseline against which a designed field can be compared, since a field that is no better than isotropic spends the fraction $(n-d)/n$ of its force on motion that violates the constraint.

Remark 7.2 (Testing is not proving)*.* Testing the inequality $\nabla g \cdot F \leq 0$ at the boundary differs from proving forward invariance. The latter needs the regularity of Theorem 7.1 and a well-defined admissible evolution. Every allowed jump of the hybrid model must also land in the safety domain, so a controller that learns an apparently tangent vector field between jumps can still violate a guard through a discrete update.

7.2 The Safe Ejection Threshold

The thermal constraint of the hub is not forward invariant. A docked pod with $T_\infty > T_{\max}$ heats toward T_∞ and will leave the set $\{T \leq T_{\max}\}$ if nothing intervenes. Viability theory

asks the weaker question whether there is a control, here the ejection command, that keeps the trajectory in the set, and the set of states from which one exists is the viability kernel [18].

Suppose that after an ejection command the pod continues to dissipate for a latency δ before it leaves the hub. During the latency the docked equation still holds.

Proposition 7.3 (Viability kernel of the hub phase)**.** *A docked pod at temperature $T < T_\infty$ can be kept below $T_{\max}$ by an ejection command if and only if $T \le T_{\text{safe}}(\delta) = T_\infty - (T_\infty - T_{\max})\, e^{\delta/\tau_h}$. The control rule of Chapter 5 keeps the pod in this kernel if and only if $T_e \le T_{\text{safe}}(\delta)$, that is, if and only if $\delta \le \delta_{\max} = \tau_h \ln \dfrac{T_\infty - T_e}{T_\infty - T_{\max}}$.*

Proof. The temperature of a docked pod increases in time because $T < T_\infty$, so the earliest and best ejection command is issued immediately. After the latency the temperature is $T_\infty - (T_\infty - T)e^{-\delta/\tau_h}$, which is at most $T_{\max}$ if and only if $T \le T_{\text{safe}}(\delta)$. If $T > T_{\text{safe}}(\delta)$, every command, including the earliest, leaves the pod above $T_{\max}$ at departure, so no admissible trajectory exists. The rule ejects at T_e, and the statement for T_e is the case $T = T_e$. Rearranging $T_e \le T_{\text{safe}}(\delta)$ for δ gives $\delta_{\max}$. □

With the pod of Table 5.1 and a limit $T_{\max} = 95\ ^\circ\text{C}$, a latency of 5 s gives $T_{\text{safe}} = 92.6\ ^\circ\text{C}$, so the threshold $T_e = 85\ ^\circ\text{C}$ has a margin of 7.6 K, and the largest tolerable latency is $\delta_{\max} = 20.7$ s. The adiabatic estimate $C(T_{\max} - T_e)/P = 18.0$ s is conservative because it neglects the loss to the hub. The result is a lumped-model result, and, as for the stall response of Chapter 8, it is exact for a single-temperature body and a safe estimate for a real junction only if the effective mass C represents the mass coupled to the junction over the latency.

7.3 Learned Controllers on the Constraint Set

A useful learned controller predicts remaining cooling time, dock acquisition delay, or the probability of a failed handshake. Its observations are partial, and the same snapshot of position and temperature can be compatible with several different future commitments, so the relevant reservation and event history must be included or the state estimate represented as a distribution. Calibration matters most near the admission threshold, where one confident error can invalidate a complete cycle.

For a smooth equality-constrained model $g(s) = 0$ whose Jacobian J has full row rank, the Euclidean tangent projector is $\Pi_T = I - J^\top (J J^\top)^{-1} J$.

Proposition 7.4. *Π_T is an orthogonal projector with $J\Pi_T = 0$. Hence for every learned velocity f_θ the projected field $\Pi_T f_\theta$ satisfies $J\,\Pi_T f_\theta = 0$ at that state, and it is the nearest tangent vector to f_θ in the Euclidean norm.*

Proof. $J\Pi_T = J - JJ^\top (JJ^\top)^{-1} J = 0$. Symmetry and $\Pi_T^2 = \Pi_T$ follow by expansion, and the nearest-point property is the characterization of an orthogonal projector onto $\ker J$. □

Finite numerical steps can still drift off a curved manifold, so a retraction or a constraint-preserving integrator is needed for the corresponding discrete guarantee. With inequalities and control bounds, solve for a feasible action

$$u^\star \in \arg\min_{u \in U} \|u - u_\theta\|^2 \quad \text{subject to} \quad \nabla g_j(s) \cdot f(s,u) \le 0 \ \ (j \text{ active}).$$

This local formulation needs a sampled-data margin or a reachable-tube check to justify operation between evaluations, and a defined outcome when the feasible set is empty. A

suspension action with its own physical guard is preferable to assuming that a safe halt is always available. A low prediction error averaged over ordinary states can coexist with a high error rate at the particular guards where decisions matter, so the learned estimate and the verified resource witness are kept distinct in the event record, and Run L1 of Chapter 13 compares equally budgeted controllers with and without the projection.

Part III

Transport by Moving Solids

Chapter 8

Parcel Streams and Sliding Carriers

The circulating pod moves a compute element through a fixed thermal path. The converse arrangement keeps the compute elements fixed and moves a solid through them. This chapter treats the second arrangement as the continuum limit of the first, develops it with the same discipline of stated assumptions, and finds that the central structural fact of Part IV, the difference between local and global admissibility, appears in it in its simplest form.

8.1 A Stream of Parcels

Consider a stream of identical parcels, each of heat capacity K, that pass a heat source one after another at throughput X per unit time. Each parcel enters the exchange region at temperature T_{in} and leaves it at T_{out}.

Proposition 8.1 (Mean heat balance of a stream)**.** *In steady operation the mean power transported by the stream is* $XK\,(T_{\mathrm{out}} - T_{\mathrm{in}}) = C\,\Delta T$ *with* $C = XK$.

Proof. Each parcel carries away $K(T_{\mathrm{out}} - T_{\mathrm{in}})$, and parcels pass at rate X. Steady operation means that no parcel accumulates between the source and the sink. □

For the pod loop of Table 5.1 at saturation, $X = N/\tau_D = 0.1022$ per second and $K = C = 900$ J/K, so the loop has the heat-capacity rate $XK = 91.9$ W/K. Across the ejection and return thresholds, $\Delta T = 40$ K, and the loop transports 3678 W of the 4000 W that the eight docked pods dissipate, the remaining 322 W being lost directly to the hub through the conductance h. A sliding carrier is the limit in which the parcels become infinitesimal and continuous, so that the heat-capacity rate is $C = \rho c A v\, m$ for a conductive solid of volumetric heat capacity ρc, cross-section A, speed v, and m strands in parallel. Only the mean balance is common to the two. The pod exchanges heat by dwelling at a dock, and the carrier exchanges it by sliding contact, so everything that depends on how the exchange happens is derived separately below.

8.2 Heat Paths in Present Accelerators

The thermal resistance of a stack is the sum of the resistances of its layers, and the layers that matter most for the present purpose are the interfaces. A review of four decades of work on thermal contact, gap, and joint resistance shows how strongly an interface depends

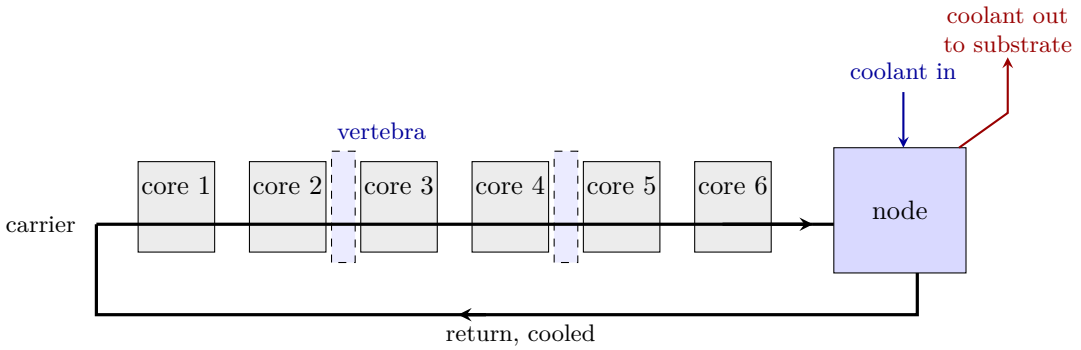


Figure 8.1: Schematic of the spine for $n = 6$. The carrier enters the first bore at temperature θ_0, takes up heat in each core in turn, reaches the node at temperature θ_n, and returns cooled. Dashed regions mark vertebrae, the coolant-bearing elements that may hold additional stations between cores.

on surface roughness, hardness, pressure, and the fluid between the surfaces [26]. A static interface is made as good as the clamping force and the filler allow, and then left alone. The interface that this chapter introduces is dynamic, and a sliding contact differs from a clamped one in that it must tolerate relative motion for the life of the module. Its conductance is therefore not a property of the materials alone, and Section 8.5 treats it as a quantity that the module must declare and that wear can degrade.

Liquid cooling placed close to the heat source reduces the stack. Water flowing in channels etched into the silicon substrate was demonstrated to remove heat from a chip at very high power density [25]. For vertically stacked dies, coolant has been routed between the layers, and the performance of such interlayer cooling has been characterized across parallel-plate, microchannel, and pin-fin structures [21]. The finding from that work that matters here is that, at small hydraulic diameters, the rise in coolant temperature due to the sensible heat it absorbs limits the cooling performance, and not the convective resistance at the wall. The same accounting appears below with the carrier in place of the coolant: the carrier's temperature rises along the spine by the total power divided by its heat-capacity rate, and every core downstream of another sees that rise.

The comparison with vapour chambers, heat pipes, and oscillating heat pipes is taken up in Section 8.7, after the model has been stated, because the comparison depends on which quantities the model makes explicit.

8.3 A Segmented Accelerator with Sliding Carriers

The accelerator is an ordered sequence of n cores along a spine. Each core is a compute element with a bore, a channel that passes through it along the axis of the spine. A carrier is a conductive rod, band, or chain that passes through the bore of every core in order. It is driven so that it moves continuously in one direction along the spine, and it closes on itself by a return path outside the cores, so that no carrier material leaves the module.

The terms of the design are fixed by the following definition. The name *vertebra* is used for the coolant-bearing element of the spine, since a series of such elements along the axis, with the carrier passing through all of them, is what gives the structure its articulated form. The *node* is the vertebra at which the carrier's heat is finally delivered to the coolant of the module, and the *substrate* is the larger body that receives the coolant afterwards.

Definition 8.2 (Segmented accelerator with sliding carriers)**.** A segmented accelerator with sliding carriers consists of the following. *Cores* $1, \dots, n$ in carrier order, with powers $P_1, \dots, P_n \geq 0$. Each core i has a junction resistance R_j between the junction and the bore wall, so that the junction is at the wall temperature plus $R_j P_i$. *Carriers*, which move at speed v and together have heat-capacity rate $C = \rho c A v\, m$, where ρc is the volumetric heat capacity of the carrier material, A the cross-section of one carrier, and m the number of carriers in parallel. *Bore contacts*, each with conductance G between the carriers and the bore wall of a core, and the dimensionless number $\varepsilon = 1 - e^{-G/C}$ that results. *Stations*, which are vertebrae in which the carrier exchanges heat with a coolant stream of capacity rate C_w and inlet temperature T_w, with effectiveness ε_N. A temperature limit $T_{\max}$ applies to every junction.

Remark 8.3 (Reciprocating variant)*.* A set of short sliders that shuttle between a hot position and a cold position transports heat with the same mean heat balance. A slider of mass M_s and specific heat c that completes f cycles per unit time, absorbing heat at its hot position and releasing it at its cold position, carries $M_s c\, \Delta T f$ per unit time, which is the form $Q = C\Delta T$ with $C = M_s c f$. Only this mean balance is claimed. Extending the profile of Proposition 8.6 and the ordering of Proposition 8.9 to a shuttle would require assumptions about when it contacts each core, how completely it cools between strokes, and the temperature ripple within a cycle. The shuttle is therefore a second architecture, whose equivalence to the continuous loop is a matter to be tested, and the analysis below is for the continuous loop.

Remark 8.4 (Electrical isolation)*.* A metal carrier that touches the bore of an active core must be electrically isolated from it. The isolating layer adds a series resistance to the contact, and the conductance G of Definition 8.2 is the conductance of the whole interface, isolation included. A carrier made of a dielectric of high thermal conductivity avoids the layer and has a lower heat capacity per unit volume. The analysis below does not depend on the choice, and the choice enters through ρc and G.

Remark 8.5 (Substrate)*.* The substrate is whatever receives the coolant after the node: a chassis plate network, a rack manifold, or a building circuit. The module's contract concerns its coolant inlet and outlet. The analysis treats the substrate as supplying coolant at temperature T_w and capacity rate C_w and does not model it further.

8.4 Steady-State Heat Transport

The model rests on four assumptions, which are stated here because the propositions are conditional on them.

The first assumption, A1, is that the module is at steady state and every core dissipates its power P_i entirely into the carrier, so that other heat paths from the cores, such as conduction into the spine structure, are neglected. The second, A2, is that each bore wall is isothermal and exchanges with the carrier stream according to the effectiveness relation for a stream passing an isothermal surface [23], so that the heat taken up in core i is $\varepsilon C(T_i - \theta_{i-1})$, where T_i is the wall temperature and θ_{i-1} the carrier temperature on entry to the bore. The third, A3, is that axial conduction along the carrier is negligible against its advection. For a copper carrier of 4×9 mm^2 total cross-section over a spine length of 0.1 m, the conductance kAm/L is about 0.14 W/K, against $C \approx 62$ W/K for the example of Section 8.4.1, a ratio near 0.2 percent. The fourth, A4, is that the carrier is returned to a fixed entry temperature θ_0 by the station that precedes the first core.

Proposition 8.6 (Carrier profile)**.** *Under A1 to A4, the carrier temperature on leaving core i is $\theta_i = \theta_{i-1} + P_i/C$, and the junction temperature of core i is*

$$T_i^{\mathrm{J}} \;=\; \theta_0 \;+\; \frac{1}{C}\sum_{j<i} P_j \;+\; P_i\, a, \qquad a = \frac{1}{\varepsilon C} + R_j.$$

Proof. At steady state the heat taken up by the carrier in core i equals P_i, so the energy balance of the carrier through the bore is $C(\theta_i - \theta_{i-1}) = P_i$. By A2 the same heat is $\varepsilon C(T_i - \theta_{i-1})$, which gives $T_i = \theta_{i-1} + P_i/(\varepsilon C)$ for the wall. Summing the balances from the first core gives $\theta_{i-1} = \theta_0 + \frac{1}{C}\sum_{j<i} P_j$, and adding the junction offset $R_j P_i$ gives the stated expression. □

The expression has two terms that depend on the powers. The term $P_i a$ depends on the core alone, and the term $\frac{1}{C}\sum_{j<i} P_j$ depends on every core upstream. The second term is what makes the assembly more than the sum of its cores.

Definition 8.7 (Local and global admissibility)**.** Core i is *locally admissible* at power P_i if $\theta_0 + P_i a \le T_{\max}$, which is the condition for the core to meet the limit when it alone dissipates. A power assignment is *globally admissible* if $T_i^{\mathrm{J}} \le T_{\max}$ for every i.

Proposition 8.8 (Global admissibility is strictly stronger)**.** *If $P_j > 0$ for some $j < i$, then T_i^{J} exceeds the local value $\theta_0 + P_i a$ by $\frac{1}{C}\sum_{j<i} P_j > 0$, so global admissibility of core i implies local admissibility of core i and the converse fails. Moreover, the node couples the cores further: the entry temperature θ_0 is itself a nondecreasing function of the total power $Q = \sum_i P_i$.*

Proof. The first statement is read directly from Proposition 8.6. For the second, the carrier delivers Q to the coolant at the node, so $Q = \varepsilon_N C_{\min}(\theta_n - T_w)$ with $C_{\min} = \min(C, C_w)$, and $\theta_n = \theta_0 + Q/C$. Eliminating θ_n gives $\theta_0 = T_w + Q\left(\frac{1}{\varepsilon_N C_{\min}} - \frac{1}{C}\right)$. Because $C_{\min} \le C$ and $\varepsilon_N \le 1$, the bracket $\frac{1}{\varepsilon_N C_{\min}} - \frac{1}{C}$ is nonnegative, so θ_0 is nondecreasing in Q. □

The proposition has the form of the general observation that local transitions can each be admissible while the global configuration is not [27]. Here the shared quantity is the carrier's heat content, and the boundary through which the cores exchange is the carrier itself.

Proposition 8.9 (Ordering)**.** *Among all orderings of a fixed multiset of core powers along the carrier, the ordering with powers in nonincreasing order minimizes the steady-state peak $\max_i T_i^{\mathrm{J}}$.*

Proof. The total power Q and hence θ_0 are the same for every ordering, so the comparison concerns the profile of Proposition 8.6 with $b = 1/C$ and $a \ge b$, where $a - b = \frac{1}{\varepsilon C} - \frac{1}{C} + R_j \ge 0$ because $\varepsilon \le 1$. Consider two adjacent cores holding $P < P'$ in that order, and let $X = \theta_0 + \frac{1}{C}S$ with S the power upstream of the pair. Before an exchange the two junctions are at $X + aP$ and $X + bP + aP'$, and the larger is $X + bP + aP'$. After exchanging, they are at $X + aP'$ and $X + bP' + aP$. The first is at most $X + bP + aP'$, and the second is at most $X + bP + aP'$ because $(bP + aP') - (bP' + aP) = (a - b)(P' - P) \ge 0$. The cores upstream of the pair see the same upstream power, and the cores downstream see the same total upstream power, so no other junction changes. Any ordering in which a smaller power precedes a larger one can be improved or left unchanged by such an exchange, and repeated exchanges reach the nonincreasing order without raising the maximum. □

The proposition concerns the peak steady-state temperature and nothing else. The dependence it describes is directional: a core heats every core downstream of it on the same pass, and the finite station returns part of the total load to the inlet temperature of the next pass. It does not say which ordering gives the best response to a stall, which depends on the thermal mass, power, and margin of each core together, as Section 8.6 shows.

Proposition 8.10 (Stations bound the accumulated rise)**.** *Suppose a station returns the carrier to θ_0 after every k cores, with the same θ_0 at each. Then for every core the junction temperature satisfies*

$$T_i^{\mathrm{J}} \;\le\; \theta_0 + \frac{(k-1)\,P_{\max}}{C} + P_{\max}\,a,$$

a bound that does not depend on n, where $P_{\max} = \max_i P_i$.

Proof. Within a group of k cores the upstream sum in Proposition 8.6 contains at most $k-1$ terms, each at most $P_{\max}$. □

The proposition is the sense in which the vertebrae matter. A single node at the end of a spine of n cores accumulates a rise of $(n-1)P/C$ for uniform power, and stations placed between cores cut the rise to $(k-1)P/C$ at the price of one liquid-cooled element for every k cores. The coolant is the other side of the same trade, and the next proposition states it.

Proposition 8.11 (Heat-grade ceiling)**.** *The temperature at which the coolant leaves the node cannot exceed the carrier temperature at the node inlet, and that temperature is itself bounded by the junction limit:*

$$T_{w,\mathrm{out}} \;\le\; \theta_n \;\le\; T_{\max} - R_j P_n - P_n\!\left(\frac{1}{\varepsilon C} - \frac{1}{C}\right).$$

Proof. A passive exchanger cannot heat the coolant above the inlet temperature of the hotter stream, which is the first inequality. For the second, $\theta_n = \theta_{n-1} + P_n/C$ while the wall of the last core is at $\theta_{n-1} + P_n/(\varepsilon C)$ and its junction at that value plus $R_j P_n$. The junction is at most $T_{\max}$, and rearranging gives the bound. □

The proposition says that the heat of the module can be delivered at a high temperature only if the cores run close to their limit. When the heat is wanted at a particular temperature, for example for a hydronic heating circuit, the temperature that the module can offer is limited by the same $T_{\max}$ that limits the cores, and the number of cores per station, k, determines how much of that limit the carrier uses.

8.4.1 A worked example

Table 8.1 gives the parameters of the example. The values are assumptions chosen to exercise the model. The contact coefficient and the junction resistance in particular are placeholders for quantities that Chapter 13 proposes to measure.

With uniform power the junction temperatures of Proposition 8.6 rise by $P/C = 4.03$ K from core to core, from 61.1 °C at the first core to 89.2 °C at the eighth, below the limit of 95 °C. The coolant leaves at 54.1 °C, which is below the ceiling of Proposition 8.11, equal to 76.5 °C for $P_n = 250$ W.

The gap between local and global admissibility is large here. Taken alone, with the carrier entering at the example's $\theta_0 = 38.6$ °C, a core meets the limit up to about 627 W. This is a local screening limit with the inlet held fixed. A single powered core on the same closed loop changes the station load and so its own inlet temperature, and with

Quantity	Value
Cores, power	$n = 8$, $P_i = 250$ W (total $Q = 2000$ W)
Carrier	copper, 3×3 mm, $m = 4$ in parallel, $v = 0.5$ m/s
Volumetric heat capacity, heat-capacity rate	$\rho c = 3.45$ MJ/(m^{3}K), $C = 62.1$ W/K
Bore contact	2.4×10^{-4} m^2 per carrier per core, 5×10^4 W/(m^{2}K), so $G = 48.0$ W/K
Effectiveness	$G/C = 0.773$, $\varepsilon = 0.538$
Junction resistance, limit	$R_j = 0.06$ K/W, $T_{\max} = 95$ °C
Coolant	water, 1.5 L/min, $C_w = 104.5$ W/K, inlet $T_w = 35$ °C
Node effectiveness	$\varepsilon_N = 0.9$
Core thermal mass (Section 8.6)	$K_c = 150$ J/K
Derived: carrier rise, entry, node inlet	$Q/C = 32.2$ K, $\theta_0 = 38.6$ °C, $\theta_n = 70.8$ °C
Derived: coolant outlet	$T_w + Q/C_w = 54.1$ °C

Table 8.1: Parameters of the worked example. All values are assumptions.

Core	1	2	3	4	5	6	7	8
Power, heavy first (W)	450	400	300	250	200	150	150	100
Junction (°C)	79.0	81.8	79.2	79.6	79.1	77.8	80.2	78.2
Power, heavy last (W)	100	150	150	200	250	300	400	450
Junction (°C)	47.6	53.7	56.1	63.0	70.7	79.2	93.1	104.0

Table 8.2: The same eight powers in opposite orders. The maximum is 81.8 °C with the heaviest cores upstream and 104.0 °C with them downstream. The limit is 95 °C.

the parameters of Table 8.1 the isolated-core limit is about 654 W. With all eight cores dissipating, the largest uniform power that keeps every junction below the limit is about 276 W, because the last core carries the accumulated rise and the node raises θ_0 with the total. At 300 W per core every core is locally admissible and the eighth reaches 100.1 °C.

Ordering matters at fixed total power. Table 8.2 gives the same 2000 W distributed unevenly over the eight cores. With the heaviest cores first, the highest junction is at 81.8 °C. With the reverse order, the same powers drive the last core to 104.0 °C and the assembly is inadmissible, though the largest single power is 450 W, well below the local limit.

Stations change the picture, as Table 8.3 shows for uniform 250 W with the return temperature imposed at 38.6 °C. A station after every second core reduces the highest junction from 89.2 °C to 65.1 °C, and the carrier reaches the station inlet at only 46.7 °C. By Proposition 8.11 the coolant leaving such a station cannot be warmer than that. The cooler cores are paid for in lower heat grade, which is the trade named after the proposition.

8.4.2 Carrier speed

At fixed contact conductance G, the carrier speed enters the model through $C \propto v$ alone, and the two terms of Proposition 8.6 respond differently to it.

Proposition 8.12 (Diminishing return of speed)**.** *Fix G, R_j, the core powers, and the coolant. As $C \to \infty$, the upstream term $\frac{1}{C}\sum_{j<i} P_j$ tends to zero and $\frac{1}{\varepsilon C} = \frac{1}{C(1-e^{-G/C})}$ tends*

Cores per station k	Highest junction (°C)	Carrier at station inlet (°C)
1	61.1	42.6
2	65.1	46.7
4	73.1	54.7
8	89.2	70.8

Table 8.3: Effect of station spacing for uniform 250 W, with the carrier returned to 38.6 °C at each station. That return temperature is imposed. A comparison at fixed coolant inlet, flow, and station effectiveness would solve for the return temperature of each configuration. The right column bounds the temperature of the coolant leaving a station.

v (m/s)	C (W/K)	ε	Highest junction (°C)	Friction power (W)
0.25	31.0	0.787	123.8	16
0.5	62.1	0.538	89.2	32
1	124.2	0.321	75.5	64
2	248.4	0.176	76.0	128
4	496.7	0.092	76.2	256

Table 8.4: Carrier speed for the example of Table 8.1, uniform 250 W. The limiting value as $v \to \infty$ is 76.5 °C. Friction heat is not included in the junction temperatures and would add to the node load.

to $\frac{1}{G}$, so the local wall-to-carrier drop tends to P_i/G and does not vanish. For $C > C_w$ the minimum capacity rate in the node is the coolant's, and θ_0 tends to $T_w + Q/(\varepsilon_N C_w)$. The friction power $\mu N v$ per contact grows linearly in v at fixed N and μ.

Proof. Both limits follow from $C(1 - e^{-G/C}) \to G$ as $C \to \infty$. The node relation of Proposition 8.8 with $C_{\min} = C_w$ gives the limit of θ_0, since $Q/C \to 0$. □

Table 8.4 gives the last core of the example over a range of speeds. Friction heat is listed and is not included in the junction temperatures. From 0.25 to 1 m/s the highest junction falls from 123.8 °C to 75.5 °C. Beyond 1 m/s it stops falling, because the upstream rise has become small and the wall drop and coolant terms dominate, while the friction power doubles with each doubling of speed, from 64 W at 1 m/s to 256 W at 4 m/s. In the example the speed at which additional motion stops buying thermal margin is near 1 m/s, and the sweep of Run H1 of Chapter 13 measures where it lies in a real module.

8.5 Friction, Wear, and the Cost of Motion

The carrier moves, so the sliding contacts dissipate power and lose material. Both effects can be bounded with simple relations, and both depend on parameters that this chapter assumes and does not measure.

Friction heat. A contact carrying normal force N at sliding speed v with friction coefficient μ dissipates $\mu N v$. For a lubricated or liquid-metal-wetted interface with $\mu = 0.1$, $N = 20$ N, and $v = 0.5$ m/s, each contact dissipates 1.0 W. The example has $8 \times 4 = 32$ contacts, so the friction power is 32 W, which is 1.6 percent of the 2000 W of the cores. It enters the node load and the carrier rise, and it is the power that the carrier drive must supply. A design in which the friction power approaches a few percent of the load would be spending a significant part of its cooling capacity on its own motion.

Wear. The classical relation attributes a wear volume $V = KNs/H$ to a contact of hardness H under load N over a sliding distance s, with a dimensionless wear coefficient K [20]. At 0.5 m/s the carrier slides 1.58×10^7 m in a year. With $N = 20$ N, $H = 1$ GPa, and the contact area 2.4×10^{-4} m^2 of the example, the depth removed per contact in a year is 1.3 μm for $K = 10^{-9}$, 131 μm for $K = 10^{-7}$, and 1.3 mm for $K = 10^{-6}$. These span three orders of magnitude, from negligible to destructive, and the span is the point: the wear coefficient decides whether the architecture is practical, and its value for the chosen pair of materials and film is an experimental question that no relation in this chapter settles.

Contact conductance over life. Wear changes the surface, and debris changes the interface. The conductance G of Definition 8.2 therefore drifts, and its drift enters the model through $\varepsilon = 1 - e^{-G/C}$. Lower G raises a and all junction temperatures by Proposition 8.6, with the largest effect on the cores that already run closest to the limit. The module's contract should state a floor on G and the rated life over which it holds, and the protocol of Chapter 13 that tests it records the drift of G with sliding distance.

8.6 Degraded Operation and the Thermal Contract

A carrier can stop. The drive can fail, a contact can seize, or debris can jam the return path. In that case A1 no longer describes the module, because the cores lose their path to the node except through whatever residual conduction the structure provides. The specification must say what the module promises when this happens, in the same way that a module of a cloud service must say what it does when a link fails [28].

Let T_i^{J} be the junction temperature of core i at the moment of the stall, let $K_{c,i}$ be an effective thermal mass for core i, and suppose the core continues at power P_i. In a lumped model in which the core is a single temperature, loses no heat, and receives no heat from its neighbours, its temperature rises at $P_i/K_{c,i}$ and reaches the limit after

$$t_i \;=\; \frac{K_{c,i}\,(T_{\max} - T_i^{\mathrm{J}})}{P_i}.$$

The formula is exact for that model. Whether it is a safe estimate for a real junction depends on two things that the model assumes and does not show: that $K_{c,i}$ represents the thermal mass actually coupled to the junction over the response interval, which for a small junction heated at high flux can be much less than the mass of the whole core, and that heat flowing in from neighbouring components or from the stalled carrier cannot raise the temperature faster than the lumped rate. A heater block, which has a larger and more uniform mass than a junction, can show a slower transient than the device it stands for. For these reasons the next proposition is stated as a result of the lumped model.

Proposition 8.13 (Safe response time, lumped model)**.** *In the lumped model, suppose the module detects a stall and reduces every core to a power that its residual path can sustain within a time t_d after the stall. Then no core exceeds $T_{\max}$ provided $t_d \le \min_i t_i$, where each t_i is computed from a positive margin $T_{\max} - T_i^{\mathrm{J}} > 0$.*

Proof. Until time t_d core i heats at $P_i/K_{c,i}$, so its temperature at t_d is $T_i^{\mathrm{J}} + P_i t_d/K_{c,i} \le T_{\max}$ when $t_d \le t_i$. After the reduction the power is sustainable by assumption. □

The proposition needs a positive margin. A core assigned exactly to $T_{\max}$ has $t_i = 0$, and no response is fast enough, so a module that is operated at its limit has no stall response to declare and must not be operated there when a stall is possible.

Mode	Observable	Effect on the model	Required response
Carrier stall or seizure	Speed sensor reads zero, drive current rises	$C \to 0$, A1 fails	Reduce power within $\min_i t_i$, report fault
Contact degradation	Rise of wall temperature at fixed power	G falls, ε falls	Derate by Proposition 8.6, report drift
Coolant loss or reduced flow	Node inlet temperature rises	C_w falls, θ_0 rises	Derate by Proposition 8.8, report
Debris or leak in the return path	Rise of drive power, particle count	μ rises, G may change	Stop in a safe state, report

Table 8.5: Failure modes of the carrier loop, their observables, their effect on the model, and the response the contract requires.

In the worked example, with $K_{c,i} = 150$ J/K assumed for every core, the margins of Proposition 8.6 give $t_1 = 20.3$ s for the first core and $t_8 = 3.5$ s for the last. The response time that this module would have to meet is 3.5 s, set by the core with the least margin per unit of $P_i/K_{c,i}$. Which core that is depends on all three quantities. The ordering of Proposition 8.9 was derived for the steady-state peak, and the lumped stall model turns out to be covered by the same argument when the effective thermal masses are equal.

Proposition 8.14 (Ordering for stall response, equal masses)**.** *Let $K_{c,i} = K_c$ for every core, and fix a multiset of powers, so that θ_0 is the same in every ordering. For each $t \geq 0$, every core retains at least t seconds of lumped-model response time after a stall if and only if*

$$\max_i \Big[\theta_0 + \frac{1}{C}\sum_{j<i} P_j + \Big(a + \frac{t}{K_c}\Big)P_i\Big] \leq T_{\max}.$$

For every t this maximum is minimized by the nonincreasing power order, so that order maximizes the guaranteed response time $\min_i t_i$, and it minimizes the whole envelope of worst-case temperatures after a stall.

Proof. The condition $t_i \geq t$ reads $T_i^{\mathrm{J}} + tP_i/K_c \leq T_{\max}$, and substituting Proposition 8.6 gives the displayed form, which is the profile with a replaced by $a' = a + t/K_c$. Since $a \geq 1/C$ and $t/K_c \geq 0$, $a' \geq 1/C$, which is the only property the adjacent-exchange argument of Proposition 8.9 uses. That argument shows that moving a larger power upstream of a smaller one does not raise the maximum, so the nonincreasing order minimizes it for every t. The set of t for which an ordering is feasible is therefore largest for the nonincreasing order. □

With position-dependent masses $K_{c,i}$ the coefficient multiplying P_i differs from core to core, and the exchange argument no longer goes through directly. Maximizing the minimum of $K_{c,i}(T_{\max} - T_i^{\mathrm{J}})/P_i$ over assignments in that case remains open. A speed sensor on the drive detects a stall within milliseconds at 0.5 m/s, which is far inside the budget, and the budget that matters is the one for actuation, since reducing power across all cores through the accelerator's own power management must also finish within it.

Table 8.5 lists the failure modes together with what each does to the model and what the module must do. For a module in a cloud service the response includes a report to the scheduler. A stall is a fault of the module, and an application running on it is subject to the same recovery contract as for any other fault of a module: a typed outcome,

either a commit before the module withdraws or a suspension that can resume from the last committed checkpoint [28]. The thermal contract adds the quantities that must be declared for the response to be checkable, namely the response time t_d and the sustainable power after a stall.

8.7 Relation to Other Heat Paths

A cold plate on a static interface has a junction temperature that is the coolant temperature plus the drops across the stack. It has no carrier rise, since the coolant flows past each cold plate in parallel, and its limit is the resistance of the stack. A vapour chamber or heat pipe spreads heat by evaporation and condensation within a sealed envelope, and moves it to a condenser with a small temperature difference. It has no moving solid, and its transport capacity is limited by the working fluid and wick. An oscillating or pulsating heat pipe moves heat by the motion of vapour plugs and liquid slugs in a capillary tube, and its behaviour is hard to predict in general. Microchannel and interlayer cooling bring the liquid to the silicon [25, 21]. The sliding carrier replaces none of these. Its distinctive feature is that heat is moved along the spine by the motion of a solid that returns, so that the interface to each core is a dynamic one, and its distinctive costs are the contact, friction, and wear terms of Section 8.5.

The comparison that would decide between these paths is a comparison of measured junction temperature at equal power, equal coolant, and equal lifetime. The model in this chapter supplies the prediction for the sliding carrier, and the protocols of Chapter 13 supply the measurement. No claim is made here that the sliding carrier performs better than any of the alternatives.

8.8 What the Two Architectures Share

The pod loop and the carrier have the same two limits on the heat they can deliver. The heat leaves the loop no hotter than T_e in the pod architecture and no hotter than the carrier inlet temperature θ_n in the carrier architecture, and both temperatures are bounded by the junction limit, as Proposition 7.3 and Proposition 8.11 show. The shared quantity through which the elements couple is the heat load on the cooling plant in the pod loop and the carrier's heat content in the carrier, and in each case a check that treats the elements one by one cannot certify the whole. The carrier makes the coupling explicit in a closed form, which is why it is the clearest example for the contract problem of Part IV. A stall of the carrier, which Proposition 8.13 treats, is the continuous counterpart of the scheduled withdrawal of a pod that leaves for cooling, and both are handled by the recovery discipline of Chapter 11.

Part IV

Contracts at Module Boundaries

Chapter 9

Local Transitions and Global Admissibility

When circulating modules are sold as a service, what is promised is a commitment about what happens at module boundaries. The first question about a boundary is what each module can check for itself and what only the assembly can check. The carrier of Chapter 8 supplied an instance in which every core was acceptable alone and the assembly was not. This chapter states the general form of the phenomenon and the remedies that exist.

9.1 Local Checks and a Shared Limit

Consider three pods with normalized cooling requests $b = (b_1, b_2, b_3)$. Each pod locally checks $0 \le b_i \le 1$. Global admission additionally requires $\sum_i b_i \le 2$. The family $b = (1, 1, 1)$ passes every local bound and fails the global one. The shared quantity is the cooling capacity of the plant, and a check that sees one pod at a time cannot see it.

Proposition 9.1 (Certified allocation)**.** *Suppose a coordinator issues reservations $r_i \ge 0$ with $\sum_i r_i \le 2$, and each pod checks $b_i \le r_i$. Then the local checks imply the global condition $\sum_i b_i \le 2$.*

Proof. $\sum_i b_i \le \sum_i r_i \le 2$. □

A global condition has thereby been transformed into local checks relative to a certified allocation, and issuing the reservations still requires coordination. Counterexamples of this kind depend on the available representation. For the carrier, a power budget for each core that is certified to satisfy the profile of Proposition 8.6 for the whole ordered assignment is the same device: each core's check $P_i \le r_i$ is local, and the coordinator's certificate is the global computation of the profile with the ordering of Proposition 8.9.

On a smooth portion of a continuous domain defined by $g(s) \le 0$, the boundary condition $\nabla g(s) \cdot f(s, u) \le 0$ when $g(s) = 0$ is again a local test. For the book's computational model the test of this inequality is distinguished from the proof of forward invariance, which needs the regularity of Theorem 7.1, and every allowed jump must land in the safety domain.

9.2 Constraints No Pairwise Check Can Reproduce

Reservations repair a counting constraint because the constraint is a sum. Some constraints are not decomposable into checks on pairs.

Proposition 9.2 (Parity example)**.** *Let each $S_i = \{0, 1\}$ for $i = 1, 2, 3$, with all restriction maps the identity, and let* $\mathrm{Adm}(s_1, s_2, s_3)$ *hold exactly when $s_1 \oplus s_2 \oplus s_3 = 0$. No family of predicates on single values and on pairs reproduces* Adm*.*

Proof. Any pair (s_i, s_j) extends to an admissible triple by choosing the third bit to fix the parity. A pairwise validator that accepts every pair occurring in some admissible state therefore accepts all four pairs, and the same holds for single values. The triple $(1, 0, 0)$ passes all pairwise and single-value validators and is inadmissible. □

The constraint here is the parity of three independent contributions, and it appears in a circulating machine whenever three modules each hold a share of a quantity whose total must satisfy a global identity.

9.3 Overlaps and the State Sheaf

Let the partition complex X be the finite poset of cells consisting of module cells and interface cells, with each interface cell above the module cells that it involves. Give X the Alexandrov topology, whose open sets are the up-sets. For a module i let U_i be the smallest open set containing its cell, namely the cell together with every interface cell that involves it. The sets U_i cover X, and $U_i \cap U_j$ consists of the interface cells involving both modules. A *state sheaf* $\mathcal{S}$ assigns to each open set a set of local states, with restriction maps. On an Alexandrov space a section over an open set is a family of local states, one per cell of the set, compatible with the restrictions along the order.

Theorem 9.3 (Gluing)**.** *Let $s_i \in \mathcal{S}(U_i)$ be given for each module. If $r_{ij} s_i = r_{ji} s_j$ on every overlap $U_i \cap U_j$, then there is a unique $s \in \mathcal{S}(X)$ restricting to each s_i. Equivalently, a global state is the equalizer of the pair of maps from $\prod_i \mathcal{S}(U_i)$ to $\prod_{i<j} \mathcal{S}(U_i \cap U_j)$.*

Proof. Define s at a cell x as the component at x of any s_i with $x \in U_i$. This is well defined, because if $x \in U_i \cap U_j$ the agreement on the overlap equates the two components. Compatibility along the order holds within each U_i, and any two comparable cells lie in a common U_i, since the larger cell belongs to the up-set generated by the smaller. Uniqueness follows because the U_i cover X. □

The theorem gives a test and a resource test: the compatibility evidence $c : \prod_{i<j} (r_{ij} s_i = r_{ji} s_j)$ and the admissibility of the glued state. A useful restriction retains the information needed by the overlap. A shared hash identifies a particular reservation record, whereas a count of reservations generally cannot determine whether two claims designate the same slot. For the parity example the remedy follows directly: a triple interface cell whose stalk carries the three bits makes the predicate checkable on a single cell, where pairwise cells could not check it. Constraints of higher order are handled by adding higher-order cells to the complex, and the price is the cost of exchanging the data those cells carry.

Exercise 9.4. Two pods each reserve dock 1 for the same time slot. Give an overlap record that detects the collision. Add a third pod and a coolant budget that cannot be verified by the collision check alone.

9.4 Repair and Its Obstruction

When overlaps disagree, a repair may be possible. Let $\delta \in I$ be the disagreement vector in a finite-dimensional space I of interface data, $A = \bigoplus_i A_i$ the space of permitted local repairs, and $D : A \to I$ the linear map sending repairs to the change in disagreement they produce, so that applying repair a changes the defect from δ to $\delta + Da$.

Theorem 9.5 (Repair obstruction)**.** *A compatible family is reachable by unrestricted permitted repairs if and only if* $-\delta \in \operatorname{im} D$*, equivalently* $\delta \in \operatorname{im} D$*, equivalently* $[\delta] = 0$ *in* $\operatorname{coker} D$*, equivalently* $\operatorname{rank}[\, D \mid \delta \,] = \operatorname{rank} D$*. If repairs are limited to a bounded feasible set* $B \subseteq A$*, a necessary and sufficient condition is* $-\delta \in D(B)$*. None of these conditions mentions the capacity of the links that carry the exchange.*

Proof. The defect after repair a is $\delta + Da$, which vanishes for some $a \in A$ exactly when $-\delta \in \operatorname{im} D$, and $\operatorname{im} D$ is a subspace so the sign is immaterial. The cokernel and rank formulations are restatements. With a bounded set the same argument gives $-\delta \in D(B)$. The statements depend on D, δ, and B alone. □

Example 9.6. Let $I = \mathbb{R}^3$, $A = \mathbb{R}^2$ with D mapping the first repair coordinate to e_1 and the second to e_2, and let $\delta = e_3$. Then $\operatorname{im} D = \operatorname{span}\{e_1, e_2\}$ does not contain δ, and $\operatorname{rank}[D \mid \delta] = 3 > 2 = \operatorname{rank} D$. No amount of link capacity changes this. The remedies are to enlarge the authorized repairs, to revise the contract, or to change the partition so that the interface structure changes.

Bandwidth can help the system discover and exchange δ sooner, but it cannot enlarge $\operatorname{im} D$, and shared resource constraints can eliminate every glued state even when the obstruction vanishes. When the repair spaces form a coefficient system over the partition complex, the cokernel is the first cohomology of the two-term complex $\bigoplus_i A_i \to \bigoplus_{i<j} I_{ij}$ defined by D and can be read as a Čech class, though the rank formulation already supplies a checkable condition for a prototype [28].

Chapter 10

Exchange Windows under Circulation

In a stationary fabric the capacity of a link is a constant, and the question whether an exchange fits is a question about contention. In a circulating machine the capacity is itself a function of time. A pod can exchange data with a fixed endpoint only while it is docked, and the dock window of Table 5.1 is 78.3 s of a 107.2 s cycle. This chapter generalizes the fluid feasibility condition to time-varying capacity, so that availability windows and contention are treated by one test.

10.1 The Interval Condition

An exchange k has a payload b_k, a release time r_k, a deadline d_k, and a latency allowance ℓ_k for validation and control, and its *effective window* is $W_k = [r_k, d_k - \ell_k]$. A link e has capacity $C_e(t) \geq 0$, piecewise constant in time, which is the link rate multiplied by the indicator that both endpoints are docked. Under the fluid idealization each payload is divisible, may be sent preemptibly at any rate the link allows, and is available in full at its release time.

Theorem 10.1 (Interval condition)**.** *For a single link, a fluid schedule delivering every payload within its effective window exists if and only if for every interval $[a, b]$,*

$$\sum_{k:\, W_k \subseteq [a,b]} b_k \;\leq\; \int_a^b C_e(t)\, dt.$$

For several links the condition remains necessary for every link and is no longer sufficient, because an exchange crossing two links must receive the same rate on both.

Proof. Necessity: exchanges whose windows lie in $[a, b]$ transmit only inside it, the total rate never exceeds C_e, and so the delivered amount cannot exceed the integral. Sufficiency: let the endpoints of all windows cut time into elementary intervals $\Delta_1, \ldots, \Delta_p$, on each of which C_e is constant, and form the bipartite network whose left nodes are exchanges, with supply b_k, whose right nodes are the elementary intervals, with capacity $\int_{\Delta_q} C_e$, and which has an edge (k, q) when $\Delta_q \subseteq W_k$. A fluid schedule is a flow saturating the supply. By the supply–demand theorem of Gale [19], equivalent to max-flow min-cut, such a flow exists if and only if for every set K' of exchanges the demand does not exceed the capacity of its neighbourhood, $\sum_{k \in K'} b_k \leq \int_{N(K')} C_e$. The neighbourhood of K' is the union of its windows, a disjoint union of maximal intervals $I_1, \ldots, I_m$. If the inequality fails for

K', it fails for the part of K' lying in some $I_s = [a, b]$, since otherwise summing over the components would give the inequality. Enlarging that part to every exchange whose window lies in $[a, b]$ only increases the demand and leaves the capacity unchanged, which exhibits a violated interval. Hence if all intervals satisfy the condition, every K' does. □

A violated interval is a witness that a customer can check by arithmetic without trusting the scheduler. It names the link, the interval, the exchanges inside it, and the two sums, and it identifies three remedies, which are to reduce the payload inside the interval, to extend the effective window, and to reserve more capacity. The amount of each that is needed follows from the same inequality.

10.2 Docking Windows

Take the link of the pod loop, available at rate B only while a pod is docked, so that $C_e(t) = B$ on dock windows and zero elsewhere. An exchange of payload b that must be completed with the single dock window of length $\tau_D = 78.31$ s has capacity $B\tau_D$, and the condition is $b \le B\tau_D$. With $b = 40$ Tb and $B = 400$ Gb/s the window carries 31.32 Tb, the shortfall is 8.68 Tb, and the minimum rate that makes the exchange fit within one window is $b/\tau_D = 510.8$ Gb/s. If instead the shortfall is carried into the next window, the transfer completes 21.7 s into the second window. That window opens one cycle after the first, so the completion time is $107.17 + 21.70 = 128.86$ s from the start of the first. The rejection record states that the binding link is the hub uplink, the interval is the first dock window, and the shortfall is 8.68 Tb, which is what a customer needs to decide between a faster link, a longer dwell, and a smaller payload.

10.3 A Worked Example

Pod A holds a reservation to compute until time 12 s, followed by a transit slot from 12 to 14 s, and pod B may dock from 14 to 26 s. Use the second set of thermal parameters of Chapter 5, with $T_\infty = 130$ °C, $\tau_h = 50$ s, and an initial temperature of 45 °C at time zero. At release

$$T_A(12) = 130 - 85\,e^{-12/50} = 63.14\ ^\circ\mathrm{C},$$

which leaves 1.41 s before the 65 °C threshold if the pod continues at the same power. That margin is conditional on the thermal model. The route reservation and the thermal margin are distinct, and a hot arrival at a blocked route may have no safe continuation even when each resource looks available over a whole cycle.

Suppose the dock cannot commit until it has received 24 Gb of checkpoint data and validated the proposal. The link can reserve 10 Gb/s during $[9, 12]$, and validation plus control latency takes 0.4 s after the data arrive, so the effective window is $[9, 11.6]$ and carries 26 Gb. At full reservation the checkpoint fits with 2 Gb of headroom. A second exchange of 4 Gb that must lie wholly inside the same window makes the demand 28 Gb against a service of 26 Gb, and the interval witness records both sums. The shortfall is 2 Gb. Restoring the condition requires a reservation of at least $28/2.6 = 10.77$ Gb/s, or a window longer by 0.2 s at the reserved rate, or 2 Gb less demand in the interval. The diagnosis says nothing about whether the received data, if delivered under a revised reservation, will establish the application invariant, which is the question of Theorem 9.5.

Exercise 10.2. Repeat the checkpoint calculation with validation that can overlap the final 0.2 s of transmission. State a dependency graph that makes the overlap assumption valid.

Find the additional service required by the competing exchange, and explain how the result changes if its release and deadline are wider than those of the checkpoint.

Chapter 11

Commit, Staleness, and Recovery

11.1 The Typed Commit

Between modules, a global commit consumes the obligations of the preceding chapters. Let $P_i(t,h)$ be the type of a proposal by module i based on epoch t and parent checkpoint digest h, containing its proposed next local state and the interface data it exposes. Let $\mathrm{Compat}(p)$ be the type of evidence that the overlaps of the proposed states agree, let $\mathrm{glue}(p,c)$ construct the global section of Theorem 9.3, and let Inv_{t+1} be the invariant of the next epoch. Define

$$\mathrm{Ready}(t,h) := \sum_{p:\prod_i P_i(t,h)} \sum_{c:\mathrm{Compat}(p)} \mathrm{Inv}_{t+1}\big(\mathrm{glue}(p,c)\big),$$

$$\mathrm{commit}_{J,t,h} : \mathrm{Ready}(t,h) \longrightarrow \mathrm{IO}\big(\mathrm{Receipt}(J,t+1) + \mathrm{StaleParent}(J,t,h)\big).$$

An inhabitant of $\mathrm{Ready}(t,h)$ is a set of proposals, a proof that they glue, and a proof that the glued state satisfies the next invariant, and the evidence is tied to the exact transition proposed because every component mentions t and h.

A proposal can be valid when produced and stale when received. Validity of the evidence says nothing about whether the job has moved on, so the commit performs an atomic compare-and-swap of the head of job J against (t,h). The evidence is checked first and the swap decides which valid proposal prevails when two coordinators race.

Proposition 11.1 (One winner)**.** *Assume the head of job J is updated only by atomic compare-and-swap, and every successful swap strictly increases the epoch. If two coordinators submit commits against the same (t,h), at most one receives a receipt, and the other receives* $\mathrm{StaleParent}(J,t,h)$.

Proof. The first swap to execute finds the head equal to (t,h) and replaces it with a pair of epoch $t+1$. The second finds a head whose epoch differs from t, so its comparison fails. □

Every attempt ends in exactly one outcome: a receipt, a rejection, or supersession. The rejections are of three classes, according to whether the evidence of gluing is defective, the invariant fails, or the schedule is infeasible in the sense of Theorem 10.1. An attempt is superseded when it passes evaluation and loses the swap. Evaluation is performed against the head the attempt names, so the class of a rejection does not depend on whether another attempt has since advanced the head. A superseded attempt is not a rejection of any class.

The word stale is used in two senses that should not be confused. A stale parent means that the head moved between the time an attempt named it and the time its swap executed, and it is decided at the swap. A stale proposal means that the state a module read has

drifted from the current state, and it is decided at admission against the staleness contract of the next section. A proposal can be stale in one sense and not in the other.

11.2 Drift and the Rejoin Condition

Between exchanges each module continues to take local steps, and the copy of its projection held by its neighbour grows stale. Suppose the exchanged state carries a metric d and that a single local step of module i changes its projection by at most δ_i.

Proposition 11.2 (Drift bound)**.** *If the neighbour last received the projection at step t_0, then after m local steps the distance between the held copy and the true projection is at most $m\delta_i$. A tolerance ε_{ij} on this distance is therefore respected whenever $m \le \lfloor \varepsilon_{ij}/\delta_i \rfloor$.*

Proof. By the triangle inequality the displacement over m steps is at most the sum of the m single-step displacements. □

The bound is conservative, since steps may partly cancel, but it is computable from a declared per-step bound and supplies the maximum interval for a contract. In a circulating machine it couples the cooling time to the contract. A pod that leaves for cooling is away for $\tau_{\text{away}} = \tau_B + 2t_x$, during which it misses $\lceil \tau_{\text{away}}/\Delta \rceil$ exchange epochs of length Δ.

Corollary 11.3 (Rejoin condition)**.** *A pod may rejoin without a refresh if and only if $\lceil \tau_{\text{away}}/\Delta \rceil \le \lfloor \varepsilon/\delta \rfloor$.*

For the pod of Table 5.1, $\tau_{\text{away}} = 22.86 + 6 = 28.86$ s. With epochs of 10 s the pod misses 3 epochs, and with epochs of 2 s it misses 15. Taking $\varepsilon = 0.05$ and $\delta = 0.004$, so that $\lfloor \varepsilon/\delta \rfloor = 12$, the first schedule rejoins freely and the second does not. The second must pay a refresh. If the refresh transfers 2 Tb at 400 Gb/s, it takes 5 s of the dock window, so the productive dwell is 73.31 s of a 107.17 s cycle and the duty cycle falls from 0.731 to 0.684. By Corollary 6.2 the pool needed to saturate eight docks rises from 11 pods to 12. The staleness contract has a price in hardware.

A rebase differs from a stale application. A stale application admits a proposal computed at an earlier state and uses it unchanged against the current one, which is permitted only when the staleness contract holds. A rebase discards the proposal and has the module read the current state and recompute, and the receipt records which operation occurred. If the head advances during a rebase, the rebased proposal still names the parent whose state the module actually read, passes evaluation against it, and is superseded at the swap, not rejected.

11.3 Receipts, Replacement, and Recovery

A replacement pod continues a job only if the record permits it. For a checkpoint h and a replacement p, a typed admission carries

$$\text{Resume}(p, h) = \sum_{z : \text{Decode}(h)} (\text{EnvOK}(p, z) \times \text{Authorized}(p, h) \times \text{ResourceOK}(p, z)).$$

Here $\text{Decode}(h)$ is the type of validated reconstructions. The type does not imply that the reconstruction will be retrieved in time, which is a separate admission obligation, and the receipt chain of Proposition 3.18 supplies the evidence that the checkpoint belongs to the history of the job.

A pod that departs a dock after submitting a proposal may lose the acknowledgment without losing the commit. The durable record must identify the winning attempt and support idempotent receipt retrieval after an ambiguous reply, because replaying the work under a new identity without checking the head can duplicate an external effect. Systems that expose such effects need a corresponding idempotency mechanism.

Definition 11.4 (Withdrawal discipline)**.** A module with a scheduled departure at time t_{dep} satisfies the withdrawal discipline if, by t_{dep}, either a commit against its last head has completed, or the job is suspended at the last durable checkpoint with that checkpoint retrievable.

Proposition 11.5. *Under the withdrawal discipline and atomic head update, at t_{dep} the head of the job is either the advanced head, with a receipt retrievable by attempt identifier, or the unchanged head, with a checkpoint from which a replacement can resume. In neither case is an epoch skipped or duplicated.*

Proof. By atomicity of the swap the head is exactly one of the two values at t_{dep}. In the first case the commit that advanced it is the unique winner by Proposition 11.1, and its receipt is durable. In the second case no commit has been applied and the discipline supplies the retrievable checkpoint. A resumed attempt names the head it finds, so it either commits the next epoch or is superseded. □

A worked failure

Return to the example of the preceding chapter. In the uncontended case the transfer completes at 11.4 s, a verifier admits the record, and the swap succeeds at 11.8 s. The dock loses the acknowledgment before releasing A at 12 s. The successor head is authoritative. If B receives a retry request naming the same operation, it queries the receipt by attempt identifier, and the receipt records a committed result. The thermal route is free to continue while the digital recovery proceeds, and the lost acknowledgment becomes an observable communication failure that does not force duplicate work.

In the contended case the job contract allows suspension after the last durable checkpoint, and there is a preauthorized departure at 12 s. The dock rejects the proposed schedule, leaves the head unchanged, and releases A. Later recovery may use B, but must verify that the old checkpoint remains retrievable and that the software and resource witnesses still hold. The branch separates a safe physical exit from a recoverable computational exit. If the scheduler omitted the old checkpoint or allowed it to expire, the physical trajectory would still succeed while the recovery promise failed.

Exercise 11.6. Design a log trace in which the physical pod returns on schedule and all route observations match the previous circuit, yet continuing the old job would be invalid. Add the smallest event or record field that lets the next dock distinguish the cases.

Part V

Synthesis

Chapter 12

Determination and Invariance

Computation, memory, entropy, geometry, identity, and agency are all derived from a single pair of a routing space and an admissible path set. The precise content of that claim is a statement about which derived quantities are determined by which part of the structure, and about which are not.

12.1 Invariants of an Admissibility Structure

An admissibility structure at class j is a finite graph $G_j = (S, E_j)$ with edge costs c and edge labels in a set Λ. An *isomorphism* of structures is a bijection of cells preserving edges, costs, and labels.

Theorem 12.1 (Invariance)**.** *Let $G = (S, E, c, \lambda)$ be a labelled structure with the edge set of a single class. Then, first, if $\varphi : S \to S'$ is an isomorphism, $N_L(s) = N'_L(\varphi s)$ for all L, $S_{\mathrm{acc}}(s) = S'_{\mathrm{acc}}(\varphi s)$, $d(s,t) = d'(\varphi s, \varphi t)$, and the trace sets of s and φs coincide. Second, if $s \sim s'$ are bisimilar by a bisimulation that matches edges of equal cost, then s and s' have the same traces and, for every bisimilarity class $[t]$, the same distance $d(s,[t]) = \min_{t' \in [t]} d(s,t')$. Third, if s, s' lie in one block of an equitable partition, then $N_L(s) = N_L(s')$ for all L and $S_{\mathrm{acc}}(s) = S_{\mathrm{acc}}(s')$.*

Proof. For the first statement, an isomorphism maps walks of each length and cost bijectively onto walks of the same length and cost, and preserves label sequences. For the second, a walk from s to t is matched step by step by a walk from s' to some $t' \sim t$ with the same label sequence and the same cost, and symmetrically, so the minimum cost to reach the class $[t]$ is the same from both, and the traces are equal. The third is Proposition 3.15. □

The converse fails, which is the sense in which the quantities are derived and not identical to the structure. Take the labelled structure G_1 with edges $s \xrightarrow{a} t$ and $t \xrightarrow{b} t$, and G_2 with the single edge $u \xrightarrow{a} u$. Both have $N_L = 1$ for every L, so $S_{\mathrm{acc}} = 0$ at every cell, while the trace ab belongs to s and not to u. Accessibility entropy is a coarser invariant than trajectory identity, and trajectory identity is coarser than the structure itself. That all of these are derived from the pair $(S, \{E_j\})$ is therefore true in the sense of Theorem 12.1, as functions of the structure that are constant on its isomorphism and bisimulation classes, and false if read as saying that any one of them determines the others.

12.2 The Coarsest Faithful Abstraction

Which distinctions must a model retain? Remove temperature from the state and a cool arrival and a hot arrival at the same dock appear equivalent, so admission needs an external

guard. Remove the parent reference and two identical route traces can hide a committed result in one run and an unfinished attempt in the other. Remove reservation identifiers and two pods can present the same local claim while trying to use the same berth. These are three failures of projection, each visible under a particular intervention. The general problem is to find the coarsest state abstraction that preserves the decisions the machine has to make.

Let X be the finite concrete state set of a verified simulator, $\mathcal{A}$ its labelled actions, and $\mathcal{P}_0$ a partition of X by the observations that the service publishes. Define the refinement step that splits a block whenever two of its members have different sets of successor blocks under some action.

Proposition 12.2 (Coarsest bisimulation)**.** *Iterating the refinement step from $\mathcal{P}_0$ terminates after at most $|X|$ strict refinements, at a partition $\mathcal{P}^*$ which is a bisimulation for the labelled system, and every bisimulation that refines $\mathcal{P}_0$ also refines $\mathcal{P}^*$.*

Proof. Every strict refinement increases the number of blocks, which is at most $|X|$, so the process terminates. At termination no block splits, so related states have the same set of successor blocks under every action, which is the bisimulation condition. For minimality, suppose $\sim$ is a bisimulation refining $\mathcal{P}_0$. By induction on the number of steps, every iterate is refined by $\sim$: the base case is the hypothesis, and if $s \sim s'$ lie in one block of the current iterate then for each action their successors are related by $\sim$ and hence lie in one block of the iterate, so the next step does not separate them. Hence $\sim$ refines $\mathcal{P}^*$ at termination. □

A new intervention expands the action set and can force more distinctions, so the appropriate abstraction is relative to what users and controllers can do. A concept earns its place in a model when it changes an admissible continuation, and the proposition gives a procedure that finds exactly the distinctions that do.

12.3 Liveness and Viability

A viability controller permits the actions that maintain safety, and a liveness controller ensures that selected progress events occur under declared assumptions. They can conflict. A pod waiting in a buffer may be perfectly safe, and a controller that routes toward a dock at every opportunity may create queueing that stops another pod from cooling in time.

Proposition 12.3 (Bounded progress)**.** *Work with a finite abstraction. Let G contain the states in which a pod has docked and acquired a valid resume witness. Under a fixed policy suppose $V \geq 0$ decreases by at least $\epsilon > 0$ at every transition outside G, and that transitions continue while the state is outside G. Then a path from s reaches G in at most $\lfloor V(s)/\epsilon \rfloor + 1$ transitions.*

Proof. After k transitions that remain outside G one has $0 \leq V \leq V(s) - k\epsilon$, so $k \leq V(s)/\epsilon$, and the next transition enters G. □

A bound in seconds also needs a bound on the waits between transitions. If docks can remain occupied indefinitely the descent condition may fail, which is why the withdrawal discipline of Chapter 11 bounds occupation.

Exercise 12.4. Construct a safe two-pod deadlock. Add a reservation policy that removes it, and state a fairness assumption needed for both pods to receive service.

12.4 Apparent Goal Direction

The behaviour of a pod that moves toward the hub when cool and retreats when hot can be called a simulated agency. The claim can be made precise without attributing a representation to the pod. Work in the overdamped model $\dot{x} = -\nabla\Phi(x)$ of Theorem 5.2, with Φ of class C^2.

Theorem 12.5 (Goal direction without representation)**.** *Let x^* be an isolated critical point of Φ and let $K = \{\Phi \leq c\}$ be a compact connected sublevel set whose only critical point is x^*. Then K is forward invariant and every trajectory starting in K converges to x^*. Consequently, if a disturbance class consists of finitely many instantaneous displacements, after each of which the state lies in K, the trajectory still converges to x^*.*

Proof. Along a trajectory $\frac{d}{dt}\Phi = -\|\nabla\Phi\|^2 \leq 0$, so Φ is nonincreasing and the sublevel set K is forward invariant, and a trajectory in the compact set K exists for all time. By LaSalle's invariance principle the ω-limit set lies in the largest invariant subset of $\{\nabla\Phi = 0\} \cap K = \{x^*\}$. For the disturbance statement, apply this from the state after the last displacement. □

A destination, a disturbance class, and the controller state sufficient to achieve it have now all been named: the destination is x^*, the disturbance class is the finite displacements into K, and the controller state is the potential Φ itself, in which x^* is not stored as a target. This is the strongest claim that the framework supports about goal direction. It is not a claim that pods have goals, and it does not extend to unbounded disturbance. The claim of apparent goal direction is also testable by intervention. A policy that follows a nominal path and does not use feedback fails after a displacement that a gradient policy survives, and a route that is blocked distinguishes a robust local policy from one that merely traces a successful nominal path.

Chapter 13

Protocols and Limits

Every claim of the preceding chapters is conditional, and the conditions are what an experiment can test. Each run below is stated in the form *initial checkpoint*, *varied*, *injected*, *recorded*, *passes when*, and each uses inert loads, resistive heaters, or small deterministic computations in place of production workloads.

13.1 Pod and Contract Runs

Run M1, one loop with one heater. *Initial checkpoint.* An inert pod with metered resistive heat and temperature sensing, bath conditions fixed. *Varied.* The commanded route is repeated identically, and the dissipated power at several levels. *Injected.* None. *Recorded.* Time-aligned temperature, drive current, position, and switching events, with heat capacity and thermal conductance estimated in dedicated calibration trials. *Passes when.* Held-out dwell and cooling times, predicted from Proposition 5.6 before the traces are inspected, agree with the measured times within a tolerance stated beforehand, with recurrent temperature errors and time errors reported separately. Thermal equilibration or route failure that prevents the predicted guard crossing falsifies the cycle model.

Run M2, a dock and a second pod. *Initial checkpoint.* Two pods and one dock with a controlled contention window. *Varied.* Initial phases and initial temperatures. *Injected.* A held dock, with the corresponding reservation renewal withheld. *Recorded.* Complete resource occupancy, commanded and actual motion, and queue delays, with the primary evidence being the event trace around the denied action. *Passes when.* Each pod reaches a declared buffer, returns safely to cooling, or violates its safe set in a way that is recorded, and total work and deadline misses agree with the single-pod model and with a reservation-based simulation within the stated tolerance.

Run M3, recoverable compute. *Initial checkpoint.* A small deterministic computation under a fixed accepted-output contract, checkpointed at the dock. *Varied.* The pod removed, and the point of removal. *Injected.* A delay after durable head advancement and before acknowledgment, and separately an unavailable checkpoint object. *Recorded.* Committed work per unit wall time, work replayed, and the provenance of the winning attempt. *Passes when.* A replacement that finds the head already advanced retrieves the receipt and does not publish the result again, and an unavailable checkpoint produces the suspension outcome of the withdrawal discipline.

Run M4, comparison of cooling architectures. *Initial checkpoint.* Three configurations under the same offered workload: fixed compute with a cold plate, fixed compute with a moving thermal carrier, and moving compute with a dedicated cooler, each with the same acceptance conditions for compute work, data transfers, and recovery. *Varied.* The configuration. *Injected.* None. *Recorded.* At matched accepted work and declared recovery behaviour, the power at the facility boundary, peak temperatures, useful cooling load and outlet grade, and downtime caused by motion, interface failure, and maintenance, with every auxiliary load measured and distributions across repeats reported along with representative traces. *Passes when.* Each setup satisfies the same service contract. No outcome of the comparison is claimed in advance, and no prototype performance follows from the calculations of this book.

Run L1, structure in learned control. *Initial checkpoint.* Two equally budgeted learned controllers, one with the tangent projection of Proposition 7.4 and one without. *Varied.* The controller, and the initial states. *Injected.* None. *Recorded.* Accepted work, violation counts, false suspensions, calibration near the admission thresholds, and time spent on admissible trajectories. *Passes when.* The comparison is reported at matched budget, with errors at the guards reported separately from averages over ordinary states.

13.2 Carrier Runs

Run H1, carrier profile. *Initial checkpoint.* Eight heater cores on a spine with the carrier at its nominal speed and the coolant at its nominal inlet temperature and flow. *Varied.* The carrier speed over at least five values, and the power per core over at least three values. *Injected.* None. *Recorded.* The wall temperature of each core, the carrier temperature at each bore inlet, the node inlet and outlet temperatures, the drive power, and the friction power at each speed. The bore-wall and carrier temperatures identify G under the model, and R_j is identified only if a temperature is also recorded on the heater side of the junction-equivalent resistance, or if that resistance is calibrated independently before the run. *Passes when.* The fitted value of G, taken from the runs at one speed, predicts the wall temperatures at the other speeds within a tolerance stated before the run, the carrier temperature rise from core to core is P/C within the stated tolerance, and the sweep reports the speed beyond which the highest junction-equivalent temperature falls by less than a stated amount per added watt of friction power, which is the measured counterpart of Proposition 8.12.

Run H2, ordering. *Initial checkpoint.* As H1 at the nominal speed. *Varied.* The assignment of a fixed set of heater powers to positions, between the order of nonincreasing power and its reverse, and at least one intermediate order. *Injected.* None. *Recorded.* The highest junction-equivalent temperature for each order. *Passes when.* The nonincreasing order produces the lowest recorded maximum, as Proposition 8.9 requires, and the recorded maxima agree with the predictions of Proposition 8.6 within the tolerance.

Run H3, stall. *Initial checkpoint.* As H1, with the heaters at an assignment that leaves a stated positive margin $\delta > 0$ below $T_{\max}$ at every core. *Varied.* The time of the stall within the run, the heater power at the time of the stall, and the margin δ over at least three values. *Injected.* The carrier is stopped at a recorded time. In a second condition the drive is released so that it resumes. *Recorded.* A junction-equivalent temperature for each core, taken on the heater side of the junction resistance and sampled fast enough to resolve

the response interval, together with the detection time, the time at which power is reduced, and the peak temperature of each core. The effective thermal mass $K_{c,i}$ is fitted from the recorded transient over the response interval and is compared with the value the contract declares. *Passes when.* The peak temperature of every core stays below $T_{\max}$, the fitted $K_{c,i}$ is not larger than the declared value, so that the rig is not slower than the device it represents, the recorded response time is at most $\min_i t_i$ computed from the fitted values and the measured margins, and the resumed carrier restores the profile of H1 within the tolerance.

Run H4, contact life. *Initial checkpoint.* As H1, with a new carrier and bores. *Varied.* The sliding distance, in stated increments up to the rated life. *Injected.* A controlled quantity of debris in the return path in one condition. *Recorded.* The conductance G fitted at each increment, the friction power, the wear depth measured on the carrier and the bore at the end, and the particle count in the loop. *Passes when.* G remains above the declared floor and the friction power below the declared ceiling throughout, and the measured wear depth is consistent with the wear coefficient that the declared life assumes.

13.3 The Status of the Results

The results of this book are of three kinds. A result is *proved* when it follows from the definitions and stated axioms alone. It is *model-conditional* when it follows from a physical model whose assumptions are listed and could fail, and it is *interpretive* when it is a reading of the mathematics that no experiment on the machine would settle. Table 13.1 classifies the principal results.

13.4 Limits and Open Problems

The parameters of every example are assumptions. The wear coefficient of the carrier alone spans the range from irrelevant to disqualifying, and the magnetic force law, thermal conductances, and reconnect reliability of the pod loop are likewise to be measured. The models are lumped and steady-state, with the transient behaviour of a hot core meeting a cold carrier outside them. The mechanical questions of sealing, particle shedding, thermal expansion, and electrical isolation are outside the analysis. Which cooling architecture performs better at matched accepted work is decided by Run M4 and not by any argument here.

A favourable result on a single pod leaves the problem of dynamically issuing reservations in a shared cooling network. A favourable fleet simulation leaves the problem of validating magnetic force, thermal conductance, and reconnect reliability over repeated trips. A favourable prototype leaves the question whether a different form of modularity buys the same continuation guarantees at lower energy or maintenance cost.

The foundations have their own open question. Given two physical implementations and a declared class of future interventions, find the smallest record that preserves the ability to resume, audit, and repair. More record may improve reconstruction and delay commitment, and more freedom to route may improve availability and make the current plan harder to certify. These trade-offs become measurable once the service names the continuations it is prepared to guarantee, which is the content of Part IV. A second open problem is the maximization of the minimum stall response time when the effective thermal masses differ from core to core, which Proposition 8.14 solves only for equal masses, and

Result	Status	Depends on
Lawvere metric, restoration, trajectory identity, equitable invariance	Proved	Axiom 3.2, determinism
Trajectory memory convergence, receipt binding	Proved	Irreducibility, collision resistance
Accessibility metric and curvature sign, stationary density	Proved	Planar smooth model
Overdamped limit	Proved with bound	Lipschitz field, $\varepsilon L < 1$
No loop in a static potential, switching ledger	Proved	Reduced mechanical model
Exact limit cycle, safe ejection threshold	Model-conditional	Lumped thermal model
Throughput bounds, utilisation ceiling	Model-conditional	Operational laws, heat balance
Carrier profile, ordering, stations, grade ceiling	Model-conditional	Assumptions A1 to A4
Stall response time and its ordering (equal masses)	Model-conditional	Lumped core, positive margin
Interval condition, repair obstruction	Proved	Fluid idealization, linear repair model
Commit uniqueness, withdrawal discipline	Proved	Atomic compare-and-swap
Rejoin condition	Model-conditional	Declared per-step bound
Goal direction without representation	Proved	Gradient model
Pattern of return, hidden curvature, simulated agency	Interpretive	Not used in any proof

Table 13.1: Status of the principal results. All numerical examples use assumed parameters and report no measurement.

a third is the extension of the ordering result to a reciprocating carrier, whose equivalence to the continuous loop is only a matter for test.

Appendix A

Selected Exercise Solutions

Return perturbations. For a margin m require $\delta/(1-r) < m$ to obtain eventual strict inclusion in that margin. A finite-time guarantee also keeps the decaying term, and an initial point outside the margin cannot be certified by the ultimate bound alone. For the example of Chapter 5, the dock temperature immediately before ejection at the fixed point is 65 °C, and $\delta/(1-r) = 1.62\,\delta$.

Alternating returns. Let $P_{12} = P_{21} = 1$. Started at state 1, the one-time law alternates between the two point masses, while the edge frequencies converge to $1/2$ each. For initial-condition dependence take two absorbing states with distinct marks. In a fleet log the first failure looks like a period that is never absorbed into a steady occupancy, and the second like two groups of pods with permanently different statistics, so interchangeability of two pods cannot be inferred from the fact that each produces a stationary log.

Directed routing. For edges $1 \to 2 \to 3 \to 1$ with unit costs the distance matrix has rows $(0, 1, 2)$, $(2, 0, 1)$, and $(1, 2, 0)$. A long occupancy time at one vertex can be caused by a forced wait even when the onward route is unique, so occupancy alone cannot identify accessibility.

Checkpoint distinction. Use two identical route words ending at the same berth. In one run the input and executable digests match the job head. In the other a previous commit advanced the head while the pod was travelling, and the pod carries an older digest. Recording the candidate parent and comparing it with the authoritative head distinguishes the two before the next commit.

Bibliography

[1] Whitehead, A. N. *Process and Reality: An Essay in Cosmology.* Macmillan, New York, 1929.

[2] Bergson, H. *Creative Evolution*, translated by A. Mitchell. Henry Holt, New York, 1911.

[3] Rescher, N. *Process Metaphysics: An Introduction to Process Philosophy.* SUNY Press, Albany, 1996.

[4] Landauer, R. Irreversibility and Heat Generation in the Computing Process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.

[5] Rosensweig, R. E. *Ferrohydrodynamics.* Cambridge University Press, Cambridge, 1985.

[6] Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[7] Dorigo, M., Maniezzo, V., and Colorni, A. Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 26(1):29–41, 1996.

[8] Jaynes, E. T. Information Theory and Statistical Mechanics. *Physical Review*, 106(4):620–630, 1957.

[9] Kolmogorov, A. N. Three Approaches to the Quantitative Definition of Information. *Problems of Information Transmission*, 1(1):1–7, 1965.

[10] Penrose, R. *The Emperor's New Mind.* Oxford University Press, Oxford, 1989.

[11] Rosen, R. *Life Itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life.* Columbia University Press, New York, 1991.

[12] Lawvere, F. W. Metric Spaces, Generalized Logic, and Closed Categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, 43:135–166, 1973.

[13] Horn, R. A. and Johnson, C. R. *Matrix Analysis.* Cambridge University Press, Cambridge, 2nd edition, 2013.

[14] do Carmo, M. P. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Englewood Cliffs, 1976.

[15] Risken, H. *The Fokker-Planck Equation: Methods of Solution and Applications.* Springer, Berlin, 2nd edition, 1989.

[16] Denning, P. J. and Buzen, J. P. The Operational Analysis of Queueing Network Models. *ACM Computing Surveys*, 10(3):225–261, 1978.

[17] Nagumo, M. Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan*, 24:551–559, 1942.

[18] Aubin, J.-P. *Viability Theory.* Birkhäuser, Boston, 1991.

[19] Gale, D. A Theorem on Flows in Networks. *Pacific Journal of Mathematics*, 7(2):1073–1082, 1957.

[20] Archard, J. F. Contact and Rubbing of Flat Surfaces. *Journal of Applied Physics*, 24(8):981–988, 1953.

[21] Brunschwiler, T., Michel, B., Rothuizen, H., Kloter, U., Wunderle, B., Oppermann, H., and Reichl, H. Interlayer Cooling Potential in Vertically Integrated Packages. *Microsystem Technologies*, 15:57–74, 2009.

[22] Dennard, R. H., Gaensslen, F. H., Yu, H.-N., Rideout, V. L., Bassous, E., and LeBlanc, A. R. Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256–268, 1974.

[23] Kays, W. M. and London, A. L. *Compact Heat Exchangers*. McGraw-Hill, New York, 3rd edition, 1984.

[24] NVIDIA Corporation. NVIDIA GB300 NVL72. Product specification page, accessed November 2026.

[25] Tuckerman, D. B. and Pease, R. F. W. High-Performance Heat Sinking for VLSI. *IEEE Electron Device Letters*, 2(5):126–129, 1981.

[26] Yovanovich, M. M. Four Decades of Research on Thermal Contact, Gap, and Joint Resistance in Microelectronics. *IEEE Transactions on Components and Packaging Technologies*, 28(2):182–206, 2005.

[27] Flyxion. Local Transitions and Global Admissibility. Public repository, standardgalactic, 2026.

[28] Flyxion. Modular GPU Clouds with Optical Interconnect: A Specification of Placement, Exchange, and Recovery. November 2026.

[29] Flyxion. Never Predict Noise. Public repository, standardgalactic, 2026.