

The Ontological Shift of Observability: From Measurement to Control

Alexey A. Nekludoff

AstraVerge Research

E-mail: an@astraverge.org
ORCID: 0009-0002-7724-5762

27 February 2026

Abstract

Observability in contemporary IT practice is commonly understood as the ability to collect, visualize, and inspect system metrics. Such an interpretation treats observability as a property of measurement availability, largely disconnected from the operational processes it is meant to support.

This paper argues that this view is ontologically incomplete. In operational environments, metrics actively influence decisions and corrective actions, forming a feedback loop between the observed system and its operators. Once this feedback is made explicit, observability must be understood as a constituent part of a broader control system rather than as a passive measurement layer.

Operational metrics are often treated as self-sufficient indicators, leading to semantic fragmentation and inconsistent interpretations across systems and layers [1].

We present a control-theoretic interpretation of observability in which dashboards, compute layers, and human operators jointly form a human-in-the-loop control loop. Within this framework, we introduce a formal model for deriving a composite control variable, termed *operational pressure*, from heterogeneous metrics. The model applies normalization and temporal operators to capture instantaneous effects, accumulated stress, and accelerating degradation, yielding a bounded and interpretable signal suitable for decision support.

The proposed framework is inherently discrete-time and aligns with the sampling and decision cycles of real-world IT operations. Classical concepts from control theory, including transfer-function reasoning and PID-like temporal decomposition, are shown to be applicable as guiding principles without requiring full automation or continuous-time assumptions.

By reframing observability as a component of control, this work provides a conceptual foundation for systematic reasoning about stability, responsiveness, and interpretability in IT operations, and outlines a path from passive monitoring toward principled operational control systems.

Keywords: Observability; Control theory; Human-in-the-loop systems; Operational pressure; Discrete-time systems; IT operations; Feedback loops

Contents

1	Introduction	3
1.1	From Measurement to Control	3
1.2	Operational Pressure as a Control Variable	3
1.3	Human-in-the-Loop Control Systems	3
1.4	Contribution and Structure	4
2	Observability as a Control System	4
2.1	Implicit Control Loops in IT Operations	4
2.2	Control-Theoretic Decomposition	4
2.3	Human-in-the-Loop Actuation	5
2.4	Transfer Functions as Conceptual Tools	5
2.5	Discrete-Time Nature of Observability	5
2.6	Linearity as a Guiding Approximation	6
2.7	Position of Temporal Operators	6
2.8	Toward Automated Decision Systems	6
2.9	From Control Structure to Metric Model	7
3	Relation to SLO and Error Budget	7
3.1	Relation to Service Level Objectives (SLO)	7
3.2	Relation to Error Budget	7
4	Control-Theoretic Interpretation	8
4.1	Pressure as State Estimate	8
4.2	Service Level Objectives as Output Constraints	9
4.3	Relation to Model Predictive Control	9
4.4	Implications for Human and Automated Control	9
4.5	Summary	10
5	Model	10
5.1	Scope and Intent	10
5.2	Observations and Normalization	10
5.3	Temporal Operators	10
5.3.1	Proportional Component	10
5.3.2	Integral Component	11
5.3.3	Derivative Component	11
5.4	Metric Contribution Function	11
5.5	Composite Pressure Signal	11
5.6	Availability and Health Mapping	11
5.7	State and Computability	12
5.8	Control Interpretation	12
5.9	Distinction from Classical PID Control	12
5.10	Stability and Parameter Selection	12
5.10.1	Integral Stability	12
5.10.2	Derivative Noise Control	13
5.10.3	Weight and Gain Constraints	13
5.10.4	Human-Centric Stability	13
5.11	Model Invariants	13

6	Implications for Observability Tools	13
6.1	Dashboards as Control Interfaces	14
6.2	Consequences of Misconfiguration	14
6.3	Necessity of a Compute Layer	15
6.4	Human-in-the-Loop as a First-Class Design Assumption	15
6.5	From Observability to Operational Control	15
7	Monitoring versus Control Systems	15
8	Conclusion	16

1 Introduction

In contemporary IT practice, the term *observability* is commonly used to denote the ability to collect, visualize, and inspect system metrics. Dashboards, time-series graphs, and alerting rules are typically treated as passive instruments for human interpretation. Within this perspective, observability is understood as a property of measurement availability rather than as a functional component of system operation.

This paper argues that such an interpretation is ontologically incomplete.

In operational environments, metrics do not merely describe system behavior. They actively influence decisions, interventions, and corrective actions, thereby closing a feedback loop between the observed system and its operators. Once this feedback is acknowledged, observability can no longer be treated as an isolated measurement layer. Instead, it must be understood as a constituent part of a broader control process.

From this perspective, observability is not an end in itself, but a necessary condition for control.

1.1 From Measurement to Control

Classical control theory distinguishes between the controlled system (plant), the measurement subsystem (sensors), the controller, and the actuator. In IT operations, these roles are typically distributed across monitoring tools, compute layers, dashboards, and human operators.

Although these components are rarely described in control-theoretic terms, their interaction forms a closed loop: system behavior affects metrics, metrics influence interpretation, interpretation drives action, and action alters system behavior.

The absence of an explicit control model does not eliminate this loop; it merely obscures it.

This paper proposes that modern observability tooling implicitly implements a human-in-the-loop control system, even when it is presented as passive monitoring. Recognizing this fact enables the application of control-theoretic reasoning to the design, evaluation, and improvement of observability systems.

1.2 Operational Pressure as a Control Variable

A key limitation of metric-centric observability is the lack of a unified control variable. Individual metrics capture heterogeneous aspects of system behavior, but do not directly express operational risk or urgency.

To address this, we introduce the notion of *operational pressure*: a composite state variable derived from normalized metrics using temporal operators. Pressure represents accumulated, instantaneous, and accelerating stress on the system, and serves as an interpretable control signal for human decision-making.

Unlike raw metrics, pressure is explicitly designed to participate in a feedback loop.

1.3 Human-in-the-Loop Control Systems

In contrast to classical automated controllers, the actuator in IT operations is typically human. Human operators exhibit delay, nonlinearity, saturation, and heuristic behavior. Rather than treating these characteristics as defects, this work treats them as intrinsic properties of the control loop.

Accordingly, the proposed model focuses on interpretive stability rather than mathematical convergence, and on decision support rather than direct actuation.

A fully automated controller is considered a subsequent evolutionary step, not a prerequisite for control-theoretic analysis.

1.4 Contribution and Structure

The contributions of this paper are threefold:

- An ontological reframing of observability as a component of system control rather than passive measurement.
- A formal model for computing operational pressure from heterogeneous metrics using temporal operators.
- A control-theoretic interpretation of observability tools, dashboards, and compute layers as elements of a human-in-the-loop control system.

The remainder of the paper is structured as follows. Section 2 interprets observability as a closed-loop control system and introduces transfer-function representations. Section 5 introduces the formal pressure model. Section 6 discusses implications for observability tool design. Section 7 contrasts monitoring and control-oriented approaches. Finally, Section 8 summarizes the findings and outlines directions toward automated decision systems.

2 Observability as a Control System

2.1 Implicit Control Loops in IT Operations

Operational IT systems are rarely described using the formal language of control theory. Nevertheless, their day-to-day operation exhibits all essential elements of a closed-loop control system.

Changes in system load, configuration, or failure modes affect observable metrics. These metrics are collected, processed, and presented to operators, who interpret them and take corrective actions. The resulting actions alter the system state, closing the feedback loop.

This loop exists independently of whether it is explicitly modeled. The absence of a formal description does not remove feedback; it merely prevents systematic reasoning about its properties.

2.2 Control-Theoretic Decomposition

To make this feedback explicit, we decompose the operational loop into standard control-theoretic components:

- **Plant G** : the operational IT system, including infrastructure, services, and workloads.
- **Measurement subsystem S** : the transformation of internal system states into observable metrics, including sampling, aggregation, and delay.
- **Controller C** : the compute layer that derives composite signals (e.g., pressure or health) from metrics.
- **Actuator A** : the human operator, whose decisions translate control signals into concrete interventions.

Together, these components form a closed-loop system. In contrast to classical automation, the actuator is not a deterministic mechanism, but a human decision process.

2.3 Human-in-the-Loop Actuation

Human actuation introduces characteristics that are atypical in automated control systems:

- non-negligible reaction delay,
- nonlinear response and saturation,
- threshold-driven behavior,
- reliance on heuristics and experience.

Rather than treating these properties as modeling deficiencies, this work considers them intrinsic to operational control. The goal of observability is therefore not to eliminate human involvement, but to provide control signals that are stable, interpretable, and actionable within human cognitive limits.

2.4 Transfer Functions as Conceptual Tools

To fix the conceptual structure of the control loop, we introduce transfer-function notation. Let:

- $G(s)$ denote the plant dynamics,
- $S(s)$ the measurement subsystem,
- $C(s)$ the compute layer,
- $A(s)$ the human actuator.

The open-loop transfer function is then given by:

$$L(s) = C(s) A(s) G(s) S(s).$$

Under negative feedback, the closed-loop behavior can be expressed as:

$$T(s) = \frac{L(s)}{1 + L(s)}.$$

These expressions are not introduced to enable exact analytical solutions. Instead, they serve to:

- make feedback explicit,
- identify sources of delay and amplification,
- reason qualitatively about stability and responsiveness.

“Figure 1 summarizes the control structure assumed throughout the remainder of the paper.”

2.5 Discrete-Time Nature of Observability

Operational observability systems are inherently discrete:

- metrics are sampled at fixed intervals,
- dashboards update periodically,
- human decisions occur at discrete times.

Consequently, the appropriate mathematical framework is discrete-time control theory. All components of the loop can be interpreted in the z -domain without modification of their conceptual roles.

The use of discrete-time models does not restrict applicability. On the contrary, it aligns the theory with the actual execution environment of observability systems.

$$T = \frac{L}{1 + L}$$

$$L = C A G S$$

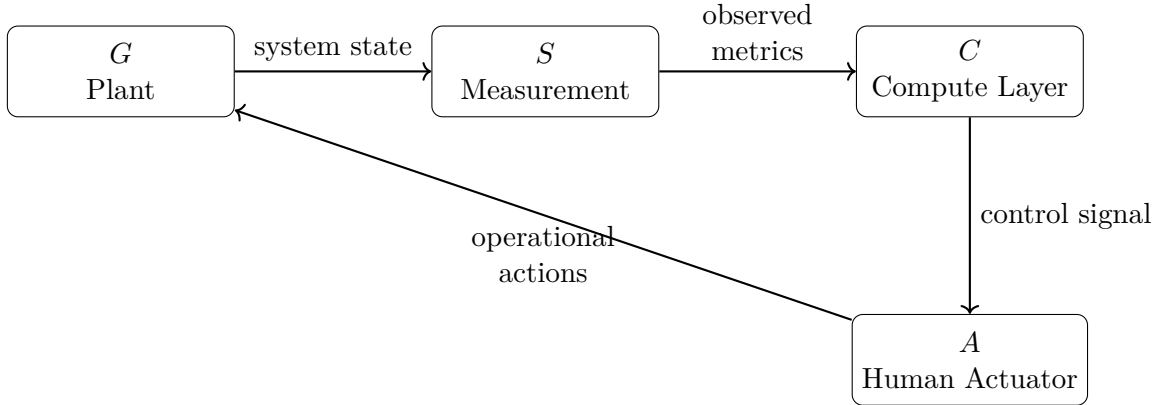


Figure 1: Human-in-the-loop control loop with open-loop composition $L = C A G S$ and conceptual closed-loop form $T = L/(1 + L)$

2.6 Linearity as a Guiding Approximation

The operational control loop is inherently nonlinear. Normalization, clipping, human decision thresholds, and saturation effects all introduce nonlinearity.

Nevertheless, linear system theory remains valuable as a guiding approximation. Local linearization around typical operating points enables reasoning about:

- relative gain between metrics and actions,
- effects of delay and aggregation,
- qualitative stability properties.

Linear theory is therefore used not as an exact description, but as a conceptual scaffold for understanding and designing observability systems.

2.7 Position of Temporal Operators

Within the control-theoretic interpretation, the temporal operators introduced in Section 5 assume a clear role.

Proportional components correspond to instantaneous feedback, integral components capture accumulated deviation, and derivative components represent trend acceleration.

These operators do not originate from a desire to replicate classical PID controllers, but from the temporal structure of operational degradation itself. PID-like decomposition emerges naturally as the minimal expressive basis for human-interpretable control signals.

2.8 Toward Automated Decision Systems

The human-in-the-loop configuration described here represents a stable and widely deployed operational regime. However, the same control structure admits gradual automation.

Replacing the human actuator A with an automated decision system does not alter the structure of the loop; it modifies only the actuation component.

This transition is deferred to future work. The present paper focuses on establishing a rigorous control-theoretic foundation for observability as practiced today.

2.9 From Control Structure to Metric Model

The control-theoretic interpretation developed in this section establishes the existence and structure of a closed operational feedback loop, but does not yet specify the form of the control signal available to the actuator.

To reason about stability, responsiveness, and interpretability, the controller component C must operate on a well-defined, bounded, and temporally meaningful variable.

Raw metrics, taken individually or as unstructured collections, do not satisfy these requirements.

Temporal misalignment between metrics, sampling intervals, and decision cycles is a structural property of observability systems, rather than an implementation artifact [2].

The following section introduces a formal model for constructing such a control variable. By normalizing heterogeneous metrics and applying temporal operators, the model derives a composite pressure signal that is suitable for participation in the control loop described above.

This model provides the concrete realization of the controller C assumed in the preceding analysis.

3 Relation to SLO and Error Budget

3.1 Relation to Service Level Objectives (SLO)

Service Level Objectives (SLOs) are typically defined as threshold-based constraints on externally observable service indicators, such as request success rate, latency percentiles, or availability over a time window. Formally, an SLO specifies an acceptable region in the space of outcome metrics, often evaluated retrospectively over fixed intervals.

The model presented in Section 5 differs from SLOs both in intent and temporal semantics.

First, SLOs operate on *outcomes*, whereas the proposed pressure model operates on *internal state indicators*. Pressure is computed from resource- and behavior-level metrics that may not immediately violate any SLO, but nonetheless indicate an increasing risk of future SLO breach.

Second, SLO evaluation is window-based and discrete, while the pressure signal is computed incrementally and reflects continuous system dynamics. As a result, pressure may increase even when all SLOs are formally satisfied.

In this sense, the pressure model can be interpreted as a *leading indicator* for SLO compliance. Rather than replacing SLOs, it provides early warning by exposing the structural and temporal causes that precede observable SLO violations.

Consequently, the relationship between pressure and SLOs is asymmetric:

- sustained high pressure increases the probability of future SLO violations,
- SLO violations imply that pressure has exceeded tolerable bounds in the past.

The pressure model thus complements SLOs by shifting attention from outcome validation to proactive state assessment.

3.2 Relation to Error Budget

The error budget framework quantifies the allowable amount of service degradation within a given time horizon. Error budgets are typically expressed as an integral measure, accumulating SLO violations over time.

The integral component of the pressure model shares conceptual similarities with error budget consumption. Both mechanisms emphasize persistence over instantaneous deviation: short-lived spikes are tolerated, while sustained degradation accumulates significance.

However, the two constructs differ in scope and resolution.

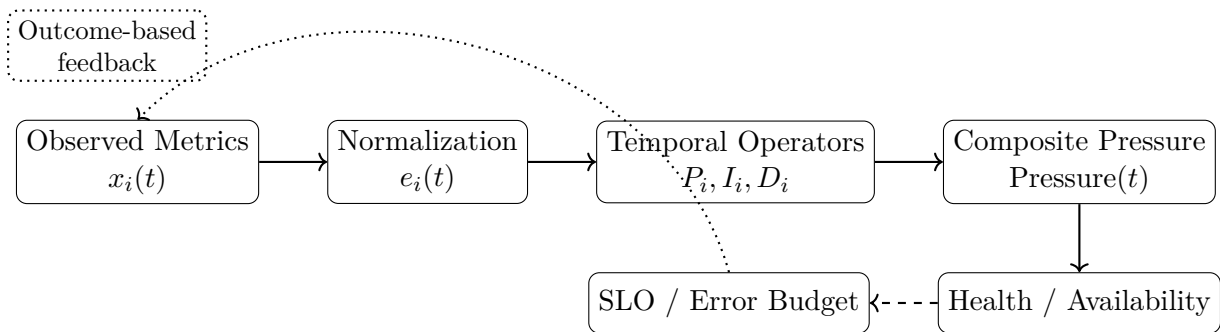


Figure 2: Conceptual structure of the pressure model and its relation to SLO and error budget. Internal metrics are transformed via proportional, integral, and derivative operators into a composite pressure signal, which is then mapped to health and availability. SLOs and error budgets operate on outcome metrics and provide delayed, retrospective feedback.

Error budgets operate on a *binary or thresholded* notion of correctness: an event either consumes budget or does not. In contrast, the pressure model accumulates continuous, normalized deviations across multiple dimensions, including those that may not yet trigger an SLO violation.

Moreover, the pressure model incorporates proportional and derivative components in addition to the integral term. This allows it to represent not only how much degradation has accumulated, but also how rapidly the system is moving toward exhaustion.

From this perspective, error budget consumption can be viewed as a *projection* of the pressure signal onto a specific outcome metric and time window. The pressure model generalizes this notion by capturing multi-metric, multi-timescale stress before it manifests as budget depletion.

Finally, while error budgets primarily support post-hoc governance decisions (e.g. release gating), the pressure model is designed for *real-time operational guidance*, supporting human decision-making prior to irreversible budget exhaustion.

In summary, error budgets formalize tolerance for failure, whereas the pressure model formalizes the dynamic approach to failure.

4 Control-Theoretic Interpretation

4.1 Pressure as State Estimate

In control-theoretic terms, the pressure signal defined in Section 5 can be interpreted as a *state estimate* of the monitored system.

Unlike externally observable outcome metrics, pressure is derived from internal indicators that reflect resource saturation, contention, and dynamic instability. Its construction explicitly incorporates temporal structure through proportional, integral, and derivative components, allowing it to capture both instantaneous stress and latent degradation.

As a state estimate, pressure serves two primary purposes:

- it provides a compact representation of system condition, aggregating heterogeneous metrics into a single interpretable signal;
- it enables forward-looking reasoning by exposing trends and accumulation effects before external failure becomes observable.

Importantly, the pressure signal is not tied to any specific service-level outcome. Instead, it represents an internal system state from which multiple outcome trajectories may emerge, depending on subsequent control actions.

4.2 Service Level Objectives as Output Constraints

Service Level Objectives (SLOs) and error budgets are most naturally interpreted as *constraints on observable outputs* rather than as control signals.

An SLO defines an acceptable region in the space of outcome metrics, typically evaluated over a fixed time window. Error budgets integrate violations of this constraint over time, providing a measure of tolerated degradation.

From a control-theoretic perspective, SLOs therefore play the role of output constraints: they specify what outcomes must be respected, but do not provide guidance on how the system should be steered to remain within those bounds.

This distinction is critical. While SLOs are well-suited for validation and governance, they are ill-suited as primary control inputs. Their windowed, threshold-based nature introduces delay and hysteresis, which can result in:

- delayed reaction to emerging degradation,
- apparent stability during periods of increasing internal stress,
- lack of early warning prior to irreversible failure.

Consequently, SLOs should not be interpreted as feedback signals for operational control, but rather as boundary conditions against which control decisions are evaluated.

4.3 Relation to Model Predictive Control

The separation between pressure as a state estimate and SLOs as output constraints is structurally analogous to reasoning employed in Model Predictive Control (MPC).

In MPC-style formulations:

- an internal state estimate is used to predict future system behavior,
- constraints define the admissible region of outputs,
- control actions are selected to minimize risk while respecting those constraints.

The model presented in this work does not implement a predictive optimizer or a receding horizon controller. However, it aligns with MPC at the level of abstraction: pressure provides the internal state signal from which future risk can be inferred, while SLOs define the externally imposed limits that must not be violated.

This structural alignment clarifies the complementary roles of pressure and SLOs: pressure enables anticipatory reasoning, whereas SLOs define the criteria for acceptability.

4.4 Implications for Human and Automated Control

In the present formulation, the actuator in the control loop is a human operator. Decisions such as scaling, throttling, restarting components, or deferring changes are informed by the pressure signal and validated against SLO constraints.

Crucially, replacing the human actuator with an automated decision system does not alter the structure of the control loop. The same distinction between state estimation and constraints applies.

Because pressure is continuous, state-based, and temporally structured, it is suitable as an input to automated control mechanisms. In contrast, SLOs and error budgets, being discrete and retrospective, are fundamentally unsuitable as direct control signals.

This observation explains a common failure mode in practice: attempts to drive automation directly from SLO alerts often result in unstable or excessively delayed responses.

By explicitly separating state estimation (pressure) from outcome validation (SLO and error budget), the model provides a foundation that supports both human-in-the-loop operation and future automation, without conflating fundamentally different roles.

4.5 Summary

The control-theoretic interpretation positions the proposed pressure model as an internal state estimator, with SLOs and error budgets acting as output constraints. This separation resolves common ambiguities in operational monitoring and clarifies why outcome-based metrics alone are insufficient for timely and stable control.

The resulting framework is compatible with established control principles, while remaining applicable to human-centered operational environments.

5 Model

5.1 Scope and Intent

We consider a monitored system observed through a finite set of metrics. The purpose of the model is not automatic control, but the computation of a composite state signal intended for *human-in-the-loop* decision making.

The model estimates operational pressure and availability degradation as functions of metric dynamics over time, explicitly accounting for heterogeneous temporal behavior, delayed effects, accumulation of stress, and acceleration toward failure.

5.2 Observations and Normalization

Let $x_i(t)$ denote the observed value of metric i at discrete time t . Each metric is normalized into a dimensionless deviation signal

$$e_i(t) \in [0, 1],$$

representing the degree of operational stress induced by metric i .

A generic normalization is defined as:

$$e_i(t) = \text{clip}\left(\frac{x_i(t) - b_i}{\ell_i - b_i}, 0, 1\right),$$

where b_i is the baseline (acceptable operating level), ℓ_i is the limit (critical or saturation level), and $\text{clip}(\cdot)$ restricts the value to $[0, 1]$.

This normalization establishes a semantic invariant: $e_i(t)$ *expresses stress rather than raw utilization*.

5.3 Temporal Operators

Each metric contributes to system pressure via temporal operators, rather than via its instantaneous value alone.

5.3.1 Proportional Component

The proportional component captures instantaneous or near-instantaneous effects:

$$P_i(t) = e_i(t - \delta_i),$$

where $\delta_i \geq 0$ is a metric-specific delay. This term models immediate impacts such as CPU saturation or packet loss spikes.

5.3.2 Integral Component

The integral component models accumulated stress over time. It is defined using an exponential moving average:

$$I_i(t) = \alpha_i e_i(t) + (1 - \alpha_i) I_i(t - \Delta t),$$

with

$$\alpha_i = 1 - e^{-\Delta t / \tau_i},$$

where τ_i is the integration time constant and Δt is the sampling interval.

This component captures persistent degradation phenomena, such as memory pressure, swap usage, or sustained I/O congestion.

5.3.3 Derivative Component

The derivative component captures acceleration toward failure:

$$D_i(t) = \frac{I_i(t) - I_i(t - \Delta t)}{\Delta t}.$$

Optionally, this term may be smoothed to suppress noise. The derivative component detects emerging instability, such as growing latency or rapidly increasing queue depth.

5.4 Metric Contribution Function

Each metric contributes a weighted combination of its temporal components:

$$c_i(t) = w_i (k_{p,i} P_i(t) + k_{i,i} I_i(t) + k_{d,i} D_i(t)),$$

where:

- w_i is the metric importance weight,
- $k_{p,i}, k_{i,i}, k_{d,i}$ are the proportional, integral, and derivative coefficients.

This formulation allows metrics to exhibit purely proportional, purely integral, purely derivative, or mixed behavior.

5.5 Composite Pressure Signal

The composite system pressure is defined as:

$$\text{Pressure}(t) = \sum_{i=1}^N c_i(t).$$

Optionally, pressure may be bounded:

$$\text{Pressure}(t) \leftarrow \text{clip}(\text{Pressure}(t), 0, 1).$$

The pressure signal represents total operational stress as interpreted by the control model.

5.6 Availability and Health Mapping

Availability or health is derived as a monotonic function of pressure:

$$\text{Health}(t) = 1 - \text{sat}(\text{Pressure}(t)),$$

or in percentage form:

$$\text{Health}_{\%}(t) = 100 \cdot (1 - \text{sat}(\text{Pressure}(t))).$$

The saturation function is interpretive rather than physical and serves to bound the output for decision support.

5.7 State and Computability

The model maintains per-metric state consisting of:

- the integral state $I_i(t)$,
- optionally the previous deviation value $e_i(t - \Delta t)$.

The total state size is $O(N)$ for N metrics. All computations are incremental and suitable for discrete-time execution.

5.8 Control Interpretation

The model defines a closed control loop with a human actuator:

- Plant: the observed system,
- Sensors: monitoring metrics,
- Controller: pressure computation,
- Actuator: human operator,
- Feedback: post-action metric evolution.

This constitutes a human-in-the-loop control system, where temporal structure guides decision-making rather than direct actuation.

5.9 Distinction from Classical PID Control

Unlike classical PID controllers:

- there is no single global setpoint,
- actuation is not automatic,
- the objective is state estimation and interpretation, not direct stabilization.

The use of proportional, integral, and derivative operators is therefore structural rather than mechanical.

5.10 Stability and Parameter Selection

Although the model does not directly actuate the system, its outputs must remain stable and interpretable to avoid oscillatory or misleading guidance.

5.10.1 Integral Stability

Integral stability is ensured by selecting finite integration time constants τ_i . For bounded inputs $e_i(t) \in [0, 1]$, the exponential moving average guarantees bounded $I_i(t)$.

To prevent excessive accumulation, τ_i should reflect the expected time scale of the underlying degradation mechanism.

5.10.2 Derivative Noise Control

The derivative component is sensitive to measurement noise. Stability requires:

- sufficiently large τ_i for the integral state,
- optional smoothing of $D_i(t)$,
- conservative choice of $k_{d,i}$.

Derivative terms are intended to indicate trend acceleration, not to dominate the composite signal.

5.10.3 Weight and Gain Constraints

To ensure interpretability and boundedness, weights and gains should satisfy:

$$\sum_i w_i(k_{p,i} + k_{i,i} + k_{d,i}) \leq 1.$$

This constraint prevents any single metric from overwhelming the composite pressure.

5.10.4 Human-Centric Stability

Because the actuator is human, stability is evaluated in terms of cognitive load:

- pressure should change smoothly under normal conditions,
- sustained degradation should be emphasized over spikes,
- acceleration signals should precede failure by a meaningful margin.

The model is therefore tuned for interpretive stability rather than mathematical convergence alone.

5.11 Model Invariants

The model enforces the following invariants:

1. Temporal heterogeneity of metrics is explicit.
2. Short-term spikes and long-term degradation are distinguishable.
3. Pressure accumulates monotonically under sustained stress.
4. Contributions are explainable per metric and per operator.

The qualitative behavior of proportional, integral, and derivative components in response to a step change in a single metric is illustrated in Figure 3.

6 Implications for Observability Tools

The control-theoretic formulation introduced in Section 5 has direct consequences for the interpretation and design of observability tools.

Within this framework, observability tools are no longer treated as passive visualization layers. Instead, they become functional components of a human-in-the-loop control system.

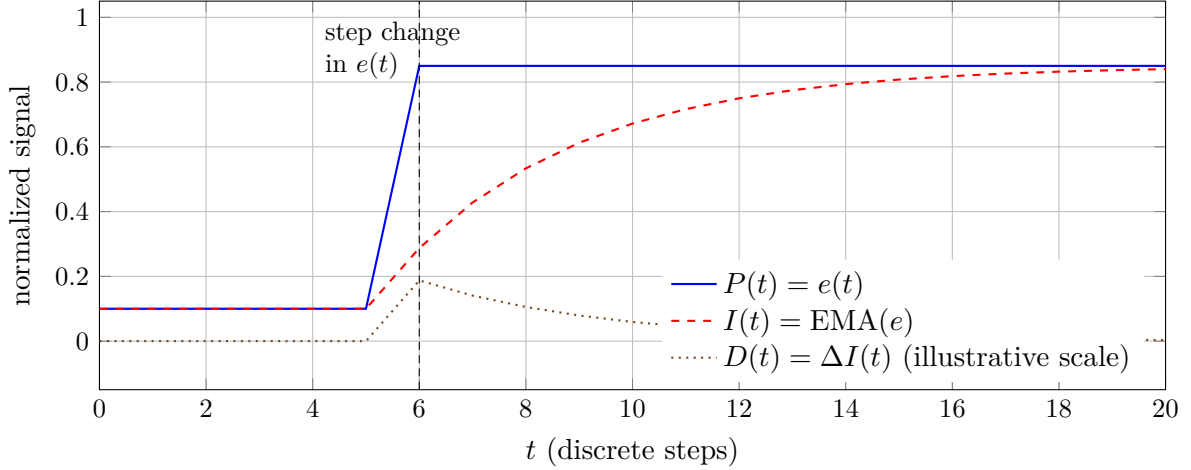


Figure 3: Illustrative time evolution of proportional, integral, and derivative components for a single normalized metric deviation $e(t)$. The proportional term reacts immediately to a step increase, the integral term accumulates gradually (EMA), and the derivative term peaks near the change point and decays as the signal stabilizes.

6.1 Dashboards as Control Interfaces

Dashboards determine how the composite pressure and health signals are presented to the human actuator. As such, they implicitly implement filtering, aggregation, scaling, and thresholding operations.

These operations affect:

- the effective gain of the control loop,
- the perceived delay of feedback,
- the balance between proportional, integral, and derivative information available to the operator.

From a control perspective, dashboard configuration is therefore equivalent to tuning parameters of the controller–actuator chain.

6.2 Consequences of Misconfiguration

Within the proposed model, common observability issues acquire precise control-theoretic interpretations:

- Excessive aggregation introduces phase delay, reducing responsiveness.
- Overemphasis on instantaneous metrics amplifies noise, leading to oscillatory operator behavior.
- Lack of trend or accumulation signals suppresses integral awareness, delaying intervention.
- Excessive alerting corresponds to gain saturation, degrading actuator effectiveness.

Thus, a poorly designed dashboard does not merely “misinform” the operator; it destabilizes the feedback loop.

6.3 Necessity of a Compute Layer

Raw metrics alone do not define a control signal. The compute layer is required to:

- normalize heterogeneous metrics into a common semantic space,
- apply temporal operators (P/I/D),
- enforce boundedness and interpretability,
- expose composite state signals suitable for decision making.

Without such a layer, observability tools provide data but fail to construct a coherent control variable.

6.4 Human-in-the-Loop as a First-Class Design Assumption

The model explicitly treats the human operator as an element of the control loop, rather than as an external observer.

This implies that:

- system stability must be evaluated in cognitive, not purely mathematical, terms;
- responsiveness must account for human reaction time;
- explainability of signals is a functional requirement, not an aesthetic one.

Observability tools designed without this assumption cannot be considered components of an operational control system.

6.5 From Observability to Operational Control

The transition from monitoring to control is not achieved by adding more metrics, but by redefining their role.

In the proposed framework, observability tools become part of an operational control system for IT landscapes, analogous to supervisory control systems in industrial automation.

7 Monitoring versus Control Systems

The distinction between monitoring and control is often blurred in contemporary IT practice. Monitoring systems are frequently extended with dashboards, alerts, and automation hooks, creating the impression that monitoring gradually evolves into control.

This section argues that the difference is not incremental but ontological. Monitoring and control systems are based on fundamentally different assumptions about the role of metrics, time, feedback, and human participation.

To make this distinction explicit, Table 1 contrasts monitoring-oriented approaches with control-oriented systems along key conceptual dimensions. The comparison highlights that transitioning from monitoring to control requires a change in system interpretation and design, rather than the mere addition of new features.

Table 1: Conceptual differences between monitoring systems and control systems

Aspect	Monitoring System	Control System
Primary purpose	Observation and reporting	State regulation and decision support
Role of metrics	Raw measurements	Normalized state variables
Temporal semantics	Implicit, visualization-driven	Explicit (P/I/D operators, delays, time constants)
Aggregation	Ad hoc or presentation-driven	Model-defined and stability-aware
Human role	Passive observer	Explicit actuator in the loop
Dashboards	Visualization tools	Control interfaces
Alerts	Threshold notifications	Control signals indicating instability or trend
Feedback interpretation	Informational	Functional (affects loop behavior)
Failure interpretation	Post hoc analysis	Predictive and preventative
System model	Implicit or absent	Explicit, formalized
Stability concern	Rarely addressed	Central design criterion
Relation to automation	Independent or optional	Hierarchically coordinated

8 Conclusion

This paper has argued that the prevailing interpretation of observability in IT systems is ontologically incomplete. Treating observability as the mere availability of metrics, dashboards, and alerts obscures its functional role in operational decision-making.

By explicitly modeling the feedback loop between system behavior, metric computation, visualization, and human action, we have shown that observability is not an isolated measurement layer, but a constituent part of a broader control system. Within this framework, dashboards function as control interfaces, compute layers act as controllers, and human operators serve as actuators in a human-in-the-loop control loop.

To support this interpretation, we introduced a formal model for constructing a composite control variable, termed operational pressure, from heterogeneous metrics. The model applies normalization and temporal operators to capture instantaneous effects, accumulated stress, and accelerating degradation. This construction yields a bounded, interpretable signal suitable for participation in a feedback loop, addressing limitations of raw metric-based observability.

The control-theoretic perspective adopted in this work does not aim to replace existing observability tooling, but to reposition it. Monitoring systems are shown to differ fundamentally from control-oriented systems in their treatment of time, feedback, aggregation, and human participation. Recognizing this distinction clarifies why the addition of dashboards, alerts, or automation hooks alone does not transform monitoring into operational control.

Importantly, the proposed framework is grounded in discrete-time execution and aligns naturally with the sampling, aggregation, and decision cycles of real-world IT operations. Classical tools from control theory, including transfer-function reasoning and PID-like temporal decomposition, are therefore applicable without modification, serving as guiding principles rather than rigid analytical constraints.

While this paper focuses on human-in-the-loop control, the resulting model provides a foundation for gradual automation. Replacing the human actuator with an automated decision system does not alter the structure of the control loop, but changes the nature of actuation. This transition, including the design of stable and explainable automated decision policies, is deferred to future work.

In reframing observability as a component of control, this work aims to shift the conceptual foundation of observability practice. Such a shift enables systematic reasoning about stability, responsiveness, and interpretability, and opens a path from passive measurement toward principled operational control of IT landscapes.

References

- [1] A. A. Nekludoff, “Availability without a model: On the semantic fragmentation of operational metrics,” 2026. DOI: 10.5281/zenodo.18618639 [Online]. Available: <https://doi.org/10.5281/zenodo.18618639>
- [2] A. A. Nekludoff, “Temporal alignment in distributed observability systems: A structural account of time semantics, ordering, and cross-layer inference,” 2026. DOI: 10.5281/zenodo.18672509 [Online]. Available: <https://doi.org/10.5281/zenodo.18672509>